$\int$ractional $\mathcal{C}$alculus
& $\mathcal{A}$pplied $\mathcal{A}$nalysis
An International Journal for Theory and Applications

ORIGINAL PAPER

Check for
updates

Impulse response of commensurate fractional-order systems: multiple complex poles

Dalibor Biolek[1,2] · Roberto Garrappa[3,4] · Viera Biolková[5]

Abstract

The impulse response of a fractional-order system with the transfer function $s^\delta/[(s^\alpha - a)^2 + b^2]^n$, where $n \in \mathbb{N}$, $a \in \mathbb{R}$, $b \in \mathbb{R}^+$, $\alpha \in \mathbb{R}^+$, $\delta \in \mathbb{R}$, is derived via real and imaginary parts of two-parameter Mittag-Leffler functions and their derivatives. With the aid of a robust algorithm for evaluating these derivatives, the analytic formulas can be used for an effective transient analysis of fractional-order systems with multiple complex poles. By some numerical experiments it is shown that this approach works well also when the popular SPICE-family simulating programs fail to converge to a correct solution.

Keywords Fractional calculus · Mittag-Leffler functions · Laplace transform · Complex poles · Commensurate systems · Impulse response

Mathematics Subject Classification 33E12 (primary) · 44A10 · 26A33

✉ Dalibor Biolek
biolek@vut.cz ; dalibor.biolek@unob.cz

Roberto Garrappa
roberto.garrappa@uniba.it

Viera Biolková
biolkova@vut.cz

1 Dept. of EE, University of Defence Brno, Kounicova 65, 66210 Brno, Czech Republic

2 Dept. of Microelectronics, Brno University of Technology, Technická 3058/10, 61600 Brno, Czech Republic

3 Dept. of Mathematics, University of Bari, Via E.Orabona 4, 70125 Bari, Italy

4 GNCS Group, Istituto Nazionale di Alta Matematica (INdAM), P.le Aldo Moro 5, 00185 Rome, Italy

5 Dept. of Radio Electronics, Brno University of Technology, Technická 3082/12, 61600 Brno, Czech Republic

1 Introduction

Evaluation of impulse or transient response of a linear fractional-order system from its transfer function is often part of designing and verifying the operation of analog circuits, especially frequency filters [1], oscillators [2], and generalized PID controllers (i.e., controllers including non-integer-order elements) [3].

In principle, this calculation can be performed in three ways: 1) in the environment of a standard analog circuit simulation program, as for instance SPICE; 2) using a universal program for scientific and technical calculations, e.g. MATLAB; 3) using customized software, e.g. SNAP, for the analysis of fractional-order circuits [4].

SPICE offers the possibility of a direct time response evaluation from a transfer function specified via controlled Laplace sources. However, serious numerical errors may arise because SPICE is not primarily designed for analyzing systems with dynamics of non-integer order. Another solution may be to use known efficient algorithms (or develop new ones) for transient analysis of fractional-order systems and implement them, for example, in MATLAB. An analog designer may prefer the analysis in a different software environment that would allow working in a similar style to SPICE, but with added capabilities such as the so-called (semi)symbolic analysis of the transfer function from the circuit diagram [4] or software-generated analytical formula of the impulse response using Mittag-Leffler (ML) functions [3].

The computation of the impulse response of a linear commensurate fractional-order system whose transfer function $K(s)$ contains the s operator with non-integer powers of the form $\alpha_k = p_k/q_k$, $p_k \in \mathbb{N}_0$, $q_k \in \mathbb{N}$, can then proceed in the following steps:

- converting the transfer function to the domain of the operator $w = s^\alpha$ with $\alpha = 1/q$ and q the least common multiple of the q_k coefficients [5];
- decomposing the transfer function $K(w)$ into partial fractions;
- converting each partial fraction from Laplace to time domain by means of ML functions and their derivatives [3].

For the case of real simple and multiple w-domain poles, commonly used correspondences between the Laplace and time domains are available, which can be summarized by the correspondence [6]

$$\frac{s^\delta}{(s^\alpha - a)^k} \triangleq \frac{t^{\alpha k - \delta - 1}}{(k-1)!} E_{\alpha, \alpha-\delta}^{(k-1)}(at^\alpha), \tag{1.1}$$

where $k \in \mathbb{N}$, $a \in \mathbb{R}$, $\alpha \in \mathbb{R}^+$, $\delta \in \mathbb{R}$ and $E_{\alpha,\beta}^{(m)}(z)$ denotes the m-th order derivative of the ML function (see Section 2).

The case of complex poles corresponding to the denominators of w-domain transfer functions with second-order polynomials has received marginal attention in the literature so far. One possible reason is that the corresponding impulse responses can be obtained indirectly also using (1.1) if decomposition into partial fractions is performed for the corresponding conjugate pairs of complex poles, i.e. with the denominators in the form of binomials with complex coefficients instead of trinomials with real coefficients. One of the few attempts at a direct solution is given in [7] for simple poles,

where a correspondence is found to exist between the Laplace transform of the form $b/(s^{2\alpha} + as^{\alpha} + b)$ and the corresponding impulse response formula expressed via the ML function. In [8], impulse responses for the transfer function $1/(s^{2\alpha} + as^{\alpha} + b)^{\gamma}$ are analyzed, but only for $0 < \gamma < 2$. Moreover, the impulse response formula is found only for a special, not a general, constellation of parameters. It is also known that the impulse response corresponding to the "three-term transfer function" or "three-term differential equation" can be expressed using an infinite series of ML functions [6]; however, using series of ML functions is not very efficient for numerical computations.

There are two main contributions in this work:

– finding the missing Laplace-time correspondence for multiple complex poles;
– suggesting a reliable algorithm for evaluating two-parameter ML functions and their derivatives.

This builds a basis for an algorithmic generation of the analytic formulas of the impulse or step responses, involving the basal functions (ML functions and their derivatives as functions of time) associated with arbitrary transfer functions $K(w)$. By evaluating these formulas, numerical results are then obtained.

Since the relation (1.1) is used to derive the above missing correspondence, its validity is first analyzed in Section 3.1 with respect to the range of values of the relevant parameters a, α and β. Then, in Sections 3.2 and 3.3, the relationships between Laplace transforms and time responses are derived for a conjugate pair of simple and also multiple complex w-domain poles. Section 4 shows examples of the numerical analysis of time responses, using the derived relations and algorithms for evaluating two-parameter ML functions and their derivatives. A comparison with the outputs of three types of SPICE simulators illustrates the robustness of the proposed procedure even in cases of a fatal failure of the SPICE algorithms.

2 The Mittag–Leffler function and its derivatives

For $\alpha \in \mathbb{C}$ with $\mathrm{Re}(\alpha) > 0$, $\beta \in \mathbb{C}$, and any complex argument z, the ML function is defined by means of its series expansion

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}, \tag{2.1}$$

and it is an entire function of order $1/\mathrm{Re}(\alpha)$ and type 1, [9]. Its derivatives can be easily determined after derivation of each term in the series as

$$E_{\alpha,\beta}^{(m)}(z) = \frac{\mathrm{d}^m}{\mathrm{d}z^m} E_{\alpha,\beta}(z) = \sum_{j=0}^{\infty} \frac{(j+m)!}{j!} \frac{z^j}{\Gamma(\alpha j + \alpha m + \beta)}, \tag{2.2}$$

where $m \in \mathbb{N}_0$. For the purpose of analyzing fractional-order systems, one can restrict the definition domain of parameters α and β to the set of real numbers. Then (2.1) and (2.2) hold for $\alpha \in \mathbb{R}^+$, $\beta \in \mathbb{R}$ and $m \in \mathbb{N}_0$.

Derivatives of the two-parameter ML function (2.2) can be more conveniently represented in terms of the three-parameter ML function (also known as the Prabhakar function [10, 11])

$$E_{\alpha,\beta}^{\gamma}(z) = \sum_{j=0}^{\infty} \frac{(\gamma)_j}{j!} \frac{z^j}{\Gamma(\alpha j + \beta)},$$

where $(\gamma)_j = \gamma(\gamma+1)\cdots(\gamma+j-1) = \Gamma(\gamma+j)/\Gamma(\gamma)$ is the Pochhammer symbol, in view of the relationship

$$E_{\alpha,\beta}^{(m)}(z) = m! E_{\alpha,\alpha m+\beta}^{m+1}(z)$$

and since for $m \in \mathbb{N}_0$ it is $\Gamma(m+1) = m!$.

Specific numerical methods [12–19] have been devised to evaluate ML functions and their derivatives. The availability of corresponding MATLAB implementations [20–23] allows to apply the outlined procedure to the transfer function and determine the impulse and step response.

3 Impulse responses in terms of ML functions

To provide a proper representation of the impulse response of a linear commensurate fractional-order system we examine different cases according to the nature of the poles of the transfer function.

3.1 Multiple real poles

After substituting (2.2) into the right-hand side of (1.1) and applying the Laplace transform, we get

$$\mathcal{L}\left(\frac{t^{\alpha k-\delta-1}}{(k-1)!} E_{\alpha,\alpha-\delta}^{(k-1)}(at^{\alpha}) \, ; \, s\right)$$

$$= \frac{1}{(k-1)!} \sum_{j=0}^{\infty} \frac{(j+k-1)! a^j}{j!} \mathcal{L}\left(\frac{t^{\alpha k+\alpha j-\delta-1}}{\Gamma(\alpha j + \alpha k - \delta)} \, ; \, s\right)$$

$$= \frac{1}{(k-1)!} \sum_{j=0}^{\infty} \frac{(j+k-1)! a^j}{j! s^{\alpha j+\alpha k-\delta}} = \frac{1}{s^{\alpha k-\delta}} \sum_{j=0}^{\infty} \binom{j+k-1}{j} \left(\frac{a}{s^{\alpha}}\right)^j$$

whenever $\mathrm{Re}(\alpha j + \alpha k - \delta) > 0$, namely for $\mathrm{Re}(\delta) < k\,\mathrm{Re}(\alpha)$. The above infinite series converges absolutely, and

$$\sum_{j=0}^{\infty} \binom{j+k-1}{j} \left(\frac{a}{s^{\alpha}}\right)^j = \left(1 - \frac{a}{s^{\alpha}}\right)^{-k}$$

for sufficiently large $|s|$, namely when

$$|s^\alpha| > |a|. \tag{3.1}$$

Hence, after rearranging some terms, one easily obtains

$$\mathcal{L}\left(\frac{t^{\alpha k - \delta - 1}}{(k-1)!} E_{\alpha,\alpha-\delta}^{(k-1)}(at^\alpha); \ s\right) = \frac{s^\delta}{(s^\alpha - a)^k} \tag{3.2}$$

which is the relation (1.1) holding for arguments $a \in \mathbb{C}$ as well. It follows from the above derivations that, when condition (3.1) is satisfied, and if we restrict (just for simplicity) to real parameters $\alpha \in \mathbb{R}^+$ and $\delta \in \mathbb{R}$, the relations (1.1) and (3.2) hold in general for any $k \in \mathbb{N}$ and $\delta < k\alpha$. Thus, (1.1) can be also used for calculations of time responses for complex w-domain poles.

Parameters $\alpha \in \mathbb{R}^+$ and $\delta \in \mathbb{R}$ can be however chosen independently, without the additional coupling condition $\delta < k\alpha$. This means, for instance, that (3.2) also holds for $\delta \geq k\alpha$ when the order of the numerator of the transfer function is greater than or equal to the order of the denominator, which is an indication that the corresponding time response will contain derivatives of the Dirac impulse of non-integer order.

The q-th order derivative of the Dirac impulse can be indeed written in the form [24, 25]

$$\mathcal{L}^{-1}\left(s^q; \ t\right) = \frac{1}{\Gamma(-q)} \frac{1}{t^{q+1}}, \quad q \in \mathbb{R}^+, \quad t > 0,$$

and these derivatives can be directly included in the time function on the right-hand side of (1.1) when $\delta > k\alpha$.

This procedure can be illustrated by a simple example for $k = 1$ and $\delta \geq \alpha$ via the following arrangement:

$$\frac{s^\delta}{s^\alpha - a} = s^{\delta-\alpha} \frac{s^\alpha}{s^\alpha - a} = s^{\delta-\alpha}\left(1 + \frac{a}{s^\alpha - a}\right) = s^{\delta-\alpha} + a\frac{s^{\delta-\alpha}}{s^\alpha - a}$$

$$\triangleq \frac{1}{\Gamma(\alpha - \delta)} \frac{1}{t^{\delta-\alpha+1}} + at^{2\alpha-\delta-1} E_{\alpha,2\alpha-\delta}(at^\alpha).$$

The first term in the time response represents the derivative of the Dirac impulse of order $\delta - \alpha$. If the inequality $\delta - \alpha \geq \alpha$ (or, equivalently, $\delta \geq 2\alpha$) holds, then this would mean that the second term could be used to split off another derivative of the Dirac impulse of order $\delta - 2\alpha$.

$$\frac{s^\delta}{s^\alpha - a} \triangleq \frac{1}{\Gamma(\alpha - \delta)} \frac{1}{t^{\delta-\alpha+1}} + \frac{a}{\Gamma(2\alpha - \delta)} \frac{1}{t^{\delta-2\alpha+1}} + a^2 t^{3\alpha-\delta-1} E_{\alpha,3\alpha-\delta}(at^\alpha).$$

The same procedure can be repeated until the time response is decomposed into a sequence of derivatives of Dirac impulses and a residual function whose corresponding

Laplace transform has a pseudo-polynomial in the denominator of higher order than in the numerator. Then the ML function occurring in the residual function has the second parameter greater than 0.

3.2 Simple complex poles

To analyze transfer functions with simple complex poles, let us consider a conjugate pair of complex w-domain poles with multiplicity 1, i.e. $w_1 = c$, $w_2 = c^\star$, where

$$c = a + ib, \quad c^\star = a - ib \tag{3.3}$$

and the corresponding transfer function with its decomposition into partial fractions is

$$\frac{b}{(w-a)^2 + b^2} = \frac{b}{(w-c)(w-c^\star)} = \frac{i}{2}\left(\frac{1}{w-c^\star} - \frac{1}{w-c}\right). \tag{3.4}$$

After substituting $w = s^\alpha$, and imposing the condition

$$|s| > |c|^{1/\alpha}, \tag{3.5}$$

the terms within the brackets in (3.4) can be converted to the time domain according to (1.1). Then

$$\frac{b}{(s^\alpha - a)^2 + b^2} = \frac{i}{2}\left(\frac{1}{s^\alpha - c^\star} - \frac{1}{s^\alpha - c}\right) \triangleq \frac{i}{2}t^{\alpha-1}\left(E_{\alpha,\alpha}(c^\star t^\alpha) - E_{\alpha,\alpha}(ct^\alpha)\right).$$

After multiplying the Laplace transform (3.4) by the term s^δ and applying the relation (1.1), we get

$$\frac{bs^\delta}{(s^\alpha - a)^2 + b^2} \triangleq \frac{i}{2}t^{\alpha-\delta-1}\left(E_{\alpha,\alpha-\delta}(c^\star t^\alpha) - E_{\alpha,\alpha-\delta}(ct^\alpha)\right).$$

For any $z \in \mathbb{C}$, by writing $z = \rho\cos\theta + i\rho\sin\theta$, with $\rho = |z|$ and $\theta = \arg(z)$, it is elementary to observe that

$$E_{\alpha,\beta}(z) - E_{\alpha,\beta}(z^\star) = \sum_{j=0}^{\infty}\left[\frac{\rho^j\cos j\theta + i\rho^j\sin j\theta}{\Gamma(\alpha j + \beta)} - \frac{\rho^j\cos j\theta - i\rho^j\sin j\theta}{\Gamma(\alpha j + \beta)}\right]$$

$$= 2i\sum_{j=0}^{\infty}\frac{\rho^j\sin j\theta}{\Gamma(\alpha j + \beta)} = 2i\,\mathrm{Im}\left(E_{\alpha,\beta}(z)\right)$$

and, hence, a minor rearrangement yields the correspondence

$$\frac{bs^\delta}{(s^\alpha - a)^2 + b^2} \triangleq t^{\alpha-\delta-1}\,\mathrm{Im}\left(E_{\alpha,\alpha-\delta}(ct^\alpha)\right) \tag{3.6}$$

which is the basic formula for the correspondence between the Laplace and time domains for the case of simple complex poles.

Since numerical algorithms for the computation of ML functions [21, 23] usually make use of complex arithmetic, the relation (3.6) can be directly used for efficient time response computations.

3.3 Multiple complex poles

For the analysis of transfer functions with multiple complex poles of multiplicity $n \in \mathbb{N}$, we consider again the conjugate pair of complex poles (3.3) and the transfer function

$$\frac{s^\delta}{\left[(s^\alpha - a)^2 + b^2\right]^n} = \frac{s^\delta}{(s^\alpha - c)^n (s^\alpha - c^\star)^n}$$

with its partial fractional decomposition

$$\frac{s^\delta}{\left[(s^\alpha - a)^2 + b^2\right]^n} = \sum_{k=1}^n P_k \left(\frac{s^\delta}{(s^\alpha - c)^k} + (-1)^k \frac{s^\delta}{(s^\alpha - c^\star)^k} \right), \quad (3.7)$$

where

$$P_k = (-1)^{n-k} \binom{2n - k - 1}{n - k} \frac{1}{(2ib)^{2n-k}}. \quad (3.8)$$

Each of the terms in the right-hand side of (3.7) can be converted to the time domain thanks to the relation (1.1), after replacing the real pole with a complex one and considering the convergence condition (3.5), i.e.

$$\frac{s^\delta}{\left[(s^\alpha - a)^2 + b^2\right]^n} \triangleq \sum_{k=1}^n P_k \frac{t^{\alpha k - \delta - 1}}{(k-1)!} \left[E_{\alpha, \alpha - \delta}^{(k-1)}(c t^\alpha) + (-1)^k E_{\alpha, \alpha - \delta}^{(k-1)}(c^\star t^\alpha) \right]. \quad (3.9)$$

By using (2.2), and following the same reasoning in the previous Section, it is possible to see that

$$E_{\alpha, \beta}^{(m)}(z) + (-1)^{m+1} E_{\alpha, \beta}^{(m)}(z^\star) = \left(1 + (-1)^{m+1} \right) \sum_{j=0}^\infty \frac{(j+m)!}{j!} \frac{\rho^j \cos j\theta}{\Gamma(\alpha j + \alpha m + \beta)} +$$

$$+ \left(1 - (-1)^{m+1} \right) i \sum_{j=0}^\infty \frac{(j+m)!}{j!} \frac{\rho^j \sin j\theta}{\Gamma(\alpha j + \alpha m + \beta)}$$

where, again, we exploited the representation $z = \rho \cos\theta + i\rho \sin\theta$ for $z \in \mathbb{C}$. Therefore, we are able to evaluate

$$
E^{(k-1)}_{\alpha,\alpha-\delta}(ct^\alpha) + (-1)^k E^{(k-1)}_{\alpha,\alpha-\delta}(c^\star t^\alpha) =
\begin{cases}
2\,\mathrm{Re}\left(E^{(k-1)}_{\alpha,\alpha-\delta}(ct^\alpha) \right) & \text{k even} \\
2\mathrm{i}\,\mathrm{Im}\left(E^{(k-1)}_{\alpha,\alpha-\delta}(ct^\alpha) \right) & \text{k odd}
\end{cases}
$$

thanks to which formula (3.9) can be rearranged into the series

$$
\frac{s^\delta}{\left[(s^\alpha - a)^2 + b^2\right]^n} \triangleq
$$
$$
\triangleq -\frac{A_1}{0!} t^{\alpha-\delta-1} \,\mathrm{Im}\left(E_{\alpha,\alpha-\delta}(ct^\alpha) \right) - \frac{A_2}{1!} t^{2\alpha-\delta-1} \,\mathrm{Re}\left(E^{(1)}_{\alpha,\alpha-\delta}(ct^\alpha) \right)
$$
$$
+ \frac{A_3}{2!} t^{3\alpha-\delta-1} \,\mathrm{Im}\left(E^{(2)}_{\alpha,\alpha-\delta}(ct^\alpha) \right) + \frac{A_4}{3!} t^{4\alpha-\delta-1} \,\mathrm{Re}\left(E^{(3)}_{\alpha,\alpha-\delta}(ct^\alpha) \right)
$$
$$
- \frac{A_5}{4!} t^{5\alpha-\delta-1} \,\mathrm{Im}\left(E^{(4)}_{\alpha,\alpha-\delta}(ct^\alpha) \right) - \frac{A_6}{5!} t^{6\alpha-\delta-1} \,\mathrm{Re}\left(E^{(5)}_{\alpha,\alpha-\delta}(ct^\alpha) \right) + \ldots
$$

whose last n-th term contains the n-th derivative of the two-parameter ML function of the real or the imaginary part of the argument, depending on whether n is even or odd. Coefficients A_k are derived from the P_k coefficients (3.8) as follows

$$
A_k = 2(-\mathrm{i})^k P_k = 2(-1)^k \binom{2n-k-1}{n-k} \frac{1}{(2b)^{2n-k}}, \quad k = 1, 2, \ldots, n. \quad (3.10)
$$

For convenience, the above series can be written in the more compact form

$$
\frac{s^\delta}{\left[(s^\alpha - a)^2 + b^2\right]^n} \triangleq
$$
$$
\triangleq t^{\alpha-1} \sum_{k=0}^{[(n-1)/2]} (-1)^{k+1} A_{2k+1} \frac{t^{2k\alpha-\delta}}{(2k)!} \,\mathrm{Im}\left(E^{(2k)}_{\alpha,\alpha-\delta}(ct^\alpha) \right) \qquad (3.11)
$$
$$
+ t^{\alpha-1} \sum_{k=0}^{[n/2-1]} (-1)^{k+1} A_{2(k+1)} \frac{t^{(2k+1)\alpha-\delta}}{(2k+1)!} \,\mathrm{Re}\left(E^{(2k+1)}_{\alpha,\alpha-\delta}(ct^\alpha) \right)
$$

with $[x]$ denoting the integer part of x.

4 Numerical simulations

The relationships (3.6) and (3.11) between the Laplace and the time domains for complex poles will be used in this section to test the accuracy and robustness of the numerical transient analysis of fractional-order systems with different types of transfer functions, with both simple and multiple w-domain poles.

To exploit the relations (3.6) and (3.11) numerical algorithms are needed to evaluate two-parameter ML functions, and their derivatives, with complex arguments and for different sets of real parameters α and β.

Despite the special role played by ML functions in analysis and computation of fractional-order systems (essentially, the same role of the exponential function for integer-order systems), programming languages and computational packages usually do not provide built-in functions for the computation of ML functions and their derivatives; therefore it is necessary to rely on routines devised in the context of other research works.

One of the first available codes is the Matlab `mlf.m` function by I. Podlubny and M. Kacenak [23]; this is a code based on a mix of asymptotic expansions and numerical integration of an integral representation of the ML function, in accordance with the analysis previously provided in [14]. It is of historical importance since it has been the first available Matlab tool for computing ML functions and it made possible to begin investigations based on this function. However the code [23] is not designed to evaluate derivatives of the ML function and therefore it is not suitable to handle systems with multiple poles.

In [3] it was devised an algorithm to evaluate derivatives of the ML function by truncating the series representation (2.2); this method, however, can suffer from unexpected numerical cancellation due to the possible presence of series terms with large absolute value and alternating sign.

The problem of specifically computing derivatives of the ML functions was faced in [13] as well; essentially this work concerns with the computation of ML functions with matrix arguments, by exploiting the Schur-Parlett algorithm [26], which, in turns, demands for the computation of (possibly high-order) derivatives of the ML function in arguments related to the eigenvalues of the matrix argument, and, hence, in the whole complex plane. The algorithm proposed in [13] is based on the numerical inversion of the Laplace transform of the derivative of the ML function, thus generalizing previous algorithms devised in [12] and [19]. To improve stability of the algorithms, when high-order derivatives are requested, the algorithm in [13] exploits a formula for the representation of higher-order derivatives in terms of lower-order derivatives. The Matlab function $\mathtt{ml\_deriv}(z, \alpha, \beta, k)$, extracted from the code devised in [13], and evaluating the k-th order derivative of the ML function with two parameters α and β at each entry of the vector z, is therefore used in the numerical experiments and the corresponding results are simply denoted as ML.

The results are hence compared with those obtained from the transient analysis in popular SPICE-family simulating programs, namely PSPICE, LTSpice, and Micro-Cap (and labelled, respectively, as PS, LT and MC), which enable a direct specification of the transfer function of a fractional-order system using the controlled Laplace source. SPICE evaluates the respective time waveforms via the convolution of the driving signal and the impulse response, the latter being computed from samples of the corresponding frequency response. The transient analysis of classical integer-order systems then leads to known errors associated with specific types of transfer functions, leading to a "no causal" impulse response.

As a benchmark for the transient analysis, we consider the transfer function

$$K(s) = s^\delta \left[\frac{a^2 + b^2}{(s^\alpha - a)^2 + b^2} \right]^n \tag{4.1}$$

with the parameters $a = -628.319$, $b = 6252$ chosen so that the transfer function (4.1) corresponds for $n = 1$ and $\alpha = 1$ to a 2-nd order system with a characteristic frequency of 1 kHz and a quality factor of 5. The time waveforms corresponding to (4.1) for this constellation of parameters are well known and the results of the transient analysis are therefore easy to verify.

Errors in the transient analysis of more complex transfer functions can also be detected by testing the limiting values of the response for $t \to 0^+$ and $t \to \infty$. The denominator in (4.1) has all s-roots in the closed left-hand complex half-plane if

$$a < \frac{b}{\tan(\alpha\pi/2)}, \quad \alpha > 0. \tag{4.2}$$

Then, according to the final value theorem, the time waveform $f(t)$, corresponding to the Laplace transform (4.1), for $t \to \infty$ converges to 0 when $\delta > -1$, and to 1 when $\delta = -1$. For $\delta < -1$ this limit is improper since the term s^δ then represents a multiple pole at zero.

The limit value of the impulse characteristic at $t \to 0^+$ is, according to the initial value theorem, $f(0^+) = 0$ for $\delta < 2n\alpha - 1$ and $f(0^+) = (a^2 + b^2)^n$ for $\delta = 2n\alpha - 1$. For $\delta > 2n\alpha - 1$ the limit takes infinity values.

In the transient analysis, the fractional-order system modeled by the transfer function (4.1) was driven by a unit step. Then, to obtain the impulse response, it is necessary to multiply the transfer function by the s operator. Since different modifications of the corresponding algorithms are implemented in the above SPICE-family programs, it was interesting to compare the results obtained.

The comparisons are shown in Figures 1 and 2 in a uniform coordinate system, namely the response $f(t)$ vs. time t.

The results (a)-(f) in Figure 1 for $n = 1$ are obtained for the fixed parameters α, a, b while varying the parameter δ. They confirm the stable behavior of procedures based on the computation of the ML function. The case (a) represents a double pole at zero, which leads to unbounded growth of the response over time. From among the SPICE programs, only Micro-Cap correctly evaluates this unstable behavior. PSpice generates a fundamentally wrong response and the simulation in LTSpice does not run at all (the Laplacian is singular at DC).

The simulations (b) and (c) satisfy the test for the limiting values $f(0^+) = 0$ and $f(\infty) = 1$. For the case (d), i.e. $\delta = 0.8$, the limiting value changes at the origin to $f(0^+) = a^2 + b^2 = 3.948 \times 10^7$, which is confirmed for all the computational methods. However, the results show that all SPICE simulators introduce an error in the steady state calculation.

The simulation for the parameters in the plot (e) for $\delta = 1$ is handled by SPICE programs only with large errors, while the ML algorithm based on the Mittag-Leffler functions gives accurate results. When the parameter δ is increased further (see the plot
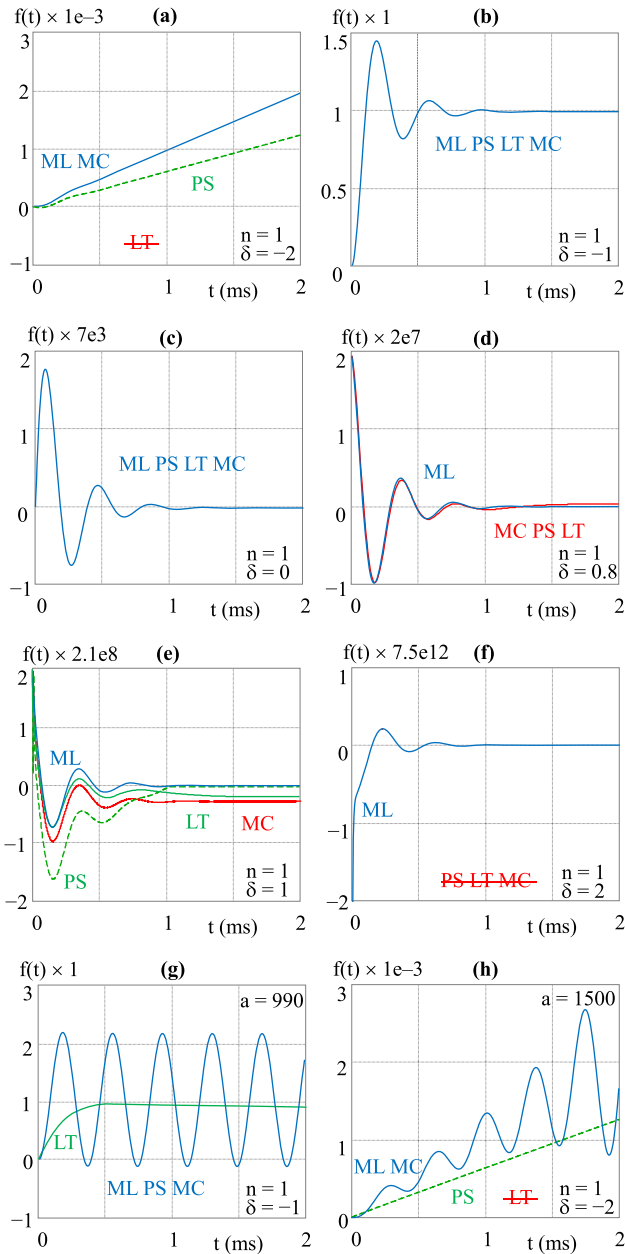
Fig. 1 Waveforms corresponding to the transfer function (4.1) with simple complex poles ($n = 1$) for $\alpha = 0.9$, $a = -628.319$ (if not specified otherwise) and $b = 6252$. The waveforms are computed in MATLAB via the algorithm [13] (ML), via the transient analysis in PSpice 16.5 (PS), LTSPICE XVII (LT), and Micro-Cap 12 (MC). Failing methods are marked by a strikeout (e.g. PS)
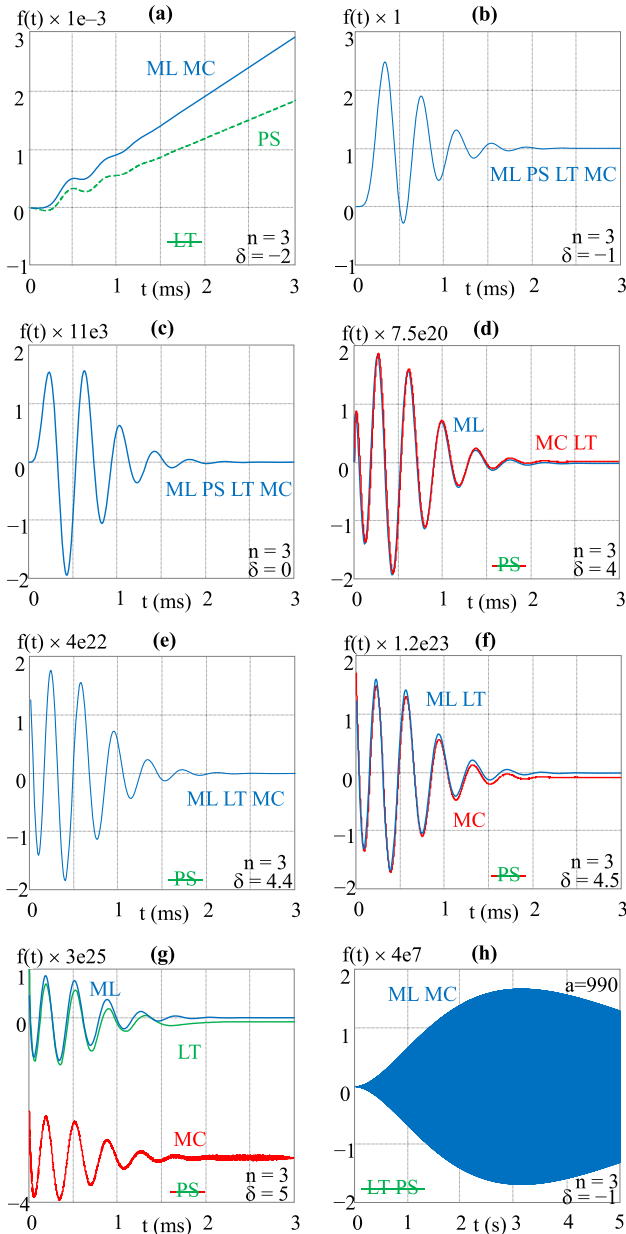
Fig. 2 Waveforms corresponding to the transfer function (4.1) with multiple complex poles ($n = 3$) for $\alpha = 0.9$, $a = -628.319$ (if not specified otherwise) and $b = 6252$. The waveforms are computed in MATLAB via the algorithm [13] (ML), via the transient analysis in PSpice 16.5 (PS), LTSPICE XVII (LT), and Micro-Cap 12 (MC). Failing methods are marked by a strikeout (e.g. PS)

(f)), the SPICE programs start to fail fatally. Only the ML method exhibits problem-free behavior.

Plot (g) shows the simulation results for $a = 990$, which is just below the 990.22 threshold between the stable and unstable response as given in (4.2). The corresponding oscillating response is confirmed by the ML method as well as by the PS and MC simulations. The PS simulation fails in this case. Finally plot (h) shows a more complex case of unstable behavior, simultaneously induced by a conjugate pair of complex poles in the instability region for $a = 1500$ and a double pole at zero. The response is correctly calculated only by the ML and MC methods.

Figure 2 shows examples of testing the methods of transient analysis for the more complex case of multiple poles, in this case $n = 3$.

Plots (a)-(g) are again ordered for increasing values of δ, which implies a successive differentiation of the responses with respect to time. All tested methods show satisfactory results for $\delta = -1$ and $\delta = 0$. Outside this interval errors can be successively observed for all SPICE programs without exception. When $\delta = 4.4$ is reached (see plot (e)), the limiting value of $f(0^+)$ goes from 0 to $(a^2 + b^2)^3 = 5.972 \times 10^{22}$, which is confirmed by the ML, LT and MC methods. The PSPICE does not converge for these parameters and even for higher values of δ. Plots (f) and (g) exhibit a high sensitivity of the results from LT and MC to δ increasing above the critical value of 4.4. Then only the ML method generates usable outputs.

Figure (h) finally shows the simulation of system (4.1) for $n = 3$ when it is set to be close to the instability limit by the parameter $a = 990$. Only the ML and MC methods passed this challenging test successfully.

The outputs of these simulations for gradually increasing values of δ also illustrate the robustness of the balancing derivatives algorithm implemented in [13] and used by the ML method; this method indeed generates correct results even for high values of δ, when instead SPICE programs fail.

5 Concluding remarks

The dictionary of current Laplace-time correspondences for fractional-order systems is completed by adding the missing correspondences (3.6) and (3.11) for single and multiple w-domain poles. This addition now allows the automated generation of analytical formulas of impulse responses using two-parameter ML functions and their derivatives for any constellation of poles, i.e., real and complex, simple and with arbitrary multiplicity. It can also be potentially used for developing useful software tools for the symbolic analysis of analog networks containing fractional-order elements.

The MATLAB code for evaluating the derivatives of two-parameter ML functions, based on the algorithm described in [13], is used and its robustness is demonstrated on benchmark simulations.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Soltan, A., Radwan, A.G., Soliman, A.M.: Fractional order Sallen-Key and KHN filters: Stability and poles allocation. Circuits, Systems, and Signal Processing **34**(5), 1461–1480 (2015). https://doi.org/10.1007/s00034-014-9925-z
2. Radwan, A.G., Elwakil, A.S., Soliman, A.M.: Fractional-order sinusoidal oscillators: Design procedure and practical examples. IEEE Transactions on Circuits and Systems I: Regular Papers **55**(7), 2051–2063 (2008). https://doi.org/10.1109/TCSI.2008.918196
3. Monje, C.A., Chen, Y., Vinagre, B.M., Xue, D., Feliu, V.: Fractional-order Systems and Controls. Springer-Verlag, London (2010)
4. Biolek, D., Kolka, Z., Brančík, L., Biolková, V., Biolek, Z.: Symbolic analysis of networks containing fractional-order elements. In: 2021 Communication and Information Technologies (KIT), pp. 1–4 (2021). https://doi.org/10.1109/KIT52904.2021.9583762
5. Petráš, I.: Stability of fractional-order systems with rational orders: a survey. Fract. Calc. Appl. Anal. **12**(3), 269–298 (2009)
6. Podlubny, I.: Fractional Differential Equations, Mathematics in Science and Engineering, vol. 198. Academic Press Inc, San Diego, CA (1999)
7. Merrikh-Bayat, F., Karimi-Ghartemani, M.: Some properties of three-term fractional order system. Fract. Calc. Appl. Anal. **11**(3), 317–328 (2008)
8. Jiao, Z., Chen, Y.: Impulse response of a generalized fractional second order filter. Fract. Calc. Appl. Anal. **15**(1), 97–116 (2012). https://doi.org/10.2478/s13540-012-0007-2
9. Gorenflo, R., Kilbas, A.A., Mainardi, F., Rogosin, S.: Mittag-Leffler Functions, Related Topics and Applications. Springer Monographs in Mathematics. Springer, Berlin (2020). https://doi.org/10.1007/978-3-662-61550-8. Second edition
10. Giusti, A., Colombaro, I., Garra, R., Garrappa, R., Polito, F., Popolizio, M., Mainardi, F.: A practical guide to Prabhakar fractional calculus. Fract. Calc. Appl. Anal. **23**(1), 9–54 (2020). https://doi.org/10.1515/fca-2020-0002
11. Prabhakar, T.R.: A singular integral equation with a generalized Mittag Leffler function in the kernel. Yokohama Math. J. **19**, 7–15 (1971)
12. Garrappa, R., Popolizio, M.: Evaluation of generalized Mittag-Leffler functions on the real line. Adv. Comput. Math. **39**(1), 205–225 (2013). https://doi.org/10.1007/s10444-012-9274-z
13. Garrappa, R., Popolizio, M.: Computing the matrix Mittag-Leffler function with applications to fractional calculus. J. Sci. Comput. **77**(1), 129–153 (2018). https://doi.org/10.1007/s10915-018-0699-5
14. Gorenflo, R., Loutchko, J., Luchko, Y.: Computation of the Mittag-Leffler function $E_{\alpha,\beta}(z)$ and its derivative. Fract. Calc. Appl. Anal. **5**(4), 491–518 (2002)
15. Valério, D., Tenreiro Machado, J.A.: On the numerical computation of the Mittag-Leffler function. Commun. Nonlinear Sci. Numer. Simul. **19**(10), 3419–3424 (2014). https://doi.org/10.1016/j.cnsns.2014.03.014

16. Seybold, H., Hilfer, R.: Numerical algorithm for calculating the generalized Mittag-Leffler function. SIAM J. Numer. Anal. **47**(1), 69–88 (2008/09). https://doi.org/10.1137/070700280

17. Ingo, C., Barrick, T.R., Webb, A.G., Ronen, I.: Accurate Padé global approximations for the Mittag-Leffler function, its inverse, and its partial derivatives to efficiently compute convergent power series. Int. J. Appl. Comput. Math. **3**(2), 347–362 (2017). https://doi.org/10.1007/s40819-016-0158-7

18. Ortigueira, M.D., Lopes, A.M., Tenreiro Machado, J.A.: On the numerical computation of the Mittag-Leffler function. Int. J. Nonlinear Sci. Numer. Simul. **20**(6), 725–736 (2019). https://doi.org/10.1515/ijnsns-2018-0358

19. Garrappa, R.: Numerical evaluation of two and three parameter Mittag-Leffler functions. SIAM J. Numer. Anal. **53**(3), 1350–1369 (2015). https://doi.org/10.1137/140971191

20. Petráš, I.: Fractional derivatives, fractional integrals, and fractional differential equations in matlab. In: A.H. Assi (ed.) Engineering Education and Research Using MATLAB, chap. 10. IntechOpen, Rijeka (2011). https://doi.org/10.5772/19412

21. Garrappa, R.: The Mittag–Leffler function. https://www.mathworks.com/matlabcentral/fileexchange/48154-the-mittag-leffler-function (2015). [Online; accessed March 27, 2022]

22. Xue, D.: FOTF toolbox. https://www.mathworks.com/matlabcentral/fileexchange/60874-fotf-toolbox (2017). [Online; accessed March 27, 2022]

23. Podlubny, I., Kačeňák, M.: Mittag–Leffler function. https://www.mathworks.com/matlabcentral/fileexchange/8738-mittag-leffler-function (2017). [Online; accessed March 27, 2022]

24. Makris, N.: The fractional derivative of the Dirac delta function and additional results on the inverse Laplace transform of irrational functions. Fractal and Fractional **5**(1) (2021). https://doi.org/10.3390/fractalfract5010018

25. Feng, Z., Ye, L., Zhang, Y.: On the fractional derivative of Dirac delta function and its application. Adv. Math. Phys. pp. Art. ID 1842945, 7 (2020). https://doi.org/10.1155/2020/1842945

26. Davies, P.I., Higham, N.J.: A Schur-Parlett algorithm for computing matrix functions. SIAM J. Matrix Anal. Appl. **25**(2), 464–485 (2003). https://doi.org/10.1137/S0895479802410815