Journal of Advanced Research 25 (2020) 191–203

Contents lists available at ScienceDirect

Journal of Advanced Research

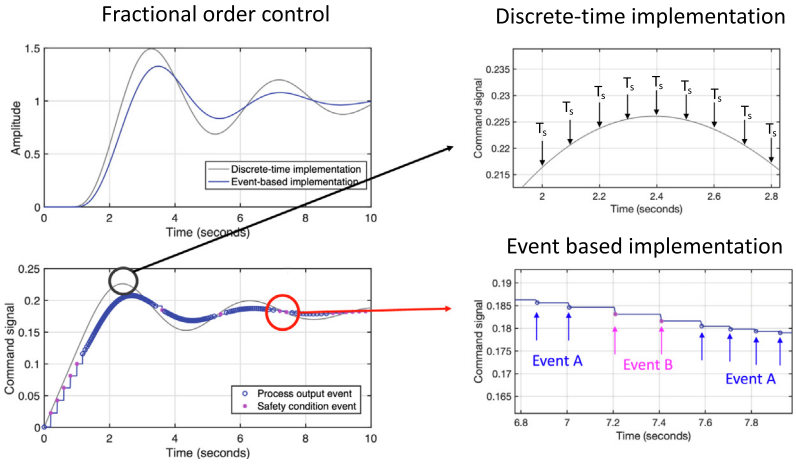
journal homepage: www.elsevier.com/locate/jare

Event-based fractional order control

Isabela Birs [a,b,c], Ioan Nascu [a], Clara Ionescu [b,c], Cristina Muresan [a,*]

[a] Technical University of Cluj-Napoca, Automation Department, Memorandumului 28, Cluj-Napoca, Romania
[b] Ghent University, Department of Electromechanical, Systems and Metal Engineering, Research Lab on Dynamical Systems and Control, Tech Lane Science Park 125, 9052 Ghent, Belgium
[c] Flanders Make, EEDT - Decision and Control Group, Tech Lane Science Park 131, 9052 Ghent, Belgium

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 16 April 2020
Revised 3 June 2020
Accepted 27 June 2020
Available online 3 July 2020

2010 MSC:
26A33
60G22
90C32

Keywords:
Fractional calculus
Fractional order control
Event-based control

ABSTRACT

The present study provides a generalization of event-based control to the field of fractional calculus, combining the benefits brought by the two approaches into an industrial-suitable control strategy. During recent years, control applications based on fractional order differintegral operators have gained more popularity due to their proven superior performance when compared to classical, integer order, control strategies. However, the current industrial setting is not yet prepared to fully adapt to complex fractional order control implementations that require hefty computational resources; needing highly-efficient methods with minimum control effort. The solution to this particular problem lies in combining benefits of event-based control such as resource optimization and bandwidth allocation with the superior performance of fractional order control. Theoretical and implementation aspects are developed in order to provide a generalization of event-based control into the fractional calculus field. Different numerical examples validate the proposed methodology, providing a useful tool, especially for industrial applications where the event-based control is most needed. Several event-based fractional order implementation possibilities are explored, the final result being an event-based fractional order control methodology.

☆ Peer review under responsibility of Cairo University.
* Corresponding author at: Technical University of Cluj-Napoca, Automation Department, Memorandumului 28, Cluj-Napoca, Romania.
E-mail addresses: Isabela.Birs@aut.utcluj.ro (I. Birs), Ioan.Nascu@aut.utcluj.ro (I. Nascu), ClaraMihaela.Ionescu@ugent.be (C. Ionescu), Cristina.Muresan@aut.utcluj.ro (C. Muresan).

https://doi.org/10.1016/j.jare.2020.06.024
2090-1232/

Introduction

Fractional calculus is a powerful tool with multiple applications in science and engineering. The main feature lies in the ability to accurately capture physical phenomena in a more natural and generalized manner than other available mathematical tools. Fractional applications spread through a multitude of fields such as physics, chemistry, mathematics, statistics, acoustics, engineering from both control and modeling point of views, providing excellent results through elegant mathematical solutions [1–4]. Its full potential is yet to be grasped by the scientific community through the multitude of papers emerging every year proving wide applicability of fractional calculus with uplifting results.

From the engineering perspective, fractional calculus has generalized the degrees of freedom of control possibilities and has revolutionized the modeling of memory based structures [5]. The ubiquity of the fractional order memory system gives a structure that is more concise in form, but encapsulates a great accuracy of complex dynamics [6]. The main contribution of fractionalization in the control field consists of the generalization of the popular Proportional Integral Derivative (PID) controller. The resulting strategy has been proven to be superior to its integer order instance. The obtained closed loop system featuring fractional order control has been proven to have increased time domain performance, stability, robustness and flexibility from the frequency response perspective [7–9]. Its diverse benefits contribute to the enhancement of multiple control possibilities such as the fractional order Internal Model Control (IMC) strategy, fractional order filter integration to available integer-order structures, fractional order Linear Quadratic Regulator (LQR), etc [10–13].

The present work focuses on introducing fractional calculus in a control branch that has not been previously explored from this perspective: the event-based control strategy. Event-based control is a relatively novel concept that triggers the computation of a new control signal through events, in comparison to classical controller implementations that trigger time based changes in the control signal [14,15]. The strategy is mostly beneficial for processes that have limited resources such as CPU, bandwidth allocation and reduced energy demand [16,17]. All of these features show the necessity of the event-based strategy in today's industrial settings [18]. Furthermore, cloud control systems are becoming more popular, the industry being the main beneficiary of such an approach that externalizes automatic control, providing a versatile solution that could be easily debugged and adapted [19,20]. Event-based control combined with cloud implementations is another powerful tool for the future of industrial control. The aim of the study is to introduce all the benefits provided by fractional order control into the event-based strategy, offering multiple possibilities to exploit these advantages to benefit the current industrial need.

The contribution of the current work to the fractional calculus field is the engineering applicability of this numerical tool into event-based control strategies. The theoretical background for event-based fractional order controllers is proposed and several implementation options are explored. The method is validated for the general fractional order PID controllers using varied numerical examples. Furthermore, the proposed approach is compared with classical, non-event based, discrete-time approaches showing the efficiency of the methodology as well as the benefits regarding the control effort.

The paper is structured as follows: Section 2 presents a brief familiarization with fractional calculus in control applications as well as current trends in fractional order control implementations; Section 3 is focused on existing event-based control approaches and provides a solid motivation of its usage in industry; Section 4 details the proposed event-based fractional order control strategy, theoretical aspects and implementation guidelines; Section 5 presents four numerical examples centered around the fractional order PID controller implemented using the proposed event-based strategy; while Section 6 concludes the paper. In addition, Appendix A provides the event-based fractional order control algorithm in a programming fashion.

Fractional calculus in control applications

Fundamentals of fractional calculus

Fractional calculus is a generalization of differentiation to any arbitrary order. The limit that integration and differentiation orders are limited to integer numbers is removed, creating a new branch of mathematics known as non-integer (fractional) calculus. The differintegral operation is generalized into the fundamental operator D_t^{α}, where $\alpha \in R$ is the fractional order [21].

The most common mathematical definition of the fractional order operator is given by Riemann–Liouville [22–24] as

$$ {}_aD_t^{-\alpha}M(t) = \int_a^x \frac{(t-\tau)^{\alpha-1}}{\Gamma(\alpha)} \tilde{M}(\tau)d\tau, \tag{1}$$

where D represents an integral for negative order and a derivative for the positive case from an arbitrary fixed base point a, and $\Gamma(\alpha)$ is Euler's Gamma function [25].

Another possibility to express the fractional differintegral operation is the Grunwald–Letnikov formula [25,26]

$$ {}_aD_t^{\alpha}M(t) = \lim_{h\to 0} h^{-\alpha} \sum_{0\leqslant j\leqslant \infty} (-1)^j \binom{\alpha}{j} M(t-jh), \tag{2}$$

where $\binom{\alpha}{j}$ represents the fractional binomial coefficient.

Applying the Laplace transform on the fractional differintegral of order α gives

$$ L\{D_t^{-\alpha}f(t)\} = s^{-\alpha}F(s) \tag{3}$$

$$ L\{D_t^{\alpha}f(t)\} = s^{\alpha}F(s) \tag{4}$$

where $F(s) = L\{f(t)\}$ and s the Laplace complex variable.

The connotations of (3) and (4) in automatic control are the extensions of the integer-order PID controller to its fractional order generalization denoted by

$$ C(s) = k_p\left(1 + \frac{k_i}{s^{\lambda}} + k_d s^{\mu}\right) \tag{5}$$

where k_p is the proportional gain, k_i and k_d are the integral and derivative terms, λ is the fractional order of integration and μ is the fractional order of differentiation. For the particular case $\lambda = 1$ and $\mu = 1$, the controller from (5) denotes the classical, integer order, PID controller. As can be seen from the form of the fractional order PID controller, the fractional orders of λ and μ lead to a more flexible structure than its integer order particularity. The PID generalization is capable of solving a wider and more restrictive set of performance specifications, having the shape of the popular PID controller, but with more power and pliability. Hence, a performing control option becomes available that proves better time response, increased stability, robustness and versatility [7,8,27,28].

Implementation possibilities of fractional order control

Analog implementation possibilities of this type of control are scarcely available, most of them based on *fractance* circuits [29]. Continuous controller implementation requires electronic components that are mostly suitable for integer order transfer functions. Some works such as [30–35] tackle analog fractional devices using

electronic systems. However, these devices are difficult to tune and also provide some restrictions.

An alternative method to implement a fractional order controller is to approximate it using finite-dimensional integer-order transfer functions. The aim of this approach is to capture the effects of fractional order operators through integer order transfer functions, both in the continuous and discrete-time domains. The focus of this paper lies on the discrete-time representation of the fractional operator. A manifold of discrete-time approximations have been studied among the years. The main advances can be classified into indirect and direct discrete-time mappers [36–39].

Indirect discrete-time fractional order approximations are methods that describe the dynamics of a fractional operator through high-order integer transfer functions in the Laplace domain [7,23,40,41]. The fractional dynamics are captured through the usage of high order transfer functions, that are further discretized using already available integer order discrete-time mappers. A wide variety of approximation methods are available, providing reliability through certain aspects, such as Continued Fraction Approximations, Oustaloup Filter Approximation, Modified Oustaloup Filter, etc. Since the Laplace representation of the fractional-order operator s^α is a straight line in both magnitude and phase plots, finding a finite order filter that fits the straight line is impossible. However, it is possible to approximate over a frequency range of interest (ω_b, ω_h) [42].

Direct discrete-time mappers use a single step approximation to map the fractional differintegral operation directly to the discrete-time domain [43]. Available studies introduce different approaches such as frequency fitting of the discrete-time model to the response of the fractional order term [44,45] or recursion based formulas [46]. Two direct discrete-time mappers relevant for the present study are introduced in [46], both of them based on the Tustin formula as the generating function with a sampling time T_s

$$s^\alpha \approx \left(\frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}\right)^\alpha. \tag{6}$$

The first mapper is based on the Muir recursion scheme. For a positive order of differintegration, $\alpha > 0$, the formula is introduced in [46] as

$$s^\alpha = \left(\frac{2}{T_s}\right)^\alpha \left(\frac{1-z^{-1}}{1+z^{-1}}\right)^\alpha = \left(\frac{2}{T_s}\right)^\alpha \lim_{n\to\infty} \frac{A_n(z^{-1},\alpha)}{A_n(z^{-1},-\alpha)}, \tag{7}$$

where $A_n(z^{-1},\alpha)$ is a polynomial computed as

$$\begin{aligned} A_0(z^{-1},\alpha) &= 1, \\ A_n(z^{-1},\alpha) &= A_{n-1}(z^{-1},\alpha) - c^n z^n A_n(z^{-1},\alpha) \end{aligned} \tag{8}$$

where $c_n = \frac{\alpha}{n}$ if n is odd, or $c_n = 0$ if n is even. Eq. (8) computes the fifth order polynomial $A_5(z^{-1},\alpha)$ as

$$\begin{aligned} A_5(z^{-1},\alpha) &= -\tfrac{1}{5}\alpha z^{-5} + \tfrac{1}{5}\alpha^2 z^{-4} - (\tfrac{1}{3}\alpha + \tfrac{1}{15}\alpha^3)z^3 + \\ &+ \tfrac{2}{5}\alpha^2 z^{-2} - \alpha z^{-1} + 1. \end{aligned} \tag{9}$$

The second direct mapper of interest throughout this study is the Continued Fraction Expansion (CFE) of the Tustin formula developed by [46]

$$s^\alpha = \left(\frac{2}{T_s}\right)^\alpha CFE\left\{\left(\frac{1-z^{-1}}{1+z^{-1}}\right)^\alpha\right\}_{p,q} = \left(\frac{2}{T_s}\right)^\alpha \frac{P_p(z^{-1})}{Q_q(z^{-1})}, \tag{10}$$

where *CFE* denotes the function generated by applying the Continued Fraction Expansion, p and q are the orders of approximation, while $P_p(z^{-1})$ and $Q_q(z^{-1})$ are polynomials in the z^{-1} variable. For $p = q = 5$, the fifth order polynomials needed by the CFE approximation are computed as

$$\begin{aligned} P_5(z^{-1}) &= (-\alpha^5 + 20\alpha^3 - 64\alpha)z^{-5} + (-195\alpha^2 + 15\alpha^4 + 225)z^{-4} + \\ &+ (-105\alpha^3 + 735\alpha)z^{-3} + (420\alpha^2 - 1050)z^{-2} - 945\alpha z^{-1} + 945, \\ Q_5(z^{-1}) &= (\alpha^5 - 20\alpha^3 + 64\alpha)z^{-5} + (-195\alpha^2 + 15\alpha^4 + 225)z^{-4} + \\ &+ (105\alpha^3 - 735\alpha)z^{-3} + (420\alpha^2 - 1050)z^{-2} + 945\alpha z^{-1} + 945. \end{aligned} \tag{11}$$

The fifth order discrete time approximations from (7) and (10) based on the integer order Tustin formula should be sufficient for most applications. These approximations will be used further to directly map the fractional order s^α into the discrete-time domain with the purpose of creating the event-based fractional order control theory.

Event-based control

Context awareness

The fourth industrial revolution, known as Industry 4.0, integrates the latest scientific advances of the 21st century into the architecture of traditional manufacturing plants. State of the art smart manufacturing systems and technologies take the industrial setting a step further into the future. Human intervention is diminished by improved automation technologies featuring state machines, self-monitoring and abilities to perform machine to machine communication. The emerging Internet of Things (IOT) protocol is the principal constituent in the smart manufacturing trend, creating connected networks of distributed systems [47].

The new industrial cloud architectures are equipped with a wide range of sensors and actuators creating a widespread network that gathers and shares data. Decisions are autonomously taken by interpreting the available information [48]. However, an overflow of unnecessary data leads to an overburden of the bandwidth and wireless network sharing. Introducing the context awareness modules into the smart manufacturing process enables the interpretation and filtering of the gathered information as well as performing decisions regarding its relevance [49]. The main idea is to process data in the lower levels of the system and forwarding the contextual information to the upper layers of the architecture. There are four main layers of the context management, as presented in Fig. 1.

Each objective of the system is analyzed and a context map is creating based on regression data. The context map determines various sets of input and output pairs that are particularly useful in reaching the objective (with green), pairs that are layabout for reaching the goal (with orange) or the pairs that hinder the main objective (with red). The next step involves an analysis of the process with respect to the previously defined sets and identifying events. These events will ultimately lead to a new context, with different pairs than the previous, by triggered the computation of a new set of control parameters in order to reach the main objective.

From the control engineering point of view, the natural solution of the context awareness paradigm lies in the event-based control methodology. The system's context is mapped based on some predefined conditions for reaching a certain objective, specific to the controlled process.

Even if traditional sampled data control, known as Riemann sampling, is the standard tool for implementing computer control, there are severe difficulties when dealing with systems having multiple sampling rates or systems with distributed computing [50]. With multi-rate sampling, the complexity of the system depends critically on the ratio of the sampling rates. For distributed systems, the theory requires that the clocks are synchronized. However, in networked distributed systems, current research revolves around solving problems such as sampling jitter, lost samples and delays on computer controlled systems [51]. Event based sampling, also known as Lebesgue sampling, is a context aware

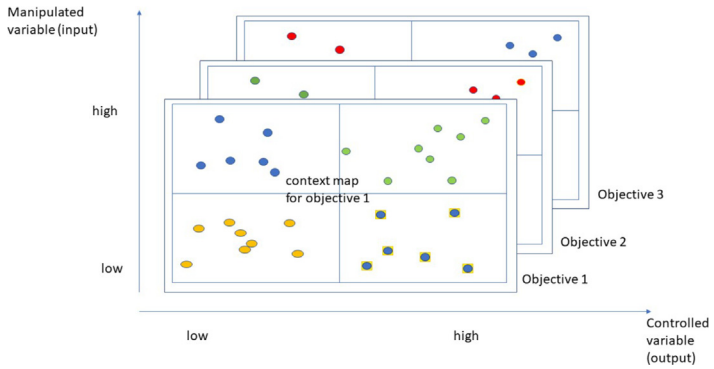


Fig. 1. Context awareness management.

alternative to periodic sampling [52]. Signals are then sampled only when significant events occurs, such as when a measured signal exceeds a limit or when an encoder signal changes. Event based control has many conceptual advantages. Control is not executed unless it is required, control by exception. For a camera based sensor it could be natural to read off the signal when sufficient exposure is obtained. Event based control is also useful in situations when control actions are expensive, for example when controlling manufacturing complexes, and when it is expensive to acquire information like in computer networks [53].

A severe drawback with event based systems is that very little theory is available. All sampled systems, periodic as well as event based, share a common property that the feedback is intermittent and that control is open loop between the samples. After an event, the control signal is generated in an open loop manner and applied to the process. In traditional sampled-data theory the control signal is simply kept constant between the sampling instants, a scheme that is commonly called a zero order hold (ZOH). In event based systems, the generation of the open loop signal is an important issue and the properties of the closed loop system depends critically on how the signal is generated [53]. There has not been much development of theory for systems with event based control, compared to the widely available strategies developed for periodic control.

Mathematical analysis related to the efficiency of the Lebesgue sampling algorithms has been realized by [53]. The study compares the performance of Riemann sampling to event based sampling and shows that the latter gives a better performance, even if the variable sampling theory is still in its infancy.

Industrial benefits

Successful industrial event based applications have been reported during recent years in works such as [14,54,55] spanning on a manifold of industrial processes from the simple tank level control [56] to microalgae cultures in industrial reactors [57,58].

Event-based or event-driven control is a strategy based on variable sampling of the control action. The concept is best explained in contrast to the classical, non-event based, discrete-time implementation. The latter implies the realization of the discrete-time controller using a control signal computed every T_s seconds, without exceptions. For step references, the control signal varies while the system is in the transient regime, afterwards the control variable is constant for most processes. The only exception to this is the presence of unwanted disturbances, when the control action tries to bring the system back to the steady-state regime. Hence, the control variable changes during the transient regime until the process variable reaches the desired reference or when rejecting disturbances, while for the other time slots it is constant. However, discrete-time control implementations compute a new value for the control signal every T_s seconds. It is obvious that the approach is inefficient and that the control effort can be reduced to the periods of time when the process variable is outside the steady-state boundaries. The efficiency problem is solved in an elegant manner by the introduction of event-based control approaches that allow the computation of a new control signal with variable sampling, removing the stress on the control circuit for unnecessary computations.

Event-based control introduces multiple benefits such as CPU resource optimization by limiting the number of times the control signal is computed. This allows the sharing of the CPU with other processes in distributed architectures and also improves energy efficiency in a control scheme [14,59].

Another trend in industrial control [19] is industrial cloud computing. The concept externalizes the process computer to a cloud-based implementation. For complex control necessities, the control signal can be computed into the cloud by a powerful computer, shared with other processes. This approach reduces the cost and removes the necessity of powerful process computers. Event-based strategies are useful in this scenario through the reduced bandwidth allocation [19,60].

Basic principles

Fig. 2. presents a schematic of the event-based implementation concept.

The process data is acquired with the sampling time h_{nom}, which is a constant value, chosen in a similar fashion as the sampling period in classical discrete-time implementations. The data is transferred into the *event detector*, whose sole purpose is to determine whether to trigger an event. The event detector can be customized with any rule that optimizes the control process [18,61].

One of the most popular event triggering rule is focused on the error variation between a predefined interval $[-\Delta_e \Delta_e]$

$$| e(t) - e(t - h_{act}) | \geqslant \Delta_e, \tag{12}$$

where h_{act} is the elapsed time since the triggering of the previous event, $e(t - h_{act})$ is the error at the previous event triggering moment and $e(t)$ is the error at the current moment. Note that the error signal at the present moment t is computed using $e(t) = Y(t) - Y_{sp}(t)$, where $Y(t)$ is the current process variable and $Y_{sp}(t)$ is the setpoint value.

Another popular event detection condition is the maximum allowed time between two consecutive events, known throughout literature as the safety condition. The maximum time is denoted by h_{max}, while the elapsed time since the previous event is h_{act} [62]. The safety condition can be expressed as

$$h_{act} \geqslant h_{max}. \tag{13}$$

The triggering of an event employs the *control input generator* to compute a new value for the control signal. The control input gener-

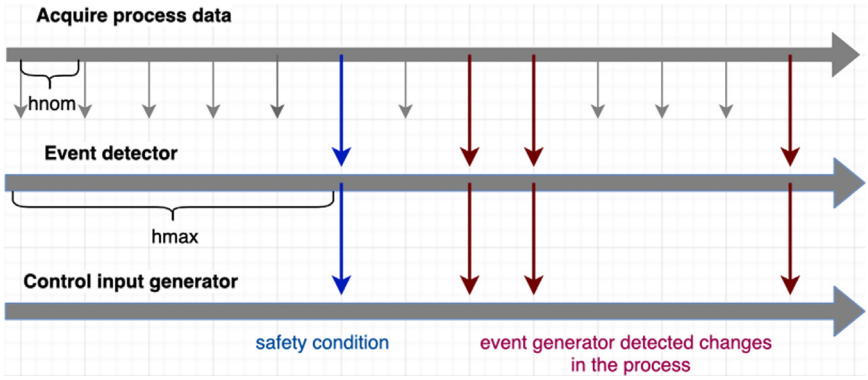


Fig. 2. Event based implementation principles.

ator encapsulates the chosen method to solve the control task. Since the action of the event-based implementation is triggered with a variable sampling h_{act}, the controller should be a discrete-time representation and the sampling time parameter for the discretization must be a variable parameter in the event-based implementation.

The variable sampling time is the challenge in extending the event-based control strategy to the fractional domain.

Event-based integer order control

A brief exemplification of the discrete-time model of the controller inside the control input generator is realized for the integer order PID controller. The transfer function is provided in (5) for the particularity $\lambda = \mu = 1$. The PID control signal in the Laplace domain can be computed using

$$U(s) = k_p\left(\beta Y_{sp}(s) - Y(s)\right) + \frac{k_p k_i}{s}\left(Y_{sp}(s) - Y(s)\right) + k_p k_d s\left(\gamma Y_{sp}(s) - Y(s)\right), \tag{14}$$

where β and γ represent weighting factors of the setpoint value for the proportional and derivative actions.

Denoting $e(t) = Y_{sp}(t) - Y(t), e_\beta(t) = \beta Y_{sp}(t) - Y(t)$ and $e_\gamma(t) = \gamma Y_{sp}(t) - Y(t)$ and splitting the control law into three individual signals: proportional $u_p(s)$, integral $u_i(s)$ and derivative $u_d(s)$ gives

$$\begin{aligned} u_p(s) &= k_p e_\beta(s), \\ u_i(s) &= \tfrac{k_i}{s} e(s), \\ u_d(s) &= k_p k_d s e_\gamma(s). \end{aligned} \tag{15}$$

The discrete-time PID control law can be computed as the sum of the three individual signals into their discrete-time approximation. Using the Tustin method from (6) where $\alpha = 1$ and replacing $s = \frac{2}{h_{act}}\frac{1-z^{-1}}{1+z^{-1}}$ in the individual components from (15) gives

$$\begin{aligned} u_p(k) &= k_p e_\beta(k), \\ u_i(k) &= h_{act}\tfrac{k_i}{2}\left(e(k) + e(k-1)\right) + u_i(k-1), \\ u_d(k) &= \tfrac{1}{h_{act}} 2k_d\left(e_\gamma(k) - e_\gamma(k-1)\right) - u_d(k-1). \end{aligned} \tag{16}$$

where $u_i(k-1)$ is the previously computed integral component and $u_d(k-1)$ is the previous derivative component. A previous computation refers to the moment of the last event. Note that the discrete sampling time, T_s, has been replaced by h_{act} which is the variable sampling time.

Eq. (16) gives the control law of the PID controller with variable sampling time. Following this model, other controllers such as the integer order IMC can be discretized and implemented inside the control input generator, as long as the sampling time is kept as a variable. In addition, the discretization method is not limited to Tustin, any viable discrete-time mapper can be used.

Event-based fractional order control

Proposed implementation methodology

The theoretical contribution of the present study is the generalization of the available event-based control strategy into the fractional order field. The challenge lies in the isolation of the sampling time and introducing its variability in the fractional order discrete-time approximation. There is not any specialized literature available that explores the possibilities of implementing fractional order controllers in the control input generator.

The development of the discrete fractional order control actions is based on fractional order direct discrete-time mappers. For an in depth analysis, both the Muir recursive and CFE discrete-time approximations from (7) and (10) will be used to compute discrete-time equivalent controllers. The aim in studying both discrete time mappers is to prove that any direct mapper whose sampling time can be isolated from the rest of the function can be successfully used in fractional order event-based implementations.

The event-based implementation of the fractional order PID controller from (5) with $\lambda, \mu \in (0, 1)$ will be exemplified further. Following this model, any fractional order controller can be discretized and implemented in the fractional order control signal generator.

The fractional order control law is written as a sum of the three individual components as $U_f(s) = u_{fp}(s) + u_{fi}(s) + u_{fd}(s)$, with $u_{fp}(s), u_{fi}(s)$ and $u_{fd}(s)$ being the proportional, integer and derivative fractional order control signals. Hence, generalizing (15) gives

$$\begin{aligned} u_{fp}(s) &= k_p e_\beta(s), \\ u_{fi}(s) &= \tfrac{k_p k_i}{s^\lambda} e(s), \\ u_{fd}(s) &= k_p k_d s^\mu e_\gamma(s). \end{aligned} \tag{17}$$

Both direct discrete-time approximations based on the Muir recursion or on the Continued Fraction Expansion approximate the fractional order term s^α to a transfer function of the form

$$s^\alpha \approx \left(\frac{2}{T_s}\right)^\alpha \frac{a_n z^{-n} + a_{n-1} z^{-n+1} + \ldots + a_0}{b_n z^{-n} + b_{n-1} z^{-n+1} + \ldots + b_0} = \left(\frac{2}{T_s}\right)^\alpha \frac{N_n(z^{-1})}{D_n(z^{-1})}. \tag{18}$$

For the Muir recursion mapper of s^α the polynomials are $N_n(z^{-1}) = A_n(z^{-1}, \alpha)$ and $D_n(z^{-1}) = A_n(z^{-1}, -\alpha)$, while for the CFE mapper the polynomials are $N_n(z^{-1}) = P_n(\alpha)$ and $D_n(z^{-1}) = Q_n(\alpha)$. The sampling time has been isolated from the rest of the model and will be treated as a variable in the event-based implementation. The $N_n(z^{-1})$ and $D_n(z^{-1})$ can be computed only once and will remain the same for every iteration through the control signal generator.

Furthermore, the fractional order terms will be treated separately and approximated to the discrete-time domain using the generalized formula from (18). The fractional order proportional

component is identical to the integer-order PID proportional term since it does not introduce any fractional order operation

$$u_{fp}(k) = k_p e_\beta(k). \tag{19}$$

The fractional order integral term can be rewritten $\frac{1}{s^\lambda} = \frac{s^{1-\lambda}}{s}$ such that the direct discretization of s^α respects the $\alpha > 0$ requirement. Hence, the discrete-time transfer function of the fractional order integral component is computed using the Tustin formula for the $1/s$ term and a direct n^{th} discrete-time formula for the $s^{1-\lambda}$ term, where $1-\lambda > 0, \lambda \in (0,1)$

$$\frac{1}{s^\lambda} = \frac{1}{s} s^{1-\lambda} = \frac{T_s}{2}\frac{1+z^{-1}}{1-z^{-1}}\left(\frac{2}{T_s}\right)^{1-\lambda}\frac{N_n(z^{-1})}{D_n(z^{-1})}. \tag{20}$$

After expanding (20), the recurrence formula of the fractional order integral component is obtained as

$$\begin{aligned} u_{fi}(k) &= \left(\tfrac{2}{h_{act}}\right)^{-\lambda}\tfrac{k_p k_i}{d_0}\mathscr{E}_{fi_n}(k) - \tfrac{1}{d_0}\mathscr{U}_{fi_n}(k), \\ \mathscr{E}_{fi_n}(k) &= n_0 e(k) + (n_0+n_1)e(k-1) + (n_1+n_2)e(k-2) + \\ &\quad + \ldots + (n_{n-1}+n_n)e(k-n) + n_n e(k-(n+1)), \\ \mathscr{U}_{fi_n}(k) &= (d_1-d_0)u_{fi}(k-1) + (d_2-d_1)u_{fi}(k-2) + \\ &\quad + \ldots + (d_n-d_{n-1})u_{fi}(k-n) - d_n u_{fi}(k-(n+1)), \end{aligned} \tag{21}$$

where $e(k-n)$ is the previous n^{th} error signal and $u_{fi_n}(k-n)$ is the previous n^{th} fractional order integral control action.

The fractional order derivative component is obtained by directly replacing $s^\mu, \mu \in (0,1)$ with any direct discrete-time mapper

$$\begin{aligned} u_{fd}(k) &= \left(\tfrac{2}{h_{act}}\right)^{\mu}\tfrac{k_p T_d}{d_0}\mathscr{E}_{fd_n}(k) - \tfrac{1}{d_0}\mathscr{U}_{fd_n}(k), \\ \mathscr{E}_{fd_n}(k) &= n_0 e_\gamma(k) + n_1 e_\gamma(k-1) + \ldots + n_n e_\gamma(k-n), \\ \mathscr{U}_{fd_n}(k) &= d_1 u_{fd}(k-1) + d_2 u_{fd}(k-2) + \ldots + d_n u_{fd}(k-n), \end{aligned} \tag{22}$$

where $e_\gamma(k-n)$ is the previous n^{th} error value whose setpoint value is pondered with the γ weighting factor and $u_{fd}(k-n)$ is the previous n^{th} fractional order derivative control value.

The fractional order PID exemplified in this section stands as a basis to any fractional order event based implementation. For example, for a fractional order IMC implementation, a similar approach can be employed. It is also possible to treat all the effects of the fractional order controller as one, but this is a reliable approach only when dealing with a single fractional-order operator inside the controller.

Another important aspect worth specifying is the fact that the actual tuning of the fractional order controller is a completely independent process from the event-based implementation. Fractional order controllers can be tuned using any available method. The present algorithm completes the fractional order controller's implementation by offering a direct discretization possibility and a fractional order event based implementation. The controller should be tuned accordingly, by imposing any performance specification as for a non-event based implementation [63–65].

Implementation guidelines

In an event-based control implementation, there are also some parameter choices that need to be done, apart from determining the direct discrete-time transfer function of the fractional order operators. The parameters that need to be customized depending upon the particularities of the process to be controlled are the nominal sampling time h_{nom}, the maximum allowed time between two consecutive events h_{max} and the error variation threshold Δ_e. The other parameters of the event-based implementation are purely implementation variables, independent of the process. It is imperative that the parameters are used correctly inside the event detector and the control input generator for an efficient implementation. For an easy and quick reference, Table 1 provides a centralization of the parameters as well as some brief clarifications.

Table 1
Event-based implementation parameters.

Parameter	Type	Description
h_{nom}	custom	nominal sampling time
h_{max}	custom	safety condition trigger time
Δ_e	custom	maximum error variation
β	custom	proportional action setpoint weighting factor
γ	custom	derivative action setpoint weighting factor
h_{act}	variable	elapsed time since the previous event
e_s	variable	previous error value

β and γ are weighting factors denoting the importance of the setpoint value in computing the error amount used to compute the fractional order proportional and derivative actions, respectively. The terms belong to the $(0,1]$ interval. If there are no error weighting requirements of the fractional order control strategy or a simple implementation is desired, both terms should be chosen $\beta = \gamma = 1$.

The nominal sampling time choice h_{nom} is identical to the sampling time in any discrete-time implementation. Shannon's theorem should be respected.

The maximum allowed time between events is denoted through h_{max}, which is a multiple of h_{nom}. This is effective in implementing the safety condition. Choosing $h_{nom} = h_{max}$ in an event-based implementation leads to a pure discrete-time effect, eliminating all the benefits brought by the event-based approach. Hence, h_{max} should always be greater than h_{nom}. The greater the value, the better the control signal computation is optimized, especially in the steady state regime. However, special attention should be given to choice of this parameter such that the safety condition is not completely eliminated. As a starting point, one should base the h_{max} choice on the dominant time constants of the process. Trial and error is also a viable option, as well as this possibility exists.

The previous error value is denoted by e_s in the available event-based literature [62]. Note that this error cannot include any weighting factor, in comparison to the error values used to compute the fractional order proportional and derivative gains respectively. The e_s variable stores the error at the previous event. Note that this value should be updated for every iteration of the control input generator. The value of e_s is used inside the event detector to check if an event should be triggered due to the error variation. The condition is usually written as $(e(k) - e_s) \in [-\Delta_e, \Delta_e]$. The $[-\Delta_e, \Delta_e]$ interval is composed by a predefined maximum error threshold. As a guideline, Δ_e should be chosen as a small percentage, usually between 1–5% to the desired setpoint value.

The parameter h_{act} is a variable that stores the actual time elapsed since the previous event. The value should be updated every h_{nom} seconds whether an event is triggered or not. The formula that increments h_{act} can be written as $h_{act} = h_{act} + h_{nom}$, making h_{act} another multiple of h_{nom}. Note that h_{act} should be reset to 0 at the triggering of an event. The purpose of h_{act} is to determine if the safety condition is met.

Numerical examples

The proposed event-based fractional order strategy is validated using numerical simulations targeting a wide range of processes from the simple first order plus dead time to a complex fractional order transfer function. The strategy is validated using fractional order PID controller case study. The case studies are intended to show the fractional order control versatility of the proposed methodology as well as to analyze its efficacy for a class of relevant processes.

All the controllers used in the numerical examples have been previously tuned using different fractional order methodologies.

The tuning procedure of each controller is irrelevant in proving the efficacy of the event based fractional order implementation. Both the Muir and CFE discrete-time approximations are tested with identical event-based implementation parameters in order to show the validity of any direct mapper for the present strategy and to asses the better discrete-time mapper choice for every example. Three implementation scenarios are performed and analyzed for every numerical example. The first test case analyzes the performance of the closed loop system obtained with the fractional order controller for a pure discrete-time implementation with the Muir recurrence-based direct discrete-time mapper from (7) and the CFE direct discrete-time mapper from (10). The test scenario validates the discrete-time fractional order controller on the selected process. The second and third test cases validate the proposed event-based fractional order control strategy, focusing on comparing the pure discrete-time realization of the fractional order controller to its event-based implementation.

First order plus dead time

For the first order plus dead time (FOPDT) model from

$$H(s) = \frac{1}{4s+1}e^{-s} \tag{23}$$

a fractional order PI controller is computed by imposing a set of frequency domain specifications targeting the gain crossover frequency $w_{gc} = 0.2$ rad/s, the phase margin $\phi_m = 44^\circ$ and a certain degree of robustness to gain variations. The controller that meets these frequency domain constraints is

$$C(s) = 0.0021\left(1 + \frac{162.6923}{s^{0.833}}\right). \tag{24}$$

The Muir and CFE 5th order discrete-time mappers from (9) and (11) are used to compute the discrete-time approximation of the fractional order PI controller. The same sampling time $T_s = 0.001$ is used for both discretizations. The pure discrete-time implementation of the closed loop with every discrete-time controller is presented in Fig. 3a. The Muir and CFE approximations lead to a similar closed-loop system response.

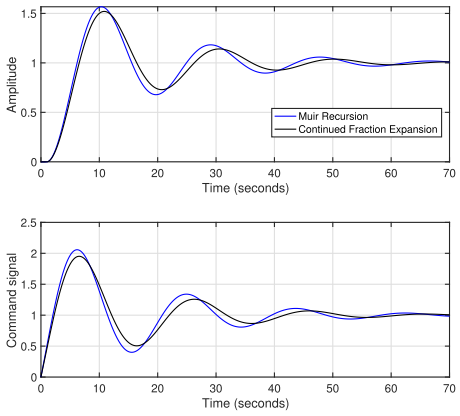
The event-based fractional order PI implementation based on the Muir scheme is shown in Fig. 3b, while the CFE approximation is given in Fig. 3c. The implementation parameters are $\beta = \gamma = 1$ for simplicity. The nominal sampling time is chosen such as $h_{nom} = T_s = 0.001$ s and the safety condition time as $h_{max} = 0.02$ s, 20 times larger than h_{nom}, while the error threshold is $\Delta_e = 0.0005$. The parameters are identical for both the Muir and CFE implementations. The simulations clearly show the cause of the control signal computations. It can be seen that in the transient response, the error triggered control prevails, whereas the safety condition is employed mostly in the steady-state regime. Both event-based fractional order implementations can be successfully used on the FOPDT process. However, during the 70 s duration of the tests, the Muir based controller computes the control signal 5824 times, the CFE controller 6508 times, improving the discrete-time control strategy which computes the control signal 70000 times.

Second order transfer function

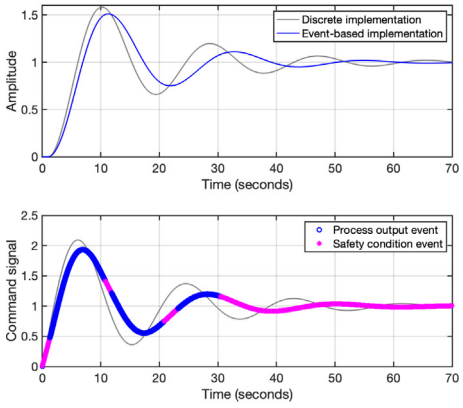
The second order transfer function (SOTF) chosen as the second numerical example

$$H(s) = \frac{78.35}{s^2 + 1.221s + 822} \tag{25}$$

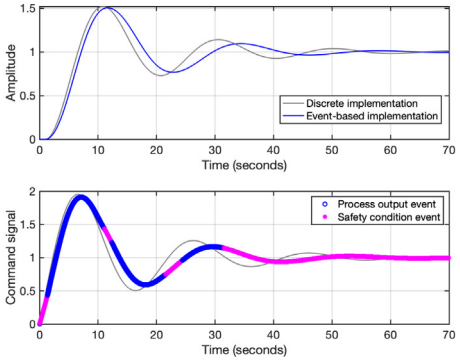
represents the dynamics of a smart beam. For the SOTF process a fractional order PID controller is tuned in [66] using an experimental optimization procedure. The obtained controller is



(a) Discrete-time controller validation on the FOPDT process for a unit step reference



(b) Event-based fractional order PI implementation based on the Muir recursion



(c) Event-based fractional order PI implementation based on the CFE recursion

Fig. 3. A fractional order PI controller applied to a FOPDT process.

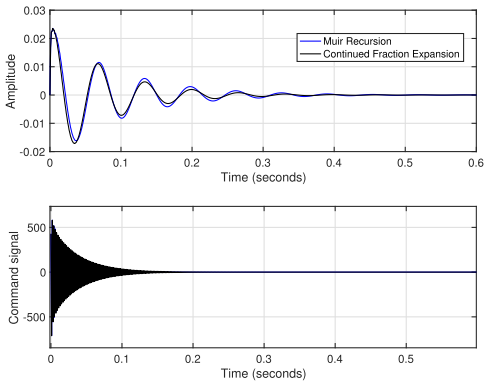
$$C(s) = 0.0288\left(1 + \frac{3.4722}{s^{0.1039}} + 28.743s^{0.822}\right). \tag{26}$$

Fig. 4a shows the impulse response of the SOTF process with the classical discrete-time implementation of the fractional order PID controller with $T_s = 0.001$ s. The dynamics of the output variable are similar for both approaches.

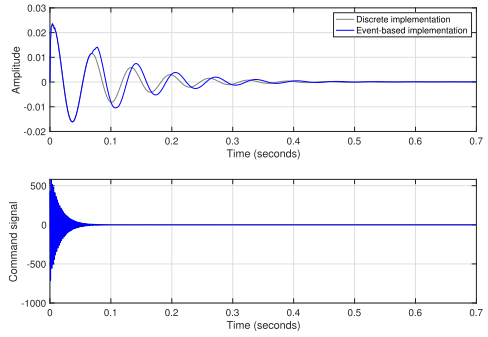
Fig. 4b shows the event-based fractional order PID controller discretized using the Muir recursion formula. The output of the closed loop system with the event-based controller is similar to the non-event-based Muir controller. The implementation parame-

ters are chosen as follows: $\beta = \gamma = 1, h_{nom} = T_s = 0.001$ s, $h_{max} = 0.005$ s - 5 times greater than h_{nom} and $\Delta_e = 10^{-4}$. The error threshold is chosen based on the amplitude of the closed-loop system response from Fig. 4a. The CFE event-based implementation is depicted in Fig. 4c. The process stabilizes with the same settling time as the Muir scenario. However, it can be observed that the amplitude of the output is 10 times larger using the CFE approximation, for the same event-based implementation parameters as in the Muir case.

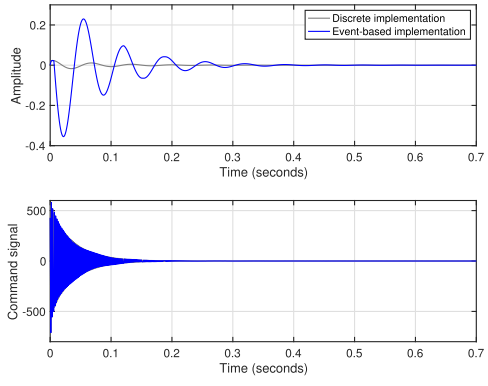
From the control effort perspective, the Muir mapper computes the control signal value 368 times, the CFE controller 494 times, while the classical discrete-time controller command is evaluated 700 times for the 0.7 s in which the tests were performed. It can be stated that for this numerical example, the Muir based controller is the better choice both from the performance and the control optimization criteria.



(a) Impulse response of the closed-loop SOTF system with the discrete-time controller implementation



(b) Event-based fractional order PID implementation based on the Muir recursion



(c) Event-based fractional order PID implementation based on the CFE recursion

Fig. 4. A fractional order PID controller applied to a SOTF process.

Second order plus time delay

A Vertical Take-Off and Landing Platform is modeled through the second order plus time delay (SOPTD) transfer function

$$H(s) = \frac{22.24}{s^2 + 0.6934s + 5.244} e^{-0.8s}. \tag{27}$$

The same tuning methodology as for the FOPDT numerical example is employed which tunes a robust fractional order PI controller that offers an open-loop system with a gain crossover frequency $\omega_{gc} = 0.4$ rad/s and a phase margin $\phi_m = 75°$

$$C(s) = 0.0422\left(1 + \frac{492.2867}{s^{0.9288}}\right). \tag{28}$$

Additional details regarding the real-life process and the control strategy can be found in [67].

Fig. 5a shows the closed-loop system response with the discrete-time Muir mapper and the CFE direct approximation using the sampling time $T_s = 0.005$ s. It can be seen that the dynamics of the closed-loop system are similar for the classical implementation of the controllers.

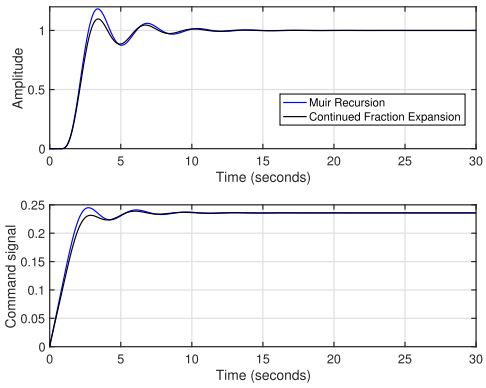
The implementation particularities of the event-based fractional order controller are $\beta = \gamma = 1$ for simplicity, $h_{nom} = T_s = 0.005$ s, $h_{max} = 0.2$ s - 20 times larger than the nominal value and $\Delta_e = 0.01$, representing 1% of the unit step reference value.

The closed-loop response of the event-based fractional order PI controller approximated using the 5th order Muir recursion from (9) is displayed in Fig. 5b. In the command signal plot it can be seen the different types of events triggered to compute the control value. The error threshold events are more prominent during the transient response, while the safety condition triggers the control value computation mostly in the steady-state regime. As can be seen in the figure, the event-based implementation obtains similar closed-loop system performance as the classical implementation.

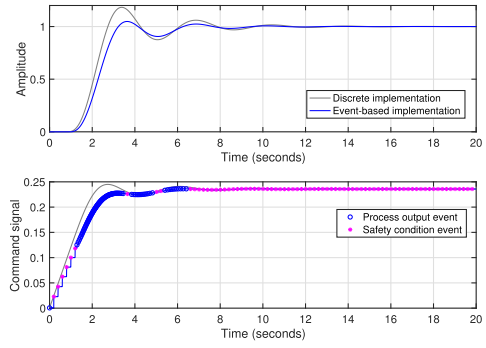
Furthermore, Fig. 5c shows the closed loop response of the event-based controller with the CFE approximation. The system reaches its steady-state value with the same sampling time as in the Muir approximation from Fig. 5b. However, the amplitude is increased from 1 to approx. 1.5. Since the dynamics of the system with the event-based controller differ from the classical approximation, an in-depth investigation is realized in Fig. 6 that analyzes the effects of the event-based parameters on the system's response. More restrictive parameters are imposed in order to verify if the event-based strategy is able to obtain a similar result as the one obtained with the Muir recursion. The implementation from Fig. 5c uses a h_{max} value which is 20 times greater than the nominal sampling time. The parameter h_{max} is varied to be 10 times and 5 times greater than $h_{nom} = 0.005$ s and the results are displayed in Fig. 6. It can be observed that lowering the h_{max} value leads to a better closed loop system response. For the current case study, it can be stated that the Muir approach provides better closed-loop system performance than the CFE implementation, for the same fractional order event-based parameters.

The validation of the control strategy developed in [67] for the SOPDT experimental platform involves the experimental endorsement of the fractional order PI controller for reference tracking, disturbance rejection and robustness. In order to validate the proposed Muir event-based control strategy for the real-life test scenarios, disturbance rejection and robustness assessment are also taken into consideration.

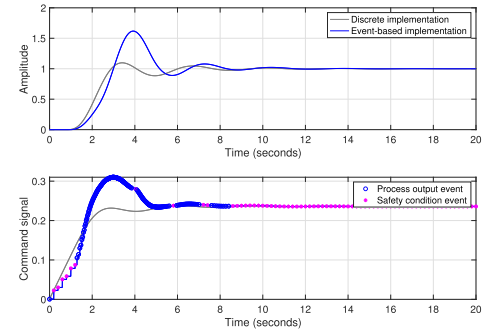
Fig. 7 presents the closed loop response of the experimental platform equipped with the event-based fractional order PI controller. A 0.2 disturbance, representing 20% from the steady state



(a) Discrete-time controller validation on the SOPTD process for a unit step reference



(b) Event-based fractional order PI implementation based on the Muir recursion



(c) Event-based fractional order PI implementation based on the CFE recursion

Fig. 5. A fractional order PI controller applied to a SOPTD process.

value, is introduced at moment t = 20 s. The event-based implementation parameters $\beta = \gamma = 1, h_{nom} = T_s = 0.005$ s, $h_{max} = 0.2$ s and $\Delta_e = 0.01$ are the same as the ones used in Fig. 5. It can be seen that the output converges to the desired set point value, rejecting the disturbance. The event detector generates mostly process output events during the transient regime, which is to be expected from the event based implementation.

Fig. 8 targets the validation of the event based fractional order controller in a robustness test scenario. The fractional order controller from (28) has been tuned in [67] with respect to the iso-damping condition, ensuring a certain degree of robustness of the closed loop system to gain variations. Hence, the gain of the SOPDT process from (27) is altered by 30% and the response of the closed loop system to a unit step reference is analyzed. The test is identical to the one used to generate Fig. 5b, featuring the same

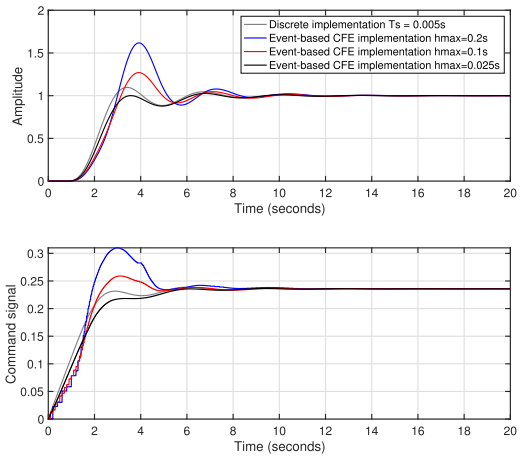


Fig. 6. Closed-loop system performance analysis for varying parameters of the event-based CFE implementation.

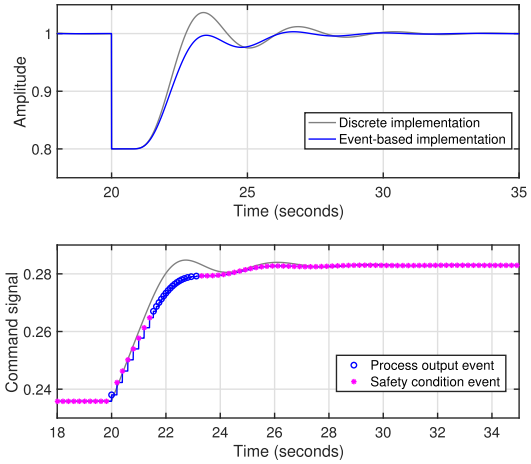


Fig. 7. Closed-loop system performance analysis for disturbance rejection of the event-based Muir implementation.

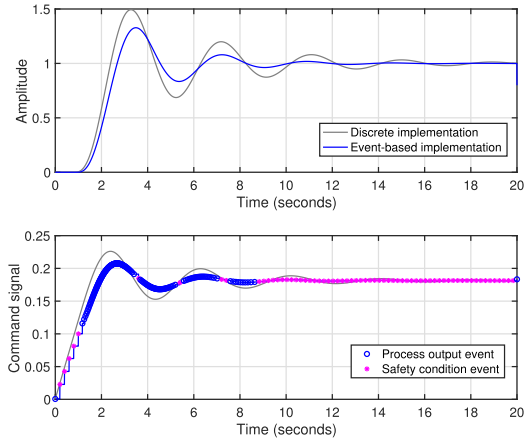


Fig. 8. Closed-loop system performance analysis for robustness of the event-based Muir implementation.

event based implementation parameters. Comparing the nominal and the altered process responses, it can be easily observed that the settling time is increased from 6 to 10 s, as well as the overshoot soaring from approx. 10% to 40%. However, the event-based fractional order PI controller has a robust behavior, despite the large process' gain modification.

The tests performed in Figs. 7 and 8 show that the event based fractional order control strategy keeps the closed loop characteristics obtained through the discrete time implementation, such as disturbance rejection capabilities and robustness to gain variations. It is important to emphasize that an event based fractional order implementation does not aid in introducing a robust character of the closed loop system, being truthful to the already existing features of the controller implemented in the control input generator.

Fractional order transfer function

The last case study is focused on proving the applicability of the proposed event-based strategy to complex fractional order (FOTF) models. The transfer function

$$H(s) = \frac{0.1}{s(0.005682s^{1.7263} + 0.11031s^{0.8682} + 1)} \tag{29}$$

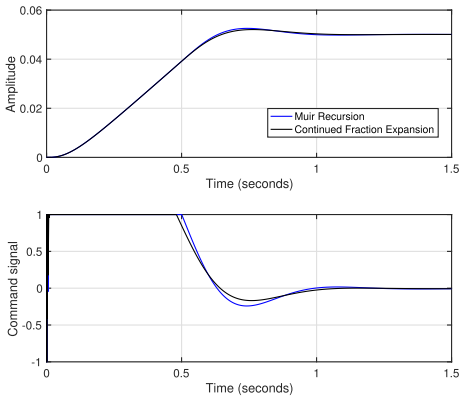
describes the braking effect of a non-Newtonian fluid, such as blood, on a transiting submersible. The main applicability of the study from [68], where the model and control have been developed, is in targeted drug delivery biomedical applications. One of the main request of the physical submersible is energy efficiency. This particular example is highly relevant to the proposed event-based fractional order methodology in order to show that the method can be useful to reduce computational effort in complex applications where energy efficiency is paramount. The control of the submersible's position is done using the fractional order PD controller

$$C(s) = 65.0028\left(1 + 0.0305s^{0.6524}\right). \tag{30}$$

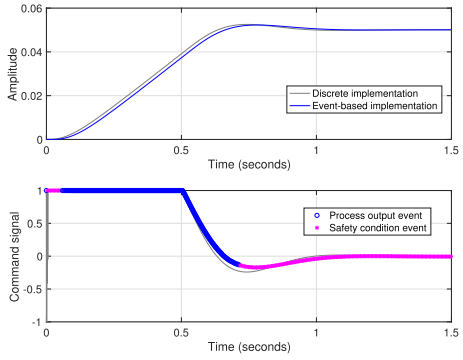
Fig. 9a shows the response of the system with the classical discrete-time implementations using 5th order Muir compared to the CFE direct mapper with $T_s = 0.001$ s. In order to be true to the real process, the control signal that represents a PWM duty ratio is saturated between $[0, 1]$. Hence, the event-based fractional order control strategy is tested on another real life situation where the control signal range is limited. It can be seen that the dynamics of the closed-loop system are similar for the three discrete-time implementations.

The implementation parameters for the event-based Muir and CFE controllers are $\beta = \gamma = 1, h_{nom} = T_s = 0.001$ s, $h_{max} = T_s = 0.005$ s - 5 times greater than h_{nom} and $\Delta_e = 0.01$ representing 1% of the desired reference value. The closed-loop system responses using the event-based fractional order controllers with the Muir and CFE mappers are shown in Fig. 9b and c, respectively. Both controllers obtain similar responses as their non-event-based implementation. However, it can be clearly observed that the Muir realization of the discrete-time fractional order PD controller is the better choice.

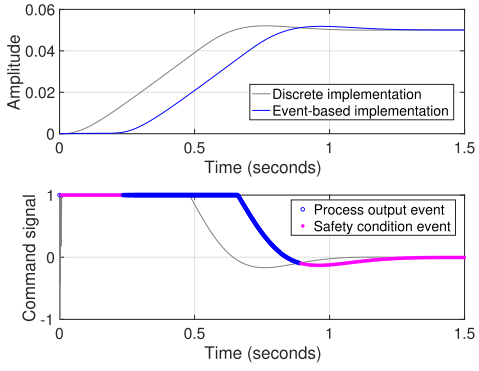
From the control effort perspective, the Muir approximation computed the command value 380 times, whereas the CFE approach computed it 473 times. Both implementations reduce the 1500 number of computations realized with the non-event-based controller, bringing an improvement of more than 70% to the control effort.



(a) Discrete-time controller validation on the FOTF process for a unit step reference



(b) Event-based fractional order PD implementation based on the Muir recursion



(c) Event-based fractional order PD implementation based on the CFE recursion

Fig. 9. A fractional order PD controller applied to a FOTF process.

Conclusion

The paper presents the theoretical background and the validation of a novel control strategy that combines the benefits of complex fractional order controllers with event-based implementations. The method is based on direct discrete-time mappers of fractional order operators. The theoretical aspects are presented for the fractional order PID controller, but any fractional order approach can be adapted given the provided guidelines. The study is validated using four numerical examples targeting various types of plants and all the fractional order PID controller variations. The methodology has been successfully validated on every case study using various direct discrete-time mappers, showing the implementation versatility of the entire concept. For all four case studies, both the Muir and CFE direct mappers obtained a similar closed-loop performance as classical discrete-time implementations with a reduction of the control effort by at least 70% in every scenario. For all the chosen test cases, the Muir recursion based approximation provided a better closed-loop system response than the CFE approach.

The proposed methodology is most beneficial in industrial settings, where resource optimization and energy efficiency can drastically reduce the costs of various plants, as well as component wear over time.

Declaration of Competing Interest

The authors have declared no conflict of interest.

Compliance with Ethics Requirements

This article does not contain any studies with human or animal subjects.

Event-based fractional order control implementation

Algorithm 1. The event detector

Result: The control signal u
Data: k_p, k_i, k_d, λ, μ, h_{nom}, h_{act}, Δ_e, Y_{sp}
$h_{act} = 0$
$e_s = Y_{sp}$

while *simulation ongoing* **do**
  read process value Y
  e = Y_{sp} - Y

  if *($|e - e_s| > \Delta_e$ or $h_{act} \geq h_{max}$)* **then**
    u = `control_input_generator`(Y_{sp}, Y) $h_{act} = 0$
    $e_s = e$
end

Algorithm 2. The fractional order control input generator

Result: The control signal u
Data: $ni_0, ni_1, ..., ni_n, di_0, di_1, ..., di_n, nd_0, nd_1, ..., nd_n, dd_0, dd_1, ..., di_n$
$e = Y_{sp} - Y$
$e_\beta = \beta Y_{sp} - Y$
$e_\gamma = \gamma Y_{sp} - Y$

`/* compute the fractional order proportional component */`
$u_{fp} = kp * e$

`/* compute the fractional order integral component */`
$a = k_p * k_i * pow(2/hact, -\lambda)$
$u_{fi} =$
$1/d_0 * (a * (ni_0 * e + (ni_0 + ni_1) * ei_1 + ... + (ni_{n-1} + ni_n) * ei_n + ni_n * ei_{n+1}) -$
$((di_1 - di_0) * ui_1 + (di_2 - di_1) * ui_2 + ... + (di_n - di_{n-1}) * ui_n - di_n * ui_{n+1}))$
$ei_{n+1} = ei_n; ei_n = ei_{n-1}; ...ei_1 = e$
$ui_{n+1} = ui_n; ui_n = ui_{n-1}; ...ui_1 = u_{fi}$

`/* compute the fractional order derivative component */`
$b = kp * kd * pow(2/hact, \mu)$
$u_{fd} = 1/dd_0 * (b * (nd_0 * e_\gamma + nd_1 * ed_1 + ... + nd_n * ed_n) - (dd_1 * ud_1 +$
$dd2 * ud_2 + ...dd_n * ud_n))$
$ed_n = ed_{n-1}; ...ed_1 = e_\gamma$
$ud_n = ud_{n-1}; ...ud_1 = u_{fd}$

`/* compute the entire fractional order control signal */`
$u = u_{fp} + u_{fi} + u_{fd}$

Acknowledgment

This work was supported by the Research Foundation Flanders (FWO) under Grant 1S04719N and by 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.

References

[1] Sun HG, Zhang Y, Baleanu D, Chen W, Chen YQ. A new collection of real world applications of fractional calculus in science and engineering; 2018. https://doi.org/10.1016/j.cnsns.2018.04.019.
[2] Dalir M, Bashour M. Applications of fractional calculus. Appl Math Sci 2010.
[3] Kilbas A, Srivastava H, Trujillo J. Theory and applications of fractional differential equations; 2006.
[4] West BJ. Fractional calculus view of complexity. Tomorrow's Sci 2016.
[5] Hilfer R. Appl Fract Calculus Phys 2000. doi: https://doi.org/10.1142/3779.
[6] Du M, Wang Z, Hu H. Measuring memory with the order of fractional derivative. Scient Rep 2013. doi: https://doi.org/10.1038/srep03431.
[7] Monje CA, Chen Y, Vinagre BM, Xue D, Feliu-Batlle V. Fundamentals of fractional-order systems. In: Fractional-order systems and controls fundamentals and applications; 2010.
[8] Chen YQ, Petráš I, Xue D. Fractional order control - A tutorial. In: Proceedings of the American control conference. doi: https://doi.org/10.1109/ACC.2009.5160719.
[9] Podlubny I. Fractional-order systems and PIλDμ-controllers. IEEE Trans Autom Control 1999. doi: https://doi.org/10.1109/9.739144.
[10] Muresan CI, Dutta A, Dulf EH, Pinar Z, Maxim A, Ionescu CM. Tuning algorithms for fractional order internal model controllers for time delay processes. Int J Control 2016. doi: https://doi.org/10.1080/00207179.2015.1086027.
[11] Amoura K, Mansouri R, Bettayeb M, Al-Saggaf UM. Closed-loop step response for tuning PID-fractional-order-filter controllers. ISA Trans 2016. doi: https://doi.org/10.1016/j.isatra.2016.04.017.
[12] Maâmar B, Rachid M. IMC-PID-fractional-order-filter controllers design for integer order systems. ISA Trans 2014. doi: https://doi.org/10.1016/j.isatra.2014.05.007.
[13] Li Y, Chen Y. Fractional order linear quadratic regulator. In: 2008 IEEE/ASME International conference on mechtronic and embedded systems and applications. p. 363–8.
[14] Grüne L, Hirche S, Junge O, Koltai P, Lehmann D, Lunze J, et al. Event-based control. In: Control theory of digitally networked dynamic systems; 2014. https://doi.org/10.1007/978-3-319-01131-8_5.
[15] Lunze J, Lehmann D. A state-feedback approach to event-based control. Automatica 2010. doi: https://doi.org/10.1016/j.automatica.2009.10.035.
[16] Visioli A. Res Trends PID Controllers 2012. doi: https://doi.org/10.14311/1656.
[17] Sánchez J, Visioli A, Dormido S. An event-based PI controller based on feedback and feedforward actions. IECON Proc (Industr Electron Conf) 2009. doi: https://doi.org/10.1109/IECON.2009.5414719.
[18] Heemels WP, Johansson KH, Tabuada P. An introduction to event-triggered and self-triggered control. In: Proceedings of the IEEE conference on decision and control. doi: https://doi.org/10.1109/CDC.2012.6425820.
[19] Vaidya S, Ambad P, Bhosle S. Industry 4.0 - A Glimpse. In: Procedia manufacturing; 2018. https://doi.org/10.1016/j.promfg.2018.02.034.
[20] Abdelaal A, Hegazy T, Hefeeda M. Event-based control as a cloud service 2017:1017–23. doi: https://doi.org/10.23919/ACC.2017.7963086.
[21] Sweilam NH, Al-Ajami TM. Legendre spectral-collocation method for solving some types of fractional optimal control problems. J Adv Res 2015;6(3): 393–403, editors and International Board Member collection. https://doi.org/10.1016/j.jare.2014.05.004.
[22] Li C, Deng W. Remarks on fractional derivatives. Appl Math Comput 2007. doi: https://doi.org/10.1016/j.amc.2006.08.163.
[23] Li C, Qian D, Chen Y. On Riemann-Liouville and Caputo derivatives. Discr Dynam Nat Soc 2011. doi: https://doi.org/10.1155/2011/562494.
[24] Munkhammar J. Riemann-Liouville fractional derivatives and the Taylor-Riemann series, UUDM project report; 2004.
[25] Ortigueira MD, Tenreiro Machado JA. What is a fractional derivative? J Comput Phys 2015. doi: https://doi.org/10.1016/j.jcp.2014.07.019.
[26] Ortigueira MD, Rodríguez-Germá L, Trujillo JJ. Complex Grünwald-Letnikov, Liouville, Riemann-Liouville, and Caputo derivatives for analytic functions. Commun Nonlinear Sci Numer Simul 2011. doi: https://doi.org/10.1016/j.cnsns.2011.02.022.
[27] Aboelela MA, Ahmed MF, Dorrah HT. Design of aerospace control systems using fractional pid controller. J Adv Res 2012;3(3):225–32. doi: https://doi.org/10.1016/j.jare.2011.07.003.
[28] Radwan A, Moaddy K, Salama K, Momani S, Hashim I. Control and switching synchronization of fractional order chaotic systems using active control technique. J Adv Res 2014;5(1):125–32. doi: https://doi.org/10.1016/j.jare.2013.01.003.
[29] Tepljakov A, Petlenkov E, Belikov J. Efficient analog implementations of fractional-order controllers 2013:377–82. doi: https://doi.org/10.1109/CarpathianCC.2013.6560573.
[30] Gonzalez E, Alimisis V, Psychalinos C, Tepljakov A. Design of a generalized fractional-order pid controller using operational amplifiers 2018:253–6. doi: https://doi.org/10.1109/ICECS.2018.8617954.
[31] Dimeas I, Petras I, Psychalinos C. New analog implementation technique for fractional-order controller: A DC motor control. AEU - Int J Electron Commun 2017. doi: https://doi.org/10.1016/j.aeue.2017.03.010.
[32] Mukhopadhyay S, Coopmans C, Chen YQ. Purely analog fractional order PI control using discrete fractional capacitors (fractors): Synthesis and experiments. In: Proceedings of the ASME design engineering technical conference. doi: https://doi.org/10.1115/DETC2009-87490.
[33] Chen YQ, Xue D, Dou H. Fractional calculus and biomimetic control. In: Proceedings - 2004 IEEE international conference on robotics and biomimetics, IEEE ROBIO 2004; 2004. https://doi.org/10.1109/robio.2004.1521904.
[34] John DA, Aware MV, Junghare AS, Biswas K. Performance analysis of solid-state fractional capacitor-based analog pi$^{\lambda}$ d$^{\mu}$ controller. Circ, Syst, Signal Process 2020;39(4):1815–30. doi: https://doi.org/10.1007/s00034-019-01255-2.
[35] Caponetto R, Dongola G, Pappalardo F, Tomasello V. Auto-tuning and fractional order controller implementation on hardware in the loop system. J Optim Theory Appl 2013;156(1):141–52. doi: https://doi.org/10.1007/s10957-012-0235-y.
[36] Vinagre B, Podlubny I, Hernandez A, Feliu V. Some approximations of fractional order operators used in control theory and applications. Fract Calc Appl Anal; 2000.
[37] Tenreiro Machado J. Discrete-time fractional-order controllers. Fract Calc Appl Anal 2001;4:47–66.
[38] Barbosa R, Tenreiro Machado J. Implementation of discrete-time fractional-order controllers based on ls approximations. Acta Polytech Hung 3; 2006.
[39] Maione G. High-speed digital realizations of fractional operators in the delta domain. IEEE Trans Autom Control 2011;56(3):697–702.
[40] Zhao Z, Li C. Fractional difference/finite element approximations for the time-space fractional telegraph equation. Appl Math Comput 2012. doi: https://doi.org/10.1016/j.amc.2012.09.022.
[41] Li J, Liu Y, Shan C, Dai C. Implementation of simplified fractional-order pid controller based on modified oustaloup's recursive filter. J Shanghai Jiaotong Univ (Sci) 2020;25(1):44–50. doi: https://doi.org/10.1007/s12204-019-2139-6.
[42] Dorcak L, Petras I, Terpák J, Zborovjan M. Comparison of the methods for discrete approximation of the fractional-order operator. In: Proc of the ICCC'2003 conference, High Tatras, Slovak Republic; 2003. p. 851–6.
[43] Pooseh S, Almeida R, Torres DF. Discrete direct methods in the fractional calculus of variations. Comput Math Appl 2013. doi: https://doi.org/10.1016/j.camwa.2013.01.045. arXiv:1205.4843.
[44] De Keyser R, Muresan CI, Ionescu CM. An efficient algorithm for low-order direct discrete-time implementation of fractional order transfer functions. ISA Trans 2018. doi: https://doi.org/10.1016/j.isatra.2018.01.026.
[45] Swarnakar J, Sarkar P, Singh LJ. Direct discretization method for realizing a class of fractional order system in delta domain – a unified approach. Autom Control Comput Sci 2019. doi: https://doi.org/10.3103/S014641161902007X.
[46] Vinagre BM, Chen YQ, Petráš I. Two direct Tustin discretization methods for fractional-order differentiator/integrator. J Franklin Inst 2003. doi: https://doi.org/10.1016/j.jfranklin.2003.08.001.
[47] Mohamed A, Imran MA, Xiao P, Tafazolli R. Memory-full context-aware predictive mobility management in dual connectivity 5G Networks. IEEE Access 2018. doi: https://doi.org/10.1109/ACCESS.2018.2796579.
[48] Truong HL, Dustdar S. A survey on context-aware web service systems. Int J Web Inform Syst 2009. doi: https://doi.org/10.1108/17440080910947295.
[49] Sun J, Yang J, Li S. Reduced-order GPIO based dynamic event-Triggered tracking control of a networked one-DOF link manipulator without velocity measurement. IEEE/CAA J Autom Sin 2020. doi: https://doi.org/10.1109/JAS.2019.1911738.
[50] Åström KJ, Bernhardsson B. Systems with Lebesgue Sampling. In: Directions in mathematical systems theory and optimization; 2007. https://doi.org/10.1007/3-540-36106-5_1.
[51] Merigo L, Beschi M, Padula F, Visioli A. A noise-filtering event generator for PIDPlus controllers. J Franklin Inst 2018. doi: https://doi.org/10.1016/j.jfranklin.2017.11.041.
[52] Dunham W. The calculus gallery: Masterpieces from Newton to Lebesgue; 2015. https://doi.org/10.5860/choice.43-0368.
[53] Åström KJ, Bernhardsson BM. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In: Proceedings of the IEEE conference on decision and control. doi: https://doi.org/10.1109/cdc.2002.1184824.
[54] Gawthrop PJ, Wang L. Event-driven intermittent control. Int J Control 2009. doi: https://doi.org/10.1080/00207170902978115.
[55] Hu Q, Shi Y. Event-based coordinated control of spacecraft formation flying under limited communication. Nonlinear Dynam 99; 12, 2019. https://doi.org/10.1007/s11071-019-05396-6.
[56] Sanchez J, Guarnes M, Dormido S. On the application of different event-based sampling strategies to the control of a simple industrial process. Sensors (Basel, Switzerland) 2009;9:6795–818. doi: https://doi.org/10.3390/s90906795.
[57] Pawlowski A, Guzmán JL, Berenguel M, Acién FG, Dormido S. Event-based control systems for microalgae culture in industrial reactors, Springer Singapore, Singapore; 2017. p. 1–48. https://doi.org/10.1007/978-981-10-1950-0_1.
[58] Rodríguez-Miranda E, Beschi M, Guzmán J, Berenguel M, Visioli A. Daytime/nighttime event-based pi control for the ph of a microalgae raceway reactor. Processes 2019;7:247. doi: https://doi.org/10.3390/pr7050247.
[59] Han X, Ma Y. Passivity analysis for singular systems with randomly occurring uncertainties via the event-based sliding mode control. Comput Appl Math 2020;39(2):99. doi: https://doi.org/10.1007/s40314-020-1086-z.
[60] Abdelaal AE, Hegazy T, Hefeeda M. Event-based control as a cloud service. In: 2017 American Control Conference (ACC). p. 1017–23.
[61] 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;450:227670. doi: https://doi.org/10.1016/j.jpowsour.2019.227670.

[62] Liu Q, Wang Z, He X, Zhou DH. A survey of event-based strategies on control and estimation 2014. doi: https://doi.org/10.1080/21642583.2014.880387.
[63] Hadian M, Aarabi A, Makvand A, Mehrshadian M. A new event-based pi controller using evolutionary algorithms. J Control, Autom Electr Syst 30 (09 2019). https://doi.org/10.1007/s40313-019-00519-1.
[64] Liu R, Wu J, Wang D. Event-based reference tracking control of discrete-time nonlinear systems via delta operator method. J Franklin Inst 2019;356(17): 10514–531, special Issue on Distributed Event-Triggered Control, Estimation, and Optimization. https://doi.org/10.1016/j.jfranklin.2018.07.022.
[65] Ruiz A, Beschi M, Visioli A, Dormido S, Jiménez J. A unified event-based control approach for foptd and iptd processes based on the filtered smith predictor. J Frank Inst 2016;354. https://doi.org/10.1016/j.jfranklin.2016.11.017.
[66] Birs I, Folea S, Prodan O, Dulf E, Muresan C. An experimental tuning approach of fractional order controllers in the frequency domain, Vol. 10; 2020. https://doi.org/10.3390/app10072379.
[67] Birs I, Muresan C, Nascu I, Folea S, Ionescu C. Experimental results of fractional order PI controller designed for second order plus dead time (SOPDT) processes. In: 2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV 2018. doi: https://doi.org/10.1109/ICARCV.2018.8581241.
[68] Birs I, Muresan C, Copot D, Nascu I, Ionescu C. Identification for control of suspended objects in non-newtonian fluids. Fract Calc Appl Anal 2020. doi: https://doi.org/10.1515/fca-2019-0072.