



Step Response-Based Identification of Fractional Order Time Delay Models

Salim Ahmed¹ 

Received: 11 December 2018 / Revised: 31 December 2019 / Accepted: 2 January 2020 /

Published online: 11 January 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

An identification method for fractional order models with time delay is presented. The proposed method, based on the output error optimization, simultaneously estimates model orders, coefficients and time delay from a single noisy step response. Analytical expressions for logarithmic derivatives of the step input are derived to evaluate the Jacobian and the Hessian required for the Newton's algorithm for optimization. A simplified initialization procedure is also outlined that assumes an integral initial order and uses estimated coefficients as the initial guess. Simulation results are presented to demonstrate the efficacy of the proposed approach. Convergence of the Newton's method and the Gauss–Newton scheme are also studied in simulation. Identification results from noisy step response data for time delay models with different structures are presented.

Keywords Step response · Fractional order · Parameter estimation · Time delay · Optimization

1 Introduction

Fractional order model identification faces an additional challenge as it requires to estimate the model orders on a large space. Moreover, if there is a time delay, the parameter estimation problem becomes further complicated due to the nonlinear appearance of the delay term in the model equation. To overcome these issues, several approaches have been adopted in the literature. Victor et al. [40] used a two-stage algorithm to estimate the orders using an optimization approach and estimated the model coefficients by solving a least-squares equation. Narang et al. [25] used a similar approach for time delay models. There are only a few methods for simultaneous estimation

✉ Salim Ahmed
sahmed@mun.ca

¹ Centre for Risk, Integrity and safety Engineering (C-RISE), Department of Process Engineering, Memorial University, St. John's, NL, Canada

of all parameters of fractional order models [36]. However, these methods are not applicable for the step input, although the characteristics of step responses of fractional order models have been studied in the literature [22,35,37]. To the best of the knowledge of the author, there is no reported method that simultaneously estimates orders, coefficients and the delay from the step response. Ahmed [1] presented a method for estimation of coefficients and delay, however, for known fractional orders.

In this article, an output error approach for simultaneous estimation of model orders, coefficients and time delay is proposed by adopting the Newton and the Gauss–Newton optimization algorithms. The optimization approach requires estimation of the output sensitivity functions. For fractional order models, these functions include the logarithmic derivatives of the input signal. Victor and Malti [39] and Victor et al. [40] commented that simulating the logarithmic derivative is not trivial and used a numerical approach to calculate the sensitivity functions. The contribution of this article lies in its presentation of the analytical expressions for the logarithmic derivatives of the step input signal and derivations of the analytical expressions for the Jacobian and the Hessian required for the Newton’s algorithm. The efficacy of the algorithm lies in its ability to identify all the parameters simultaneously.

The output error approach is the most commonly used methodology for fractional order model identification [10,17,31,39,40]. Other approaches include an extension [18] of the so-called simplified refined instrumental variable algorithm [41], use of the state variable filter [8], fractional Laguerre basis function modeling [3], frequency domain identification [26], ARX model development using fractional order and orthonormal basis filter [23], integral-based approach [34] and so on. Fractional order nonlinear system identification problem has also been addressed in the literature [19,33]. The time delay adds an extra degree of complexity to any identification method due to its nonlinear appearance. Narang et al. [25] extended the linear filter method [2] for fractional order model identification that iteratively estimates the model parameters and the delay. Tavakoli-Kakhki and Tavazoei [36] and Yuan et al. [42] also considered time delay estimation.

Motivations for addressing the fractional order identification problem arises from the advantages of the use of the fractional order models [4,20] and the use of such models for real-life applications. Reported applications include thermal processes [12], processes involving diffusive mass transfer [5], Archimedes wave swing [38], vibration suppression [24], health monitoring [15] and other engineering applications, see, e.g., [22,40]. Another motivation is the use of fractional order controllers. With introduction of the TID controller [16], the $PI^\lambda D^\mu$ controller [30], the CRONE controller [29] and the lead-lag compensator [32], fractional order representation of process and controllers are gaining more and more attention.

The remainder of the article is organized as follows. The proposed methodology is outlined in Sect. 2 that includes the optimization algorithm, expressions for the logarithmic derivatives and the initialization procedure. Simulation results along with the simulation method for fractional order differentiation and integration are presented in Sect. 3. Concluding remarks are drawn in Sect. 4.

2 Mathematical Formulations

The proposed identification method is outlined in this section. The method is based on the output error optimization. Analytical expressions for logarithmic derivatives of the step input required to evaluate the Jacobian and the Hessian are also presented.

For convenience in presentation, both time and Laplace domain expressions are used interchangeably. The Laplace domain expressions are presented in terms of ‘s’ while an equivalent symbol ‘p’ is used to represent the derivative operator in the time domain. Considering that step response methods are typically used for models with parsimonious structures, mathematical derivations are provided for two structures as in (1) and (2). We will refer to these two structures as Class I and Class II models, respectively, throughout the manuscript. Although the method is demonstrated using these two structures, the same approach can be followed for a general structure with more parameters.

$$\text{Class I: } G(s) = \frac{be^{-\delta s}}{s^\alpha + a} \quad (1)$$

$$\text{Class II: } G(s) = \frac{be^{-\delta s}}{s^{\alpha_2} + a_1 s^{\alpha_1} + a_0} \quad (2)$$

2.1 Parameter Estimation

For a single input single output system, the relation between the input and the output can be expressed using the following Laplace domain expression.

$$Y(s) = G(s, \boldsymbol{\theta})U(s) + E(s) \quad (3)$$

where $Y(s)$ and $U(s)$ are the input and the output, respectively, $G(s, \boldsymbol{\theta})$ is the model transfer function with $\boldsymbol{\theta} = [\theta_1, \dots, \theta_n]$ as the set of parameters with n being the total number of parameters. $E(s)$ represents the noise in the output measurements. For a fractional order model, the parameter vector contains the coefficients of the numerator and the denominator polynomials, the time delay as well as the fractional orders of the derivatives. The objective of an identification algorithm is to estimate the parameters $\boldsymbol{\theta}$ from a set of time domain measurements $[u(t_k) \ y(t_k)]$, $k = 1, 2, \dots, N$ with N being the number of available data points. The goal of the output error approach is to estimate $\boldsymbol{\theta}$, by minimizing a norm of the errors between measured and model outputs.

$$E(s, \boldsymbol{\theta}) = Y(s) - G(s, \boldsymbol{\theta})U(s) \quad (4)$$

An equivalent time domain expression for the error is given by

$$e(t, \boldsymbol{\theta}) = y(t) - G(p, \boldsymbol{\theta})u(t) \quad (5)$$

The lower case letters correspond to the variables in the time domain. Using the notation $e_k = e(t_k, \boldsymbol{\theta})$, the goal of an output error algorithm can be defined as to minimize the following objective function

$$f(\boldsymbol{\theta}) = \sum_{k=1}^N e_k^2 = \|\mathbf{e}\|^2 \quad (6)$$

A number of different approaches can be taken for solution of the optimization problem. We follow the Newton's algorithm to simultaneously estimates all the parameters. In this approach, estimated parameters at an iteration step $i + 1$ is given by

$$\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i - [\mathbf{H}(\boldsymbol{\theta}^i)]^{-1} g(\boldsymbol{\theta}^i) \quad (7)$$

where $g(\boldsymbol{\theta})$ is the gradient of $f(\boldsymbol{\theta})$ given by

$$g(\boldsymbol{\theta}) = 2\mathbf{A}\mathbf{e} \quad (8)$$

with \mathbf{A} being the Jacobian matrix.

$$\mathbf{A} = [\nabla e_1 \ \nabla e_2 \ \cdots \ \nabla e_N] \quad (9)$$

The columns of \mathbf{A} are the first derivative vectors ∇e_k of the components of \mathbf{e} , i.e.,

$$A_{jk} = \frac{\partial e_k}{\partial \theta_j} \quad j = 1, \dots, n \quad k = 1, \dots, N \quad (10)$$

where θ_j is the j -th element of $\boldsymbol{\theta}$. The Hessian, \mathbf{H} is defined as

$$\mathbf{H} = \nabla^2 f(\boldsymbol{\theta}) \quad (11)$$

For the Hessian, it requires

$$\begin{aligned} \frac{\partial^2}{\partial \theta_l \partial \theta_j} \sum_{k=1}^N e_k^2 &= 2 \frac{\partial}{\partial \theta_l} \sum_{k=1}^N e_k \frac{\partial e_k}{\partial \theta_j} \\ &= 2 \sum_{k=1}^N \frac{\partial e_k}{\partial \theta_l} \frac{\partial e_k}{\partial \theta_j} + 2 \sum_{k=1}^N e_k \frac{\partial^2 e_k}{\partial \theta_l \partial \theta_j} \end{aligned} \quad (12)$$

So the Hessian is given by

$$\mathbf{H} = 2\mathbf{A}\mathbf{A}^T + 2 \sum_{k=1}^N e_k \mathbf{R}_k \quad (13)$$

where \mathbf{A}^T is the transpose of \mathbf{A} , and

$$\mathbf{R}_k = \begin{pmatrix} \frac{\partial^2 e_k}{\partial \theta_1^2} & \frac{\partial^2 e_k}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 e_k}{\partial \theta_1 \partial \theta_n} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{\partial^2 e_k}{\partial \theta_n \partial \theta_1} & \frac{\partial^2 e_k}{\partial \theta_n \partial \theta_2} & \cdots & \frac{\partial^2 e_k}{\partial \theta_n^2} \end{pmatrix} \quad (14)$$

The main disadvantage of the above approach, namely the Newton's method is that it requires formulae from which the second derivative matrix can be evaluated. However, there are methods closely related to this approach which use only the first derivatives. One such approach is the finite difference Newton method. In this approach, to estimate $\mathbf{H}(\theta^i)$, increments in each coordinate direction of the output error are taken by differences in gradient vectors [11]. Another approach is the so-called quasi-Newton methods which approximate $[\mathbf{H}(\theta)]^{-1}$ by a symmetric positive definite matrix and update it as the iteration proceeds.

The second term in (13) contains e_k as multipliers which are reasonably small especially as the iteration approaches the optimum. This led to the assumption that \mathbf{H} can be approximated as

$$\mathbf{H} \approx 2\mathbf{A}\mathbf{A}^T \quad (15)$$

Thus, the basic Newton's method becomes the Gauss–Newton method when (15) is used. Accordingly, the solution of θ in the $(i + 1)$ -th iteration step is given by

$$\theta^{i+1} = \theta^i - [\mathbf{A}\mathbf{A}^T]^{-1} \mathbf{A}e \quad (16)$$

The basic Newton's method as well as the Gauss–Newton method may not be suitable for many cases since \mathbf{H} may not be positive definite when θ^i is remote from the solution. Moreover, convergence may not occur even when \mathbf{H} is positive definite [11,21]. To avoid the latter case, Newton's method with line search can be implemented where the Newton correction is used to generate a search direction [11]. Following this approach (7) is modified as

$$\theta^{i+1} = \theta^i - \lambda^i [\mathbf{H}(\theta^i)]^{-1} g(\theta^i) \quad (17)$$

The Gauss–Newton solution can also be modified similarly. The advantages and disadvantages of Newton's method and the Gauss–Newton method have been widely addressed in the literature [11]. An advantage of the Gauss–Newton method is that the second derivative matrix is approximated using the first-order derivatives. On the other hand, the Gauss–Newton method is equivalent to making a linear approximation of the residuals and hence the method is valuable either for the residuals or the degree of nonlinearity to be small [11]. While a detailed study of applicability of these two approaches is beyond the scope of this article, observations from extensive simulation results will be presented in the result section.

The optimization step follows a standard procedure. Using an initial guess of the parameters, the Jacobian and Hessian are evaluated and the parameters are iteratively updated until convergence. Next sections provide the analytical expressions for the matrices required to evaluate the Jacobian and the Hessian.

2.1.1 Class I Model

The parameter vector for the Class I model in (1) is given by $\theta = [a \ b \ \delta \ \alpha]^T$. So, the column elements of the Jacobian can be expressed in the Laplace domain as

$$\nabla E_k(s) = \left[\frac{be^{-\delta s}}{(s^\alpha+a)^2} \quad \frac{-1e^{-\delta s}}{s^\alpha+a} \quad \frac{bs e^{-\delta s}}{s^\alpha+a} \quad \frac{bs^\alpha e^{-\delta s}}{(s^\alpha+a)^2} \ln(s) \right]^T U(s) \tag{18}$$

An equivalent time domain expression is given by

$$\nabla e_k = \left[\frac{b}{(p^\alpha+a)^2} \quad \frac{-1}{p^\alpha+a} \quad \frac{bp}{p^\alpha+a} \quad \frac{bp^\alpha \ln(p)}{(p^\alpha+a)^2} \right]^T u(t_k - \delta) \tag{19}$$

Here, $\ln(s)$ and $\ln(p)$ are logarithms of the derivative operator, expressed in the Laplace and time domain, respectively. The logarithmic operation is on the operator s or p . Similarly, the matrix \mathbf{R}_k can be obtained in the time domain as

$$\mathbf{R}_k = \begin{bmatrix} \frac{-b}{(p^\alpha+a)^3} & \frac{1}{(p^\alpha+a)^2} & \frac{-bp}{(p^\alpha+a)^2} & \frac{-bp^\alpha \ln(p)}{(p^\alpha+a)^3} \\ \frac{1}{(p^\alpha+a)^2} & 0 & \frac{p}{(p^\alpha+a)} & \frac{p^\alpha \ln(p)}{(p^\alpha+a)^2} \\ \frac{-bp}{(p^\alpha+a)^2} & \frac{p}{(p^\alpha+a)} & \frac{-bp^2}{(p^\alpha+a)} & \frac{-bp^{\alpha+1} \ln(p)}{(p^\alpha+a)^2} \\ \frac{-bp^\alpha \ln(p)}{(p^\alpha+a)^3} & \frac{p^\alpha \ln(p)}{(p^\alpha+a)^2} & \frac{-bp^{\alpha+1} \ln(p)}{(p^\alpha+a)^2} & \frac{-bp^\alpha (p^\alpha - a) \ln(p) \ln(p)}{(p^\alpha+a)^3} \end{bmatrix} u(t_k - \delta) \tag{20}$$

2.1.2 Class II Model

For the Class II model as in (2), the parameter vector $\theta = [a_1 \ a_0 \ b \ \delta \ \alpha_2 \ \alpha_1]^T$. Denoting $D(p) = p^{\alpha_2} + a_1 p^{\alpha_1} + a_0$, the expressions for the elements of \mathbf{A} and \mathbf{R}_k for the Class II model can be given as

$$\nabla e_k = \left[\frac{bp^{\alpha_1}}{[D(p)]^2} \quad \frac{b}{[D(p)]^2} \quad \frac{-1}{D(p)} \quad \frac{bp}{D(p)} \quad \frac{bp^{\alpha_2} \ln(p)}{[D(p)]^2} \quad \frac{ba_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} \right]^T u(t_k - \delta) \tag{21}$$

$$\mathbf{R}_k = \begin{bmatrix} \frac{-2bp^{2\alpha_1}}{[D(p)]^3} & \frac{-2bp^{\alpha_1}}{[D(p)]^3} & \frac{p^{\alpha_1}}{[D(p)]^2} & \frac{-bp^{\alpha_1+1}}{[D(p)]^2} \\ \frac{-2bp^{\alpha_1}}{[D(p)]^3} & \frac{-2b}{[D(p)]^3} & \frac{1}{[D(p)]^2} & \frac{-bp}{[D(p)]^2} \\ \frac{p^{\alpha_1}}{[D(p)]^2} & \frac{1}{[D(p)]^2} & 0 & \frac{p}{D(p)} \\ \frac{-bp^{\alpha_1+1}}{[D(p)]^2} & \frac{-bp}{[D(p)]^2} & \frac{p}{D(p)} & \frac{-bp^2}{D(p)} \\ \frac{-2bp^{\alpha_2+\alpha_1} \ln(p)}{[D(p)]^3} & \frac{-2bp^{\alpha_2} \ln(p)}{[D(p)]^3} & \frac{p^{\alpha_2} \ln(p)}{[D(p)]^2} & \frac{-bp^{\alpha_2+1} \ln(p)}{[D(p)]^2} \\ \frac{bp^{\alpha_1} (p^{\alpha_2} - a_1 p^{\alpha_1} + a_0) \ln(p)}{[D(p)]^3} & \frac{-2ba_1 p^{\alpha_1} \ln(p)}{[D(p)]^3} & \frac{a_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} & \frac{ba_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} \\ \frac{-2bp^{\alpha_2+\alpha_1} \ln(p)}{[D(p)]^3} & \frac{bp^{\alpha_1} (p^{\alpha_2} - a_1 p^{\alpha_1} + a_0) \ln(p)}{[D(p)]^3} & \frac{-2ba_1 p^{\alpha_1} \ln(p)}{[D(p)]^3} & \frac{a_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} \\ \frac{-2bp^{\alpha_2} \ln(p)}{[D(p)]^3} & \frac{a_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} & \frac{ba_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} & \frac{ba_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} \\ \frac{p^{\alpha_2} \ln(p)}{[D(p)]^2} & \frac{ba_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} & \frac{ba_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} & \frac{ba_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} \\ \frac{bp^{\alpha_2} \ln(p)}{[D(p)]^2} & \frac{ba_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} & \frac{ba_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} & \frac{ba_1 p^{\alpha_1} \ln(p)}{[D(p)]^2} \\ \frac{bp^{\alpha_2} (-p^{\alpha_2} + a_1 p^{\alpha_1} + a_0) \ln(p) \ln(p)}{[D(p)]^3} & \frac{-2ba_1 p^{\alpha_2+\alpha_1} \ln(p) \ln(p)}{[D(p)]^3} & \frac{-2ba_1 p^{\alpha_2+\alpha_1} \ln(p) \ln(p)}{[D(p)]^3} & \frac{-2ba_1 p^{\alpha_2+\alpha_1} \ln(p) \ln(p)}{[D(p)]^3} \\ \frac{-2ba_1 p^{\alpha_2+\alpha_1} \ln(p) \ln(p)}{[D(p)]^3} & \frac{ba_1 p^{\alpha_1} (p^{\alpha_2} - a_1 p^{\alpha_1} + a_0) \ln(p) \ln(p)}{[D(p)]^3} & \frac{ba_1 p^{\alpha_1} (p^{\alpha_2} - a_1 p^{\alpha_1} + a_0) \ln(p) \ln(p)}{[D(p)]^3} & \frac{ba_1 p^{\alpha_1} (p^{\alpha_2} - a_1 p^{\alpha_1} + a_0) \ln(p) \ln(p)}{[D(p)]^3} \end{bmatrix} u(t_k - \delta) \tag{22}$$

Evaluation of A and R_k needs estimation of $\ln(p)u(t_i)$ as well as $\ln(p) \ln(p)u(t_i)$. Victor et al. [40] suggested numerical estimation of the Jacobian as logarithm of the derivative operator is not trivial to simulate. We use analytical expressions to evaluate the logarithmic derivatives of the input signal.

2.2 Evaluation of the Logarithmic Derivative

The above methodology is applicable irrespective of the input type. As this article is concerned with the step input, analytical expressions for logarithmic derivatives of the step, expressed as in (23), is derived.

$$u(t) = h\Omega(T) \quad (23)$$

where h is the size of the step input and $\Omega(T)$ is the unit step function defined as

$$\Omega(T) = \begin{cases} 0 & \text{for } t < T \\ 1 & \text{for } t \geq T \end{cases} \quad (24)$$

The log derivative, $\ln(p)$ of a constant, c , is expressed [4,14] as

$$\ln(p)c = -c(\gamma + \ln t) \quad (25)$$

where γ is the Euler–Mascheroni constant [27] given by

$$\gamma = \lim_{m \rightarrow \infty} \left(\sum_{q=1}^m \frac{1}{q} - \ln m \right) \quad (26)$$

The numerical value for the Euler-Mascheroni constant can be approximated as $\gamma \approx 0.57721$. Following the above expressions, $\ln(p)$ of a step signal can be obtained as

$$\ln(p)h\Omega(T) = -h(\gamma + \ln t)\Omega(T) \quad (27)$$

Also, estimation of R_k requires evaluation of $\ln(p) \ln(p)$ which can be obtained from

$$\ln(p) \ln(p)h\Omega(T) = \ln(p)[-h(\gamma + \ln t)\Omega(T)] \quad (28)$$

[4] derived the logarithmic derivative of the logarithmic function as

$$\ln(p) \ln t = -\zeta(2) - (\gamma + \ln t) \ln t \quad (29)$$

where ζ is the Riemann zeta function, also known as the Hurwitz function [27] given as

$$\zeta(v) = \lim_{m \rightarrow \infty} \sum_{q=1}^m \frac{1}{q^v} \quad (30)$$

Following (30), $\zeta(2)$, can be obtained as [13]

$$\zeta(2) = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \dots = \frac{\pi^2}{6} \approx 1.645 \quad (31)$$

Using (28) and (29), $\ln(p) \ln(p)$ for the step input signal can be obtained as

$$\ln(p) \ln(p) h \Omega(T) = h \left[\zeta(2) + (\gamma + \ln t)^2 \right] \Omega(T) \quad (32)$$

2.3 Implementation Issues

2.3.1 Initialization

Initialization plays an important role and poses a significant challenge for optimization schemes. In the proposed methodology, initialization of model orders, coefficients and the delay are required. For orders, we propose to initiate the optimization algorithm assuming integer values. For example, for a Class I model an order of 1 is used as the initial guess. For Class II models, a second-order model is used as the initial guess. Regarding the coefficients, we propose to estimate an integer order model using conventional identification method. In this article, the integral equation approach [9] is used to estimate the initial coefficients assuming a model without delay. The estimation procedure is detailed below. For the time delay, the initial guess can be obtained from the step response. We suggest to use a small initial value for the delay.

To describe the integral equation approach let us take an example differential equation representing the input–output relation of a process.

$$y(t) = \frac{v}{p^2 + \mu_1 p + \mu_0} u(t) + \varepsilon_1(t) \quad (33)$$

Here, v is the numerator coefficient and μ_1 and μ_0 are the denominator coefficients in the second-order model. The relation can be presented in the equation error form as

$$\frac{d^2 y(t)}{dt^2} + \mu_1 \frac{dy(t)}{dt} + \mu_0 y(t) = v u(t) + \varepsilon_2(t) \quad (34)$$

The equation is then integrated to get

$$y(t) + \mu_1 y^{[1]}(t) + \mu_0 y^{[2]}(t) = v u^{[2]}(t) + \varepsilon(t) \quad (35)$$

where for any variable, $y(t)$, $y^{[j]}(t)$ is its j -th order integral for time limit 0 to t . The estimation equation (35) can be reformulated to get it in a least-squares solution form

$$\psi(t) = \phi^T(t) \vartheta + \varepsilon(t) \quad (36)$$

where

$$\psi(t) = y(t), \quad \phi^T(t) = [-y^{[1]}(t) -y^{[2]}(t) u^{[2]}(t)], \quad \vartheta = [\mu_1 \mu_0 v]^T$$

Equation (36) can be written for $t = t_1, t_2 \dots t_N$ and combined to give the estimation equation

$$\Psi = \Phi \vartheta + \epsilon \tag{37}$$

with

$$\Psi(t) = \begin{bmatrix} \psi(t_1) \\ \psi(t_2) \\ \dots \\ \psi(t_N) \end{bmatrix}, \quad \Phi(t) = \begin{bmatrix} \phi^T(t_1) \\ \phi^T(t_2) \\ \dots \\ \phi^T(t_N) \end{bmatrix} \tag{38}$$

The parameter vector ϑ is then obtained as the solution of the least-squares equation as

$$\vartheta = (\Phi^T \Phi)^{-1} \Phi^T \Psi \tag{39}$$

2.4 Overall Algorithm

The overall methodology is summarized as Algorithm 1 taking the Class II model as an example. The same algorithm is applicable for model with other structures when appropriate equations are used.

3 Simulation Results

3.1 Simulation Environment

For the general fractional differentials and integrals of a function $\omega(t)$, the Grunwald-Letnikov (GL) definition (41) is commonly used; see, e.g., [28].

$${}_{t_0}D_t^\rho \omega(t) = \lim_{\eta \rightarrow 0} \frac{1}{\eta^\rho} \sum_{j=0}^{\lfloor \frac{t-t_0}{\eta} \rfloor} (-1)^j \binom{\rho}{j} \omega(t - j\eta) \tag{41}$$

Here, t_0 and t are the limits of the operator, η is the step size and ρ is the order with $\rho > 0$ means a derivative operation and $\rho < 0$ means integral operation. Also $\lfloor \cdot \rfloor$ means the integer part and

$$\binom{\rho}{j} = \frac{\Gamma(\rho + 1)}{\Gamma(j + 1)\Gamma(\rho - j + 1)} \tag{42}$$

with $\Gamma(\cdot)$ being the Euler’s Gamma function.

Algorithm 1: Identification algorithm for Class II models.

Step 1-Initialization

- $i = 0$
- Assume initial orders as integers; $[\alpha_2^0 \alpha_1^0] = [2 \ 1]$.
- obtain initial estimates $[\alpha_2^0 \alpha_1^0 \ b^0] = [\mu_2 \ \mu_1 \ v]$ by solving (39).
- Assume an initial delay δ^0 .
- $\theta^0 = [a_2^0 \ a_1^0 \ b^0 \ \delta^0 \ \alpha_2^0 \ \alpha_1^0]^T$.
- Choose λ^0 .

Step 2-Parameter estimation

- Evaluate ∇e_k using (21) and R_k using (22).
- Evaluate A using (9) and $g(\theta)$ using (8); evaluate H using (13) for the Newton’s method or (15) for the Gauss–Newton method.
- Estimate θ^{i+1} using (17).

Step 3-Repetition

- Estimate fractional change of each parameter as

$$\Delta\theta_j^{i+1} = \left| \frac{\theta_j^{i+1} - \theta_j^i}{\theta_j^{i+1}} \right| \tag{40}$$

for all j where θ_j^i is the i -th estimate of the j -th element of θ .

- while $\max(\Delta\theta_1^{i+1}, \Delta\theta_2^{i+1}, \dots) > tolerance, i = i + 1$, update λ^i , if required, and repeat Steps 2-3; otherwise, go to Step 4.

Step 4-Termination: When all of the parameters converge, θ^i is taken as the final estimate.

For numerical computation, a revised version of (41), presented in [6] is used where

$${}_{t_0}D_t^\rho \omega(t) = \lim_{\eta \rightarrow 0} \frac{1}{h^\rho} \sum_{j=0}^{\lfloor \frac{t-t_0}{\eta} \rfloor} w_j^{(\rho)} \omega(t - j\eta) \tag{43}$$

where $w_j^{(\rho)}$ can be evaluated recursively from

$$w_0^{(\rho)} = 1 \tag{44}$$

$$w_j^{(\rho)} = \left(1 - \frac{\rho + 1}{j} \right) w_{j-1}^{(\rho)} \quad j = 1, 2, \dots \tag{45}$$

In this work, MATLAB is used to perform the required numerical calculations. The step input is considered to be noise free. The step responses generated for corresponding fractional order models are corrupted with white Gaussian noise. Monte Carlo simulations (MCS) are performed by changing the ‘seed’ for the noise signal. The noise-to-signal ratio (NSR) is defined as the ratio of the variance of the noise to that of the signal. The results presented in this section are from 100 MCS. The presented

estimated parameters are the mean of 100 estimates. Parameters are presented along with their standard deviations. The model structures are assumed to be known.

3.2 Convergence of Iteration Schemes

The previous section outlines both the Newton's algorithm and the Gauss–Newton algorithm. Convergence issues with both of the approaches have been discussed in the optimization literature [7, 11]. In this section, a set of simulation results are presented to show the convergence of the two schemes. The following Class I model is used for this purpose.

$$G(s) = \frac{1.2e^{-7s}}{20s^\alpha + 1} \quad (46)$$

A set of models having different values of α are used. For model (46), the parameter vector is identified as $\theta^T = [a \ b \ \delta \ \alpha] = [0.05 \ 0.06 \ 7 \ \alpha]$. In the above format $G(s) = \frac{K}{\tau s^\alpha + 1}$, the parameter vector is presented as $[\tau \ K \ \delta \ \alpha] = [20 \ 1.2 \ 7 \ \alpha]$. Although the coefficients are identified as a , b , we present those as τ and K for the sake of numerical convenience and clarity in interpretation. A total of 1000 data points are used in each cases with a sampling interval 0.1. A unit step signal is used. The order is initialized with a value of 1. An integer order model is identified using the integral equation approach, and the estimated values are used for initialization. The delay is initialized with a value of 1. λ is set at 0.5.

Figure 1 shows the trajectories of all parameters and the error from initialization to convergence. The Newton's algorithm was found to show smooth trajectories for all the parameters. On the other hand, the Gauss–Newton algorithm required less iterations for responses without overshoots. For the responses with overshoots ($\alpha > 1$) both algorithms required almost the same number of steps.

The above results are obtained using noise free data. To demonstrate the performance of the two schemes for noisy data, results are shown in Table 1 for data with NSR 10%. Mean values of 100 MCS are presented along with corresponding standard deviations for model (46) with $\alpha = 1.7$.

These results show that both of the algorithms perform comparably in terms of quality of the estimates. The convergence rates are 100% for both cases with an initial guess of 1 for the delay. The convergence rate is defined as the % of times out of 100 MCS for which the algorithm converged. To compare the performance of the two algorithms in terms of convergence rate, results are presented in Fig. 2 for model (46) with $\alpha = 1.7$. This model represents a step response with overshoot. As seen from the figure, the Newton's algorithm showed high convergence rate of 100% or close to it for a wide range of initial guess of delay from 0.1 to 13. On the other hand, for the Gauss–Newton approach, the convergence rate was close to 100% only for the range of initial delay from 1 to 4. Similar data are generated for a model with $\alpha = 0.7$, which represents a step response without any overshoot; for this case, the convergence rates are at or near 100% for a range of initial delay from 1 to 12 for both algorithms. However, the required number of iterations for the Newton's algorithm is twice as many as that of the Gauss–Newton algorithm.

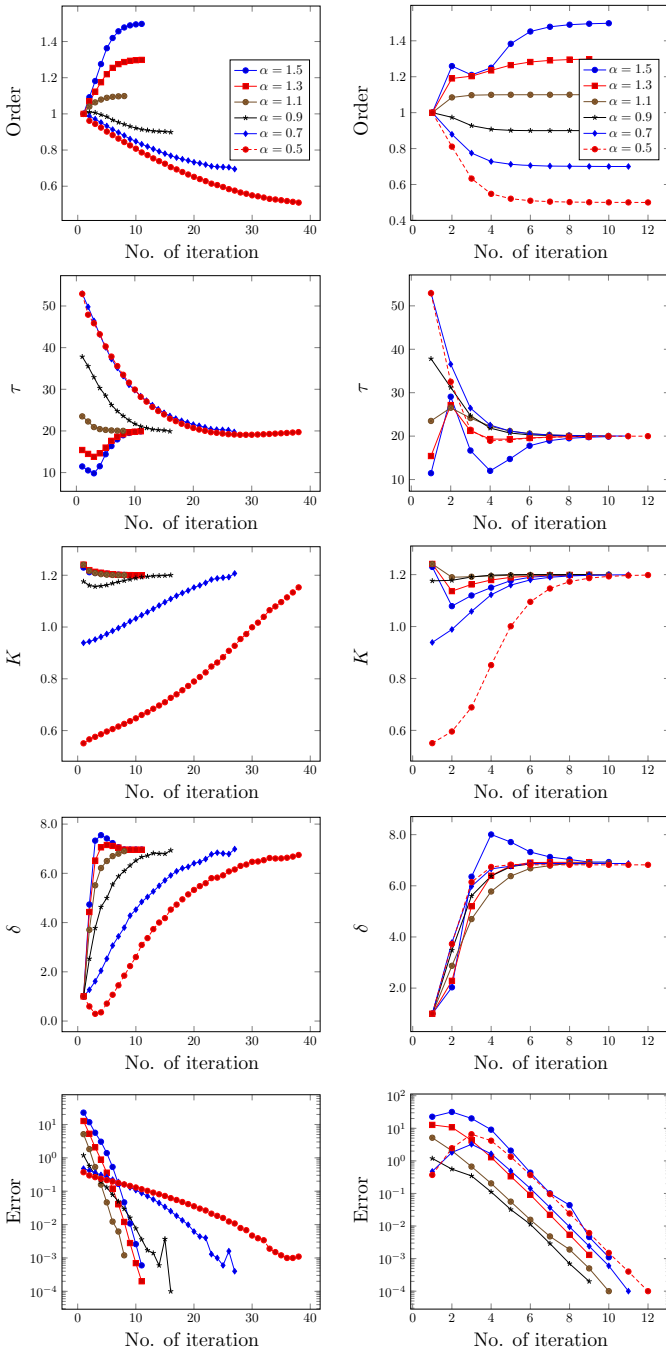
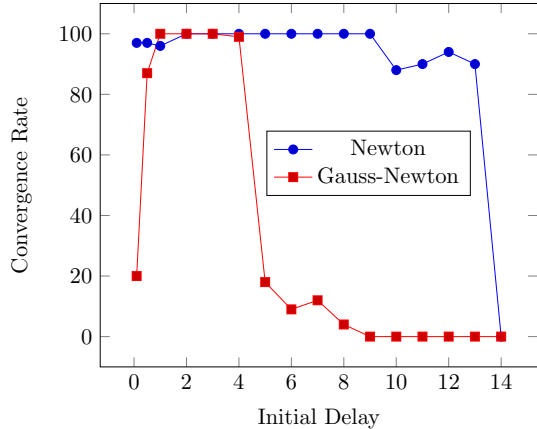


Fig. 1 Convergence of parameters and error for the Newton's algorithm (left) and Gauss–Newton algorithm (right)

Table 1 Performance of Newton's algorithm and Gauss–Newton algorithm

Algorithm	Parameter				Iteration required
	$\alpha \equiv 1.7$	$\tau \equiv 20$	$K \equiv 1.2$	$\delta \equiv 7$	
Newton's	1.699 (0.0055)	19.933 (0.3609)	1.2 (0.0044)	6.95 (0.129)	12.3 (1.1)
Gauss–Newton	1.699 (0.0057)	19.937 (0.3631)	1.2 (0.0045)	6.95 (0.13)	11 (1.3)

Fig. 2 Convergence rate for the Newton's and Gauss–Newton algorithm with different initial delay for model (46) with $\alpha = 1.7$ 

Based on extensive simulation results, the following remarks can be made

- With a suitable initial guess of the delay and the initial assumption of an integer order model, both of the schemes are able to provide satisfactory estimates of the parameters.
- For responses with overshoots, the Newton's algorithm converges for a wider range of initial guess of the delay compared to the Gauss–Newton scheme.
- For responses without overshoots, the Gauss–Newton algorithm requires less number of iterations.

Based on these observations, the Gauss–Newton method is used for the following studies.

3.3 Effect of Data Quality

Two important quality measures for sampled data, namely the length of data and the noise-to-signal ratio are considered to study the effects of data on parameter estimates.

Table 2 shows the identified parameters and their standard deviations for different data lengths. For all these cases, the NSR is 10%. The results show that satisfactory estimates are obtained for data length as low as 100 for the Class I model. As expected, the results show better consistency for larger data sets. Also the required numbers of iteration decreased slightly with the increase in data length.

Table 2 Effect of data length and sampling interval on parameter estimates

Parameter	Estimated values				
	$N = 100$ $\Delta t = 1$	$N = 250$ $\Delta t = 0.4$	$N = 500$ $\Delta t = 0.2$	$N = 1000$ $\Delta t = 0.1$	$N = 2000$ $\Delta t = 0.05$
$\alpha \equiv 1.4$	1.3928 (0.1472)	1.4018 (0.0226)	1.4009 (0.0163)	1.3975 (0.0116)	1.4005 (0.0086)
$K \equiv 1.2$	1.1893 (0.1211)	1.1999 (0.0105)	1.1991 (0.0068)	1.1999 (0.0052)	1.1991 (0.0036)
$\tau \equiv 20$	20.17 (4.185)	20.37 (1.766)	20.10 (1.239)	19.92 (0.8645)	20.08 (0.6577)
$\delta \equiv 7$	6.455 (1.134)	6.862 (0.4756)	6.852 (0.3519)	6.924 (0.2271)	6.958 (0.1561)
Iteration required	15.4 (9.2)	12.8 (5.6)	11.9 (3.9)	10.9 (2.0)	10.4 (1.1)

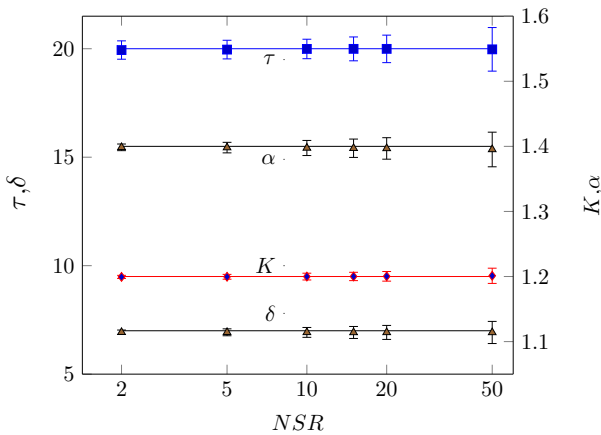


Fig. 3 Effect of noise on parameter estimates for the model (46) with $\alpha = 1.4$

Figure 3 shows the effect of noise on parameter estimates. A total of 1000 data points are used for all cases. The results show satisfactory performance of the algorithm in terms of mean and standard deviation for NSR as high as 50%.

3.4 Identification of Different Class I Models

A list of Class I models with different orders is considered in this section. The parameters presented are the means of 100 MCS with the corresponding standard deviations in the parentheses. NSRs for each cases are 10% and 1000 data points are used. The convergence rates for each of the cases are 100%. The number of iterations varies between 11 and 18 with less iterations required for higher values of α . Initial guess of the time delay is 1 for all cases and an integer order of 1 is used as initial guess. Initial

Table 3 Identification results for different Class I models

True models	Estimated models
$\frac{0.2}{s+0.1} e^{-3s}$	$\frac{0.2(\pm 0.024)}{s^{1(\pm 0.04)}+0.1(\pm 0.01)} e^{-2.94(\pm 0.26)}$
$\frac{0.1}{s^{1.22}+0.05} e^{-6s}$	$\frac{0.1(\pm 0.007)}{s^{1.218(\pm 0.017)}+0.0502(\pm 0.003)} e^{-5.91(\pm 0.286)}$
$\frac{0.5}{s^{0.7}+0.2} e^{-2s}$	$\frac{0.51(\pm 0.055)}{s^{0.71(\pm 0.035)}+0.2(\pm 0.019)} e^{-1.96(\pm 0.25)}$
$\frac{0.06}{s^{1.8}+0.05} e^{-7s}$	$\frac{0.06(\pm 0.001)}{s^{1.8(\pm 0.004)}+0.05(\pm 0.001)} e^{-6.96(\pm 0.12)}$

Table 4 Class II model identification results

True models	Estimated models
$\frac{0.1}{s^2+0.25s+0.1} e^{-2.4s}$	$\frac{0.1(\pm 0.01)}{s^{2.01(\pm 0.1)}+0.25(\pm 0.04)s^{1(\pm 0.06)}+0.1(\pm 0.01)} e^{-2.35(\pm 0.18)}$
$\frac{0.04}{s^{2.39}+0.2s^{1.18}+0.02} e^{-3s}$	$\frac{0.04(\pm 0.008)}{s^{2.39(\pm 0.09)}+0.2(\pm 0.02)s^{1.18(\pm 0.03)}+0.02(\pm 0.004)} e^{-2.94(\pm 0.38)}$
$\frac{0.01}{s^{1.8}+0.5s^{0.8}+0.01} e^{-4s}$	$\frac{0.01(\pm 0.002)}{s^{1.8(\pm 0.1)}+0.049(\pm 0.02)s^{0.78(\pm 0.12)}+0.01(\pm 0.02)} e^{-3.89(\pm 0.64)}$

guess of coefficients are obtained assuming an integer order model. The mean values of the estimated parameters match quite well with the corresponding true values; also the standard deviations are comparable for a NSR of 10%.(Table 3)

3.5 Identification of Class II Models

Table 4 shows the identification results for a set of Class II models. Models with both integer and fractional orders are considered. For each case, 2000 data points are used and data are corrupted with a noise having NSR 10%. An initial guess of 1 is used for the delay and integer orders $[\alpha_2 \ \alpha_1] = [2 \ 1]$ is used for initialization. Convergence rates are 100% for all cases. The obtained results are quite satisfactory in terms of the mean and standard deviations.

4 Concluding Remarks

The step has not been used as input for simultaneous estimation of orders, coefficients and the delay for fractional order models. Also none of the optimization methods for fractional order identification estimates the logarithmic derivatives of input signals which are required to evaluate the Jacobian and the Hessian. This article provides analytical expressions for logarithmic derivatives of the step input and those for the Jacobian and the Hessian; the resulting optimization scheme does not need to resort to numerical approximations. A simplified initialization procedure is also presented which, in fact, requires a choice of the delay only; however, convergence for a wide range of initial guess of the delay is observed. The orders are initially assumed to be integers. Initial value of the coefficients is obtained assuming an integral order model. Simulation results show the robustness of the optimization scheme in the presence of

noise. Satisfactory results are obtained for data length as low as 100 and for NSR as high as 50% for different models. The required number of iteration steps is in the range 10–20. Quality of parameter estimates and low number of iteration required show the performance and efficiency of the algorithm.

Acknowledgements The author acknowledges the financial supports from Research and Development Corporation (RDC) of Newfoundland and Labrador and Natural Sciences and Engineering Research Council (NSERC) of Canada.

Compliance with Ethical Standards

Conflicts of interest The author declares that there is no conflict of interest.

Human Participants The study does not involve any human participants and/or animal. The manuscript has not been published previously (partly or in full).

References

1. S. Ahmed, Parameter and delay estimation of fractional order models from step response, in *9th IFAC Symposium on Advanced Control of Chemical Processes*, Whistler, BC, Canada (2015), pp. 942–947
2. S. Ahmed, B. Huang, S.L. Shah, Parameter and delay estimation of continuous-time models using a linear filter. *J. Process Control* **16**(4), 323–331 (2006)
3. M. Aoun, R. Malti, F. Levron, A. Oustaloup, Synthesis of fractional Laguerre basis for system approximation. *Automatica* **43**, 1640–1648 (2007)
4. D. Babusci, G. Dattoli, *On the logarithm of the derivative operator* arXiv e-prints (2011)
5. A. Benchellal, T. Poinot, C. Trigeassou, Approximation and identification of diffusive interfaces by fractional systems. *Signal Process.* **86**(10), 2712–2727 (2006)
6. Y. Chen, I. Petras, D. Xue, Fractional order control—A tutorial, in *2009 American Control Conference*, St. Louis, USA (2009), pp. 1397–1411
7. E.K. Chong, S.H. Zak, *An Introduction to Optimization* (Wiley, New York, 1996)
8. O. Cois, A. Oustaloup, T. Poinot, J. Battaglia, Fractional state variable filter for system identification by fractional model, in *European Control Conference*, Porto, Portugal (2001)
9. J.E. Diamesis, A new method for determining the parameters of physical systems, in *Proceedings of the IEEE* (1965), pp. 205–206
10. S.M. Fahim, S. Ahmed, S.A. Imtiaz, Fractional order model identification using the sinusoidal input. *ISA Trans.* **83**, 35–41 (2018). <https://doi.org/10.1016/j.isatra.2018.09.009>
11. R. Fletcher, *Practical Methods of Optimization. Vol. 1: Unconstrained Optimization* (Wiley, New York, 1980)
12. J.D. Gabano, T. Poinot, H. Kanoun, Identification of a thermal system using continuous linear parameter-varying fractional modelling. *IET Control Theory Appl.* **5**(7), 889–899 (2011)
13. E.V. Hayngworth, K. Goldbe, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, chap. Bernoulli and Euler Polynomials- Riemann Zeta Function, U.S. Department of Commerce, NIST, Washington DC (1972) pp. 803–820
14. J. Hines, Operator mathematics II. *Math. Mag.* **28**(4), 199–207 (1955)
15. K. Leyden, B. Goodwine, Fractional-order system identification for health monitoring. *Nonlinear Dyn.* **92**(3), 1317–1334 (2018)
16. B. Lurie, Three-parameter tunable tilt-integral-derivative (TID) controller. US Patent 5371670 (1994)
17. R. Malti, S. Victor, A. Oustaloup, Advances in system identification using fractional models. *J. Comput. Nonlinear Dyn.* **3**(2), 0214011–7 (2008)
18. R. Malti, S. Victor, A. Oustaloup, H. Garnier, An optimal instrumental variable method for continuous-time fractional order model identification, in *Proceedings 17th IFAC World Congress*, Seoul, Korea (2008), pp. 14379–14384
19. A.K. Mani, M.D. Narayanan, M. Sen, Parametric identification of fractional-order nonlinear systems. *Nonlinear Dyn.* **93**(2), 945–960 (2018)

20. R. Mansouri, M. Bettayeb, S. Djenjone, Approximation of high order integer systems by fractional order reduced parameter models. *Math. Comput. Model.* **51**, 53–62 (2010)
21. W.F. Mascarenhas, Newton iterates can converge to non-stationary points. *Math. Program. Ser. A* **112**, 327–334 (2008)
22. C. Monje, Y. Chen, B. Vinagre, D. Xue, V. Feliu-Battle, *Fractional-order Systems and Controls* (Springer, London, 2010)
23. M. Muddu, A. Narang, S. Patwardhan, Development of ARX models for predictive control using fractional order and orthonormal basis filter parameterization. *Ind. Eng. Chem. Res.* **48**(19), 8966–8979 (2009)
24. C.I. Muresan, S. Folea, I.R. Birs, C. Ionescu, A novel fractional-order model and controller for vibration suppression in flexible smart beam. *Nonlinear Dyn.* **93**(2), 525–541 (2018). <https://doi.org/10.1007/s11071-018-4207-0>
25. A. Narang, S.L. Shah, T. Chen, Continuous-time model identification of fractional-order models with time delays. *IET Control Theory Appl.* **15**(5), 900–912 (2010)
26. P. Nazarian, M. Haeri, M.S. Tavazoei, Identifiability of fractional-order systems using input output frequency contents. *ISA Trans.* **49**, 207–214 (2010)
27. K. Oldham, J. Myland, J. Spanier, *An Atlas of Functions*, 2nd edn. (Springer, New York, 2000)
28. K. Oldham, J. Spanier, *The Fractional Calculus* (Academic Press, New York, 1974)
29. A. Oustaloup, X. Moreau, M. Nouillant, The CRONE suspension. *Control Eng. Pract.* **4**(8), 1101–1108 (1996)
30. I. Podlubny, Fractional order systems and $PI^\lambda D^\mu$ controllers. *IEEE Trans. Autom. Control* **44**(1), 208–214 (1999)
31. T. Poinot, J.C. Trigeassou, Identification of fractional systems using an output-error technique. *Nonlinear Dyn.* **38**(1), 133–154 (2004)
32. H. Raynaud, A. ZergaInoh, State-space representation for fractional order controllers. *Automatica* **36**, 1017–1021 (2000)
33. L. Sersour, T. Djamah, M. Bettayeb, Nonlinear system identification of fractional wiener models. *Nonlinear Dyn.* **92**(4), 1493–1505 (2018). <https://doi.org/10.1007/s11071-018-4142-0>
34. M. Tavakoli-Kakhki, M. Haeri, M. Tavazoei, Simple fractional order model structures and their applications in control system design. *Eur. J. Control* **16**(6), 680–694 (2010)
35. M. Tavakoli-Kakhki, M. Haeri, M.S. Tavazoei, Over- and under-convergent step responses in fractional-order transfer functions. *Trans. Inst. Meas. Control* **32**(4), 376–394 (2010)
36. M. Tavakoli-Kakhki, M. Tavazoei, Estimation of the order and parameters of fractional order models from a noisy step response data. *J. Dyn. Syst. Meas. Control* **136**(3), 0310201–6 (2014)
37. M.S. Tavazoei, Overshoot in the step response of fractional-order control systems. *J. Process Control* **22**, 90–94 (2012)
38. D. Valerio, M.D. Ortigueira, J.S. da Costa, Identifying a transfer function from a frequency response. *J. Comput. Nonlinear Dyn.* **3**(2), 0212071–7 (2008)
39. S. Victor, R. Malti, Model order identification for fractional-order models, in *European Control Conference*, Zurich, Switzerland (2013), pp. 3470–3475
40. S. Victor, R. Malti, H. Garnier, A. Oustaloup, Parameter and differential order estimation of fractional-order models. *Automatica* **49**, 926–935 (2013)
41. P.C. Young, Optimal IV identification and estimation of continuous-time TF models, in *IFAC World Congress*, Barcelona, Spain (2002), pp. 337–358
42. L. Yuan, Q. Yang, C. Zeng, Chaos detection and parameter identification in fractional-order chaotic systems with delay. *Nonlinear Dyn.* **73**(1), 439–448 (2013)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.