Variable-order fractional derivatives and their numerical approximations

Duarte Valério*, José Sá da Costa

IDMEC/IST, TULisbon, Portugal

**A B S T R A C T**

This paper addresses the different possible definitions of variable-order derivatives and their numerical approximations; both approximations based upon the definitions and approximations consisting of non-linear transfer functions (in particular combining existing approximations of constant-order fractional derivatives, such as the Crone approximation, with fuzzy logic) are considered. There are different possible configurations, implementing variable-order fractional derivatives both with and without memory of past values of the time-dependent differentiation order.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

In Calculus we learn to give a meaning to \( \frac{df(t)}{dt^n} \) for all \( n \in \mathbb{Z} \) — provided that we identify indefinite integrals (anti-derivatives) with negative orders of differentiation and the function itself with the zero-order derivative, and that \( f(t) \) is well-behaved enough for the limits in the definitions of derivatives and integrals to converge. In Fractional Calculus we extend the meaning of \( \frac{df(t)}{dt^z} \) for any \( z \), real or complex. Variable-order derivatives \( \frac{df(t)}{dt^z} \) become thus possible. This paper addresses the different possible definitions of variable-order derivatives and their numerical approximations; both approximations based upon the definitions and approximations consisting of non-linear transfer functions (in particular combining existing approximations of constant-order fractional derivatives with fuzzy logic) are considered: the latter may be more useful, for instance, for control purposes or on-line calculations.

While there are plenty of engineering applications of constant-order fractional derivatives with real orders — e.g. in control, diffusion or viscoelasticity; a complete bibliographical review falls clearly outside the scope of this paper — they have however been used to develop the so-called third generation Crone robust controllers \([3,4]\). Variable-order fractional integrals (which are but derivatives of negative order) are understood to be a particular case.

* Research for this paper was partially supported by Grant PTDC/EME-CRO/70341/2006 of FCT, funded by POCI 2010, POS C. FSE and MCTES; and by the Portuguese Government and FEDER under program “Programa de Financiamento Plurianual das Unidades de I&D da FCT para as atividades de investigação do laboratório associado LAETA” (POCTI-SFA-10-46-IDMEC).

* Corresponding author. 
E-mail addresses: duarte.valerio@ist.utl.pt (D. Valério), sadacosta@dem.ist.utl.pt (J. Sá da Costa).

The fractional in Fractional Calculus is due to historical reasons: fractions were the object of the first attempts of generalisation. But we keep on speaking about fractional derivatives even when \( z \notin \mathbb{Q} \). Fractional derivatives

\[ \frac{df(t)}{dt^z} \]
derivatives are a more recent area: while the definition given below as RL1 was studied in the 1990s (in [5] and in other papers listed therein), the seminal paper in what the comparison of different possible definitions is concerned is [6], which addresses mainly variable negative orders, considering positive orders only in passing. Applications are given in [7,8] to viscoelasticity, to the processing of geographical data in [9], to signature verification in [10] and to diffusion in [11,12]. Applications to adaptive control are likely feasible. Possible numerical implementations of variable-order fractional derivatives are given in [10,13,14]; these papers, however, pass over the study of the memory that the operator has of previous values of the differentiation order (see Section 4 below).

In this paper, Section 2 sums up basic facts about fractional calculus for reference purposes and defines variable-order derivatives enlarging the definitions given in [6]; Section 3 presents the approximations, first for real variable-orders, then for complex ones; Section 4 gives some numerical results by way of illustration; Section 5 draws some conclusions. The two major innovations in this paper are the systematic combination in Section 2 of different definitions of fractional derivatives with different manners of taking into account past values of order; and the several ways of combining existing approximations for constant orders to obtain approximations for variable orders, given in Section 3.

2. Fractional calculus

2.1. Definitions

In this subsection the basic definitions and results of Fractional Calculus are summed up for reference purposes. No proofs are given. For details, see for instance [16–19]. Below, function \( f(t) \) is assumed well-behaved enough for all necessary operations to be performed. 

**Definition 1 (Functional D for \( n \in \mathbb{Z} \)).**

\[
\mathcal{D}_c^n f(t) = \begin{cases} 
\frac{d^n f(t)}{dt^n} & \text{if } n \in \mathbb{N} \\
 f(t) & \text{if } n = 0 \\
 \int_c^t \mathcal{D}_c^{n+1} f(\tau) \, d\tau & \text{if } n \in \mathbb{Z}^- 
\end{cases}
\]

When \( n \in \mathbb{Z}^+ \) the operator \( D^n \) is local, and hence subscripts \( c \) and \( t \) are useless (e.g. \( \mathcal{D}_c^2 f(t) = \frac{d^2 f(t)}{dt^2} = -\infty \mathcal{D}_t^2 f(t) \)). When \( n \in \mathbb{Z}^- \), the operator is no longer local; changing the value of \( c \) will change the result. In what follows the operator \( D \) will be used instead of the notations \( d/dt \) and \( \int dt \) for derivatives and integrals.\(^5\)

**Theorem 1 (Derivatives of order \( n \in \mathbb{N} \)).**

\[
D^n f(t) = \frac{d^n f(t)}{dt^n} = \lim_{h \to 0} \frac{\sum_{k=0}^{n} (-1)^k \binom{n}{k} f(t-kh) h^n}{h^n}
\]

**Theorem 2 (Cauchy's formula for \( n \in \mathbb{N} \)).**

\[
\mathcal{D}_c^n f(t) = \int_c^t \cdots \int_c^t f(\tau) \, d\tau \cdots d\tau = \int_c^t \frac{(t-\tau)^{n-1}}{(n-1)!} f(\tau) \, d\tau
\]

\( n \) integrations

**Theorem 3 (Law of exponents for \( m, n \in \mathbb{Z} \)).** The equality

\[
\mathcal{D}_c^n \mathcal{D}_c^m f(t) = \mathcal{D}_c^{m+n} f(t)
\]

holds in each of the three following cases:

\( m, n \in \mathbb{Z}^+ \)

\( m \in \mathbb{Z}_0, n \in \mathbb{Z}^+ \)

\( m \in \mathbb{Z}^+, n \in \mathbb{Z}^- \)

**Theorem 4 (Factorials).**

\( n! = \Gamma(n+1), n \in \mathbb{Z}_0^+ \)

Function \( \Gamma(z) \) has poles at \( z \in \mathbb{Z}_0^- \), is defined for all other complex numbers, is real-valued if \( z \) is real, and complex-valued otherwise.

**Definition 2 (Combinations).**

\[
\binom{a}{b} = \begin{cases} 
\frac{\Gamma(a+1)}{\Gamma(b+1)\Gamma(a-b+1)} & \text{if } a, b, \, a-b \in \mathbb{C}, \mathbb{Z}^- \\
\frac{(-1)^b\Gamma(a)}{\Gamma(b+1)\Gamma(-a)} & \text{if } a \in \mathbb{Z}^- \land b \in \mathbb{Z}_0^+ \\
0 & \text{if } (b \in \mathbb{Z}^- \lor b-a \in \mathbb{N}) \land a \notin \mathbb{Z}^- 
\end{cases}
\]

There are several alternative definitions of fractional derivatives, of which the three main ones are considered here.

**Definition 3 (Riemann–Liouville fractional derivatives).**

\[
\mathcal{D}_c^\alpha f(t) = \begin{cases} 
\int^t_0 \frac{(t-\tau)^{-\alpha-1}}{\Gamma(-\alpha)} f(\tau) \, d\tau & \text{if } \Re(\alpha) \in \mathbb{R}^- \\
f(t) & \text{if } \alpha = 0 \\
\frac{d^\alpha f(t)}{dt^\alpha} \mathcal{D}_c^\alpha f(t) & \text{if } \Re(\alpha) \in \mathbb{R}^+ \\
\frac{d \mathcal{D}_c^{-|\alpha|} f(t)}{dt^{-|\alpha|}} & \text{if } \Re(\alpha) = 0 \land \alpha \neq 0
\end{cases}
\]

When \( z \in \mathbb{Z} \) (10) reduces to (1). For \( \Re(\alpha) < 0 \) (10) generalises Cauchy’s formula (3). For \( \Re(\alpha) \geq 0 \) (10) generalises the law of exponents (4).

\(^3\) Notice that we are only considering the case \( c \leq t \), far more needed in applications. It is possible to handle the case \( \mathcal{D}_c^\alpha f(t), c < t \) as well; this will not be done in this paper, but is addressed in references given.
\textbf{Definition 4} (Caputo fractional derivatives).
\[
\mathcal{D}_t^\alpha f(t) = \begin{cases} 
\int_t^{(t-T)^{\alpha-1}} f(c) \, dc & \text{if } \Re(\alpha) \in \mathbb{R}^- \\
 f(t) & \text{if } z = 0 \\
\int_t^{\frac{z}{z+1}} f(c) \, dc & \text{if } \Re(\alpha) \in \mathbb{R}^+ \\
\int_t^{\frac{z}{z+1}} f(c) \, dc & \text{if } \Re(\alpha) = 0 \land z \neq 0 
\end{cases}
\]  
(11)

This is a variation of the Riemann–Liouville definition; it differs only for \( \Re(\alpha) \geq 0 \) (in which case it no longer generalises the law of exponents). It was developed because the Laplace transform of (10) includes fractional derivatives as initial conditions when \( \Re(\alpha) \geq 0 \); the Laplace transform of (11) includes only integer order derivatives (easier to calculate or measure) as initial conditions. When \( z \in \mathbb{Z} \) (11) also reduces to (1).

\textbf{Definition 5} (Gr"unwald–Letnikoff fractional derivatives).
\[
\mathcal{D}_t^\alpha f(t) = \lim_{h \to 0} \frac{\sum_{k=0}^{[\alpha]} (-1)^{k} \binom{\alpha}{k} f(t-kh)}{h^\alpha}
\]  
(12)

This definition is a generalisation of (2). Once more, it can be shown that, if \( z \in \mathbb{Z} \), (12) reduces to (1).

For functions well-behaved enough, the Riemann–Liouville and Gr"unwald–Letnikoff definitions provide the same result; the Caputo definition will often provide different results when \( \Re(\alpha) \geq 0 \). Whatever the definition used, it is clear that if \( z \notin \mathbb{Z} \), the operator is not local (thus fractional derivatives look more like the integrals, not the derivatives, known from Calculus); and that \( z \in \mathbb{C} \setminus \mathbb{R} \) implies in general that the result will be complex.

\textbf{Theorem 5} (Laplace transforms of fractional derivatives).
When initial conditions are equal to zero, the Laplace transform of \( \mathcal{D}_t^\alpha f(t) \) is \( \mathcal{L}[\mathcal{D}_t^\alpha f(t)] = s^{\alpha} \mathcal{L}[f(t)] \), irrespective of the definition employed.\footnote{As mentioned, the initial conditions involved do depend on the definition employed; see the references given for details.}

\textbf{Theorem 6} (Time responses). The impulse response of \( G(s) = s^\alpha \) is
\[
g(t) = \frac{t^{\alpha-1}}{\Gamma(\alpha)}, -z \notin \mathbb{Z}_0
\]  
(13)

and the corresponding step response is
\[
g(t) = \frac{t^{\alpha}}{\Gamma(\alpha+1)}, -z + 1 \notin \mathbb{Z}_0
\]  
(14)

2.2. Transfer functions: frequency responses

Fractional transfer functions are those with non-integer powers of the Laplace transform variable \( s \). Their frequency behaviour can be found, as usual, by replacing \( s \) with \( j \omega \), where \( \omega \) is the frequency under concern. Thus, for \( G(s) = s^{a+jb}, a, b \in \mathbb{R} \), it is found after some calculations that (see Fig. 1)
\[
20 \log_{10} \left| G(j\omega) \right| = 20 a \log_{10} \omega + 20 \log_{10} e^{-\pi b/2}
\]  
(15)

\[
\angle G(j\omega) = \frac{\alpha \pi}{2} + \text{bog}(10) \log_{10} \omega
\]  
(16)

(If \( z \in \mathbb{R} \Rightarrow b = 0 \), the last term disappears in both expressions.) Because for control purposes complex values are not possible, third-generation Crone controllers handle separately the real and imaginary parts of \( G(s) \):
\[
\hat{G}_r(s) = \Re(s^{a+jb}) = \Re(s^a \cos(b) + js^b \sin(b)) = s^a \cos(b)
\]  
(17)

\[
\hat{G}_i(s) = \Im(s^{a+jb}) = \Im(s^a \cos(b) + js^b \sin(b)) = s^b \sin(b)
\]  
(18)

For reasons advanced below, the following transfer functions are also considered:
\[
\hat{G}_r(s) = s^a \sec(b)
\]  
(19)

\[
\hat{G}_i(s) = s^b \csc(b)
\]  
(20)

After some calculations it is seen that
\[
20 \log_{10} \left| \hat{G}_r(j\omega) \right| = 20 a \log_{10} \omega
\]
\[
+ 20 \log_{10} \sqrt{\cos(\pi b) + \cos(2 \log_{10} \omega)}
\]  
(21)

\[
20 \log_{10} \left| \hat{G}_i(j\omega) \right| = 20 a \log_{10} \omega
\]
\[
+ 20 \log_{10} \sqrt{\cos(\pi b) - \cos(2 \log_{10} \omega)}
\]  
(22)

\[
20 \log_{10} \left| \hat{G}_r(j\omega) \right| = 20 a \log_{10} \omega
\]
\[
- 20 \log_{10} \sqrt{\cos(\pi b) + \cos(2 \log_{10} \omega)}
\]  
(23)

\[
20 \log_{10} \left| \hat{G}_i(j\omega) \right| = 20 a \log_{10} \omega
\]
\[
- 20 \log_{10} \sqrt{\cos(\pi b) - \cos(2 \log_{10} \omega)}
\]  
(24)

\[
\angle \hat{G}_r(j\omega) = \frac{\alpha \pi}{2} - \arctan \left[ \frac{\tan(\pi b/2)}{\cos(\log_{10} \omega)} \right]
\]  
(25)

\[
\angle \hat{G}_i(j\omega) = \frac{\alpha \pi}{2} + \arctan \left[ \frac{\tan(\pi b/2)}{\sin(\log_{10} \omega)} \right]
\]  
(26)

\[
\angle \hat{G}_r(j\omega) = \frac{\alpha \pi}{2} - \arctan \left[ \frac{\tan(\pi b/2)}{\sin(\log_{10} \omega)} \right]
\]  
(27)

\[
\angle \hat{G}_i(j\omega) = \frac{\alpha \pi}{2} + \arctan \left[ \frac{\tan(\pi b/2)}{\cos(\log_{10} \omega)} \right]
\]  
(28)

Unlike (15)–(16), (21)–(28) do not correspond to linear Bode diagrams, but all of them turn out to be almost linear\footnote{The arc of tangent in (25)–(28) need never originate discontinuities, since both the sine and the cosine are known. This corresponds to the use of an algorithm implemented as atan2 in most programming languages, and seems not to have been taken into account in [3,4].}; asymptotic values can easily be found. Concerning gains (21)–(24), the non-linearity is caused by the cosine.
The cosine that originates the oscillations has a period $\frac{\pi}{2}$.

Concerning phases (25)–(28), they would be linear if $
abla j^2 \log_{10} \tanh o$ also verifies inequalities (29)–(30); in all cases the amplitude of the oscillations is

$\omega > 0, b > 0$

$\omega > 0, b < 0$

$\omega < 0, b > 0$

$\omega < 0, b < 0$

Fig. 1. Bode diagram of $G(j\omega)$.

The tangent that originates the oscillations has a period $\pi$ and an argument $\log_{10} \tanh o = \log_{10} \tanh(\beta \pi/2)$; this means that $\omega$ is a sequence of steps. Thus the maximum amplitude of the oscillations is

$T_{\text{phase}} = \frac{\pi}{\log_{10}(10)}$ decades

(39)

Phases (25)–(28) are monotonic, but if $b \to 0$ they approach a sequence of steps. Thus the maximum amplitude of the oscillations is

$A_{\text{phase}} = ||b|| \log_{10}(10^{95/\log_{10}(10)}) = \pi$

(40)

Because $\lim_{b \to \infty} \tanh(b \pi/2) = \pm 1$ and thus $\lim_{b \to \infty} 2 \log_{10} \tanh(b \pi/2) = 0$, it is clear that, the larger $b$ is, the closer to linearity the Bode diagrams of transfer functions (17)–(20) will be. For the purpose of third-generation Crone control, they are combined as

$G_{\text{cr}}(s) = \begin{cases} s^a \cos(b \log_{10}) & \text{if } b \leq 0 \\ s^b \sec(b \log_{10}) & \text{if } b > 0 \end{cases}$

(41)

$G_{\text{c}}(s) = \begin{cases} s^b \sin(b \log_{10}) & \text{if } b \leq 0 \\ s^b \csc(b \log_{10}) & \text{if } b > 0 \end{cases}$

(42)

In (41)–(42) the sign of $a$ commands the sign of the slope of the gain and the sign of $b$ commands the sign of the slope of the phase. The slopes are thus the same obtained
with $G(s)$. Thus the frequency responses $\hat{G}_1(j\omega)$ and $\hat{G}_2(j\omega)$ can be viewed as approximations of $G(j\omega)$—in spite of being shifted upwards or downwards, and in spite of the ripples. The example in Fig. 2 shows for a particular case the difference between the exact frequency behaviour and the linearised approximations.

2.3. Crone approximation

While it is possible to implement fractional transfer functions using hardware (see e.g. [20]), nearly all implementations, both in hardware and simulation, rely on approximations using integer order derivatives and integrals only. Among the existing ones, the one considered in this paper is the widely used and well-established Crone approximation, that makes use of a recursive placement of $N$ poles and $N$ zeros within a frequency range $[0, \omega_0]$:

$$G(s) = s^\alpha + j\beta \approx C \prod_{m=1}^{N} \frac{1 + \frac{s}{\Omega_{z,m}}}{1 + \frac{s}{\Omega_{p,m}}}$$  \hspace{1cm} (43)

$$\Omega_{z,m} = \Omega_l \left( \frac{\Omega_h}{\Omega_l} \right)^{2m-1 - (z + j\beta)/2N}$$  \hspace{1cm} (44)

$$\Omega_{p,m} = \Omega_l \left( \frac{\Omega_h}{\Omega_l} \right)^{2m-1 + (z + j\beta)/2N}$$  \hspace{1cm} (45)

The correct gain and phase at 1 rad/s must be set\(^8\) by adjusting $C$. In practice (43) is valid in the frequency range $[10\omega_0, \omega_0/10]$. Even within that range, both gain and phase have ripples. The quality of the approximation increases as $N \to \infty$, $\alpha \to 0$ and $b \to 0$. For this reason it is usual to approximate only orders $z \in \Omega = \{w \in \mathbb{C} : -1 \leq \Re(w) \leq 1 \land \Im(w) \leq 1\}$. For orders such that $|\Re(z)| > 1$ and $-1 \leq \Im(w) \leq 1$ it is possible to make $s^2 = s^2 + s^2$, $z_1 \in \mathbb{Z}$, $z_2 \in \Omega$ and approximate $s^2$ only.

Discrete approximations exist that directly provide transfer functions which are rational functions of the delay operator\(^9\) $z^{-1}$. But discretising continuous approximations (such as the one above) often leads to equally good or even better results [21].

If $z \in \mathbb{R}$, the zeros (44), the poles (45) and $C$ are all real.\(^{10}\) If $z \in \mathbb{C} \setminus \mathbb{R}$, $C$ will in general need to have an imaginary part, and zeros and poles are complex and do not appear in pairs of complex conjugates. Hence (43) will be a ratio of two polynomials with complex coefficients.

\(^8\) Doing so for frequency 1 rad/s makes calculations easier, but if 1 rad/s is outside the frequency range of interest any other suitable frequency will of course have to be used instead.

\(^9\) To avoid confusing the delay operator $z^{-1}$ with the variable order $z$, a different font is being employed.

\(^{10}\) For $z \in \mathbb{R}$ the phase behaviour at 1 rad/s is approximately correct, hence $C$ will have to be adjusted to correct the gain only; the correction needed for the phase would be minimal and would force a complex value for $C$, and thus complex coefficients, which are completely unnecessary and better avoided.
The best way to use such approximations for computational purposes is to discretise them, thus allowing the computation of their responses by the usual process (a linear combination of past values of inputs and outputs, weighted by the coefficients).

Since the Bode diagram of \(G(s)\) is similar to the linearised Bode diagrams of \(G_L(s)\) or \(G_R(s)\), the Crone approximation can be used to approximate them as well. But the ripples in both phase and gain of \(G_L(s)\) or \(G_R(s)\) given by (21)–(28) are neglected, while (unrelated) ripples resulting from the finite number of approximation zeros and poles \(N\) appear.

2.4. Variable order derivatives

If the differentiation order \(z\) is not constant with time, there are three obvious ways [6] of dealing with branch \(\Re(z(t)) \in \mathbb{R}^+\) in (10): (1) to let the argument of \(z\) be the current time instant \(t\); (2) to let the argument of \(z\) be the same as that of \(f\), which is the integration variable \(t\); (3) to let the argument of \(z\) be the difference between the two above, as in the kernel of the integral. The choice will also affect the \(\Re(z(t)) \in \mathbb{R}^+\) branch, since it is defined at the expense of the former.

**Definition 6** (Riemann–Liouville variable-order derivatives \(1\) (RL1)).

\[
\begin{align*}
^C_D\!t^z_0f(t) &= \begin{cases}
\int_{(t-z)^{-\alpha(t)-1}}^{(t-z)^{\alpha(t)-1}} f(\tau) d\tau, & \text{if } \Re(z(t)) \in \mathbb{R}^- \\
f(t), & \text{if } z(t) = 0 \\
\frac{1}{\Gamma(1 - \alpha(t))} D_t^{1-\alpha(t)} f(t), & \text{if } \Re(z(t)) \in \mathbb{R}^+ \\
\frac{d}{dt} D_t^{1-\alpha(t)} f(t), & \text{if } \Re(z(t)) = 0 \land z(t) \neq 0
\end{cases}
\end{align*}
\]

**Definition 7** (Riemann–Liouville variable-order derivatives \(2\) (RL2)).

\[
\begin{align*}
^C_D\!t^z_1f(t) &= \begin{cases}
\int_{(t-z)^{-\alpha(t)-1}}^{(t-z)^{\alpha(t)-1}} f(\tau) d\tau, & \text{if } \Re(z(t)) \in \mathbb{R}^- \\
f(t), & \text{if } z(t) = 0 \\
\frac{1}{\Gamma(1 - \alpha(t))} D_t^{1-\alpha(t)} f(t), & \text{if } \Re(z(t)) \in \mathbb{R}^+ \\
\frac{d}{dt} D_t^{1-\alpha(t)} f(t), & \text{if } \Re(z(t)) = 0 \land z(t) \neq 0
\end{cases}
\end{align*}
\]

**Definition 8** (Riemann–Liouville variable-order derivatives \(3\) (RL3)).

\[
\begin{align*}
^C_D\!t^z_2f(t) &= \begin{cases}
\int_{(t-z)^{-\alpha(t)-1}}^{(t-z)^{\alpha(t)-1}} f(\tau) d\tau, & \text{if } \Re(z(t)) \in \mathbb{R}^- \\
f(t), & \text{if } z(t) = 0 \\
\frac{1}{\Gamma(1 - \alpha(t))} D_t^{1-\alpha(t)} f(t), & \text{if } \Re(z(t)) \in \mathbb{R}^+ \\
\frac{d}{dt} D_t^{1-\alpha(t)} f(t), & \text{if } \Re(z(t)) = 0 \land z(t) \neq 0
\end{cases}
\end{align*}
\]

The same three variations can be applied to (11) and (12) as well:

**Definition 9** (Caputo variable-order derivatives \(1\) (C1)).

\[
\begin{align*}
^C_D\!t^z_0f(t) &= \begin{cases}
\int_{(t-z)^{-\alpha(t)-1}}^{(t-z)^{\alpha(t)-1}} f(\tau) d\tau, & \text{if } \Re(z(t)) \in \mathbb{R}^- \\
f(t), & \text{if } z(t) = 0 \\
\frac{1}{\Gamma(1 - \alpha(t))} D_t^{1-\alpha(t)} f(t), & \text{if } \Re(z(t)) \in \mathbb{R}^+ \\
\frac{d}{dt} D_t^{1-\alpha(t)} f(t), & \text{if } \Re(z(t)) = 0 \land z(t) \neq 0
\end{cases}
\end{align*}
\]

**Definition 10** (Caputo variable-order derivatives \(2\) (C2)).

\[
\begin{align*}
^C_D\!t^z_1f(t) &= \begin{cases}
\int_{(t-z)^{-\alpha(t)-1}}^{(t-z)^{\alpha(t)-1}} f(\tau) d\tau, & \text{if } \Re(z(t)) \in \mathbb{R}^- \\
f(t), & \text{if } z(t) = 0 \\
\frac{1}{\Gamma(1 - \alpha(t))} D_t^{1-\alpha(t)} f(t), & \text{if } \Re(z(t)) \in \mathbb{R}^+ \\
\frac{d}{dt} D_t^{1-\alpha(t)} f(t), & \text{if } \Re(z(t)) = 0 \land z(t) \neq 0
\end{cases}
\end{align*}
\]

**Definition 11** (Caputo variable-order derivatives \(3\) (C3)).

\[
\begin{align*}
^C_D\!t^z_2f(t) &= \begin{cases}
\int_{(t-z)^{-\alpha(t)-1}}^{(t-z)^{\alpha(t)-1}} f(\tau) d\tau, & \text{if } \Re(z(t)) \in \mathbb{R}^- \\
f(t), & \text{if } z(t) = 0 \\
\frac{1}{\Gamma(1 - \alpha(t))} D_t^{1-\alpha(t)} f(t), & \text{if } \Re(z(t)) \in \mathbb{R}^+ \\
\frac{d}{dt} D_t^{1-\alpha(t)} f(t), & \text{if } \Re(z(t)) = 0 \land z(t) \neq 0
\end{cases}
\end{align*}
\]

**Definition 12** (Grünwald–Letnikov variable-order derivatives \(1\) (GL1)).

\[
^G_D\!t^z_0f(t) = \lim_{h \to 0^+} \sum_{k=0}^{\lfloor (t-z)/h \rfloor} (-1)^k \left( \begin{array}{c}
z(t) \\ k\end{array} \right) f(t-kh)
\]

**Definition 13** (Grünwald–Letnikov variable-order derivatives \(2\) (GL2)).

\[
^G_D\!t^z_1f(t) = \lim_{h \to 0^+} \sum_{k=0}^{\lfloor (t-z)/h \rfloor} \frac{(-1)^k}{k!} \left( \begin{array}{c}
z(t-kh) \\ k\end{array} \right) f(t-kh)
\]

**Definition 14** (Grünwald–Letnikov variable-order derivatives \(3\) (GL3)).

\[
^G_D\!t^z_2f(t) = \lim_{h \to 0^+} \sum_{k=0}^{\lfloor (t-z)/h \rfloor} \frac{(-1)^k}{k!} \left( \begin{array}{c}
z(kh) \\ k\end{array} \right) f(t-kh)
\]

(Notice that in definitions GL2 and GL3 the denominator must be inside the summation.)

Different definitions lead to operators with different memories of past values of \(z(t)\), ranging from no memory
at all to a strong memory,11 as seen below in Section 4. If \( z \) is constant with time, all definitions reduce to (10), (11) or (12) as the case may be [6].

### 3. Approximations of variable-order fractional derivatives

#### 3.1. Approximations from definition

The Riemann–Liouville and the Caputo definitions can be implemented for numerical purposes approximating derivatives with finite differences and integrals with trapezoidal numerical integration.

The Grünwald–Letnikoff definitions can be implemented approximating the limit by replacing the vanishing of some small, finite sampling time \( T_\varepsilon \), thus GL1 becomes

\[
\varepsilon D_t^{\alpha/\varepsilon} f(t) \approx \sum_{k=0}^{\max} \left(-\frac{1}{k}\right)^k \frac{Z(t)}{k}s^{\alpha/\varepsilon} k^{k}(t-kT_\varepsilon)
\]

(GL2) and GL3 are handled similarly. But it must be taken into account that in all machines the \( f(t) \) function returns \( +\infty \) for relatively small values of \( t \). Let \( k_{\max} \in \mathbb{N} \) be the smallest integer for which \( f(k_{\max}+1) \) returns \( +\infty \). If \( [t-c)/T_\varepsilon > k_{\max} \), some terms will be neglected in the summation of (55), which are not necessarily small: this truncation corresponds to a change in the value of \( c \). Increasing \( T_\varepsilon \) would decrease the accuracy of the approximation. To minimize the negative effect of increasing \( T_\varepsilon \), we can do as follows. We assume that \( k_{\max} \) is even and that only integer multiples of \( T_\varepsilon \) can be used (in other words, \( T_\varepsilon \) is fixed and all we can do is resample the data). Then we replace (55) with

\[
\varepsilon D_t^{\alpha/\varepsilon} f(t) \approx \sum_{k=0}^{k_{\max}} \left(-\frac{1}{k}\right)^k \frac{Z(t)}{k}s^{\alpha/\varepsilon} k^{k}(t-kT_\varepsilon)
\]

(GL2) and GL3 are handled similarly. But it must be taken into account that in all machines the \( f(t) \) function returns \( +\infty \) for relatively small values of \( t \). Let \( k_{\max} \in \mathbb{N} \) be the smallest integer for which \( f(k_{\max}+1) \) returns \( +\infty \). If \( [t-c)/T_\varepsilon > k_{\max} \), some terms will be neglected in the summation of (55), which are not necessarily small: this truncation corresponds to a change in the value of \( c \). Increasing \( T_\varepsilon \) would decrease the accuracy of the approximation. To minimize the negative effect of increasing \( T_\varepsilon \), we can do as follows. We assume that \( k_{\max} \) is even and that only integer multiples of \( T_\varepsilon \) can be used (in other words, \( T_\varepsilon \) is fixed and all we can do is resample the data). Then we replace (55) with

\[
\varepsilon D_t^{\alpha/\varepsilon} f(t) \approx \sum_{k=0}^{k_{\max}} \left(-\frac{1}{k}\right)^k \frac{Z(t)}{k}s^{\alpha/\varepsilon} k^{k}(t-kT_\varepsilon)
\]

If \( [t-c)/mT_\varepsilon > k_{\max} \), we need to keep summation indexes \( k \) always below \( k_{\max} \).

#### 3.2. Crone-based approximations: real orders

In this subsection Crone approximations of fractional derivatives are used to build approximations of variable order derivatives. It is expedient to describe first approximations with orders \( z \in [-1,1] \); the next subsection will deal with the more generic case \( z \in \Omega \).

##### 3.2.1. Fuzzy approximation without memory (local feedback)

In Boolean logic there are only two possible truth values, \( \lambda = 0 \) (falsehood) and \( \lambda = 1 \) (truth). Given a set of real numbers \( A \), its membership function \( \mu_k \) is a function that takes the value \( \mu_k(x) = 1 \) if \( x \in A \) and the value \( \mu_k(x) = 0 \) if \( x \notin A \). A set can be defined from its membership function, and hence any function \( \mu_k \) mapping \( \mathbb{R} \) into \([0,1]\) defines a set of real numbers. A partition of set \( A \) is a set of sets \( A_1, A_2, \ldots, A_n \) such that \( \forall a \in A \) and \( A \cap A' = \emptyset \) and \( \bigcup_{k=1}^{n} A_k = A \), or, which is the same, such that \( \sum_{k=1}^{n} \mu_k = \mu_k \). When using fuzzy logic all values \( \lambda \in [0,1] \) are admissible as truth values; hence membership functions are those mapping \( \mathbb{R} \) into \([0,1]\). Such membership functions allow defining fuzzy sets (by opposition to sets in the classical sense) of real numbers. The membership degree of a real number \( x \) in a fuzzy set \( A \) is defined by a membership function \( \mu_k \) that can thus assume any value between 0 (no membership) and 1 (full membership). A partition of a fuzzy set \( A \) is a set in the classical sense of fuzzy sets \( A_1, A_2, \ldots, A_n \) such that \( \sum_{k=1}^{n} \mu_k = \mu_k \). For instance, Fig. 3 shows the membership function of set (in the classical sense) \( A = [-1,1] \) and the membership functions of a partition of this set into 11 fuzzy sets. This partition is uniform in the sense that all membership functions are the result of a translation of a triangular template; the membership functions it consists of are said to be triangular.

Consider one such partition of \( A = [-1,1] \) into \( n_{mf} \) triangular fuzzy membership functions \( \mu_k, k = 1 \ldots n_{mf} \).

---

11 Because \( D_t^{\alpha} \) is not a local operator (save only when \( z \in \mathbb{Z}_+ \)), it is said to have memory of past values of \( f(t) \). This is not to be confused with the memory of past values of \( z(t) \).

12 All uniform partitions result in triangular membership functions, but a partition into triangular membership functions with different widths is possible and is not uniform. Nevertheless a uniform partition is the most reasonable choice here.
Let \( z_k, k=1\ldots n_{mf} \), be the absicssa for which \( \mu_{A_k}(z_k) = 1 \). We will say that the truth value of the fuzzy proposition “\( z \) is \( z_k \)” is \( \mu_{A_k}(z) \). If \( z \) is one of the \( z_k \) there is only one such proposition that is true, all the others being false; for all the other values of \( z \in \mathbb{A} \) there are two propositions that are partially true (there are two membership functions which are not zero).

Suppose that we obtain \( n_{mf} \) Crane approximations of fractional derivatives \( G_{k}(z) \), such that \( G_k(z) = s^q, \ k=1\ldots n_{mf} \). If \( z \) is one of the \( z_k \), clearly we can say that “\( z \) is \( z_k \)” will be true.

Thus, if we let \( s^q \approx \mu_{A_k}(z) G_k(z) \), \( k=1\ldots n_{mf} \), because all these \( n_{mf} \) fuzzy rules will be false save one, which is (completely) true. If \( z_k < z < z_{k+1} \), two of the fuzzy rules above will be partially true, and we will say that \( s^q \approx \mu_{A_k}(z) G_k(z) + \mu_{A_{k+1}}(z) G_{k+1}(z) \). Since all the other membership functions are zero we can sum the extent and say that \( s^q \approx \theta \sum_{k=1}^{n_{mf}} \mu_k(z) G_k(z) \); this is the meaning attributed to the \( n_{mf} \) fuzzy rules above. Notice that, the membership functions being triangular, the result is a linear interpolation between several possible results.

The fuzzy approximations of variable-order derivatives without memory considered in this paper are like the ones above, but Crane approximations used are discretised, as this is necessary for other approximations described below, as well as whenever the orders are complex. Consequently, using discretised approximations as a basis for all approximations of variable-order derivatives allows more meaningful and fair comparisons.

Let us consider that these approximations have \( Z \) zeros, \( P \) poles, denominator coefficients \( a \), numerator coefficients \( b \), zeros \( \zeta \), poles \( \pi \), gains \( K \), and use a sampling time \( T_s \):

\[
s^q \approx \hat{G}_k(z^{-1}) = \sum_{m=0}^{Z} b_{k,m} z^{-m} + \sum_{m=1}^{P} a_{k,m} z^{-m} = K_k \prod_{m=1}^{Z} \frac{1-z_k \zeta_m^{-1}}{1-z_k \pi_m^{-1}}, \quad k=1\ldots n_{mf} \tag{57}
\]

Thus, if we let \( \hat{G}_k(t) = \hat{G}_k(z^{-1}) f(t) \),

\[
\hat{G}_k(t) = \sum_{m=0}^{Z} b_{k,m} f(t-m T_s) - \sum_{m=1}^{P} a_{k,m} \hat{G}_k(t-m T_s), \quad k=1\ldots n_{mf} \tag{58}
\]

Our fuzzy model will consist of \( n_{mf} \) fuzzy rules, have two inputs—the differentiated variable \( f(t) \) and the differentiation order \( z(t) \)—and one output, which is \( \hat{G}(t) \approx G_t^{D^q} f(t) \). The conditions the rules depend upon are functions of \( z(t) \) only; each will provide an output \( \hat{G}_k(t) \) which is the rule’s contribution to \( \hat{G}(t) \):

If \( z(t) = z_k \) then \( \hat{G}_k(t) = \sum_{m=0}^{Z} b_{k,m} f(t-m T_s) - \sum_{m=1}^{P} a_{k,m} \hat{G}_k(t-m T_s), \quad k=1\ldots n_{mf} \tag{59}
\]

\[13\] Because the conclusion that can be taken from any single fuzzy rule is not fuzzy, these constitute what is called a Takagi–Sugeno fuzzy model, by opposition to a linguistic model, in which the conclusion is itself a fuzzy proposition [15].
The output \( \hat{G}(t) \) is given by

\[
b_m(t) = \sum_{k=1}^{N} \mu_{\omega_k}(z(t)) b_{k,m}(t), \quad m = 0 \ldots Z
\]

\[
a_m(t) = \sum_{k=1}^{N} \mu_{\omega_k}(z(t)) a_{k,m}(t), \quad m = 1 \ldots P
\]

\[
\hat{G}(t) = \sum_{m=0}^{Z} b_m(t) f(t-mT_s) - \sum_{m=1}^{P} a_m(t) \hat{G}(t-mT_s)
\]

Replacing (63)–(64) in (65) we obtain

\[
\hat{G}(t) = \sum_{m=0}^{Z} \sum_{k=1}^{N} \mu_{\omega_k}(z(t)) b_{k,m}(t) f(t-mT_s)
\]

\[
- \sum_{m=1}^{P} a_m(t) \hat{G}(t-mT_s)
\]

which is what we obtain replacing (61) in (60); in other words, the fuzzy interpolation of coefficients leads to the same result of global feedback (which corresponds to easier calculations). Consequently, only global feedback will be mentioned in what follows.

3.2.4. Third fuzzy approximation with memory (fuzzy interpolation of poles, zeros and gains)

Instead of using fuzzy logic to interpolate coefficients \( a \) and \( b \), it is possible to use it to interpolate transfer function zeros, poles and gain. Rules become

If \( z(t) \) is \( z_k \) then \( \hat{z}_m(t) = z_{k,m} \), \( m = 1 \ldots Z \) and \( \pi_m(t) = \pi_{k,m} \), \( m = 1 \ldots P \) and \( K(t) = K_{k} \), \( k = 1 \ldots n_{mf} \)

\[
\hat{z}_m(t) = \sum_{k=1}^{n_{mf}} \mu_{z_k}(z(t)) z_{k,m}(t), \quad m = 1 \ldots Z
\]

\[
\pi_m(t) = \sum_{k=1}^{n_{mf}} \mu_{z_k}(z(t)) \pi_{k,m}(t), \quad m = 1 \ldots P
\]

\[
K(t) = \sum_{k=1}^{n_{mf}} \mu_{z_k}(z(t)) K_{k}(t)
\]

\[
\hat{G}(t) = \left[ K(t) \prod_{m=1}^{Z} \left( 1 - \hat{z}_m(t) z_{k,m}^{-1} \right) \right] f(t)
\]

3.2.5. Approximations with \( C^\infty \) interpolations

Rather than using fuzzy logic to interpolate in-between the zeros and poles of a small number of Crone approximations, it is possible to use in each instant the values of poles and zeros given by (44)–(45), after discretisation. If Tustin discretisation method \( s \approx (2/T_s)(1-z^{-1})/(1+z^{-1}) \) is employed, then (43) becomes

\[
s^2 \approx C \prod_{m=1}^{N} \left[ 1 + \frac{2}{\omega_{m,T_s}} \frac{1-z^{-1}}{1+z^{-1}} \right]
\]

From (72) and from (44)–(45), we find that discrete, variable-order poles and zeros are given by

\[
\hat{z}_m(t) = \sum_{m=0}^{Z} b_m(t) f(t-mT_s)
\]

\[
- \sum_{m=1}^{P} a_m(t) \hat{G}(t-mT_s)
\]

\[
= C \prod_{m=1}^{N} \left[ 1 - \frac{2}{\omega_{m,T_s}} \frac{1-z^{-1}}{1+z^{-1}} \right]
\]

3.3. Crone-based approximations: complex orders

The approximations from the previous subsection can be used for complex orders \( z \in \Omega \) with slight adaptations.

3.3.1. Fuzzy approximation without memory (local feedback)

Suppose that we obtain \( n_{mf}^2 \) Crone approximations of fractional derivatives \( \hat{G}_{k_1,k_2}(s) \), such that \( \hat{G}_{k_1,k_2}(s) \approx s^{\alpha_{k_1,k_2}} = s^{\alpha_{k_1}} + j \omega_{k_2} \), \( k_1, k_2 = 1 \ldots n_{mf} \), and that the \( \alpha_k \) and the \( b_{k_j} \) are the abscissas for which the triangular membership functions that are a partition of set \( A \) verify \( \mu_{\alpha_k}(\alpha_k) = 1 \) and \( \mu_{b_{k_j}}(b_{k_j}) = 1 \). The partition being the same in both cases, this implies that the \( \alpha_k \) are uniformly distributed in \( \Omega \) as the nodes of a grid. Eq. (57) becomes

\[
\hat{G}_{k_1,k_2}(z(t)) = \hat{G}_{k_1,k_2}(z^{-1}) f(t), \text{ and the } n_{mf}^2 \text{ fuzzy rules will be}
\]

\[
\text{If } \Re[z(t)] \text{ is } \alpha_{k_1} \text{ and } \Im[z(t)] \text{ is } b_{k_2} \text{ then } \hat{G}_{k_1,k_2}(t)
\]

\[
= \sum_{m=0}^{Z} b_{k_1,k_2,m}(f(t-mT_s)) - \sum_{m=1}^{P} a_{k_1,k_2,m} \hat{G}_{k_1,k_2}(t-mT_s),
\]

\( k_1, k_2 = 1 \ldots n_{mf} \)

The truth value of the fuzzy proposition in the antecedent will be \( \mu_{\alpha_{k_1}}(\Re[z(t)]) \mu_{b_{k_2}}(\Im[z(t)]) \). The fuzzy model’s output \( \hat{G}(t) \) will be

\[
\hat{G}(t) = \sum_{k_1=1}^{n_{mf}} \sum_{k_2=1}^{n_{mf}} \mu_{\alpha_{k_1}}(\Re[z(t)]) \mu_{b_{k_2}}(\Im[z(t)]) \hat{G}_{k_1,k_2}(t)
\]

\[14 \text{ This is the same as partitioning } \Omega \text{ into } n_{mf}^2 \text{ fuzzy sets with pyramidal membership functions.} \]
3.3.2. First fuzzy approximation with memory (global feedback)

Fuzzy rules (61) are replaced by

\[
\text{If } R \frac{z(t)}{C_{138}} \text{ is } a_{k_1} \text{ and } \Im \frac{z(t)}{C_{138}} \text{ is } b_{k_2} \text{ then } \hat{G}_{k_1,k_2}(t)
\]

\[
= \sum_{m=0}^{Z} b_{k_1,k_2,m} f(t-mT_s) - \sum_{m=1}^{P} a_{k_1,k_2,m} \hat{G}(t-mT_s),
\]

\[k_1,k_2 = 1 \ldots n_{mf}\]

(78)

The fuzzy model output \(\hat{G}(t)\) is given by (77).

3.3.3. Second fuzzy approximation with memory (fuzzy interpolation of coefficients)

Eqs. (63) and (64) become

\[
b_m(t) = \sum_{k_1 = 1}^{n_{mf}} \sum_{k_2 = 1}^{n_{mf}} H_{A_{k_1}}(R \frac{z(t)}{C_{138}}) H_{A_{k_2}}(\Im \frac{z(t)}{C_{138}}) b_{k_1,k_2,m}(t), \quad m = 0 \ldots Z
\]

(79)

\[
a_m(t) = \sum_{k_1 = 1}^{n_{mf}} \sum_{k_2 = 1}^{n_{mf}} H_{A_{k_1}}(R \frac{z(t)}{C_{138}}) H_{A_{k_2}}(\Im \frac{z(t)}{C_{138}}) a_{k_1,k_2,m}(t), \quad m = 1 \ldots P
\]

(80)

Fig. 5. Derivatives of a unit step without memory of past values of \(z(t)\); top left: \(z(t) \in \mathbb{R}\); other plots: \(z(t) \in \mathbb{C} \setminus \mathbb{R}\).
The fuzzy model output $\hat{G}(t)$ is given by (65). This leads to the same result of global feedback for the same reasons mentioned in Section 3.2.3.

### 3.3.4. Third fuzzy approximation with memory (fuzzy interpolation of poles, zeros and gains)

Eqs. (68)–(70) become

$$
\zeta_m(t) = \sum_{k_1=1}^{n_{\text{real}}} \sum_{k_2=1}^{n_{\text{real}}} \mu_{A_{k_1}}(9\beta(t)) \mu_{A_{k_2}}(3\beta(t)) \zeta_{k_1,k_2,m}(t), \quad m = 1 \ldots Z
$$

(81)

$$
\pi_m(t) = \sum_{k_1=1}^{n_{\text{real}}} \sum_{k_2=1}^{n_{\text{real}}} \mu_{A_{k_1}}(9\beta(t)) \mu_{A_{k_2}}(3\beta(t)) \pi_{k_1,k_2,m}(t), \quad m = 1 \ldots P
$$

(82)

$$
K(t) = \sum_{k_1=1}^{n_{\text{real}}} \sum_{k_2=1}^{n_{\text{real}}} \mu_{A_{k_1}}(9\beta(t)) \mu_{A_{k_2}}(3\beta(t)) K_{k_1,k_2}(t)
$$

(83)

The fuzzy model output $\hat{G}(t)$ is given by (71).

### 3.3.5. Approximations with $C^\infty$ interpolations

The sole difference to note is that $K(t)$ must be calculated so that $\hat{G}(s) = \mathcal{L}[G(t)]$ verifies $\hat{G}(f) = j^{\text{real}}$, $\forall \beta(t) \in C$, as mentioned in Section 2.3.

### 4. Numerical results

The numerical results in this section were obtained for $Z=5$, $P=5$, $[\alpha_0, \alpha_\text{H}] = [10^{-2}, 10^2]$ rad/s (this means that a good approximation can be expected in the
If \( z(t) \) is real and positive, or complex with a positive real part, definitions C1 and GL1 still provide an operator without memory of past values of \( z(t) \), but RL1 does not.\(^\text{16}^\) Definitions RL2, C2 and GL2 no longer lead to the same result. Definitions RL3 and C3 provide the same response, which is different from that of GL3.

If \( z(t) \) is a pure imaginary number, definitions behave as if it had a positive real part, which is not surprising as the branches for positive and non-existent real parts are similar for definitions RL1 to RL3 and C1 to C3.

These differences compound if the sign of \( z(t) \) changes with time.

\(^\text{15}^\) It is linked at from the first author’s webpage.

\(^\text{16}^\) Notice the memoryless RL1 curve in the top left plot of Fig. 5, which oscillates with an increasing amplitude. The RL1 curve in the left top plot of Fig. 6 is its derivative, and must thus also oscillate with an increasing amplitude; if it were memoryless, its oscillations would have to decrease.
controller will be overshoot or settling time change with time, the value of \( M \) response is always the same, irrespective of the actual phase margin, and thus that the overshoot of a step control plant order derivatives to control. Suppose we want to carry on this task because they are so heavy and slow that real time implementation is impossible.

The implementations based upon the definitions have over Crone-based approximations the great disadvantage of being significantly heavier from the computational point of view. Among the latter, the global feedback approximation appears to be the one that is most stable when the sampling time changes; the fuzzy interpolation of poles, zeros and gains leads to poor results for large sampling times and the \( C^n \) interpolation may lead to numerical instability for small sampling times.

A final example concerns the application of variable order derivatives to control. Suppose we want to control plant \( G(s) = 1/Ms^2 \approx (1/M)(0.0001z^{-1} + 0.0001z^{-3})/(2-4z^{-1} + 2z^{-2}), T_s=0.01 \), where \( M \) may change with time, using a feedback loop with controller \( C(s) = s^2, \alpha \in \mathbb{R} \) (implemented using a discretised Crone approximation), which ensures that the open loop \( C(s)G(s) \) has a constant phase margin, and thus that the overshoot of a step response is always the same, irrespective of the actual value of \( M(t) \) [3]. If the requirements on maximum overshoot or settling time change with time, the controller will be \( C(s) = s^{\alpha(t)} \). Fig. 8 shows simulation results where all methods from Section 3.2 perform acceptably and fast enough, irrespective of having or not memory of past values of \( \alpha \); methods from Section 3.1 are unable to carry on this task because they are so heavy and slow that real time implementation is impossible.

5. Conclusions

In this paper it is shown how a simple fuzzy inference engine can be used together with approximations of fractional derivatives of constant order \( z \) to approximate fractional derivatives of varying order, and how the different use of the fuzzy inference engine allows having or not memory of past values of \( z(t) \). Concerning the examples given, it is good to repeat that varying-order approximations obtained cannot be better than the constant-order approximations used to build them. Very accurate approximations can be obtained with many rules and consequent transfer functions with many zeros and poles: they will also be computationally heavy. The exact relation they have with the theoretical definitions that also present a memory of past values of \( z(t) \) must still be established.

References