

Trabajo Final de Master

Máster en Investigación Matemática

Universidad Politécnica de Valencia
Departamento de Matemática Aplicada

A Study of Fractional Calculus Applicability to System Identification and Modeling for Control Design Purposes

First Adviser
Prof. J. Alberto Conejero Casares

Second Adviser
Prof. Enric Picó i Marco

Student
Jonathan Franceschi

Academic Year 2018/2019

Abstract

We analyze some of the existent software tools to identify fractional-order transfer functions from experimental data, and, more broadly, their use in the context of Systems and Control Engineering.

The main goal of this Master Thesis is to compare fractional-order transfer functions to ordinary ones, the latter being supported by many years of experience in their use and development, in order to determine whether the former could provide practical advantages, at least for certain kinds of systems, like those which present diffusion phenomena, delays, and other characteristics hard to capture.

Resumen

Se propone analizar las herramientas sofware existentes para identificar funciones de transferencia fraccionales a partir de datos experimentales y su uso en el contexto de la ingenieria de sistemas y control.

El objetivo principal es compararlas con funciones de transferencia ordinarias, para las que hay herramientas con muchos años de experiencia en su uso y desarrollo, para determinar si las nuevas herramientas ofrecen alguna ventaja práctica al menos para determinado tipo de sistemas como puedan ser aquellos que presentan fenómenos de difusión, retardos y otras características dificiles de capturar.

Sommario

Vengono analizzati alcuni strumenti software per identificare funzioni di trasferimento frazionarie a partire da dati sperimentali e, più in generale, per il loro uso nel campo dell'Ingegneria di Sistemi e di Controllo.

L'obiettivo principale è comparare le funzioni di trasferimento di ordine non intero a quelle ordinarie, che sono sostenute da decenni di esperienza nel loro uso e sviluppo, per verificare se questi nuovi strumenti possano fornire un vantaggio pratico, almeno per determinati tipi di sistemi, come quelli che presentano fenomeni di diffusione, ritardi e altre caratteristiche difficili da catturare.

Acknowledgments

This Master Thesis is the first part of a joint work by the universities of Valencia, Spain (Universitat Politècnica de València, Universitat de València) and Ferrara, Italy (Università degli Studi di Ferrara), in the scope of a double master degree which will conclude in 2020 in Ferrara.

First of all, I would like to thank my two advisers, Professors José Alberto Conejero Casares and Enric Picó i Marco, for their valuable help and guidance through the development of this project. I would also like to thank Professor Jesús Andrés Picó Marco for his participation in the early stage of the work and to have provided really useful experimental data, as did Professor Juan Manuel Herrero Dura, who I wish to thank, either.

I wish to express my gratitude to Professor Chiara Boiti, for her great support from the very beginning of this double degree journey, a journey which I would have never undertaken without her help. Her counterparts in Valencia have been Professors Casares (UPV) and Pep Mulet (UV), I also thank them for their administrative advice.

I will never thank my father enough. Thank you for resisting and for being there whenever I have needed to. To paraphrase the Herald, you are my Enforcer. You have always been ours.

Finally, thank you Anna Chiara. Without you, I would have flown back in a second, no doubt. If a time machine would made me come back, you are reasons number 1 to 10 for redoing everything again.

When I cried, you were with me, when I laughed, it was because of you.

Contents

Introduction

Fractional calculus is a branch of mathematical Analysis with three centuries of history that recently has seen strong reception in a variety of fields, from engineering [31, 20, 2] to biology [25, 17]. The flexibility given by a noninteger order of derivation or integration seem to help understanding underlying dynamics of many systems in an unprecedented way. In particular, two of the main sectors of employment of fractional modeling are control theory and system identification theory, both in real-life industrial scenarios and in more theoretical works. In control theory, the impression is that the non-integer character of the operators allows to capture nonlinearities, at least partially, in order to design a control system.

However, such flexibility comes at a cost, in terms of computational resources and conceptual effort in managing a much more sophisticated operator than the traditional one. Therefore, there is room for wondering whether such cost is justified by the enhanced capability of noninteger instruments or whether in the end the investment outweigh the earnings, so that traditional integer-order operator could be preferred without significant performance losses. After all, ordinary techniques have been playing the main role in the field for almost 80 years [3], so it does not seem unfair asking for tangibles improvements.

In order to find insights which prove useful in answering the question, in this Master Thesis we present three numerical results developed in the context of system identification, more specifically time-domain transfer function identification. The first two come from data collected in an experimental setting, while the last one deals with computer-simulated data.

The first example is a furnace, a typical thermal system currently easily approximated by ordinary differential equations for control purposes. Its most significant characteristic is that its gain depends on the input size (the voltage applied to the heating resistance) and also on the sign of the input step (heating is not the same as cooling). For relatively small variations of the input in a neighborhood of a suitable working point, i.e., one of the possible equilibrium points of the system, this latter effect has greater impact.

The second example is a Peltier cell, chosen precisely because it presents significant diffusion phenomena, often (see e.g., [25, 26]) believed to be one of the cases where fractional operators can actually provide advantages.

The main goal of these works is testing whether fractional-order transfer functions outperform their integer-order counterparts when the former are approximated with high-order rational transfer functions where every operator of the form s^α, $\alpha \in \mathbb{R}^+$ is in the end replaced by a suitable s^n, with $n \in \mathbb{N}$.

The tests were conducted using the MATLAB toolbox FOMCON, by Aleksei Tepljakov [32], a popular toolbox for fractional-order system identification and control.

The thesis outline is the following: in the first Chapter (1) we propose a brief and mostly nontechnical overview to feedback and control theory, while in the subsequent Chapter (2) we introduce in detail the mathematical basis of fractional calculus and fractional differential equations. In Chapter 3 we synthesize the topic of system identification and in particular the topic of transfer functions identification with respect of both integer and noninteger orders, and we also discuss the two most popular approaches to approximate fractional operators, i.e., Oustaloup's filter [27, 21, 28] and Grünwald-Letnikov approximation [13, 32, 21]. Then (4) we discuss the principal toolboxes employed in the field for fractional system identification and control, and provide a brief outline of the one we effectively used for our numerical computations, i.e. the FOMCON [33]. In the fifth Chapter (5) we present and discuss the numerical results we found; finally in Chapter 6 conclusions are drawn.

Chapter 1

Feedback and Control

The purpose of this Chapter is to provide a tutorial overview of the engineering topics that lie beneath this thesis, i.e., above all feedback and control.

1.1 Feedback

We will start with a well-known basic example. Let us consider the heating system of a house: we would like to set the room temperature at a desired value and keep it constant in spite of external undesired influences. There are some factors which we need to take into account, above all the current room temperature and outside temperature; of course one has to consider also the presence of individuals in the room, the walls material composition, the actual kind of construction of the heater etc, but for our purposes (i.e., control) we will see it is enough focusing only on the first two factors, i.e., the two temperatures.

We would like the room temperature to be, for instance, 20°C: we will call it the *reference* value, and we will denote it by r. To achieve this goal, we would use a suitably implemented device: typically it would obtain information about the variables we want to control by measuring some related physical magnitude[1]; we will refer to it as the *output* of the system and we will denote it by y.

Now let us focus our attention on the heater: for the sake of simplicity, let assume it is an electric resistance. The higher the voltage, the more intense the heat transmitted. We will refer to the voltage, being the physical magnitude we are actually going to manipulate, as an *input*, and we will denote it by u.

Unfortunately, the room is not isolated. Ambient temperature also can be (greatly) influential, but we in principle cannot manipulate it. Besides, in most application other than for example this one, there is no way to measure the external

[1] The measured variables could coincide with the ones we would like to control, like in the example of the heater, but there is no compelling reason for this to happen in every case.

variable affecting the system behavior. Hence, we will call such an uncontrollable, and often unmeasurable, entity *disturbance*, and we will denote it by d.

We now have all the elements to set up the process: a sensor measures the room temperature y, we compare it to the reference value r; their difference $y - r$ will be the *error*, denoted by e. Fed with the error e and, implicitly, with the information about the underlying dynamics of the system (which was used to design the controller itself), the *controller* actuates over the heater suitably, with the goal of reducing the error, in spite of effects of external disturbances. A new y is then obtained, which leads us to the starting point.

What we have described is a basic example of (closed) *feedback loop*: which can be represented as in Figure 1.1, top half. In the bottom half we have zoomed on one particular of the loop: the controller, described mathematically as a function $K(s)$, maps the input u to the function $G(s)$, which represents the dynamics of the system. However, we shall stress that the controller is actually a microprocessor that sends signals to the *house*, including also all those variables we ignored in our introduction above, so there is a fairly significant difference between the two.

Figure 1.2 shows another possibility, named *open-loop feedback*: What if we consider the bottom half of Figure 1.1 with the reference value r instead of the error e as the variable entering in the controller? (That is, there is no feedback). Then to achieve the desired relation $y = r$ it would be enough to set $K(s) = G^{-}-1(s)$. But, as we said before, $G(s)$ only models those aspects of the system dynamics that are relevant for our control problems, otherwise it would be too complex for control design, and anyway perfect models do not exist[2]. It will never be the real thing, our house. Therefore, this approach will generally fail. Closed-loop feedback, instead, can continually adjust for errors made at previous iterations, compensating them through time. For this reason closed-loop has become a key idea of control engineering.

Now, let us deepen the mathematical interpretation of a system like the house heating system. We assume that it can be described by an ordinary differential equation with state variable $x = x(t)$ and input variable $u = u(t)$, so that the state derivative is equal to a function f of both x and u:

$$\dot{x} = f(x, u), \tag{1.1}$$

We consider the set of possible equilibrium points (x_0, u_0) of the system, i.e., those such that $\dot{x} = 0$. In a neighborhood of such points the system can be well

[2]"Essentially, all models are wrong, but some are useful."—Box, George E. P.; Norman R. Draper (1987). Empirical Model-Building and Response Surfaces, p. 424, Wiley. ISBN 0471810339.

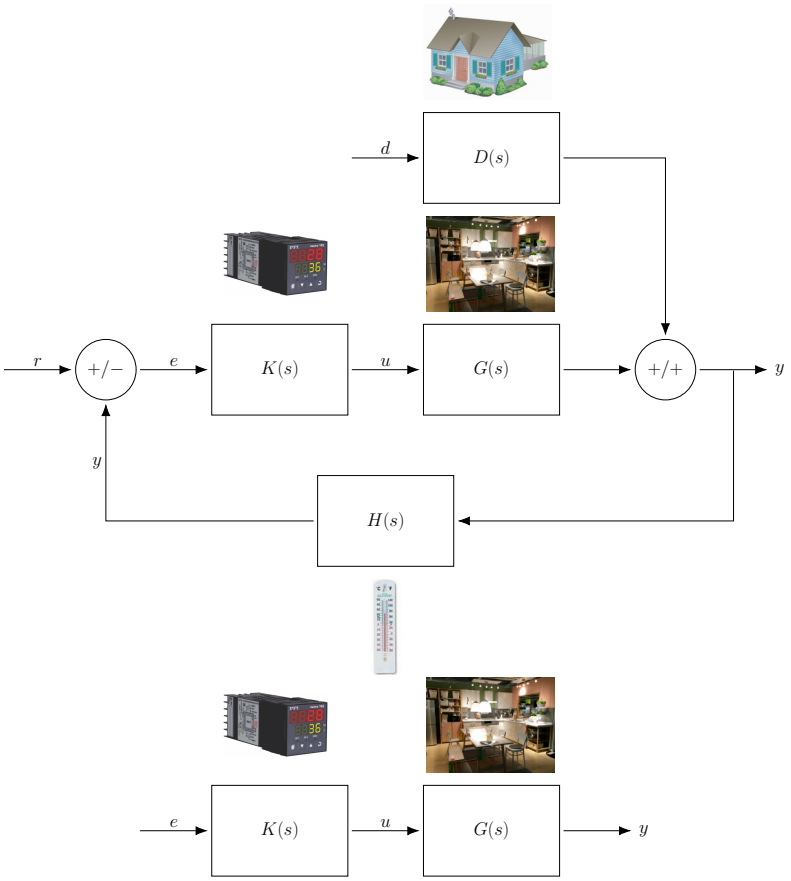

Figure 1.1: Top half: Closed-loop feedback system. $K(s)$ represents the controller and $G(s)$ encapsulate system's dynamics. $D(s)$ models the way disturbances affect the output; while the way the measured output y is compared with the error e at the beginning of a loop iteration is given by another function, $H(s)$ which represents the behavior as physical objects of the sensors used to measure the system output.
Bottom half: Zoom on the phase from the controller to the objective system (the actual room).

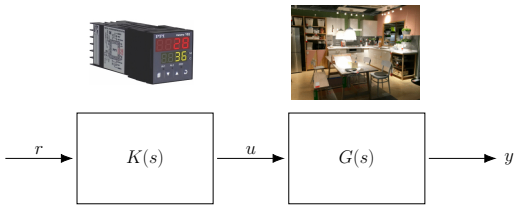

Figure 1.2: Open-loop feedback system. The dining room over the second block represents all the details and variables inherent to the system which must be predicted and modeled, a practically insurmountable challenge.

approximated by its first-order Taylor expansion:

$$\dot{x} \approx f(x_0, u_0) + \left.\frac{\partial f}{\partial x}\right|_{(x_0,u_0)} (x - x_0) + \left.\frac{\partial f}{\partial u}\right|_{(x_0,u_0)} (u - u_0) \tag{1.2}$$

Although it is possible to have a large number of equilibrium points, in studying systems to design real-life controllers it is common practice to consider only the ones that are of interest for a certain application. Consider for instance that we need to store some perishable foods in a room for industrial purposes: we would need to keep the room at a certain temperature (for example 4°C), so we consider only those points that are of equilibrium themselves or are close enough to equilibrium points in a neighborhood of 4°C. A point of such subset is called *working point* to underline that we want our system to operate around it.

Suppose now we want a linear approximation of the system in a neighborhood of a suitable working point $(\bar{x}_0, \bar{u}_0)$. We would have:

$$\begin{cases} \dfrac{\partial f}{\partial x}(\bar{x}_0, \bar{u}_0) & = a \in \mathbb{R}, \\ \dfrac{\partial f}{\partial u}(\bar{x}_0, \bar{u}_0) & = b \in \mathbb{R}, \end{cases} \tag{1.3}$$

so that we can write

$$\dot{x} \approx a(x - x_0) + b(u - u_0). \tag{1.4}$$

We are one step away from the linear approximation we wanted: a simple change of coordinates $(\bar{x}, \bar{u}) = (x - x_0, u - u_0)$, finally, leads to:

$$\dot{\bar{x}} \approx a\bar{x} + b\bar{u}. \tag{1.5}$$

At this time, it is common practice to consider the system as actually described by $\dot{x} = ax + bu$, where, in an abuse of notation, bars have dropped.

Then a standard tool in mathematical Analysis, the *Laplace transform*, defined as

$$\mathscr{L}\{g\}(s) \coloneqq \int_0^\infty g(t)\mathrm{e}^{st}\,\mathrm{d}t,$$

is applied to obtain:

$$x(s) = \underbrace{\frac{b}{s+a}}_{\substack{\text{Transfer}\\ \text{Function}}} \cdot u(s) \tag{1.6}$$

In this case, thus, the *transfer function* $G(s)$ of the system would be $b/(s+a)$.

Applying the *inverse Laplace transform* $\mathscr{L}^{-1}$, gives us the time-domain equation:

$$x(t) = u(t) \cdot \frac{b}{a}(1 - \mathrm{e}^{-at}). \tag{1.7}$$

More generally, a system like the house's heating one can be described by a transfer function of the form:

$$G(s) = \frac{K}{1+\tau s}\mathrm{e}^{\theta s}, \tag{1.8}$$

where k is the *gain*, i.e., the relation between input and output when the input is a step and the system is in steady state; τ is the time constant and θ is the delay between the instant in which the input changes and the instant in which a steep response in the output starts. In other words, the gain tell us how much we must change the input in order to change the output of a fixed value, the time constant how much time we need to wait to have the system complete its response and the delay how much time we need to wait to have the system start to respond the controller commands.

What happens in practice in the field of Control Engineering is that only the transfer function is analyzed because, since through decades it has been studied the dependence between the coefficients in (1.8) and its associated ODE's solution, it provides all the information needed to understand the system dynamics.

Example 1.1. As an example, consider Figure 1.3: after 5 seconds the input u is raised and after a delay of 2 s the output starts to increase steeply, until it tends asymptotically to a stable value, called *steady state value*. The time needed to pass from the initial exponential increase to the final plateau is called *settling time* and it is usually equal to 4 or 5 times the amount of time needed to get to the 63.2% of the steady state value, which is called *time constant* and it is usually denoted by τ.

In the example, $u(t)$ is a step of width 5 s. Its associated $u(s)$ is $u(s) = 5/s$ and the whole system is described by:

$$x(s) = \frac{b}{s+a} \cdot \frac{5}{s}. \tag{1.9}$$

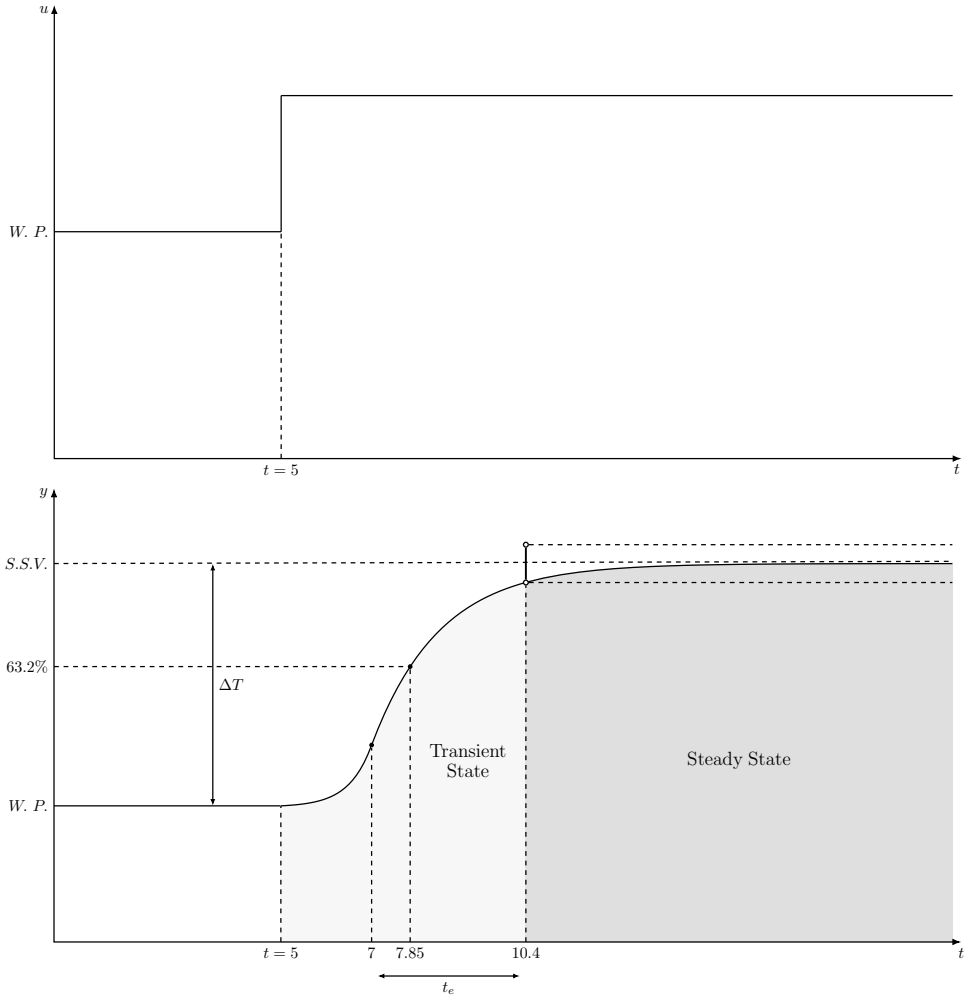

Figure 1.3: Example of system response under a step input.

1.2 Control

The kind of control used in real life for the previous example is the following: if the room temperature is lower than the reference value, it turns on the heater, otherwise the controller turns it off. Unsurprisingly, this kind of control action is called *on-off control* and is mathematically described by the following law:

$$u = \begin{cases} u_{\max} & \text{if } e > 0, \quad \text{i.e., the measured temperature is lower than } r, \\ u_{\min} & \text{if } e < 0, \quad \text{i.e., the measured temperature is higher than } r, \end{cases}$$

where, as said above, the control error $e = r - y$ is the difference between the reference value and the output y of the system, while u is the actuation command (the input).

Actually, the most used type of controller in real-world scenarios [31] is the so called *PID* controller, that stands for Proportional Integral Derivative controller. It has the proportional term that depends on the current (instantaneous) value of the error, $k_{\mathrm{p}}\, e(t)$; the integral term that considers all the past history of the error

$$k_{\mathrm{i}} \int_0^t e(\tau)\, \mathrm{d}\tau;$$

and finally it has the derivative term, which try to predict the future behavior following a simple linear extrapolation:

$$e(t + T_d) \approx e(t) + T_d \frac{\mathrm{d}e(t)}{\mathrm{d}t}.$$

We can, hence, express mathematically the PID controller by means of the equation

$$u(t) = k_{\mathrm{p}}\, e(t) + k_{\mathrm{i}} \int_0^t e(\tau)\, \mathrm{d}\tau + k_{\mathrm{d}} \frac{\mathrm{d}e(t)}{\mathrm{d}t}.$$

Of course, in light of what we said in the introduction, it is quite natural wondering whether the PID could be generalized via the use of fractional-order operators. The answer is affirmative, and such controllers are called *FOPID* (Fractional Order PID).

This means that if ordinary differential equations (ODE) are the mathematical objects used to model and design PID controller, the equations involved in the study of FOPID controller are *fractional differential equations* (FDE).

1.3 Motivation

In Mathematical Physics, the subject of study are *closed* systems, i.e., such that it is not subjected to any interaction whose source is external to the system.

However, sometimes there is some sort of exchange (either in form of mass, energy, information,...) with entities that we are not capable of isolate totally, and it has a significant impact on the behavior of the system. Indeed, let us consider we are studying the motion of a set of celestial bodies: we know from empirical observations that there are some other objects, planets[3] for instance, which are far away from our system but not enough to not influence its dynamics. In celestial mechanics, and successively in the whole Mathematical Physics, the solution to this problem has been modeling external influences by introducing parameters dependent explicitly on time. This *implicit* way of dealing with uncontrollable effects from the outside of the system is what distinguish between autonomous and non-autonomous systems.

In Control Engineering, systems that are autonomous from the point of view of Mathematical Physics are still considered non-autonomous, since external influences (inputs) are not taken into account explicitly. Those inputs appear explicitly in the equations as variables, in addition to the state variables which are presents in the derivatives of the system of equations which define the behavior of the system.

Since there are more variables than equations, we are dealing with a underdetermined system, mathematically speaking, which is said to be an *open* system, in engineering jargon. Like before there will be open systems which are also time-invariant (i.e., with constant coefficients) or they will be time-variant (i.e., with varying coefficients).

This premise should provide a reference for what mathematicians and engineers mean when they refer to a system.

We now remind the reader that the transfer function $G(s)$ of a system is a relation between its inputs and outputs that is normally obtained via the Laplace transform (for continuous systems). As said above, engineers use it habitually to predict the system dynamics through time.

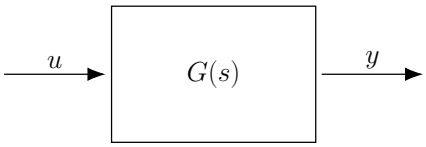

Figure 1.4: Transfer function.

Another possibility is to study system's behavior with respect to the *frequency* of the input. In practice it is very important to know how a system amplifies inputs at various frequencies. In performance analysis of control systems this information allows one to quantify the influence of high frequency measurement noise on the steady-state response of the system, or how close a closed-loop system will track

[3] A planet situated outside of the system being considered is called *exoplanet*.

low-frequency reference signals. Usually, this information is provided by *Bode diagrams*, graphs of gain (or magnitude) and output's amplitude (or phase) when the system is subjected to periodic signals; see for example Figure 1.5

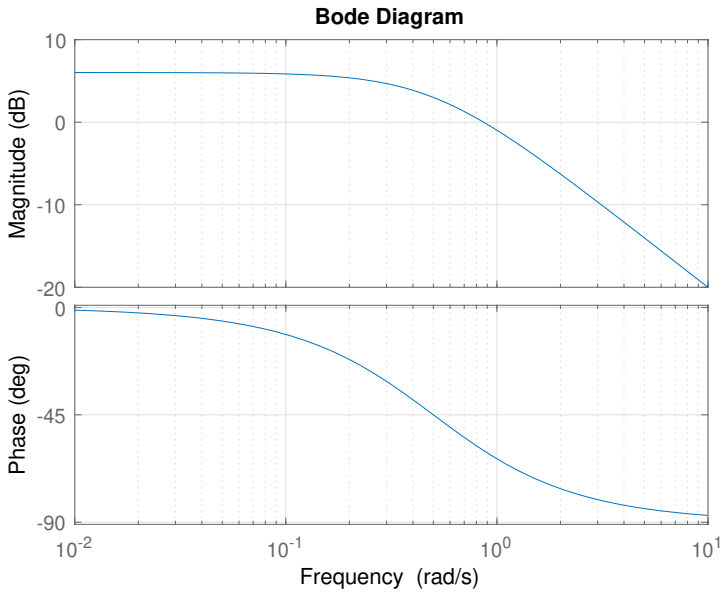

Figure 1.5: Example of Bode diagram.

In particular, the Bode magnitude plot is a graphical representation of the gain with which the system amplifies harmonic signals at various frequencies.

Bode diagrams and more generally the whole branch of frequency analysis, are not an exclusive feature of ordinary systems. Indeed, in Control Engineering there is a fractional analog of frequency analysis, which in principle could lead to significant scientific advances.

Fractional-order operators have seen in recent years great reception in many applied fields for the flexibility they bring about in modeling systems dynamics, especially when it comes to capture system's nonlinearities.

Moreover, for fractional-order open linear systems there exists an analogue of the Engineering version of the *superposition principle* which asserts that the output of a sum of two inputs equals the sum of the outputs of the single inputs.

Actually, when it comes to the specific task of solving ordinary differential equations (ODEs) there are two ways to employ the superposition principle, according to the perspective of each field:

- In Mathematics, the superposition principle is applied to solve an inhomogeneous ODE as the sum of the family of its homogeneous counterpart plus a particular solution of the original one;

- In Engineering, the superposition principle, referred to open systems where external influences are explicitly considered, is represented by

$$\begin{array}{rcl} u_1 & \to & y_1 \\ u_2 & \to & y_2 \end{array} \quad \rightsquigarrow \quad u_1 + u_2 \to y_1 + y_2,$$

  that is, if the input u_1 produces the output y_1 and the input u_2 produces the output y_2, then the sum of the two outputs $y_1 + y_2$ is obtained as sum of the two inputs $u_1 + u_2$.

For linear systems, the two coincide. For nonlinear ones there is an analogue [5, 12], although for *closed*, isolated systems, which are the ones Mathematical Physics has been always, from an historical point of view, focused on. Indeed, it is still an open problem whether there exist a superposition principle in the fashion of the Engineering field for *open* systems.

Coming back to frequency analysis, one of the peculiarities of linear systems is that their Bode magnitude plot depends only on the frequency. On the other hand, the nonlinear ones are characterized by the dependence on input's magnitude: in every nonlinear system, the gain depends on input's magnitude. Besides, there is a certain class of nonlinear systems which allow a nonlinear frequency response function which characterizes all steady-state solutions corresponding to harmonic excitations at various amplitudes and frequencies [29], thus extending the conventional frequency response function defined for linear systems. The elements of such a class are called *convergent systems* and they are also characterized by the so-called *input entrainment* i.e., a periodic input generates a periodic output and also with the same frequency. These nonlinear systems are thus perfect candidates to be analyzed through a suitable extension of Bode diagrams, with enormous benefits (also from an industrial point of view, where there is little room for controllers whose response to various frequencies is hard to predict). However, as stressed by Pavlov *et al.* in [29], the gain in steady state will depend not only on the frequency, as in the linear case, but also on the amplitude of the excitation.

Yet, after careful examination, the second adviser of this Master Thesis found that fractional Bode diagrams do not present any evidence of dependence on the amplitude of the excitation. They look no different than ordinary, standard Bode diagrams of linear systems, so it could not be possible for them to capture any significant nonlinearity, as if it that was the case there would be a related evidence in the corresponding magnitude plot.

This could be due to the fact that they are obtained via an approximation with high-order linear systems, defeating the whole analysis purpose.

As a consequence, in this work we verify whether fractional operators (needing to be approximated for practical reasons) are more capable of capturing nonlinearities of a system than ordinary ones. We do so by identifying the same systems

(in the time domain) with both ordinary transfer functions and fractional transfer functions and then comparing the results. We will discuss them in Chapter 5.

Chapter 2

Fundamentals of Fractional Calculus

In this chapter we develop the theory in detail to provide a reference for the interested reader, the calculations refer to the most interesting properties of fractional operators and they do not claim to be a complete exposition. This chapter has been greatly inspired by mainly [10] and partly [21].

We will start by giving a general historical context of the development of fractional calculus, then we will introduce the three main approaches to noninteger fractional operators, providing for each of them their most important properties. Finally we conclude with an example of application where the use of derivatives of order $k \in \mathbb{R}_+$ emerges naturally.

2.1 Brief Historical Overview

From the very beginning of the development of calculus there was some concern about the nature of the order of the differential operator $\mathrm{d}^n/\mathrm{d}x^n$, where, classically, $n \in \mathbb{N}$. Indeed, was Leibniz himself that in a letter to L'Hôpital in 1695 wondered "Can the meaning of derivatives with integer order be generalized to derivatives with non-integer orders?" In his reply, L'Hôpital proposed "What if the order will be 1/2?" Leibniz on September 30, 1695 famously declared "It will lead to a paradox, from which one day useful consequences will be drawn."

The first systematic studies seem to have been made by Liouville (first half of XIX century), who expanded functions in series of exponentials and defined the nth-order derivative of such a series by operating term-by-term, although still with $n \in \mathbb{N}$. Riemann, in the same years, instead proposed a definition based on definite integrals that was applicable also to non-integer exponents, and finally Holmgren. Later, Grünwald (1867) and Krug unified the results of Liouville and Riemann. Grünwald, arrived at definite-integral formulas for the nth-order derivative using as starting point the definition of a derivative as the limit of a (backward) difference quotient. Krug, working through Cauchy's integral formula for ordinary

derivatives, showed that Riemann's definite integral had to be interpreted as having a finite lower limit while Liouville's definition corresponded to a lower limit $-\infty$.

The first application of the fractional calculus was made by Abel in 1823. He discovered that the solution of the integral equation for the tautochrone[1] problem could be obtained via an integral in the form of a derivative of order one half. Later in the nineteenth century, important stimuli to the use of fractional calculus were provided by the development by Boole of symbolic methods for solving linear differential equations of constant coefficients, or the operational calculus of Heaveside developed to solve certain problems in electromagnetic theory such as transmission lines. In the twentieth century contributions have been made to both the theory and applications of fractional calculus by very well known scientists such as Erdélyi [11] (integral equations, 1854), Riesz [30] (functions of more than one variable, 1973), Caputo and Mainardi [6, 7] (dissipation in geophysics, 1971), or Oldham and Spanier [25] (electrochemistry and general transport problems, 1974).

In the last decades of the last century there was continuing growth of the applications of fractional calculus mainly promoted by the engineering applications in the fields of feedback control, systems theory, and signals processing [31, 20, 2]. In particular, in control theory efforts have been devoted to improve the performances of traditional controllers using fractional operators in place of integer ones, in order to achieve more flexible and smooth effects. However, the applicability of such methods is still under development and it is an open research field.

2.2 The Foundations

There are very many possible generalizations of $\mathrm{d}^n/\mathrm{d}x^n f(x)$ to the case $n \notin \mathbb{N}$. We shall only discuss three of them, the Riemann–Liouville derivative, the Grünwald-Letnikov derivative and the Caputo derivative. The former concept is historically the first (developed in works of Abel, Riemann and Liouville in the first half of the nineteenth century) and the one for which the mathematical theory has been established the most. However, it has certain features that lead to difficulties when applying it to "real world" problems. As a consequence, the latter concepts was developed. They are closely related to the Riemann–Liouville idea, but certain modifications were introduced in order to avoid the above-mentioned difficulties. Yet, the mathematical implications of these modifications have not been investi-

[1] A *tautochrone* (from Greek prefix tauto- meaning *same* and chrono *time*) is the curve for which the time taken by an object sliding without friction to its lowest point is independent of its starting point. The curve is a cycloid, as it was proved geometrically by Huygens in 1659 and later confirmed with a variational approach by Euler and Lagrange in the subsequent century.

gated fully so far.

The Grünwald-Letnikov derivative is especially used when precise numerical approximations of the operator are needed, since its allows an immediate discretization of the derivative from its very definition. The Caputo derivative is compatible for initial values problems [13], so it is mostly used in that context.

Before delving into details, we shall recall some classical function spaces where we will set our discussion (Lebesgue space and k-regular functions space):

Definition 2.1: *Let $0 < \mu \leq 1$, $k \in \mathbb{N}_0$ and $p \in \mathbb{R}$, $p \geq 1$.*

$$L_p[a,b] \coloneqq \left\{ f\colon [a,b] \to \mathbb{R};\ f \text{ is measurable on } [a,b] \text{ and } \int_a^b |f(x)|^p \,\mathrm{d}x < \infty \right\},$$
$$L_\infty[a,b] \coloneqq \{ f\colon [a,b] \to \mathbb{R};\ f \text{ is measurable and bounded a. e. on } [a,b] \},$$
$$C^k[a,b] \coloneqq \{ f\colon [a,b] \to \mathbb{R};\ f \text{ has a continuous } k\text{-th derivative} \},$$
$$C[a,b] \coloneqq C^0[a,b],$$

We shall occasionally also use the following set of functions.

Definition 2.2: *By A^n or $A^n[a,b]$ we denote the set of functions with an absolutely continuous $(n-1)$st derivative, i.e., the functions f for which there exists (almost everywhere) a function $g \in L_1[a,b]$ such that*

$$f^{(n-1)}(x) = f^{(n-1)}(a) + \int_a^x g(t) \,\mathrm{d}t.$$

In this case we call g the generalized nth derivative *of f, and we simply write $g = f^{(n)}$.*

Definition 2.3: *Let $f\colon [a,b] \to \mathbb{R}$ be a differentiable function. We define*

(i) We denote by $\mathscr{D}$ the operator that maps a differentiable function onto its derivative, i.e.,

$$\mathscr{D} f(x) \coloneqq f'(x).$$

(ii) We denote by $\mathscr{I}_a$ the operator that maps a function f, assumed to be Riemann-integrable on the closed interval $[a,b]$, onto its primitive centered at a, i.e.,

$$\mathscr{I}_a f(x) = \int_a^x f(t) \,\mathrm{d}t$$

for $a \leq x \leq b$.

(iii) For $n \in \mathbb{N}$ we use the symbols $\mathscr{D}^n$ and $\mathscr{I}_a^n$ to denote the n-fold iterates of $\mathscr{D}$ and $\mathscr{I}_a$, respectively, i.e., we set $\mathscr{D}^1 \coloneqq \mathscr{D}$, $\mathscr{I}_a^1 \coloneqq \mathscr{I}_a$ and $\mathscr{D}^n \coloneqq \mathscr{D}\,\mathscr{D}^{n-1}$ and $\mathscr{I}_a^n \coloneqq \mathscr{I}_a\,\mathscr{I}_a^{n-1}$ for $n \geq 2$.

Our focus will be on how can we extend the concepts of Definition (iii) to $n \in \mathbb{R}_+$.

The results we will present in the following sections show that, if we restrict ourselves to suitable spaces of functions, we may unify the definitions of fractional differential operators into one, denoted by $\mathscr{D}_a^n$, which enjoys the following properties:

- If $f(x)$ is analytic, then the derivative $\mathscr{D}_a^n f(x)$ is, too.
- If $n \in \mathbb{N}$ is a positive integer, then $\mathscr{D}_a^n f(x)$ coincides with the usual $\mathscr{D}^n f(x)$.
- If $n = 0$, then the 0-order operator $\mathscr{D}_a^0$ coincides with the identity operator $\mathbb{I}$.
- Given constants $\lambda, \mu \in \mathbb{R}$, $\mathscr{D}_a^n[\lambda f(x) + \mu g(x)] = \lambda \cdot \mathscr{D}_a^n f(x) + \mu \cdot \mathscr{D}_a^n g(x)$ must hold, i.e., we require $\mathscr{D}_a^n$ to be *linear*.
- If $n > 0$ and $m > 0$ are any two positive real numbers, then $\mathscr{D}_a^n \left(\mathscr{D}_a^m f(x) \right) = \mathscr{D}_a^m \left(\mathscr{D}_a^n f(x) \right) = \mathscr{D}_a^{n+m} f(x)$ must hold, i.e., $\mathscr{D}_a^n$ enjoy the *semigroup property*.

2.3 Riemann-Liouville Integrals

We shall remark that the Fundamental Theorem of Calculus reads, in our notation,

$$\mathscr{D}\,\mathscr{I}_a f = f$$

which implies that

$$\mathscr{D}^n\,\mathscr{I}_a^n f = f \tag{2.1}$$

for $n \in \mathbb{N}$, i.e., $\mathscr{D}^n$ is the left inverse of $\mathscr{I}_a^n$ in a suitable space of functions.

In order to extend this property, we begin with the integral operator $\mathscr{I}_a^n$, in the case $n \in N$.

Lemma 2.4: *Let f be Riemann integrable on $[a, b]$. Then, for $a \leq x \leq b$ and $n \in \mathbb{N}$, we have:*

$$\mathscr{I}_a^n f(x) = \int_0^x \frac{(x-t)^{n-1} f(t)}{(n-1)!}\,\mathrm{d}t.$$

Proof. Let us apply operator $\mathscr{I}_a$ to f twice; we get:

$$\mathscr{I}_a^2 f(x) = \int_a^x \left(\int_0^t f(y)\,\mathrm{d}y \right) \mathrm{d}t. \tag{2.2}$$

Equation (2.2) can be regarded as a double integral, thus, if we switch the order of integration, we can reinterpret it as:

$$\mathscr{I}_a^2 f(x) = \int_a^x \left(\int_y^t f(y)\,\mathrm{d}t \right) \mathrm{d}y \tag{2.3}$$

and, since $f(y)$ is constant with respect to t, the integral in (2.3) becomes

$$\mathscr{I}_a^2 f(x) = \int_a^x (x-t) f(t)\,\mathrm{d}t.$$

Iterating the process, we obtain:

$$\mathscr{I}_a^3 f(x) = \frac{1}{2} \int_a^x (x-t)^2 f(t)\,\mathrm{d}t$$

and more generally we have

$$\mathscr{I}_a^n f(x) = \int_a^x \frac{(x-t)^{n-1} f(t)}{(n-1)!}\,\mathrm{d}t. \qquad \square$$

Lemma 2.5: *Let $m, n \in \mathbb{N}$ such that $m > n$, and let f be a function having a continuous n-th derivative on the interval $[a, b]$. Then,*

$$\mathscr{D}^n f = \mathscr{D}^m \mathscr{I}_a^{m-n} f.$$

Proof. By (2.1), we have $f = \mathscr{D}^{m-n} \mathscr{I}_a^{m-n} f$. Applying the operator $\mathscr{D}^n$ to both sides of this relation and using the fact that $\mathscr{D}^n \mathscr{D}^{m-n} = \mathscr{D}^m$, the statement follows. $\square$

We shall remark also that if $f \in L_1[a, b]$, then the Fundamental Theorem of Calculus reads in the form $\mathscr{D} \mathscr{I}_a f = f$ almost everywhere on $[a, b]$.

We introduce the well-known generalization of the factorial function.

Definition 2.6: *The function $\Gamma\colon (0, \infty) \to \mathbb{R}$, defined by*

$$\Gamma(x) \coloneqq \int_0^\infty t^{x-1} \mathrm{e}^{-t}\,\mathrm{d}t,$$

is called Euler's Gamma function.

We remark here that the following equality holds

$$\Gamma(n) = (n-1)!, \qquad \forall n \in \mathbb{N}$$

Now we see that Lemma 2.4 suggests the following noninteger generalization of the n-fold integral:

Definition 2.7: *Let $n \in \mathbb{R}_+$. The operator $\mathscr{I}_a^n$, defined on $L_1[a,b]$ by*

$$\mathscr{I}_a^n f(x) \coloneqq \frac{1}{\Gamma(n)} \int_a^x (x-t)^{n-1} f(t)\,\mathrm{d}t$$

for $a \le x \le b$, is called the Riemann-Liouville fractional integral operator of order n.

For $n = 0$, we set $\mathscr{I}_a^0 \coloneqq \mathbb{I}$, the identity operator.

It is evident that the Riemann-Liouville fractional integral coincides with the classical definition of $\mathscr{I}_a^n$ in the case $n \in \mathbb{N}$, (note also that we have extended the domain from Riemann integrable functions to Lebesgue integrable functions). Moreover, in the case $n \ge 1$ it is easily seen that the integral $\mathscr{I}_a^n f(x)$ exists for every $x \in [a,b]$ because the integrand is the product of an integrable function f and the continuous function $(x - \cdot)^{n-1}$. In the case $0 < n < 1$ the following result asserts that this definition is well posed.

Theorem 2.8: *Let $f \in L_1[a,b]$ and $n > 0$. Then, the integral $\mathscr{I}_a^n f(x)$ exists for almost every $x \in [a,b]$. Moreover, the function $\mathscr{I}_a^n f$ itself is also an element of $L_1[a,b]$.*

One important property of integer-order integral operators is preserved by our generalization:

Theorem 2.9: *Let $m, n \ge 0$ and $\phi \in L_1[a,b]$. Then,*

$$\mathscr{I}_a^m \mathscr{I}_a^n \phi = \mathscr{I}_a^{m+n} \phi$$

holds almost everywhere on $[a,b]$. If additionally $\phi \in C[a,b]$ or $m + n \ge 1$, then the equality holds everywhere on $[a,b]$.

Proof. We have

$$\mathscr{I}_a^m \mathscr{I}_a^n \phi(x) = \frac{1}{\Gamma(m)\Gamma(n)} \int_a^x (x-t)^{m-1} \int_a^t (t-\tau)^{n-1} \phi(\tau)\,\mathrm{d}\tau\,\mathrm{d}t.$$

In view of Theorem 2.8, the integrals exist, and by Fubini's theorem we may interchange the order of integration, obtaining

$$\begin{aligned}\mathscr{I}_a^m\,\mathscr{I}_a^n\,\phi(x) &= \frac{1}{\Gamma(m)\Gamma(n)}\int_a^x\int_\tau^t (x-t)^{m-1}(t-\tau)^{n-1}\phi(\tau)\,\mathrm{d}t\,\mathrm{d}\tau \\ &= \frac{1}{\Gamma(m)\Gamma(n)}\int_a^x \phi(\tau)\int_\tau^x (x-t)^{m-1}(t-\tau)^{n-1}\,\mathrm{d}t\,\mathrm{d}\tau.\end{aligned}$$

The substitution $t=\tau+s(x-\tau)$ yields

$$\begin{aligned}\mathscr{I}_a^m\,\mathscr{I}_a^n\,\phi(x) &= \frac{1}{\Gamma(m)\Gamma(n)}\int_a^x \phi(t)\int_0^1 [(x-\tau)(1-s)]^{m-1}\cdot \\ &= \frac{1}{\Gamma(m)\Gamma(n)}\int_a^x \phi(t)(x-\tau)^{m+n-1}\int_0^1 (1-s)^{m-1}s^{n-1}\,\mathrm{d}s\,\mathrm{d}\tau,\end{aligned}$$

where we notice that $\int_0^1(1-s)^{m-1}s^{n-1}\mathrm{d}s$ is Euler's β function, so that

$$\int_0^1 (1-s)^{m-1}s^{n-1}\mathrm{d}s = \frac{\Gamma(m)\Gamma(n)}{\Gamma(n+m)}$$

and thus

$$\mathscr{I}_a^m\,\mathscr{I}_a^n = \frac{1}{\Gamma(m+n)}\int_a^x \phi(\tau)(x-\tau)^{m+n-1}\,\mathrm{d}\tau = \mathscr{I}_a^{m+n}\,\phi(x)$$

almost everywhere on $[a,b]$.

Moreover, if $\phi\in C[a,b]$ then also $\mathscr{I}_a^n\,\phi\in C[a,b]$, and therefore $\mathscr{I}_a^m\,\mathscr{I}_a^n\,\phi\in C[a,b]$, and $\mathscr{I}_a^{m+n}\,\phi\in C[a,b]$, too. Thus, since two continuous functions coincide almost everywhere, they must coincide everywhere.

Finally, if $\phi\in L_1[a,b]$ and $m+n\geq 1$ we have, by the result above,

$$\mathscr{I}_a^m\,\mathscr{I}_a^n\,\phi = \mathscr{I}_a^{m+n} = \mathscr{I}_a^{m+n-1}\,\mathscr{I}_a^1\,\phi$$

almost everywhere. Since we have that $\mathscr{I}_a^1\,\phi$ is continuous, we also have that $\mathscr{I}_a^{m+n}\,\phi = \mathscr{I}_a^{m+n-1}\,\mathscr{I}_a^1\,\phi$ is continuous, and once again we may conclude that the two functions on either side of the equality almost everywhere are continuous; thus they must coincide everywhere. □

Corollary 2.10: *Under the assumptions of Theorem 2.9,*

$$\mathscr{I}_a^m\,\mathscr{I}_a^n\,\phi = \mathscr{I}_a^n\,\mathscr{I}_a^m\,\phi$$

There is an algebraic way to state this result.

Theorem 2.11: *The operators* $\{ \mathscr{I}_a^n \colon L_1[a,b] \to L_1[a,b];\ n \geq 0\}$ *form a commutative semigroup with respect to concatenation. The identity operator* $\mathscr{I}_a^0$ *is the neutral element of this semigroup.*

The following result will be the basis to obtain the analyticity property.

Theorem 2.12: *Let* $n > 0$. *Assume that* $(f_k)_{k=1}^{\infty}$ *is a uniformly convergent sequence of continuous functions on* $[a,b]$. *Then we may interchange the fractional integral operator and the limit process, i.e.,*

$$(\mathscr{I}_a^n \lim_{k\to\infty} f_k)(x) = (\lim_{k\to\infty} \mathscr{I}_a^n f_k)(x).$$

In particular, the sequence of functions $(\mathscr{I}_a^n f_k)_{k=1}^{\infty}$ *is uniformly convergent.*

Proof. We denote the (pointwise) limit of the sequence $(f_k)_{k=1}^{\infty}$ by f. It is well known that f is continuous. We then find

$$\begin{aligned} |\,\mathscr{I}_a^n f_k(x) - \mathscr{I}_a^n f(x)| &\leq \frac{1}{\Gamma(n)} \int_a^x |f_k(t) - f(t)|(x-t)^{n-1}\,\mathrm{d}t \\ &\leq \frac{1}{\Gamma(n)} \|f_k - f\|_\infty \int_a^x (x-t)^{n-1}\,\mathrm{d}t \\ &= \frac{1}{\Gamma(n+1)} \|f_k - f\|_\infty (x-a)^n \\ &\leq \frac{1}{\Gamma(n+1)} \|f_k - f\|_\infty (b-a)^n \end{aligned}$$

which converges to zero as $k \to \infty$ uniformly for all $x \in [a,b]$. □

Corollary 2.13: *Let* f *be analytic in* $(a-h, a+h)$ *for some* $h > 0$, *and let* $n > 0$. *Then*

$$\mathscr{I}_a^n f(x) = \sum_{k=0}^{\infty} \frac{(-1)^k\,(x-a)^{k+n}}{k!\,(n+k)\,\Gamma(n)}\,\mathscr{D}^k f(x)$$

for $a \leq x < a + h/2$, *and*

$$\mathscr{I}_a^n f(x) = \sum_{k=0}^{\infty} \frac{(x-a)^{k+n}}{\Gamma(k+1+n)}\,\mathscr{D}^k f(a)$$

for $a \leq x < a + h$. *In particular,* $\mathscr{I}_a^n f$ *is analytic in* $(a, a+h)$.

Proof. For the first statement, we use the definition of the Riemann-Liouville integral operator $\mathscr{I}_a^n$, namely,

$$\mathscr{I}_a^n f(x) = \frac{1}{\Gamma(n)} \int_a^x f(t)(x-t)^{n-1}\,\mathrm{d}t,$$

and expand $f(t)$ into a power series about x. Since $x \in [a, a + h/2)$, the power series converges in the entire interval of integration. Thus, by Theorem 2.11, we are allowed to exchange summation and integration. Then we use again Euler's β function to derive the representation:

$$\begin{aligned}\mathscr{I}_a^n f(x) &= \frac{1}{\Gamma(n)} \int_a^x (t-a)^\beta (x-t)^{n-1} \,\mathrm{d}t \\ &= \frac{1}{\Gamma(n)} (x-a)^{n+\beta} \int_0^1 s^\beta (1-s)^{n-1} \,\mathrm{d}s = \frac{\Gamma(\beta+1)}{\Gamma(n+\beta+1)} (x-a)^{n+\beta},\end{aligned}$$

and we find the final result.

For the second statement, we proceed in a similar way, but we now expand the power series at a and not at x. This allows us again to obtain the convergence of the series in the required interval.

The analyticity of $\mathscr{I}_a^n f$ follows immediately from the second statement □

2.4 Riemann-Liouville Derivatives

Having established these fundamental properties of Riemann-Liouville integral operators, we now come to the corresponding differential ones. To motivate the definition coming up, we recall Lemma 2.5 that (under certain conditions) states the identity

$$\mathscr{D}^n f = \mathscr{D}^m \mathscr{I}_a^{m-n} f$$

where m and n were integers such that $m > n$. Now assume that n is not an integer: then we may still choose an integer m such that $m > n$. We hence come to the following definition if we choose the value of the integer m to be as small as possible:

Definition 2.14: *Let $n \in \mathbb{R}_+$ and $m = \lceil n \rceil$, i.e., m is the smallest integer greater or equal than n. The operator $\mathscr{D}_a^n$, defined by*

$$\mathscr{D}_a^n f \coloneqq \mathscr{D}^m \mathscr{I}_a^{m-n} f$$

is called the Riemann-Liouville fractional differential operator of order n.

For $n = 0$, we set $\mathscr{D}_a^0 \coloneqq \mathbb{I}$, the identity operator.

Once again we see that, as a consequence of Lemma 2.5, the newly defined operator $\mathscr{D}_a^n$ coincides with the classical differential operator $\mathscr{D}^n$ whenever $n \in \mathbb{N}$.

In Lemma 2.5 we had not required m to be as small as possible; indeed arbitrary natural numbers for m were allowed as long as the inequality $m > n$ was satisfied. A similar statement holds here.

Lemma 2.15: *Let $n \in \mathbb{R}_+$ and let $m \in \mathbb{N}$ such that $m > n$. Then,*

$$\mathscr{D}_a^n = \mathscr{D}^m \, \mathscr{I}_a^{m-n}.$$

Proof. Our assumptions on m imply that $m \geq \lceil n \rceil$. Thus,

$$\mathscr{D}^m \, \mathscr{I}_a^{m-n} = \mathscr{D}^{\lceil n \rceil} \, \mathscr{D}^{m-\lceil n \rceil} \, \mathscr{I}_a^{m-\lceil n \rceil} \, \mathscr{I}_a^{\lceil n \rceil - n} = \mathscr{D}^{\lceil n \rceil} \, \mathscr{I}_a^{\lceil n \rceil - n} = \mathscr{D}_a^n$$

in view of the semigroup property of booth integer differentiation and fractional integration and (2.1). □

We have seen in Theorem 2.9 that the Riemann-Liouville integral operators form a semigroup. Therefore it is natural to ask whether and, if so, when the Riemann-Liouville differential operators have got such a structure. We begin our investigations in this direction with a positive result.

Theorem 2.16: *Assume that $n_1, n_2 \geq 0$. Moreover let $\phi \in L_1[a,b]$ such that $f = \mathscr{I}_a^{n_1+n_2} \phi$. Then,*

$$\mathscr{D}_a^{n_1} \, \mathscr{D}_a^{n_2} \, f = \mathscr{D}_a^{n_1+n_2} \, f.$$

Proof. Note that in order to apply this identity we do not need to know the function ϕ explicitly; it is sufficient to know that such a function exists.

By our assumption on f and the definition of the Riemann-Liouville differential operator,

$$\mathscr{D}_a^{n_1} \, \mathscr{D}_a^{n_2} \, f = \mathscr{D}_a^{n_1} \, \mathscr{D}_a^{n_2} \, \mathscr{I}_a^{n_1+n_2} \, \phi = \mathscr{D}^{\lceil n_1 \rceil} \, \mathscr{I}_a^{\lceil n_1 \rceil - n_1} \, \mathscr{D}^{\lceil n_2 \rceil} \, \mathscr{I}_a^{\lceil n_2 \rceil - n_2} \, \mathscr{I}_a^{n_1+n_2} \, \phi.$$

The semigroup property of the integral operators allows us to rewrite this expression as

$$\begin{aligned} \mathscr{D}_a^{n_1} \, \mathscr{D}_a^{n_2} \, f &= \mathscr{D}^{\lceil n_1 \rceil} \, \mathscr{I}_a^{\lceil n_1 \rceil - n_1} \, \mathscr{D}^{\lceil n_2 \rceil} \, \mathscr{I}_a^{\lceil n_2 \rceil + n_1} \, \phi \\ &= \mathscr{D}^{\lceil n_1 \rceil} \, \mathscr{I}_a^{\lceil n_1 \rceil - n_1} \, \mathscr{D}^{\lceil n_2 \rceil} \, \mathscr{I}_a^{\lceil n_2 \rceil} \, \mathscr{I}_a^{n_1} \, \phi. \end{aligned}$$

Because of (2.1) and the fact that the orders of the integral and differential operators involved are natural numbers, we find that this is equivalent to

$$\mathscr{D}_a^{n_1} \, \mathscr{D}_a^{n_2} \, f = \mathscr{D}^{\lceil n_1 \rceil} \, \mathscr{I}_a^{\lceil n_1 \rceil - n_1} \, \mathscr{I}_a^{n_1} \, \phi = \mathscr{D}^{\lceil n_1 \rceil} \, \mathscr{I}_a^{\lceil n_1 \rceil} \, \phi$$

where we have once again used the semigroup property of fractional integration. We may now use (2.1) one more time and find that

$$\mathscr{D}_a^{n_1} \, \mathscr{D}_a^{n_2} \, f = \phi$$

The proof that $\mathscr{D}_a^{n_1+n_2} \, f = \phi$ goes along similar lines. □

Recall that one of the key features that we wanted to keep also in fractional operators was (2.1). It turns out that the Riemann-Liouville definitions indeed have this property.

Theorem 2.17: *Let $n \geq 0$. Then, for every $f \in L_1[a,b]$,*

$$\mathscr{D}_a^n \, \mathscr{I}_a^n \, f = f$$

almost everywhere.

Proof. The case $n = 0$ is trivial for then $\mathscr{D}_a^n$ and $\mathscr{I}_a^n$ are both the identity operator.

For $n > 0$ we proceed as in the proof of Theorem 2.16: let $m = \lceil n \rceil$. Then, by the definition of $\mathscr{D}_a^n$, the semigroup property of fractional integration and (2.1) (which may be applied here since $n \in \mathbb{N}$),

$$\mathscr{D}_a^n \, \mathscr{I}_a^n \, f(x) = \mathscr{D}^m \, \mathscr{I}_a^{m-n} \, \mathscr{I}_a^n \, f(x) = \mathscr{D}^m \, \mathscr{I}_a^m \, f(x) = f(x). \qquad \square$$

Essentially this result and its proof have already been known to Abel even though he has not denoted the operators involved as integrals and derivatives of fractional order, respectively.

Now we come to an analogue of Theorem 2.11.

Theorem 2.18: *Let $n > 0$. Assume that $(f_k)_{k=1}^{\infty}$ is a uniformly convergent sequence of continuous functions on $[a,b]$, and that $\mathscr{D}_a^n \, f_k$ exists for every k. Moreover assume that $(\mathscr{D}_a^n \, f_k)_{k=1}^{\infty}$ converges uniformly on $[a+\varepsilon, b]$ for every $\varepsilon > 0$. Then, for every $x \in (a,b]$, we have*

$$(\lim_{k\to\infty} \mathscr{D}_a^n \, f_k)(x) = (\mathscr{D}_a^n \lim_{k\to\infty} f_k)(x).$$

Proof. We recall that $\mathscr{D}_a^n = \mathscr{D}^{\lceil n \rceil} \, \mathscr{I}_a^{\lceil n \rceil - n}$. Theorem 2.11 guarantees that the sequence $(\mathscr{I}_a^{\lceil n \rceil - n} \, f_k)_k$ is uniformly convergent, and we may interchange the limit operation and the fractional integral. By assumption, the $\lceil n \rceil$th derivative of this series converges uniformly on every compact subinterval of $(a,b]$. Thus, by a standard theorem from Analysis, we may also interchange the limit operator and the differential operator whenever $a < x \leq b$. $\square$

We can immediately deduce an analogue of Corollary 2.13.

Corollary 2.19: *Let f be analytic in $(a-h, a+h)$ for some $h > 0$, and let $n > 0$, $n \notin \mathbb{N}$. Then $D_a^n f$ is analytic in $(a, a+h)$.*

Proof. It suffices to use Corollary 2.13 and the definition of the operator $\mathscr{D}_a^n$,

$$\mathscr{D}_a^n \, f(x) = \mathscr{D}^{\lceil n \rceil} \, \mathscr{I}_a^{\lceil n \rceil - n} \, f(x).$$

Then it follows that

$$\mathscr{D}_a^n f(x) = \sum_{k=0}^{\infty} \frac{(x-a)^{k-n}}{\Gamma(k+1-n)} \mathscr{D}^k f(a)$$

and the statement follows. □

The next result assures that generalizing the operators to a non-integer order nevertheless preserves their linearity.

Theorem 2.20: *Let f_1 and f_2 be two functions defined on $[a,b]$ such that $\mathscr{D}_a^n f_1$ and $\mathscr{D}_a^n f_2$ exist almost everywhere. Moreover, let $c_1, c_2 \in \mathbb{R}$. Then, $\mathscr{D}_a^n(c_1 f_1 + c_2 f_2)$ exists almost everywhere, and*

$$D_a^n(c_1 f_1 + c_2 f_2) = c_1 \mathscr{D}_a^n f_1 + c_2 \mathscr{D}_a^n f_2.$$

Proof. This linearity property of the fractional differential operator is an immediate consequence of the definition of $\mathscr{D}_a^n$. □

We now introduce a notation that will be useful in later proofs.

Definition 2.21: *Let f be a m-times differentiable function in a open neighborhood of $a \in \mathbb{R}$ and let $m \in \mathbb{N}$; we denote by $T_m[f;a]$ the Taylor expansion of f centered at a, i.e.,*

$$T_m[f;a] = \sum_{k=0}^{m} \frac{f^{(k)}(a)}{k!}(x-a)^k.$$

Remark 2.22. There is one more fundamental difference between differential operators of integer order and the Riemann-Liouville fractional derivatives: The former are *local* operators, the latter are not, in the sense that in order to calculate $\mathscr{D}^n f(x)$ for $n \in \mathbb{N}$, it is sufficient to know f in an arbitrarily small neighborhood of x. This follows from the classical representation of $\mathscr{D}^n$ as a limit of a difference quotient. However, to calculate $\mathscr{D}_a^n f(x)$ for $n \notin \mathbb{N}$, the definition tells us that we need to know f throughout the entire interval $[a,x]$.

Having established a theory of Riemann-Liouville differential and integral operators separately, we now investigate how they interact. A very important first result in this context has already been shown in Theorem 2.17 above: $\mathscr{D}_a^n$ is the left inverse of $\mathscr{I}_a^n$. Yet, we cannot claim that it is the right inverse. More precisely, we have the following situation.

Theorem 2.23: *Let $n > 0$. If there exists some $\phi \in L_1[a,b]$ such that $f = \mathscr{I}_a^n \phi$, then*

$$\mathscr{I}_a^n \mathscr{D}_a^n f = f$$

almost everywhere.

Proof. This is an immediate consequence of the previous result: We have, by definition f f and Theorem 2.17, that

$$\mathscr{I}_a^n \mathscr{D}_a^n f = \mathscr{I}_a^n[\mathscr{D}_a^n \mathscr{I}_a^n \phi] = \mathscr{I}_a^n \phi = f. \qquad \square$$

If f is not as required in the assumptions of Theorem 2.23, then we obtain a different representation for $\mathscr{I}_a^n \mathscr{D}_a^n f$.

Theorem 2.24: *Let $n > 0$ and $m = \lfloor n \rfloor + 1$. Moreover, assume f is such that $\mathscr{I}_a^{m-n} f \in A^m[a,b]$. Then,*

$$\mathscr{I}_a^n \mathscr{D}_a^n f(x) = f(x) - \sum_{k=0}^{m-1} \frac{(x-a)^{n-k-1}}{\Gamma(n-k)} \lim_{z \to a^+} \mathscr{D}^{m-k-1} \mathscr{I}_a^{m-n} f(z).$$

Specifically, for $0 < n < 1$ we have

$$\mathscr{I}_a^n \mathscr{D}_a^n f(x) = f(x) - \frac{(x-a)^{n-1}}{\Gamma(n)} \lim_{z \to a^+} \mathscr{I}_a^{1-n} f(z).$$

Proof. We first note that the limits on the right-hand side exist because of our assumption on f that implies the continuity of $\mathscr{D}^{m-1} \mathscr{I}_a^{m-n} f$. Moreover, because of this assumption, there exists some $\phi \in L_1$ such that

$$\mathscr{D}^{m-1} \mathscr{I}_a^{m-n} f = \mathscr{D}^{m-1} \mathscr{I}_a^{m-n} f(a) + \mathscr{I}_a^1 \phi.$$

This is a classical differential equation of order $m-1$ for $\mathscr{I}_a^{m-n} f$; its solution is of the form

$$\mathscr{I}_a^{m-n} f(x) = \sum_{k=0}^{m-1} \frac{(x-a)^k}{k!} \lim_{z \to a^+} \mathscr{D}^k \mathscr{I}_a^{m-n} f(z) + \mathscr{I}_a^m \phi(x). \tag{2.4}$$

Thus, by definition of $\mathscr{D}_a^n$,

$$\begin{aligned}
\mathscr{I}_a^n \mathscr{D}_a^n f(x) &= \mathscr{I}_a^n \mathscr{D}^m \mathscr{I}_a^{m-n} f(x) \\
&= \mathscr{I}_a^n \mathscr{D}^m \left[\sum_{k=0}^{m-1} \frac{(\cdot - a)^k}{k!} \lim_{z \to a^+} \mathscr{D}^k \mathscr{I}_a^{m-n} f(z) + \mathscr{I}_a^n \phi \right](x) \\
&= \mathscr{I}_a^n \mathscr{D}^m \mathscr{I}_a^m \phi(x) + \sum_{k=0}^{m-1} \frac{\mathscr{I}_a^n \mathscr{D}^m[(\cdot - a)^k](x)}{k!} \lim_{z \to a^+} \mathscr{D}^k \mathscr{I}_a^{m-n} f(z) \\
&= \mathscr{I}_a^n \phi(x)
\end{aligned} \tag{2.5}$$

because of Theorem 2.17 (note that $\mathscr{D}^m$ annihilates every summand in the sum). Next we apply the operator $\mathscr{D}_a^{m-n}$ to both sides of (2.4) and find, in view of Theorem (2.17), that

$$\begin{aligned} f(x) &= \sum_{k=0}^{m-1} \frac{\mathscr{D}^m[(\cdot - a)^k](x)}{k!} \lim_{z\to a^+} \mathscr{D}^k \, \mathscr{I}_a^{m-n} f(z) + \mathscr{D}_a^{m-n} \, \mathscr{I}_a^m \, \phi(x) \\ &= \sum_{k=0}^{m-1} \frac{\mathscr{D}^m[(\cdot - a)^k](x)}{k!} \lim_{z\to a^+} \mathscr{D}^k \, \mathscr{I}_a^{m-n} f(z) + \mathscr{D}_a^1 \, \mathscr{I}_a^{1-m+n} \, \mathscr{I}_a^m \, \phi(x). \end{aligned}$$

We now recognize the fraction in the summation to be

$$\frac{\mathscr{D}^m[(\cdot - a)^k](x)}{k!} = \frac{(x-a)^{k+n-m}}{\Gamma(k+n-m+1)}$$

so that after replacing it, and applying the semigroup property of fractional integration and Theorem 2.17 to manipulate the remaining term we have

$$f(x) = \sum_{k=0}^{m-1} \frac{(x-a)^{k+n-m}}{\Gamma(k+n-m+1)} \lim_{z\to a^+} \mathscr{D}^k \, \mathscr{I}_a^{m-n} f(z) + \mathscr{I}_a^n \, \phi(x). \tag{2.6}$$

Finally we substitute k in the sum by $m-k-1$, solve for $\mathscr{I}_a^n \, \phi(x)$ and combine the result with (2.5) to obtain

$$\mathscr{I}_a^n \, \mathscr{D}_a^n \, f(x) = \mathscr{I}_a^n \, \phi(x) = f(x) - \sum_{k=0}^{m-1} \frac{(x-a)^{n-k-1}}{\Gamma(n-k)} \lim_{z\to a^+} \mathscr{D}^{m-k-1} \, \mathscr{I}_a^{m-n} \, f(z)$$

as desired. □

2.5 Grünwald-Letnikov Operators

In the classical calculus it is well known that derivatives can be expressed as differential quotients, i.e., as limits of backward difference quotients:

$$\mathscr{D}^1 \, f(x) = \lim_{h\to 0} \frac{1}{h}[f(x) - f(x-h)].$$

Applying the operator twice leads to the second-order derivative of $f(x)$:

$$\mathscr{D}^2 \, f(t) = \lim_{h\to 0} \frac{1}{h^2}[f(x) - 2f(x-h) + f(x-2h)].$$

Iterating the process n-times (i.e., applying the operator n-times) we have:

Theorem 2.25: *Let $n \in \mathbb{N}$, $f \in \mathbb{C}^n[a,b]$ and $a < x \leq b$. Then*

$$\mathscr{D}^n f(x) = \lim_{h \to 0} \frac{\Delta_h^n f(x)}{h^n}$$

where

$$\Delta_h^n f(x) := \sum_{k=0}^{n} (-1)^k \binom{n}{k} f(x - kh). \tag{2.7}$$

This result is actually not only useful for analytical investigations; by using a finite positive value for h instead of performing the limit operation $h \to 0$ it also gives us a straightforward numerical approximation for the derivative. In view of these advantages of this representation, it is evidently desirable to have an analogue also for the fractional case. Such a construction is possible; it dates back to the work of Grünwald and Letnikov. Indeed, all we have to do is to give a meaning to the finite difference in (2.7) for $n \notin \mathbb{N}$. To this end we recall the definition of the binomial coefficient with non-integer upper coefficients:

$$\binom{n}{k} := \frac{n(n-1)\cdots(n-k+1)}{k!}, \quad n \in \mathbb{R}_+,\ k \in \mathbb{N}.$$

We also note that $\binom{n}{k} = 0$ if $n \in \mathbb{N}$ and $n < k$. Thus for $n \in \mathbb{N}$, equation (2.7) is equivalent to

$$\Delta_h^n f(x) := \sum_{k=0}^{\infty} (-1)^k \binom{n}{k} f(x - kh), \tag{2.8}$$

indeed, all terms from $k = n + 1$ on vanish.

We observe that the representation (2.8) introduces two problems in the case $n \notin \mathbb{N}$ where none of the binomial coefficients vanishes, so that this expression really represents an infinite series:

- In order to evaluate the expression in (2.8) for all $x \in (a,b]$, the function f needs to be defined on $(-\infty, b]$.
- The function f must be such that the series converges.

These two problems can be resolved simultaneously by a simple concept: Given a function $f\colon [a,b] \to \mathbb{R}$, define a new function

$$f^*\colon (-\infty, b] \to \mathbb{R}, \qquad x \mapsto \begin{cases} f(x) & \text{if } x \in [a,b], \\ 0 & \text{if } x \in (-\infty, a), \end{cases}$$

and use this function instead of the original f. In view of the fact that f and f^* coincide on the interval where both functions are defined, we interpret f^* as a

continuation of f and, with a slight abuse of notation, we will from now on write f in place of f^*.

This leads us to the required generalization of the concept of differential quotients. For the sake of simplicity, we impose a restriction in the way that $h \to 0$; specifically for the value of x under consideration we assume that h takes only the values $h_N = (x-a)/N$, $N = 1, 2, \ldots$, even if in principle such an assumption is not necessary, actually.

Definition 2.26: *Let $n > 0$, $f \in C^{\lceil n \rceil}[a,b]$ and $a < x \leq b$. Then*

$$\widetilde{\mathscr{D}}_a^n f(x) = \lim_{N\to\infty} \frac{\Delta_{h_N}^n f(x)}{h_N^n} = \lim_{N\to\infty} \frac{1}{h_N^n} \sum_{k=0}^{N} (-1)^k \binom{n}{k} f(x - kh_N)$$

with $h_N = (x-a)/N$ is called the Grünwald-Letnikov fractional derivative *of order n of the function f.*

The following results (for which we omit the quite cumbrous proofs [10]) explain the relation between this new notion of a fractional derivative and the one that we already know.

Theorem 2.27: *Let $n > 0$, $m = \lceil n \rceil$ and $f \in C^m[a,b]$. Then, for $x \in (a,b]$,*

$$\widetilde{\mathscr{D}}_a^n f(x) = \mathscr{D}_a^n f(x)$$

Theorem 2.28: *Let $n > 0$, $f \in C[a,b]$ and $a \leq x \leq b$. Then, with $h_N = (x-a)/N$, we have*

$$\mathscr{I}_a^n f(x) = \lim_{N\to\infty} h_N^n \sum_{k=0}^{N} (-1)^k \binom{n}{k} f(x - kh_N)$$

2.6 Caputo's Approach

It turns out that the Riemann-Liouville derivatives have certain disadvantages when trying to model real-world phenomena with fractional differential equations. We shall therefore now discuss a modified concept of a fractional derivative. As we will see below when comparing the two ideas, this second one seems to be better suited to such tasks.

We start with a preliminary definition.

Definition 2.29: *Let $n \geq 0$ and $m = \lceil n \rceil$. Then, we define the operator $\widehat{\mathscr{D}}_a^n$ by*

$$\widehat{\mathscr{D}}_a^n f := \mathscr{I}_a^{m-n} \mathscr{D}^m f$$

whenever $\mathscr{D}^m f \in L_1[a,b]$.

First off, let us look at the case $n \in \mathbb{N}$. Here we have $m = n$ and hence our definition implies

$$\widehat{\mathscr{D}}_a^n f = \mathscr{I}_a^0 \mathscr{D}^n f = \mathscr{D}^n f,$$

i.e., we recover the standard definition in the classical case.

The key to the construction of the alternative differential operator that we are looking for is the following identity involving Riemann-Liouville derivatives on one hand and the newly defined operator on the other.

Theorem 2.30: *Let $n \geq 0$ and $m = \lceil n \rceil$. Moreover assume that $f \in A^m[a,b]$. Then,*

$$\widehat{\mathscr{D}}_a^n f = \mathscr{D}_a^n [f - T_{m-1}[f;a]]$$

almost everywhere. Here, $T_{m-1}[f;a]$ denotes the Taylor polynomial of degree $m-1$ for the function f, centered at a; in the case $m = 0$ we define $T_{m-1}[f;a] := 0$.

Note that the expression on the right-hand side of the equation exists if $\mathscr{D}_a^n f$ exists and f possess $m-1$ derivatives at a, the latter condition making sure that the Taylor polynomial exists. This condition is weaker than the previous condition that $f \in A^m$. Therefore we will, from now on, use the latter expression. A formalization is given as follows.

Proof. In the case $n \in \mathbb{N}$ the statement is trivial because, both sides of the equation reduce to $\mathscr{D}^n f$. We therefore only have to consider the case $n \notin \mathbb{N}$, which implies that $m > n$.

In this case we have

$$\begin{aligned} \mathscr{D}_a^n [f - T_{m-1}[f;a]](x) &= \mathscr{D}^m \mathscr{I}_a^{m-n} [f - T_{m-1}[f;a]](x) \\ &= \frac{\mathrm{d}^m}{\mathrm{d}x^m} \int_a^x \frac{(x-t)^{m-n-1}}{\Gamma(m-n)} (f(t) - T_{m-1}[f;a](t)) \, \mathrm{d}t. \end{aligned} \tag{2.9}$$

A partial integration of the integral is permitted and yields

$$\begin{aligned} &\int_a^x \frac{1}{\Gamma(m-n)} (f(t) - T_{m-1}[f;a](t))(x-t)^{m-n-1} \, \mathrm{d}t \\ &= -\frac{1}{\Gamma(m-n+1)} [(f(t) - T_{m-1}[f;a](t))(x-t)^{m-n}]_{t=a}^{t=x} \\ &+ \frac{1}{\Gamma(m-n+1)} \int_a^x (\mathscr{D} f(t) - \mathscr{D} T_{m-1}[f;a](t))(x-t)^{m-n} \, \mathrm{d}t. \end{aligned}$$

The term outside the integral is zero (the first factor vanishes at the lower bound, the second vanishes at the upper bound). Thus,

$$\mathscr{I}_a^{m-n} [f - T_{m-1}[f;a]] = \mathscr{I}_a^{m-n+1} \mathscr{D} [f - T_{m-1}[f;a]].$$

Under our assumptions, we may repeat this process a total number of m times, and this results in

$$\mathscr{I}_a^{m-n}[f-T_{m-1}[f;a]] = \mathscr{I}_a^{2m-n}\,\mathscr{D}^m[f-T_{m-1}[f;a]] = \mathscr{I}_a^m\,\mathscr{I}_a^{m-n}\,\mathscr{D}^m[f-T_{m-1}[f;a]].$$

We note that $\mathscr{D}^m\,T_{m-1}[f;a] \equiv 0$ because T_{m-1} is a polynomial of degree $m-1$. Thus, the last identity can be simplified to

$$\mathscr{I}_a^{m-n}[f - T_{m-1}[f;a]] = \mathscr{I}_a^m\,\mathscr{I}_a^{m-n}\,\mathscr{D}^m\,f.$$

This may be combined with (2.9) to obtain

$$\mathscr{D}_a^n[f - T_{m-1}[f;a]](x) = \mathscr{D}^m\,\mathscr{I}_a^m\,\mathscr{I}_a^{m-n}\,\mathscr{D}^m\,f = \mathscr{I}_a^{m-n}\,\mathscr{D}^m\,f = \widehat{\mathscr{D}}_a^n f$$

in view of (2.1). □

Definition 2.31: *Assume that $n \geq 0$ and that f is such that $\mathscr{D}_a^n[f - T_{m-1}[f;a]]$ exists, where $m = \lceil n \rceil$. Then we define the function $\mathscr{D}_{*a}^n f$ by*

$$\mathscr{D}_{*a}^n f \coloneqq \mathscr{D}_a^n[f - T_{m-1}[f;a]].$$

The operator $\mathscr{D}_{*a}^n$ is called the Caputo differential operator of order n.

Actually this concept has been introduced independently by many authors, including Caputo [6, 7] and Rabotnov [22]. We follow the most common convention and we will name the derivative after Caputo only.

Once again we note for $n \in \mathbb{N}$ that $m = n$ and hence

$$\mathscr{D}_{*a}^n f = \mathscr{D}_a^n[f - T_{n-1}[f;a]] = \mathscr{D}^n f - \mathscr{D}^n(T_{n-1}[f;a]) = \mathscr{D}^n f$$

because $T_{n-1}[f;a]$ is a polynomial of degree $n-1$ that is annihilated by the classical operator $\mathscr{D}^n$. So in this case we recover the usual differential operator as well. In particular, $\mathscr{D}_{*a}^0$ is once again the identity operator.

Remark 2.32. As in the case of the Riemann-Liouville operators, we see that the Caputo derivatives are not local either.

Another representation for the Caputo operator can be obtained by combining its definition with Theorem 2.27:

Lemma 2.33: *Let $n > 0$, $m = \lceil n \rceil$ and $f \in C^m[a,b]$. Then, for $x \in (a,b]$,*

$$\mathscr{D}_{*a}^n f(x) = \lim_{N\to\infty} \frac{1}{h_N^n} \sum_{k=0}^{N} (-1)^k \binom{n}{k} [f(x - kh_N) - T_{m-1}[f;a](x - kh_N)]$$

with $h_N = (x-a) = /N$.

This representation has proven to be useful for numerical work [16, 9].

Lemma 2.34: *Let $n \geq 0$ and $m = \lceil n \rceil$. Assume that f is such that both ${}_{*a}\mathscr{D}^n f$ and ${}_a\mathscr{D}^n f$ exist. Then,*

$$ {}_a\mathscr{D}^n f = {}_{*a}\mathscr{D}^n f $$

holds if and only if f has an m-fold zero at a, i.e., if and only if

$$ \mathscr{D}^k f(a) = 0 \quad \text{for} \quad k = 0, 1, \ldots, m-1. $$

When it comes to the composition of Riemann-Liouville integrals and Caputo differential operators, we find that the Caputo derivative is also a left inverse of the Riemann-Liouville integral:

Theorem 2.35: *If f is continuous and $n \geq 0$, then*

$$ {}_{*a}\mathscr{D}^n \, {}_a\mathscr{I}^n f = f. $$

A proof involving estimations for Riemann-Liouville integrals of μ-Lipschitz functions can be found in [10].

Once again, we find that the Caputo derivative is not the right inverse of the Riemann-Liouville integral:

Theorem 2.36: *Assume that $n \geq 0$, $m = \lceil n \rceil$, and $f \in A^m[a, b]$. Then*

$$ {}_a\mathscr{I}^n \, {}_{*a}\mathscr{D}^n f(x) = f(x) - \sum_{k=0}^{m-1} \frac{\mathscr{D}^k f(a)}{k!}(x-a)^k. $$

Proof. By Theorem 2.30 and Definition 2.29, we have

$$ {}_{*a}\mathscr{D}^n f = {}_a\widehat{\mathscr{D}}^n f = {}_a\mathscr{I}^{m-n} \, \mathscr{D}^m f. $$

Thus, applying the operator ${}_a\mathscr{I}^n$ to both sides of this equation and using the semigroup property of fractional integration, we obtain

$$ {}_a\mathscr{I}^n \, {}_{*a}\mathscr{D}^n f = {}_a\mathscr{I}^n \, {}_a\mathscr{I}_{m-n} \, \mathscr{D}^m f = {}_a\mathscr{I}^m \, \mathscr{D}^m f. $$

By the classical version of Taylor's theorem we have that

$$ f(x) = \sum_{k=0}^{m-1} \frac{D^k f(a)}{k!}(x-a)^k + {}_a\mathscr{I}^m \, \mathscr{D}^m f(x). $$

Combining these two equations we derive the claim. □

Theorem 2.37: *Let $f \in C^{\mu}[a,b]$ for some $\mu \in \mathbb{N}$. Moreover let $n \in [0,\mu]$. Then,*

$$\mathscr{D}_a^{\mu-n}\,\mathscr{D}_{*a}^{n}\,f = \mathscr{D}^{\mu}\,f.$$

Notice that the operator $\mathscr{D}^{\mu}$ appearing on the right-hand side of the claim is a classical (integer-order) differential operator.

Proof. If n is an integer then both differential operators on the left-hand side reduce to integer-order operators and hence we obtain the desired result by an application of the definition of the iterated operators, namely Definition (iii).

If n is not an integer then we may invoke Theorem 2.30 to conclude that

$$\mathscr{D}_{*a}^{n}\,f = \widehat{\mathscr{D}}_a^{n} f = \mathscr{I}_a^{\lceil n\rceil - n}\,\mathscr{D}^{\lceil n\rceil}\,f.$$

Combining this with the definition of the Riemann-Liouville derivative and using the semigroup property of fractional integration and equation (2.1) we find

$$\begin{aligned}\mathscr{D}_a^{\mu-n}\,\mathscr{D}_{*a}^{n}\,f &= \mathscr{D}^{\mu-\lceil n\rceil+1}\,\mathscr{I}_a^{n+1-\lceil n\rceil}\,\mathscr{I}_a^{\lceil n\rceil-n}\,\mathscr{D}^{\lceil n\rceil}\,f = \mathscr{D}^{\mu-\lceil n\rceil+1}\,\mathscr{I}_a^{1}\,\mathscr{D}^{\lceil n\rceil}\,f \\ &= \mathscr{D}^{\mu-\lceil n\rceil}\,\mathscr{D}^{\lceil n\rceil}\,f = \mathscr{D}^{\mu}\,f.\end{aligned}$$

□

Caputo differential operators are linear, too.

Theorem 2.38: *Let $f_1, f_2\colon [a,b] \to \mathbb{R}$ be such that $\mathscr{D}_{*a}^{n} f_1$ and $\mathscr{D}_{*a}^{n} f_2$ exist almost everywhere and let $c_1, c_2 \in \mathbb{R}$. Then $\mathscr{D}_{*a}^{n}(c_1 f_1 + c_2 f_2)$ exists almost everywhere, and*

$$\mathscr{D}_{*a}^{n}(c_1 f_1 + c_2 f_2) = c_1\,\mathscr{D}_{*a}^{n} f_1 + c_2\,\mathscr{D}_{*a}^{n} f_2.$$

Proof. This linearity property of the fractional differential operator is an immediate consequence of the definition of D_{*a}^{n}. □

2.7 Laplace and Fourier Transforms

Laplace and Fourier integral transforms are fundamental tools in systems and control engineering. For this reason, we will give here the equation of these transforms for the defined fractional-order operators. Notice that we can take these integral transforms since we saw before that fractional operators of L_1 functions

are themselves L_1 functions. These equations are [21]:

$$\mathscr{L}[\mathscr{I}_a^n f(t)] = s^{-n}F(s), \tag{2.10}$$

$$\mathscr{L}[\mathscr{D}_a^n f(t)] = s^n F(s) - \sum_{k=0}^{m-1} s^k [\mathscr{D}_a^{n-k-1} f(t)]_{t=0}, \tag{2.11}$$

$$\mathscr{L}[\mathscr{D}_{*a}^n f(t)] = s^n F(s) - \sum_{k=0}^{m-1} s^{n-k-1} f^{(k)}(0), \tag{2.12}$$

$$\mathscr{L}[\tilde{\mathscr{D}}_a^n f(t)] = s^n F(s), \tag{2.13}$$

$$\mathscr{F}[\mathscr{I}_a^n f(t)] = \mathscr{F}\left[\frac{t_+^{n-1}}{\Gamma(n)}\right] \mathscr{F}\{f(t)\} = (j\omega)^{-n}F(\omega), \tag{2.14}$$

$$\mathscr{F}[\mathscr{D}^n f(t)] = \mathscr{F}\{\mathscr{D}^m \mathscr{I}^{m-n} f(t)] = (j\omega)^n F(\omega), \tag{2.15}$$

where $m - 1 \leq n < m$.

2.8 An Example Application of Fractional Calculus

As a conclusion, let us take a brief look at a simple but not unrealistic example of a model arising in mechanics where fractional derivatives can be used successfully. The model has been originally proposed as theoretical work by Nutting [23, 24]; Scott Blair et al. [4] were among the first to confirm its value in practice.

Specifically, we want to describe the behavior of certain materials under the influence of external forces. Traditionally, laws of Hooke and Newton are employed. We are interested in the relation between stress $\sigma(t)$ and strain $\varepsilon(t)$, both of which are considered as functions of time t. If we are dealing with viscous liquids, then Newton's law

$$\sigma(t) = \eta \mathscr{D}^1 \varepsilon(t) \tag{2.16}$$

is the preferred tool. Here the material constant η is the so-called *viscosity* of the material. Hooke's law

$$\sigma(t) = E \mathscr{D}^0 \varepsilon(t) \tag{2.17}$$

on the other hand is the correct way of modeling the stress-strain relationship for elastic solids. The constant E is known as the modulus of *elasticity* of the material. Here we mention the operator $\mathscr{D}^0$ (i.e., the identity operator) in(2.1) explicitly to stress the formal similarity between the two laws.

Now consider an experiment where the strain is manipulated in a controlled fashion such that, say, $\varepsilon(t) = t$ for $t \in [0, T]$ with some $T > 0$. It then follows that the stress behaves as

$$\sigma(t) = Et$$

in the case of an elastic solid and

$$\sigma(t) = \eta$$

for a viscous liquid. We may summarize the equations in the form

$$\psi_k = \frac{\sigma(t)}{\varepsilon(t)} t^k \tag{2.18}$$

where $\psi_0 = E$ and $\psi_1 = \eta$. Evidently the case $k = 0$ corresponds to Hooke's law for solids and $k = 1$ refers to Newton's law for liquids.

In practice it is not uncommon to find so-called *viscoelastic* materials[2] that exhibit a behavior somewhere between the pure viscous liquid and the pure elastic solid, i.e., where one would observe a relationship of the form (2.18) with $0 < k < 1$. In this case it is appropriate to interpret k as a second material constant in addition to ψ_k. Classical examples are polymers, but some types of biological tissue may also share this property as well as a number of metals (aluminum, for example) at least under certain temperature and pressure conditions. It should be noted that for the case of a constant strain ε, the stress in such a material would develop according to the formula

$$\sigma(t) = c \cdot t^{-k},$$

where c is a real constant, and thus converges to zero for long observation times. In this respect it once again lies between a viscous liquid for which σ vanishes identically and an elastic solid whose stress σ is a nonzero constant.

It seems therefore natural to assume that it is also possible to model the relation between stress and strain for such a viscoelastic material via an equation of the form

$$\sigma(t) = \nu \, \mathscr{D}^k \, \varepsilon(t) \tag{2.19}$$

where ν is a material constant and $k \in (0, 1)$ is the parameter introduced above. This equation "interpolates" between (2.16) and (2.17) in a similar spirit. In view of the above mentioned theoretical foundations of (2.19) laid by Nutting, this relation is frequently called *Nutting's law*.

[2] E.g., some polymers, bitumen, non-Newtonian fluids, but also tendons and ligaments.

Chapter 3

System Identification: Ordinary vs Fractional

3.1 Introduction

In this chapter we will discuss system identification as an approach for modeling phenomena. In particular, we will introduce the topic from a broad perspective and then we will go into the detail of the so-called *system identification loop* for ordinary, integer-order models, following [15]. Finally, we will look at the approximation of fractional systems describing Oustaloup's filter method [27], which will be the keystone of the final chapter.

3.1.1 Dynamical Systems and Models

We can think a *system* as a pair box-observer. External stimuli can act on the box, and in this case they are divided into control *inputs* and *disturbances* according to the influence the observer can exert on them; this interaction exterior-box produces signals, and those of interest for the observer are called *outputs*.

Once inputs and outputs are identified, the observer needs to know the relationship among the variables located inside the box and how they respond to both inputs and disturbances to produce the corresponding outputs. Indeed, it is assumed that the final goal of the observer is to *predict* system outputs with sufficient precision to make the whole process purposeful. We will call the relationship external stimuli-box's variables-outputs a *model* of the system.

Different levels of abstraction and formalism may be required when representing a system. Some of them (*mental models*) are easily understood without relying on any kind of mathematical formalization. Some other (like linear systems) are appropriately described by numerical tables and/or plots. We shall refer to them as *graphical models*. Sometimes instead the most proficient description comes in

terms of mathematical expressions like difference or differential equations. We shall call such models *mathematical* (*or analytical*) *models*. According to the degree of sophistication, these latter descriptions can be further specified; this is just a side effect of the fact that mathematical models are ubiquitous in all branches of modern sciences, both technical and nontechnical.

3.1.2 The Construction of a Model

Regardless of its kind, devising a model is an inductive process: the observer gather instances of outputs and use them to infer the whole law. The route that she follows to do so, however, depends on the characteristics of the system. Mental models can be constructed from direct experience, while graphical models are made up from suitable measurements. For what concerns mathematical models, instead, the approach is twofold. A possibility is to divide the system into subsystems that are well understood upon empiric considerations from various branches (physics, engineering, biology, etc.); the global model is then built gluing every contributions from the bottom up with mathematical laws. This first approach is called *first principle modeling* and in recent years it has been greatly boosted by computer science, so that a more precise term to connote a system description devised this way would be *software model*.

The alternative route to build a mathematical model is to record and analyze input and output signals from the system in order to infer a model from the data. This method is called *experimental identification* and it will be our focus in this chapter.

3.2 The System Identification Loop

The general procedure of system identification consists of the following stages.

1. Design the experiment. For dynamic systems, this typically involves collecting transient response data in the time domain or frequency response in the frequency domain.

2. Record the *data set*, which should be the most informative possible in order to represent the whole system with the most accuracy possible.

3. Choose a set of *candidate models*, whose elements are the different models that will be tested against the data set. Depending on the *a priori* knowledge the model set can be called in different ways: for example one could assume nothing on the physical internal structure of the system, so that all the inferred conclusions come from experimentally collected data only. Such

choice is called *black box model* and it is the one we will focus on throughout the thesis.

4. Choose the structure of the model, (e.g., whether a first-order or a second-order model) based on known characteristics of the system and experience joint with prior free identification.

5. Find model parameters choosing the ones that fit the experimental data the most using optimization techniques.

6. Validate the obtained model, keeping in mind that a model can never be accepted as a final and true description of the system, but, rather, it can at best be regarded as a description of some of its interesting properties that is good enough to be useful. To do so generally requires the method to pass certain tests, which are known as *model validation*. A typical example of validation is to identify a model on a *portion* of the initial data set, which is known as *identification set* or, especially in the contest of the sets of models called *neural networks*, as *training set*, and then make it perform against the rest of the data set, which is known accordingly as *validation set*.

7. If the model meet all requirements, use it for the desired purpose. Otherwise, revise modeling/identification strategy and repeat the above steps.

3.3 Open-Loop Identification in the Time Domain

Definition 3.1: *Formally, a single-input-single-output (SISO) black box model can be defined starting from a map* $\psi\colon \mathscr{I} \to \mathscr{O}$*, where* $(\mathscr{I}, \mathscr{O}) \subset \mathbb{R}^2$ *denote the measured input and output signals, respectively. Moreover, every signal can be seen as a function of time* t*, such that the relationship between inputs and outputs can be written as:*

$$z(t) = \psi(v(t)) + \mathfrak{R}, \tag{3.1}$$

where $z(t)$ *denotes the system output, and* $v(t)$ *denotes the system input, and* $\mathfrak{R}$ *denotes measurement noise, which usually is modeled using a random distribution.*

If we suppose that the sampling rate for the system is uniform $t_s = t_{k+1} - t_k \equiv$ constant, with k an integer index, then we refer to system's inputs as $u_k = v(kt_s)$ and to system's output as $y_k = z(kt_s) + \mathfrak{R}$, so that the data set will be:

$$Z_n = \{u_0, y_0, u_1, y_1, \ldots, u_N, y_N, t_s\}, \tag{3.2}$$

where $k = 0, 1, \ldots, N$. Since zero initial conditions are assumed, if $y_0 \neq 0$, the offset will be removed from each of the collected output samples by translating the

whole vector of the quantity y_0:

$$y_k \rightsquigarrow y_k - y_0, \quad k = 0, 1, \ldots, N. \tag{3.3}$$

3.3.1 Transfer Functions

Throughout the rest of the thesis we will focus *Linear Time Invariant* (LTI) systems. If we consider a SISO system with input $u(t)$ and output $y(t)$, it is said to be *linear* if its output response of a linear combination of input signals equals the same linear combination (i.e., with the same coefficients) of the output responses to the individual inputs. It is said to be *time invariant* if its output response to a certain input signal does not depend on absolute time. Finally, if the output at a certain time depends on the input up at that specific time only, the system is also said to be *causal*.

It is well known that a causal LTI system can be described by its *impulse response* $g(\tau)$ as follows:

$$y(t) = \int_0^{+\infty} g(\tau)u(t-\tau)\,\mathrm{d}\tau. \tag{3.4}$$

If we assume that the output $y(t)$ is sampled at the *sampling instants* $t_k = kT$, $k = 0, 1, \ldots$, where T is the *sampling interval*, and we also assume that the input signal $u(t)$ is kept constant between the sampling instants, i.e., $u(t) = u_k$ for $t \in [kT, (k+1)T)$, we can simplify equation (3.4) and obtain:

$$\begin{aligned} y(kT) &= \int_{\tau=0}^{+\infty} g(\tau)u(kT-\tau)\,\mathrm{d}\tau = \sum_{\ell=1}^{+\infty} \int_{\tau=(\ell-1)T}^{\ell T} g(\tau)u(kT-\tau)\,\mathrm{d}\tau \\ &= \sum_{\ell=1}^{+\infty} \left[\int_{\tau=(\ell-1)T}^{\ell T} g(\tau)\,\mathrm{d}\tau \right] u_{k-\ell} = \sum_{\ell=1}^{+\infty} g_T(\ell)u_{k-\ell}, \end{aligned} \tag{3.5}$$

where, following Ljung [15], we defined

$$g_T(\ell) = \int_{\tau=(\ell-1)T}^{\ell T} g(\tau)\,\mathrm{d}\tau. \tag{3.6}$$

The sequence $\{g_T\ell)\}_{\ell=1}^{+\infty}$ is called the *impulse response* on the system.

Moreover, if T is one time unit and (with slight abuse of notation) we indicate with t the sampling instants we have that becomes:

$$y(t) = \sum_{k=1}^{+\infty} g(k)u(t-k). \tag{3.7}$$

To streamline the notation a bit, let us introduce the *forward shift operator* and its inverse, the *backward shift operator*, again following [15]:

$$\mathrm{q}\, u(t) \coloneqq u(t+1) \quad \mathrm{q}^{-1}\, u(t) = u(t-1). \tag{3.8}$$

Reexamining equation (3.7) we get:

$$\begin{aligned} y(t) = \sum_{k=1}^{+\infty} g(k) u(t-k) = \sum_{k=1}^{+\infty} g(k) \big(\mathrm{q}^{-k}\, u(t) \big) \\ \Big[\sum_{k=1}^{+\infty} g(k)\, \mathrm{q}^{-k} \Big] u(t) = G_T(q) u(t), \end{aligned} \tag{3.9}$$

where we introduced the notation:

$$G_T(q) \coloneqq \sum_{k=1}^{\infty} g(k)\, \mathrm{q}^{-k}, \tag{3.10}$$

which we shall call the *transfer function* of the system (3.7).

Working with a continuous-time representation of (3.4), one fundamental device in system identification is the *Laplace transform* $\mathscr{L}$. Applying a reasoning similar to the one expressed above, we can apply the Laplace transform to both inputs and outputs, obtaining the relationship

$$y(t) = G_c(p) u(t), \tag{3.11}$$

where G_c is the Laplace transform of the impulse response function $g(\tau)$ and p is the differentiation operator. Two important things follow:

Remark 3.2.

- Equations (3.9) and (3.11) describe the output at all values for discrete-time and continuous-time systems, respectively. This means that identifying such systems can be reduced to identify their transfer functions.

- It is possible to go from continuous-time to discrete time representation in several ways. One possibility is to approximate the differentiation operator p by a difference approximation, so that we have the so-called Euler approximation

  $$G_T(q) \approx G_c\Big(\frac{q-1}{t} \Big). \tag{3.12}$$

  From now on we will refer to the continuous-time transfer function G_c as simply G.

3.3.2 Identifying Transfer Functions

Let us consider a system regardless of errors and disturbances as

$$y(t) = G(q)u(t). \tag{3.13}$$

The transfer function G is determined by a certain set of numerical values, or *coefficient*, whose knowledge it is often not possible *a priori*. This means that actually, when describing a system through a transfer function, one should keep in mind that it is fact a function of two variables: $G(q) = G(q, \theta)$, where θ is the vector of all coefficients. Therefore, θ specifies a *set of models* and to identify a system θ is to be determined. The identification method aims to find the θ which minimizes the squared Euclidean norm of output error. The corresponding problem is stated as:

$$\min_{\theta} e^2, \tag{3.14}$$

where $e = y_k - \hat{y}_k$ and $\hat{y}_k = \hat{\psi}(u_k, \theta)$ denotes the response of the estimated system model $\hat{\psi}$ under the input signal u_k, $k = 0, \ldots, N$.

In those cases where nonlinear least-squares estimation of model parameters θ is required, the following are the most employed algorithms:

1. *Trust Region Reflective*: especially useful for large-scale problems. Bounds are given for parameter sets as $\theta_b = \{\theta_{\min}, \theta_{\max}\}$ [32].

2. *Levenberg-Marquardt*: common in the context of black-box model identification.

3.3.3 The ODE model

One especially important example is given by a system described by an ordinary differential equation (ODE):

$$\sum_{k=0}^{n} a_k \frac{\mathrm{d}^k}{\mathrm{d}t^k} y(t) = \sum_{k=0}^{m} b_j \frac{\mathrm{d}^j}{\mathrm{d}t^j} u(t). \tag{3.15}$$

The parameters in this case are:

$$\theta = [a_0, a_1, \ldots, a_n, b_0, b_1, \ldots, b_m]$$

Applying the Laplace transform to both sides we have:

$$y(t) = \frac{Y(s)}{U(s)} u(t) \coloneqq \frac{\sum_{j=0}^{m} b_j s^j}{\sum_{k=0}^{n} a_k s^k} = G(s)u(t),$$

that is, we have expressed the model via a rational complex transfer function, whose numerator and denominator are called *zero polynomial* and *pole polynomial*, respectively.

The identification problem is then to estimate a set of parameters θ of the model in (3.15), where θ is formed by:

$$\begin{aligned} a_p &= [a_n, a_{n-1}, \cdots, a_0], \quad \alpha_p = [n, n-1, \cdots, 0], \\ b_z &= [b_m, b_{m-1}, \cdots, b_0], \quad \beta_z = [m, m-1, \cdots, 0], \end{aligned} \tag{3.16}$$

a_p and b_z denote pole and zero polynomial differential operator coefficients, α_p and β_z denote the corresponding exponents (orders of differentiation), respectively; if $\alpha_0 = \beta_0 = 0$, then the system static gain is identified as $K = b_0/a_0$; and for θ there exists 9 possible parameter sets depending on the chosen identification method:

- Full model parameter identification, $\theta = [a_p, \alpha_p, b_z, \beta_z]$;
- Fixed orders, unknown coefficients, $\theta = [a_p, b_z]$;
- Fixed coefficients, unknown orders, $\theta = [\alpha_p, \beta_z]$.

With pole polynomial fixed:

- Full zero polynomial identification, $\theta = [b_z, \beta_z]$;
- Fixed orders, unknown zero polynomial coefficients, $\theta = b_z$;
- Fixed coefficients, unknown zero polynomial orders, $\theta = \beta_z$.

With zero polynomial fixed:

- Full pole polynomial identification, $\theta = [a_p, \alpha_p]$;
- Fixed orders, unknown pole polynomial coefficients, $\theta = a_p$;
- Fixed coefficients, unknown pole polynomial orders, $\theta = \alpha_p$.

Generally speaking, if there is a considerable amount of parameters to identify, the identification process may be slow; fixing or bounding parameters around particular values (maybe supported by engineering or physical insight) can alleviate the workload significantly.

3.3.4 Residual Analysis

The following discussion deals with the assessment of the quality of the identified model.

Denote by y_r the experimental plant output, and by y_m the identified model output,so that they are two vectors of size $N \times 1$. In the following, we address the problem of statistical analysis of modeling residuals. Analysis is partially due to Ljung [15]. Residuals are given by a vector containing the model output error

$$\varepsilon = y_r - y_m. \tag{3.17}$$

The percentage fit may be expressed as

$$\text{Fit} = \Big(1 - \frac{\|\varepsilon\|}{\|y_r - \bar{y}_r\|}\Big) \times 100\%, \tag{3.18}$$

where $\|\cdot\|$ is the Euclidean norm, and $\bar{y}_r$ denotes the mean value of y_r.

Basic statistical data may be computed first, such as maximum absolute error

$$\varepsilon_{\max} = \max_k |\varepsilon(k)|, \tag{3.19}$$

and mean squared error

$$\varepsilon_{\text{MSE}} = \frac{1}{N}\sum_{k=1}^{N} \varepsilon_k^2 = \frac{\|\varepsilon\|_2^2}{N}. \tag{3.20}$$

Assuming normal distribution of residuals the confidence band $\hat{\eta}$ is then approximated for a confidence percentage $p_{\text{conf}} \in (0, 1]$ around zero mean as an interval

$$\hat{\eta} = \Big[\frac{0 - \phi^{-1}(c_p)}{\sqrt{N}}, \frac{0 + \phi^{-1}(c_p)}{\sqrt{N}}\Big], \tag{3.21}$$

where $c_p = 1 - 0.5(1 - p_{\text{conf}})$ and $\phi^{-1}(x) = \sqrt{2}\,\text{erf}^{-1}(2x - 1)$ is the quantile function.

Based on these considerations, we may draw two conclusions: maximum absolute error $\varepsilon_{\max}$ shows the maximum deviation from the expected behavior of the model over the examined time interval, however, it may be misleading when adjusting for noise or disturbance; mean squared error ε_{MSE} may serve as a general measure of model quality. The lower it is, the more likely the model represents an adequate description of the studied process.

3.4 Fractional-Order Models

A fractional-order continuous-time dynamic system can be expressed by a fractional differential equation of the following form:

$$a_n \mathscr{D}^{\alpha_n} y(t) + a_{n-1} \mathscr{D}^{\alpha_{n-1}} y(t) + \cdots + a_0 \mathscr{D}^{\alpha_0} y(t) = \\ = b_m \mathscr{D}^{\beta_m} u(t) + b_{m-1} \mathscr{D}^{\beta_{m-1}} u(t) + \cdots + b_0 \mathscr{D}^{\beta_0} u(t), \quad (3.22)$$

where y_i, u_j are functions of time, $(a_i, b_j) \in \mathbb{R}^2$ and $(\alpha_i, \beta_j) \in \mathbb{R}_+^2$. The system will be called of *commensurate-order* if in (3.22) all the orders of derivation are integer multiples of a base order α such that $\alpha_k, \beta_k = k\alpha$, $\alpha \in \mathbb{R}^+$. The system can then be expressed as

$$\sum_{k=0}^{n} a_k \mathscr{D}^{k\alpha} y(t) = \sum_{k=0}^{m} b_k \mathscr{D}^{k\alpha} u(t). \quad (3.23)$$

If in (3.23) is $\alpha = 1/q$, $q \in \mathbb{Z}^+$, the system will be said of *rational* order. This allows for a neat classification of linear time-invariant systems based on their order kind:

$$\text{LTI Systems} \begin{cases} \text{Non-integer} \begin{cases} \text{Commensurate} \begin{cases} \text{Rational} \\ \text{Irrational} \end{cases} \\ \text{Non-commensurate} \end{cases} \\ \text{Integer} \end{cases}$$

Applying the Laplace transform to (3.22) with zero initial conditions, the input-output representation of a (continuous) fractional-order system can be obtained in the form of a transfer function of the form:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots a_0 s^{\alpha_0}}. \quad (3.24)$$

We shall call the number of fractional poles in (3.24) the *pseudo-order* of the system. In the case of a system with commensurate order α, we can take $\sigma = s^\alpha$ and consider the pseudo-rational transfer function

$$H(\sigma) = \frac{\displaystyle\sum_{k=0}^{m} b_k \sigma^k}{\displaystyle\sum_{k=0}^{n} a_k \sigma^k}. \quad (3.25)$$

3.5 Approximation of Fractional-Order Operators

3.5.1 Grünwald-Letnikov Approximation

Numerical computations of fractional-order derivatives can be obtained by means of an approximation of Grünwald-Letnikov definition [13, 32]

$$ {}_{t_0}\mathscr{D}_t^{\alpha} f(t)\Big|_{t=kh} = \frac{1}{h^{\alpha}} \sum_{k=0}^{[\frac{t-a}{h}]} w_j^{(\alpha)} f(t-jh), \tag{3.26} $$

where h is the computation step-size and

$$ w_j^{(\alpha)} = (-1)^j \binom{\alpha}{j}. $$

Since the approximation in (3.26) is defined via a summation, we note that the larger t becomes, the more terms we need to add, which can easily lead to memory problems. On the other hand, the Grünwald-Letnikov definition sees its coefficients corresponding to values of f near the initial point to bring little contribution for large t. This fact is summarized in the *short memory principle*: is it possible to approximate the derivative by using only "local" information. More formally, this implies that to compute $\mathscr{D}^{\alpha} f(t)$ with initial point t_0, it actually suffices to operate in $[t-L, t]$, with L being the maximum length of memory acting as a (moving) lower limit. Thus we have

$$ \mathscr{D}^{\alpha} f(t) \approx {}_{t-L}\mathscr{D}^{\alpha} f(t), \qquad t > L, \tag{3.27} $$

and L/h represents the upper bound on the number of terms in the summation. Of course, the memory length L influence the quality of the approximation.

For what concerns the calculation of the coefficients in the case of a fixed α, they can be evaluated recursively from

$$ w_0^{(\alpha)} = 1, \quad w_j^{(\alpha)} = \Big(1 - \frac{\alpha+1}{j}\Big) w_{j-1}^{(\alpha)}, \qquad j = 1, 2, \ldots \tag{3.28} $$

while for a varying α the most used techniques involves the use of the fast Fourier transform (FFT).

To obtain a numerical solution for the equation in (3.22), the signal $\hat{u}(t)$ should be obtained first, using the algorithm in (3.26), where

$$ \hat{u}(t) = b_m \, \mathscr{D}^{\beta_m} u(t) + b_{m-1} \, \mathscr{D}^{\beta_{m-1}} u(t) + \cdots + b_0 \, \mathscr{D}^{\beta_0} u(t). \tag{3.29} $$

The time response of the system can then be obtained using the following equation:

$$y(t) = \Big(\sum_{i=0}^{n} \frac{a_i}{h^{\alpha_i}}\Big)^{-1}\Big[\hat{u}(t) - \sum_{i=0}^{n} \frac{a_i}{h^{\alpha_i}} \sum_{j=1}^{[\frac{t-a}{h}]} w_j^{(\alpha)} y(t-jh)\Big]. \tag{3.30}$$

The presented method is a fixed step method. The accuracy of simulation therefore may depend on the step size.

3.5.2 Oustaloup's Filter Approximation

Due to practical limitations, it is often necessary to approximate a fractional-order operator with a more manageable one of often much higher order, though. Such replacement is the high-order *rational* approximation of the fractional-order operator, and the method to derive it that we will adopt throughout the thesis is due to Oustaloup [27].

Oustaloup's recursive filter gives a very good approximation of fractional operators in a specified frequency range. It is a well-established method and is often used for practical implementation of fractional-order systems and controllers [28, 19, 32]. It is summarized next.

In order to approximate a fractional differentiator of order α or a fractional integrator of order $(-\alpha)$ by a conventional transfer function one may compute the zeros and poles of the latter using the following equations:

$$s^\alpha \approx K \prod_{k=1}^{N} \frac{s+\omega'_k}{s+\omega_k}, \tag{3.31}$$

where

$$\omega'_k = \omega_b \cdot \omega_u^{(2k-1-\alpha)/N}, \tag{3.32}$$

$$\omega_k = \omega_b \cdot \omega_u^{(2k-1+\alpha)/N}, \tag{3.33}$$

$$K = \omega_h^\alpha, \quad \omega_u = \sqrt{\omega_h/\omega_b}, \tag{3.34}$$

with ω_u being the unit gain frequency and the central frequency of a band of frequencies geometrically distributed around it. That is, $\omega_u = \sqrt{\omega_h \omega_b}$, ω_h, ω_b are the high and low transitional frequencies.

One strong advantage of Oustaloup's filter is that the amount of necessary computations grows linearly with the order of approximation N, indeed, only a limited history of the process is considered. The bigger the N the better the approximation of the differentiator s^α in its frequency band.

Besides, we shall remark that it suffices to consider $\alpha \in (0,1)$, since we can always write:

$$s^\alpha = s^n s^\gamma, \tag{3.35}$$

where $n = \alpha - \gamma$ denotes the integer part of α, so that $\gamma \in (0, 1)$, since fractional and integer-order derivatives commute for fractional orders $\alpha \geq 1$, and s^γ is obtained by the Oustaloup approximation by using (3.31).

Thus, every operator in (3.24) may be approximated using (3.35) and replaced by the obtained approximation, yielding as a final result a *conventional integer-order* transfer function. This suggests that, after all, identifying in the time domain a system with a fractional-order model would lead to substantially the same results of those provided by a classical integer-order model of possibly (very) high order, at the price of a significantly larger amount of computations. Before study the correctness of this claim in the next chapter, we conclude the current one with a remark.

Remark 3.3. Everything that has been said for the ODE model (3.15) still holds for a fractional-order one, with the modification that in this latter case the order of derivation are generic nonnegative real numbers. Moreover, if the system under investigation is of commensurate order $0 < \gamma < 2$, an initial guess model should be generated

$$\alpha_p = [\gamma n \ \gamma(n-1) \ \cdots \ 0] \tag{3.36}$$

and

$$\beta_z = [\gamma m \ \gamma(m-1) \ \cdots \ 0], \tag{3.37}$$

such that

$$\big\{ (n, m) \in \mathbb{Z}_+^2 \colon n \geq m \big\}, \tag{3.38}$$

where n determines the pseudo-order of the system; the orders should be fixed and only model coefficients estimated. Of course, a commensurate initial model may also be used for identification of all parameters.

Chapter 4

Fractional-Order Modeling and Control Toolboxes

The recent increase of interest in fractional calculus by applied mathematics and science fields led to an increase in the demand of numerical tools for the computation of fractional integration/differentiation, and the simulation of fractional order-systems. Accordingly, much software programs were produced and distributed, so that today there is not an established standard "go-to" software, each of the currently used tools has its own strengths and weaknesses.

To perform the calculations and simulations needed to complete our numerical results, we chose to use the MATLAB toolbox FOMCON, developed by Aleksej Tepljakov [33], which is based on three other popular toolboxes.

In this chapter we give an outline of each of them and we also provide a table of comparison as a reference. Next we describe the structure of the FOMCON more in detail, concluding with an hands-on example of its usage. This summary is largely based on a survey paper by Zhuo Li *et al.* [14], so something could have changed.

4.1 FOTF

FOTF (Fractional Order Transfer Function) is a control toolbox for fractional order systems developed by Xue et al. [8] which extends many MATLAB built-in functions. The reference text for the FOTF toolbox is [37], where the author explains thoroughly all its commands and applications.

The toolbox employs a programming technique called overload to enable the related methods of the MATLAB built-in functions to deal with fractional-order models. For instance, the transfer function objects generated from FOTF can interact with those generated from the MATLAB transfer function class. However, overload presents also negative sides, in fact functions such as `impulse()`, `step()`,

etc, lost the plotting functionality. There are easy workarounds, though, usually defining a time vector as second input suffices.

FOTF toolbox also supports time delay in the transfer function, enabled by calling for example `fotf(a,na,b,nb,delay)`. It does not directly support transfer function matrix, hence, multiple-input-multiple-output (MIMO) systems cannot be simulated directly. Yet, it does provides Simulink block encapsulation of function `fotf()`, so multiple input/output relationship can be set up by adding loop interactions in Simulink block diagrams manually.

As reported by Zhuo Li *et al.* [14] FOTF sampling time can have relatively great impact on the accuracy.

FOTF approximate fractional differential operators by means of a discretization of the Grünwald-Letnikov definition of noninteger derivative, but other approximation methods are possible [8].

It seems to lack a system identification module.

4.2 Ninteger

Ninteger (Non-integer) is a toolbox for MATLAB intended to help developing non-integer order controllers for single-input, single-output plants, and assess their performance. It was originally developed by Duarte Valério and José Sá da Costa [34] to face the lack of availability of toolboxes for fractional calculus and control.

Ninteger provides Simulink block encapsulation of the involved functions, such as `nid` and `nipid` blocks. Moreover, it offers a user-friendly GUI for fractional order PID controller design. It uses integer-order approximations of fractional-order transfer function, mainly based on Oustaloup's filter; more generally the whole toolbox has been inspired by the original CRONE one, from which Ninteger imported several methods.

There are compatibility issues with Ninteger toolbox in MATLAB version 2013a or later: it has conflicts with some built-in functions due to the overload editing of the Matlab built-in function `isinteger()`.

Its last update dates back to March 2008.

4.3 ooCRONE

The CRONE (*Commande Robuste d'Ordre Non Entier*, robust command of non-integer order) Toolbox, developed since the nineties by the CRONE team [28], is a MATLAB and Simulink toolbox dedicated to applications of non integer derivatives in engineering and science. It started as a script-based toolbox, while later evolved into the current object-oriented version [19].

CRONE is the first developed toolbox for the management of fractional operators. Several other toolboxes are inspired by CRONE, e.g., Ninteger (as said above) and FOMCON. Moreover, many approximations techniques proposed by the CRONE team are considered as foundational in the literature (see e.g., [21, 32]). For instance Oustaloup's (leader of the CRONE team) method of approximation of transfer functions was one of the cornerstones of the original CRONE toolbox.

It has a fairly enhanced support for MIMO fractional transfer functions. For example, executing `sysMIMO=[sys,sys;sys2,sys2]` generates a two-input-two-output transfer function matrix. Yet CRONE does not allow incorporation of time delay into the generated fractional-order transfer function, neither by manual multiplication.

CRONE is a toolbox much more powerful than merely simulating fractional order systems. Besides this basic functionality, it is also capable of fractional order system identification and robust control analysis and design. However, user may experience difficulties about the availability of the toolbox, since it is not hosted on MathWorks, but the license of use and the product itself have to be distributed by the CRONE team itself. It also does not have a GUI, for the time being.

4.4 FOMCON

The FOMCON toolbox for MATLAB is a fractional-order calculus based toolbox for system modeling and control design. Aleksei Tepljiakov [33, 32] developed it upon the core of FOTF. Consequently, the main object of analysis in FOMCON is a fractional-order transfer function of the form:

$$G(s) = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots a_0 s^{\alpha_0}},$$

and its main aim is to extend conventional control schemes, like PID controllers, with concepts of fractional calculus and to provide tools to implement fractional-order systems and controllers.

FOMCON is also related to other existing fractional-order calculus oriented MATLAB toolboxes, such as CRONE [28] and Ninteger [34] (indeed, the author refers to its work as the "missing link" between them) through either system model conversion features or shared code, and this relation is depicted in Fig. 4.1.

FOMCON was initially developed in order to facilitate the research of fractional-order systems. This involved writing convenience functions, e.g., the polynomial string parser and building a GUI. However, a full suite of tools was also desired due to certain limitations in existing toolboxes. Once the core of basic function of FOMCON was established, it was then extended with advanced features, such

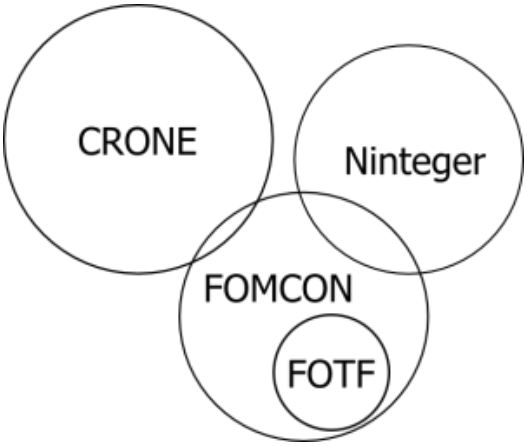

Figure 4.1: Fractional-calculus based toolboxes relations

as fractional-order system identification and FOPID controller design. This makes the toolbox suitable for both beginners and more demanding, experienced users.

The toolbox also supports sophisticated modeling approach and real-time control application through Simulink blockset. A plus is also FOMCON's (possibly partial) portability among different platforms such as Scilab and Octave, made possible by the availability of the source code, hosted e.g. in the official website [1].

4.4.1 Structure of the Toolbox

The toolbox has an interconnected modular structure with most features supported by graphical user interfaces. It currently consists of:

- Main module (core-fractional system analysis);
- Identification module (system identification in both time and frequency domains);
- Control module (FOPID controller design, tuning and optimization tools, as well as some additional features);
- Implementation module (continuous and discrete time approximations, implementation of corresponding analog and digital filters).

4.4.2 Dependencies

FOMCON employs two standard MATLAB toolboxes, i.e., Control System toolbox, required for most features, and Optimization toolbox, required for time domain identification and PID tuning.

To further underline the connection lying between FOMCON and CRONE, we remark that is also possible to export fractional-order systems to CRONE format, once this latter toolbox has been installed on the main operative system.

4.4.3 Identification Module

Since the numerical results that we will present in the next chapter deal with system identification, we devote some space here to illustrate FOMCON's identification module.

It provides the following main features:

- Time domain identification:
  - Commensurate and non-commensurate order system identification;
  - Parametric identification;
  - Approximation of fractional systems by conventional process models.
- Frequency domain identification:
  - Commensurate transfer function identification;
  - Best fit algorithm for choosing an optimal commensurate order and pseudo-orders of the fractional transfer function.

In addition, coefficients and orders of the obtained model can be manipulated, (e.g., truncated, rounded, normalized etc), via ad-hoc functions; likewise FOMCON also has functions for validating the models and carry out residual analysis.

4.4.4 Example

Here we illustrate the use of the `fid` function of FOMCON toolbox using one of the examples provided by Tepljakov himself in [32]. Functions explanations can be found at [1].

The task is to identify a system from an experimental signal to verify the identification algorithm.

An excitation signal (a PRBS7 sequence with amplitude of 1 applied for 30 s and immediately followed by a sine wave with an amplitude of 1 centered around zero with frequency of 20 Hz lasting 30 s) is applied to the input of the system and output samples are collected with a sample rate of 200 Hz.

The system under study is described by

$$G(s) = \frac{-1.3333s^{0.63} + 2.6667}{1.3333s^{3.501} + 2.5333s^{2.42} + 1.7333s^{1.798} + 1.6667s^{1.31} + 1}. \tag{4.1}$$

Assuming the variables u, y, and t hold the experimental input, output, and sample time vector, respectively, and denoting also with c and α the generic model coefficient and the generic model order, respectively, the initial model structure and parameters are chosen as

$$G_i(s) = \frac{s+1}{s^3 + s^{2.5} + s^{1.5} + s + 1}, \qquad c \in [-100, 1000] \qquad \alpha \in [10^{-9}, 5]. \tag{4.2}$$

The model is simulated by an Oustaloup's recursive filter with $\omega \in [0.0001, 10000]$, $N = 5$, and the Trust-Region-Reflective optimization algorithm is used to optimize the fit.

We now go through the code line by line:

```
% Setup: Create the fractional identification dataset
id1 = fidata (y, u, t);
```

Function `fidata()` returns an object containing correctly sized identification parameters and additionally sampling interval based on the provided time vector. It is called as `fidata(y,u,t)`, where the arguments stand for, in order, observed system output, observed system input, observations time vector.

```
% Initial model structure and parameters
g_i = fotf ('s+1', 's^3 + s^2.5 + s^1.5 + s + 1');
```

Function `fotf()` creates a new fractional-orderr transfer function object. It can be called in multiple ways; here it has been chosen the fashion `fotf('s')`, where `'s'` stands for a pair of symbolic expression in the variable s that represents the pair pole polynomial-zero polynomial of the transfer function. In all other cases, `fotf()` must be given the information about the transfer function providing coefficients and orders and delay, using coupled vectors or polynomial strings. The function can also be called with argument another `fotf` object.

```
% Use Oustaloup approximation for system simulation
fsp = fsparam(g_i, 'oust', [0.0001 10000], 5);
```

Function `fsparam()` creates a new simulation parameters structure used for FOTF system simulation using one of the following approximations: Oustaloup filter, refined Oustaloup filter. It is called as `fsparam(plant, approx, w, N)`, where `plant` is a `fotf` object or a symbolic expression in the variable s which evaluate to a valid transfer function; `approx` is the approximation method employed, can be either `'gl'`, `'oust'` or `'ref'` that indicate Grünwwald-Letnikov, Oustaloup's filter and refined Oustaloup's filter, respectively (default: `'oust'`); `w` is the approximation frequency range in form `[wb; wh]` (default: `[1e-3;1e3]`); `N` is approximation order (default `5`);

```
% Model is assumed to have a unitary static gain and no delay
gp = {1, []};
% Optimization algorithm: Trust-region-reflective
op.IdentificationAlgorithm = 'trr';
lim = {[-100; 1000], [1e-9 5]};     % Bounds
% Run the identification: G_id1 is the identified model
[~,~,~,~,~,G_id1] = fid(fsp, gp, id1, [], [], [], lim, op);
```

Function `fid()` identifies SISO systems transfer functions of fractional order. It is called as

```
[a,na,b,nb,l,gid] = fid(fsim|g,gparam,idd,...
                        npoints,type,fixpoly,limits,op)}
```

The first argument can be either a `fsparam` structure or the initial `fotf` object. `gparam` is a cell array with explicit model parameters: `{K, L}`, where `K` is the static gain[1] and `L` is the time delay in seconds. `idd` is a `fidata` structure with the collected system samples.

All the remaining arguments are optional: `npoints` stands for the number of points to use for identification (default: `0` or, equivalently, `[]`); `type` stands for the type of identification, to be either `'n'`, i.e., free identification, `'c'`, i.e., coefficients will be kept fixed, or `'e'`, i.e., exponents will be kept fixed (default: `'n'`). `fixpoly` allows to fix one polynomial or both, it is in form of a two-valued vector `[bfix,afix]`, nonzero values will fix corresponding polynomials (default: `[0; 0]`). `limits` is a cell array of the form `[CMIN;CMAX],[EMIN;EMAX]`, containing polynomial coefficients and exponents (default: `[]`). Finally `op` is a string indicating the optimization algorithm employed: `'trr'` sets Trusted-Region-Reflective, while `'lm'` set Levenberg-Marquadt.

For invoking the Levenberg-Marquardt algorithm the following commands may be used[2]

```
% Optimization algorithm: Levenberg-Marquardt
op.IdentificationAlgorithm = 'lm'; op.Lambda = 100;
% Run the identification : G_id2 is the identified model
[~,~,~,~,~,G_id2] = fid(fsp, gp, id1, [], [], [], [], op);
```

A summary of the achieved results is provided in Table 4.1. The following models are obtained with the powers truncated:

$$G_{id1}(s) = \frac{-1.281s^{0.656} + 2.657}{1.396s^{3.495} + 2.145s^{2.471} + 2.736s^{1.817} + 1.199s^{1.176} + 1} \tag{4.3}$$

[1] Note that static gain will be identified only in case of free identification, i.e., all parameters of both polynomials are identified.

[2] Note that `'lm'` algorithm does not handle bound constraints, so the limits of search will be discarded with this option.

Algorithm	%Fit	ε_{MSE}	NoIter	FunEval	τ (min)
TRR	99.98	1.50×10^{-8}	84	1020	02 : 26
LM	99.07	4.24×10^{-5}	97	1227	02 : 50

Table 4.1: Identification of a complex fractional system: results for different estimation algorithms

and

$$G_{id2}(s) = \frac{0.014s^{4.617} + 2.627}{0.899s^{4.922} + 5.003s^{3.409} + 6.519s^{2.059} + 1.71s^{0.962} + 1} \tag{4.4}$$

Clearly, using the Trust-Region-Reflective estimation algorithm leads to a more accurate result in this particular case.

4.5 Summary

All the toolboxes described in this chapter are known and widespread in the fractional calculus community. We opted for the FOMCON for its usability (although we did not make use of its capable GUI) and updated state. The reliability of its built-in techniques derives from established ideas also implemented in the other toolboxes, so it was not necessary to have a trade-off between performance and ease. It was also essential that the toolbox had an identification module, of course.

A brief dashboard with the major characteristics of each toolbox follows.

Toolbox	Identification	Control	GUI	TF Approximation	MIMO	Maintenance
FOTF	✗	✓	✗	G-L	✓	07/2017
Ninteger	✓	✓	✓	Oustaloup	✗	03/2008
ooCRONE	✓	✓	✗	Oustaloup	✓	$\geq$ 01/2010
FOMCON	✓	✓	✓	Oustaloup	✓	04/2018

Table 4.2: Summary of toolboxes characteristics: the listed methods employed to approximate transfer functions (TF) are the main ones used in each toolbox; as said above they are not the only ones. For maintenance, the last known update time was used as reference; concerning ooCRONE it is not clear when it was updated the last time, so the date of the license was provided as a (probably very) lower bound.

Chapter 5

Numerical Results

In this chapter we will present some numerical results on three cases of system identification, the first two coming from experimental data and the last one simulated at random. The goal is to investigate whether fractional-orders models, being more flexible than their ordinary integer-order counterparts, are actually more effective in approximating a given nonlinear system; and if that is the case, whether the computational cost to generate them is worth it in real-life scenarios. As we noted in the previous chapter, we know that these systems can be completely described by a rational transfer function G. Thus, our purpose is to identify G as an ordinary, integer-order transfer function and obtain an approximation of the system. Then, we will employ MATLAB FOMCON toolbox by Tepljakov [32] to identify the systems as fractional-order models, using G as starting point. In both cases we simulate the outputs for identification data and validation data and we parallel them to the experimental data, estimating the error committed as the squared norm of their difference.
Finally we compare the performance of the fractional-order model against the one of the integer-order model, with particular focus on validation: the concern is that the flexibility of fractional models can cause them to over-fit the identification data, weakening their performance against validation sets.

5.1 Furnace

We start with experimental data collected from an industrial furnace. The system is considered to be well approximated by an LTI model, with inputs the voltages applied to furnace's heating system and outputs its temperatures.

5.1.1 Data Set

The data set comes in form of a table whose columns are ordered as:

1. Time instants;
2. Inputs u (voltages);
3. Outputs y (temperatures).

We chose to divide the data set with an $80:20$ split between identification set and validation set, as can be seen in Figure 5.1.

```
portion_id = round(0.8*num_data);
data_id = data(1:portion_id,:);
data_val = data(portion_id+1:num_data,:);

% Identification data
time_id = data_id(:,1);
inputs_id = data_id(:,2);
outputs_id = data_id(:,3);

% Validation data
time_val = data_val(:,1);
inputs_val = data_val(:,2);
outputs_val = data_val(:,3);
```

5.1.2 Integer-Order Identification

Before starting the identification process, it is convenient to interpolate the time instants and remove possible offsets, then simulate the collected data to assure them to be evenly sampled and with zero initial condition. The same holds for validation data, of course.

```
Ts = 12

% Identification data set
time_id_int = [0:Ts:time_id(end)-time_id(1)] + time_id(1); $ time
u_id = interp1(time_id,inputs_id,time_id_int); % inputs
y_id = interp1(time_id,outputs_id,time_id_int); % outputs

% Offset removal
u_id = u_id - inputs_id(1);
y_id = y_id - outputs_id(1);

% Validation data set
```

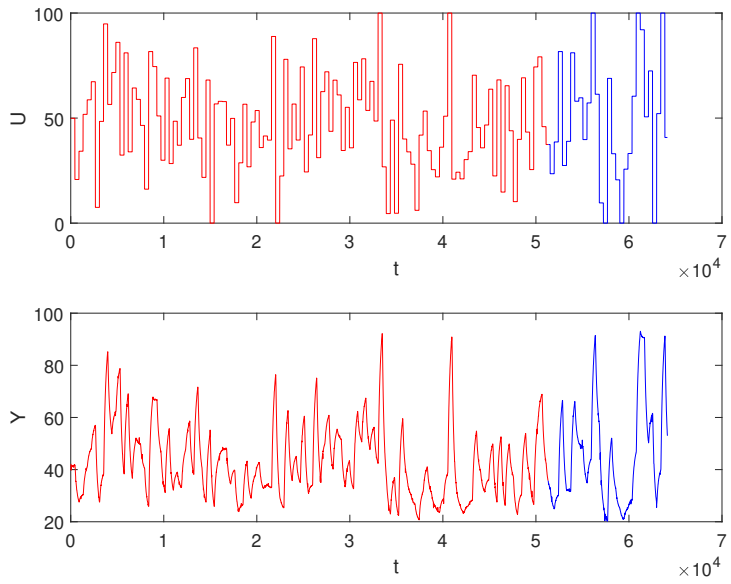

Figure 5.1: Furnace: data set. Identification data are plotted in red, validation data in blue.

```
time_val_int = [0:Ts:time_val(end)-time_val(1)] + time_val(1); % time
u_val = interp1(time_val,inputs_val,time_val_int); % inputs
y_val = interp1(time_val,outputs_val,time_val_int); % outputs

% Offset removal
u_val = u_val - inputs_id(1);
y_val = y_val - outputs_id(1);
```

In this and every other case in this chapter, the parametric identification develops with fixed orders, so that we only estimate the unknown coefficients; in particular, here we chose the numerator of G to be of order 1 and the denominator to be of order 2 with a time step $T_s = 12$.
The integer-order transfer function is provided by the MATLAB function `tfest`:

```
data_id = iddata(y_id',u_id',Ts);
sys = tfest(data_id,2,1);
```

and for the furnace system it returns:

$$G_I(s) = \frac{0.003207\, s - 2.166 \times 10^{-8}}{s^2 + 0.003965\, s + 8.054 \times 10^{-8}} \tag{5.1}$$

Its associated normalized root mean squared error (NRMSE) percentage of the fit of the estimation data is of 70.91%, with an overall mean squared error (MSE) of 14.79. Moreover, the norm of the difference between approximated and experimental data is 302.1166. In conclusion, not extraordinary, but sufficient for our purposes.

5.1.3 Fractional-Order Identification

The next phase is fractional-order identification. We employ the MATLAB FOMCON toolbox by Tepljakov [32]. The process starts with an initial transfer function: we choose G_I we found earlier in (5.1) to be the input of the FOMCON function `fotf`.

```
frac_id = fidata(y_id',u_id,time_id_int);
G_starting = fotf(G_ord)
```

Then we use the FOMCON function `fsparam` to determine all the parameters: in fact, when performing fractional-order identification we do not fix the orders, in contrast with our choices about integer-order identification. Here we leave fairly wide freedom of search, setting parameters bounds as low as -10^2 and as high 10^3, and setting the order of derivation to be in the range $[10^{-9}, 10]$. Concerning the frequency range, instead, we allow searching in the range of $[10^{-4}, 10^3]$ radiants per second. As remarked in the preceding chapter, fractional-order transfer functions need to be approximated to be used. We choose therefore to approximate them via Oustaloup's filter, described earlier.

```
frac_param = fsparam(G_starting, 'oust', [0.0001 1000], 10);
```

System identification is an optimization process; we thus need a (nonlinear) optimization algorithm to help us realize it. In the case of the industrial furnace we choose the Trusted Region Reflective. We also assume that the model has static gain.

```
gp = {1, []}; % static gain
op.IdentificationAlgorithm = 'trr'; % Trusted Region Reflective
lim = {[-100 ; 1000], [1e-9  10]}; % Search bounds
[a, na, b, nb, l, G_frac] = fid(frac_param, gp, frac_id,...
[],[],[], lim, op);
```

The fractional-order transfer function we obtained is:

$$G_F(s) = \frac{-5.6539\, s^{0.91322} + 0.86611}{24.911\, s^{2.8497} + 115.81\, s^{0.85359} + 1} \tag{5.2}$$

Identification error			Validation error		
Integer	Fractional	Relative %	Integer	Fractional	Relative %
302.1166	315.4941	4.24%	235.4625	251.9559	6.55%

Table 5.1: Furnace: approximation errors. Computed as $\|y - \tilde{y}\|$, where y are the experimental data, $\tilde{y}$ are the approximated data and $\|\cdot\|$ is the Euclidean norm. The relative percentage express the difference of the errors committed by the two models in proportion of the highest between them.

5.1.4 Validation

Both G_I and G_F are employed to simulate data to parallel against the validation set; the results are discussed next.

5.1.5 Comparison

Fractionally approximated data actually perform slightly worse than their ordinary integer-order counterpart, in both identification and validation set. Table 5.1 reports the errors committed, expressed as the norm of the difference between approximated and experimental data.
What really is remarkable is that the two approximations differ just a little, in the end the returned model seem to be the same, practically speaking. Figures 5.2–5.7 show the comparison between the two models, with respect to experimental data of both identification and validation sets, and between each other.

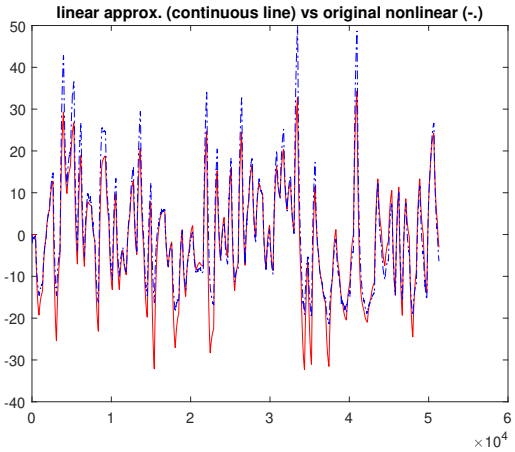

Figure 5.2: Furnace: integer-order identification. Approximated data are plotted in red, experimental data in blue (dashed).

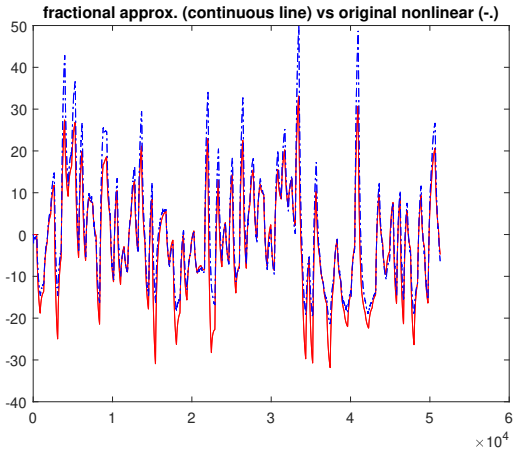

Figure 5.3: Furnace: fractional-order identification. Approximated data are plotted in red, experimental data in blue (dashed).

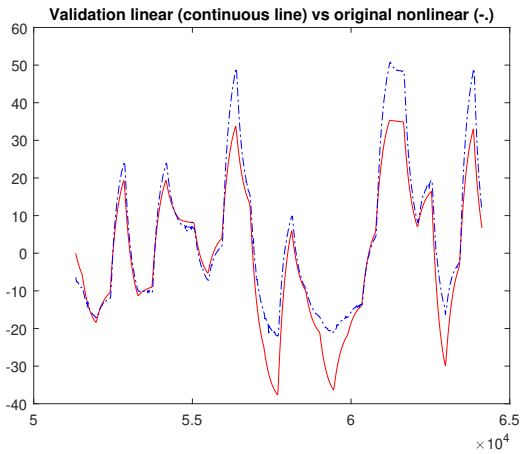

Figure 5.4: Furnace: integer-order validation. Approximated data are plotted in red, experimental data in blue (dashed).

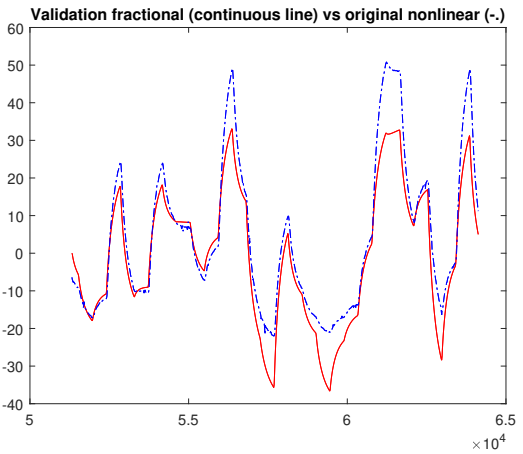

Figure 5.5: Furnace: fractional-order validation. Approximated data are plotted in red, experimental data in blue (dashed).

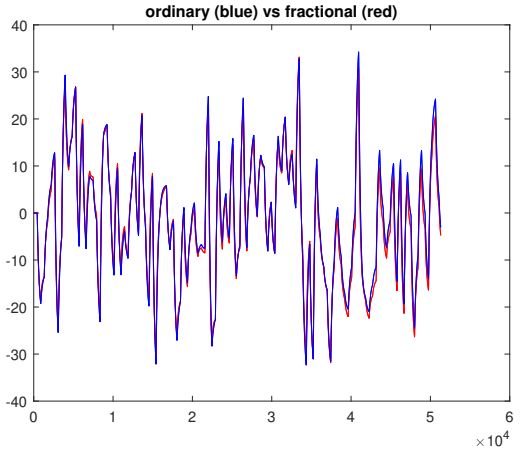

Figure 5.6: Furnace: identification comparison. Fractional approximation data are plotted in red, integer approximation data in blue.

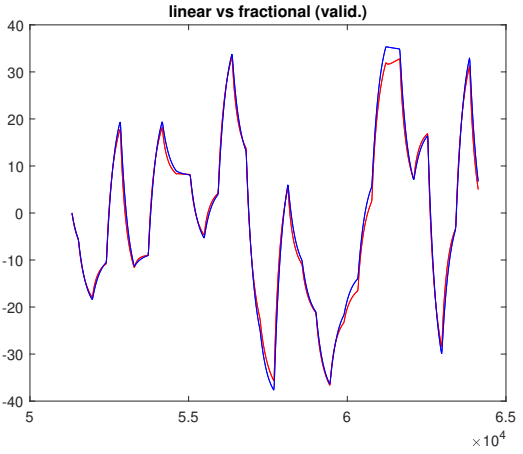

Figure 5.7: Furnace: validation comparison. Fractional approximation data are plotted in red, integer approximation data in blue.

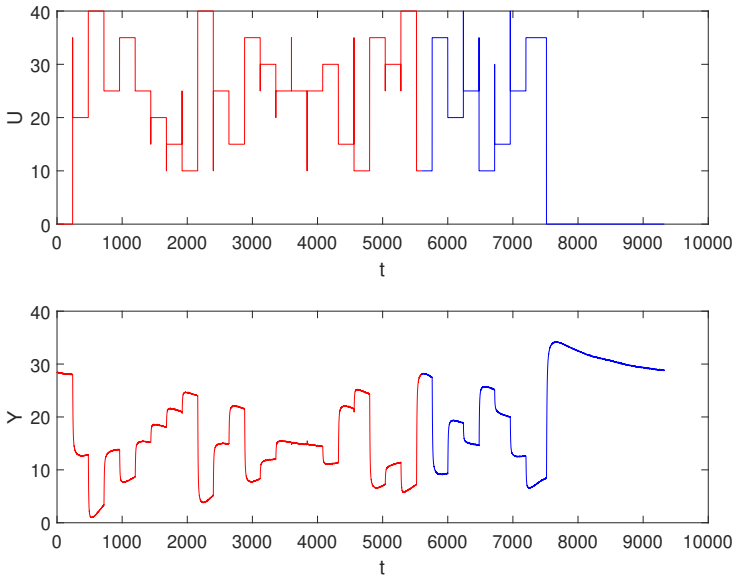

Figure 5.8: Peltier cell: data set. Identification data are plotted in red, validation data in blue.

5.2 Peltier Cell

The next data set comes from a Peltier cell, a thermoelectric device. In the experimental context of these data collection, heat diffusion effects were studied. It is thus interesting to compare the approximation performances of integer-order against fractional-orders models, since in the literature fractional-order models are appraised for their great capability of identifying systems involved in fluid diffusion, especially.

5.2.1 Data Set

The only things that change with respect at the analogous previous subsection are that in this case the data set is much bigger (about three times larger), and in the last 20% of its output column it presents a monotonic slight increase after an initial sudden raising. Therefore we opt for a 60 : 40 split in order to preserve variability in the validation data as shown in Figure 5.8.

5.2.2 Integer-Order Identification

Like we did before in the case of the furnace we choose to identify the system with a rational transfer function with a first-order numerator and a second-order

denominator. This time, however, we interpolate with a time step $T_s = 7$. We get:

$$G_I(s) = \frac{-0.119\,s - 0.0001706}{s^2 + 0.1701\,s + 0.0003153}. \tag{5.3}$$

In this case the fit is much better, with a fit of 87.11%, a MSE of 0.7803 and a norm of the difference between approximation and experimental data of 22.1407.

5.2.3 Fractional-Order Identification

We keep unvaried all the settings for the FOMCON toolbox that we discussed in the case of the industrial furnace. The fractional transfer function found for the Peltier cell is:

$$G_F(s) = \frac{-123.35\,s^{0.078998} + 6.8456}{953.83\,s^{0.8569} + 103.21\,s^{4.9443\times10^{-6}} + 1}. \tag{5.4}$$

5.2.4 Validation

Like before, we employed the two transfer functions to approximate also the validation data and compare the performances.

5.2.5 Comparison

First off, this time the errors are significantly lower. The fractional model perform slightly worse than the integer one over the identification data, while the former outdoes the latter by a neck over the validation set. Table 5.2 reports the errors committed, expressed as the norm of the difference between approximated and experimental data, like in the case of the furnace.
Again, we find that the two different models are practically the same in terms of approximation.
Figures 5.9–5.14 show the comparison between the two models, with respect to experimental data of both identification and validation sets, and between each other.

Identification error			Validation error		
Integer	Fractional	Relative %	Integer	Fractional	Relative %
22.1407	25.3970	12.82%	50.7180	49.6075	2.19%

Table 5.2: Peltier cell: approximation errors. Computed as $\|y - \tilde{y}\|$, where y are the experimental data, $\tilde{y}$ are the approximated data and $\|\cdot\|$ is the Euclidean norm. The relative percentage express the difference of the errors committed by the two models in proportion of the highest between them.

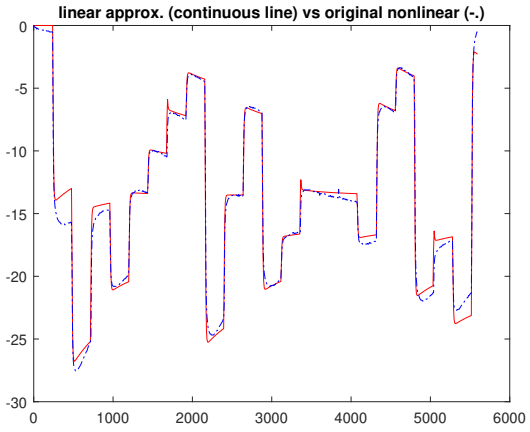

Figure 5.9: Peltier cell: integer-order identification. Approximated data are plotted in red, experimental data in blue (dashed).

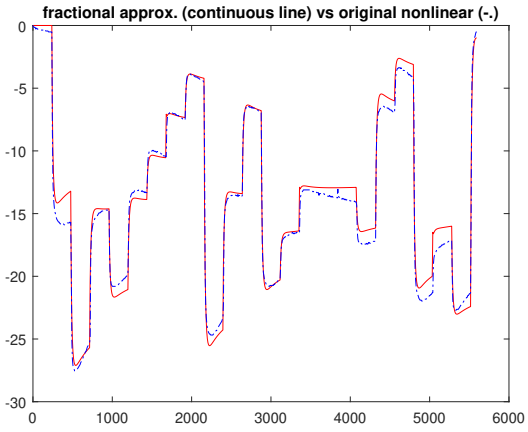

Figure 5.10: Peltier cell: fractional-order identification. Approximated data are plotted in red, experimental data in blue (dashed).

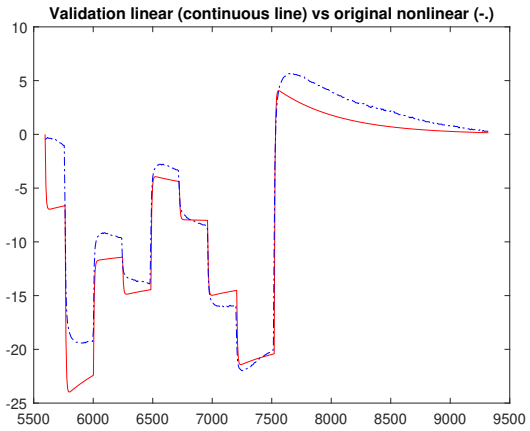

Figure 5.11: Peltier cell: integer-order validation. Approximated data are plotted in red, experimental data in blue (dashed).

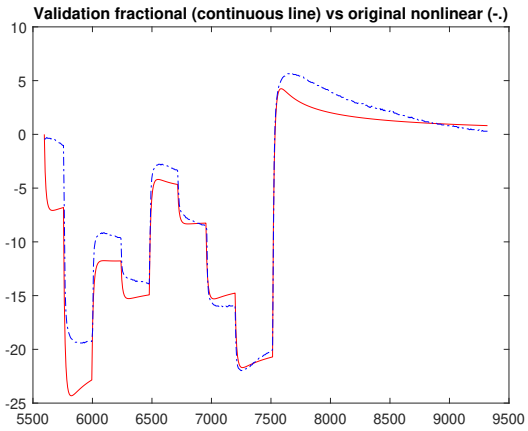

Figure 5.12: Peltier cell: fractional-order validation. Approximated data are plotted in red, experimental data in blue (dashed).

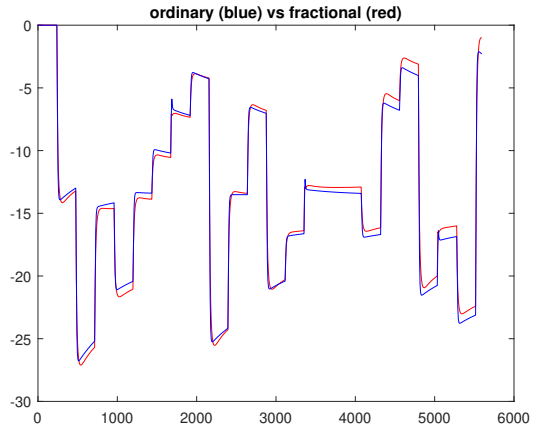

Figure 5.13: Peltier cell: identification comparison. Fractional approximation data are plotted in red, integer approximation data in blue.

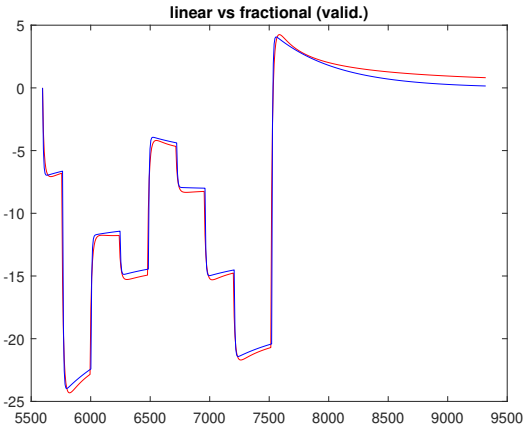

Figure 5.14: peltier: validation comparison. Fractional approximation data are plotted in red, integer approximation data in blue.

5.3 Simulated Data

The last experimental result is different from the previous two. In this case we start from a transfer function with a strong fractional character, i.e., such that its exponents are not close to integer numbers:

$$G(s) = \frac{-1.3333\, s^{0.63} + 2.6667}{1.3333\, s^{3.501} + 2.5333\, s^{2.42} + 1.7333\, s^{1.798} + 1.6667\, s^{1.31} + 1}. \tag{5.5}$$

This function is one of the examples employed by Tepljakov in [32] to illustrate the use of the `fid` function in the FOMCON.

```
G = fotf('-1.3333 s^0.63 + 2.6667',...
 '1.3333 s^3.501 + 2.5333 s^2.42 +...
 1.7333 s^1.798 + 1.6667 s^1.31 + 1');
```

5.3.1 Data Set

We simulate input data at random to mimic the one obtained in a real experimental context. In this way we can use G to simulate the outputs data that are supposed to be associated to the random inputs.

```
time = 0:0.5:4999.5;
num_instant = length(time);
constant = 500;
num_samples = num_instant/constant;
% Inputs are designed to be piecewise constant.
inputs = [];
for i = 1:num_samples
    valore = 100.*rand(1);
    inputs = [inputs valore.*ones(1,constant)];
end
outputs = lsim(G,inputs,time);
```

5.3.2 Integer-Order Identification

As before, we interpolate the data (here $T_s = 10$) before performing system identification.

```
Ts = 10;
time_int = [0:Ts:time(end)];
u = interp1(time,inputs,time_int);
y = interp1(time,outputs,time_int);
```

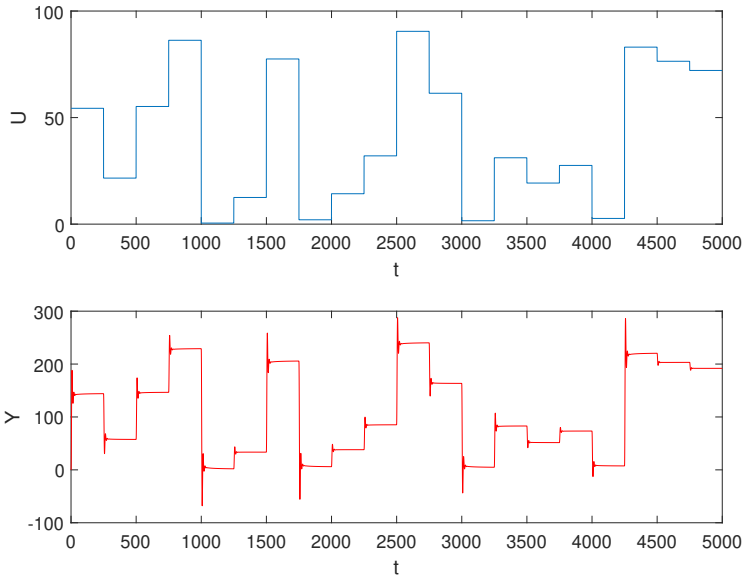

Figure 5.15: Simulated Data: data set. Inputs are plotted in blue, outputs in red.

Next we identify G with an integer-order model: it will be our reference. We choose a fourth-order numerator and a sixth-order denominator.

```
data = iddata(y',u',Ts);
sys = tfest(data,6,4)
G_ord=tf(sys);
y_id_ordinary = lsim(G_ord,u,time_int)
```

What we obtained is:

$$G_I(s) = \frac{15.63\,s^4 + 4.076\,s^3 + 0.1355\,s^2 + 0.0008051\,s + 8.178 \times 10^{-7}}{s^6 + 24.38\,s^5 + 8.618\,s^4 + 1.644\,s^3 + 0.05196\,s^2 + 0.0003044\,s + 3.07 \times 10^{-7}}. \tag{5.6}$$

The quality of the approximation is the highest so far, with a fit percentage of 99.84%, a MSE of 0.01765 and a norm of the difference between approximated and simulated of 7.9039.
This fact alone tells us that an integer-order transfer function can approximate a fractional-order one with absolute precision without the need of excessively high order of derivation or computational resources.

5.3.3 Oustaloup's Filter

The final step is to use Oustaloup's approximation of G provided by the function `oustapp` to simulate another set of outputs and then compare it against both the

simulated experimental one and integer-order one. We set the coefficients to be bounded in $[10^{-3}, 10^{3}]$ and the orders to be less or equal than 20.

```
G_oustapp = oustapp(G,0.001,1000,20,'oust');
y_id_oustapp = lsim(G_oustapp,u,time_int);
```

Oustaloup's approximation of `G_oustapp` is quite cumbrous, too much to report it here; we will make to say that it has 262 terms suffice.

5.3.4 Comparison

Like the previous ones, in this case also the more sophisticated approximation is the one that performs worse, with an error of 59.4048 (against 7.9039). Anyway, the two models are again very similar to each other when looking at the provided outputs, as Figures 5.16–5.18 show.

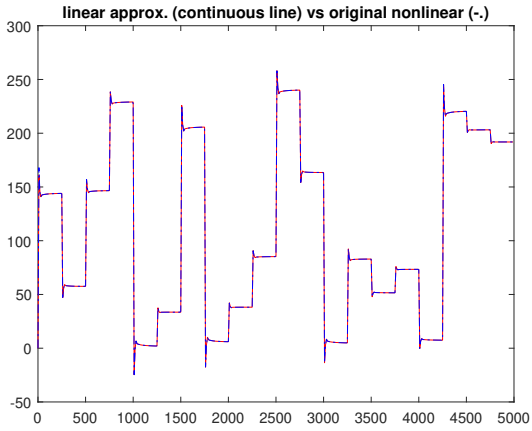

Figure 5.16: Simulated Data: integer-order identification. Approximated data are plotted in red, experimental data in blue (dashed).

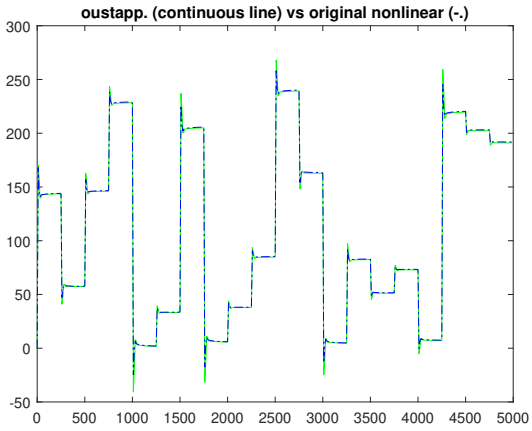

Figure 5.17: Simulated Data: Oustaloup's approximation. Approximated data are plotted in green, experimental data in blue (dashed).

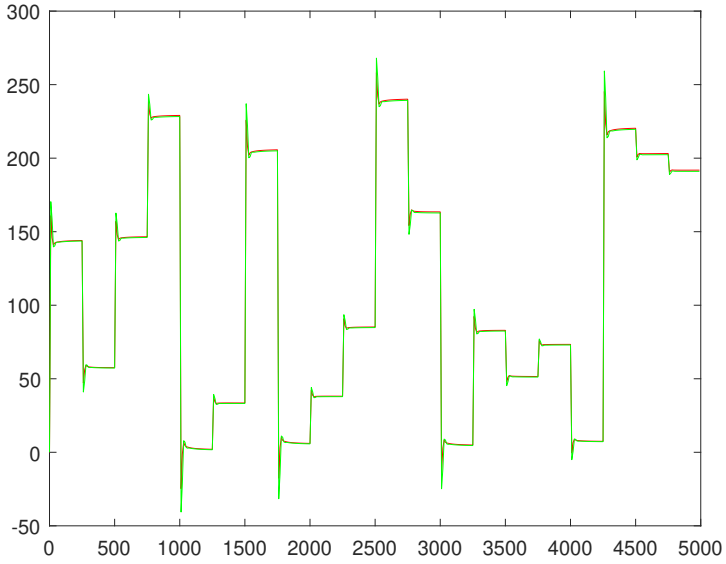

Figure 5.18: Simulated Data: identification comparison. Integer-order approximation data are plotted in red, Oustaloup's approximation in green.

Chapter 6

Conclusions

Fractional-order operators have been gaining popularity in the field of Control Engineering in recent years, especially for their performances in capturing systems nonlinearities. What is more, the whole domain of frequency analysis has a fractional-order counterpart that could be explored to achieve significant advances in the whole branch of control theory. However, using the fractional variant of Bode diagram to analyze a nonlinear system modeled with a fractional transfer function showed no evidence of dependence on the amplitude in the magnitude plot, thus questioning whether any significant nonlinearity could have been captured without leaving no trace in the diagram. This fact, noticed by the second adviser of this Master Thesis, inspired the three experiments we explained in the previous chapter.

In every of the three cases examined in the previous chapter, fractional-order identification of transfer function failed to provide significant improvements with respect to the performance of standard integer-order identification, even in the case where fractional operators capabilities where expected the most, i.e., the one of the Peltier cell.

In all cases fairly wide freedom was given to parameters search in order to give the chance to fractional operators to show their strength at their best. However, both when approximation error was high and when it was low, the results provided by fractional transfer functions were more deficient (even if not very much) than the ones provided by integer-order transfer functions, surprisingly.

Randomly generated data also confirmed this trend when they were used to simulate outputs based on a fractional transfer function: a relatively low-order classical integer transfer function was more effective in fitting the data than Oustaloup's approximation, that is, more effective than what is popularly used to approximate fractional transfer functions themselves.

However, as stressed before, what really stands out is that every time the two kinds of model approximation proposed are nearly indistinguishable, virtually the same.

This may be due to the fact that to approximate fractional transfer functions in an efficient manner (for instance using Oustaloup's filter), one in the end works with integer-order transfer function, possibly of (very) high order, though.
In conclusion, these results suggest that fractional-order methods may be useful to take into account the system nonlinearities provided that ordinary linear approximations are not used except for the implementation phase. Yet if a filter like Oustaloup's one is employed at any time, all the theoretical advantage given by non-integer-order operators may end up frustrated in an approximation, making the adoption of fractional calculus useless.

Bibliography

[1] *Fomcon toolbox reference manual*. `http://docs.fomcon.net`.

[2] *Fractional differential equations*, vol. 198 of Mathematics in Science and Engineering, Elsevier, 1999, p. iii.

[3] S. Bennett, *A History of Control Engineering 1930-1955*, 1993.

[4] G. W. S. Blair and M. Reiner, *The rheological law underlying the nutting equation*, Applied Scientific Research, (1951), pp. 2–225.

[5] D. Blazquez-Sanz, *Differential Galois Theory and Lie-Vessiot Systems*, PhD thesis, 07 2008.

[6] M. Caputo and F. Mainardi, *A new dissipation model based on memory mechanism*, Pure and Applied Geophysics, 91 (1971), pp. 134–147.

[7] M. Caputo and F. Mainardi, *Linear models of dissipation in anelastic solids*, La Rivista del Nuovo Cimento, 1 (1971), pp. 161–198.

[8] Y. Chen, I. Petráš, and D. Xue, *Fractional order control - a tutorial*, 07 2009, pp. 1397 – 1411.

[9] K. Diethelm, *An algorithm for the numerical solution of differential equations of fractional order*, Electronic Transactions on Numerical Analysis, 5 (1998).

[10] ——, *The Analysis of Fractional Differential Equations*, Springer International Publishing, 01 2010.

[11] A. Erdélyi, *Higher Transcendental Functions*, vol. 175, 01 1954.

[12] I. Garcia, H. Giacomini, and J. Giné, *Generalized nonlinear superposition principles for polynomial planar vector fields*, Journal of Lie Theory, 1 (2005).

[13] R. Garrappa, *Numerical solution of fractional differential equations: A survey and a software tutorial*, 6 (2018), p. 16.

[14] Z. Li, L. Liu, S. Dehghan, Y. Chen, and D. Xue, *A review and evaluation of numerical tools for fractional calculus and fractional order control*, International Journal of Control, (2015).

[15] L. Ljung, *System Identification: Theory for the User*, Prentice Hall Information and System Sciences Series, 2 ed., 1999.

[16] C. Lubich, *Discretized fractional calculus*, SIAM J. Math. Anal., 17 (1986), pp. 704–719.

[17] R. Magin, *Fractional calculus in bioengineering*, Critical reviews in biomedical engineering, 32 (2004), pp. 1–104.

[18] R. Malti, M. Aoun, J. Sabatier, and A. Oustaloup, *Tutorial on system identification using fractional differentiation models*, IFAC Proceedings Volumes, 39 (2006), pp. 606 – 611. 14th IFAC Symposium on Identification and System Parameter Estimation.

[19] R. Malti, P. Melchior, P. Lanusse, and A. Oustaloup, *Object oriented crone toolbox for fractional differential signal processing*, Signal, Image and Video Processing, 6 (2012).

[20] K. S. Miller and B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations*, John Wiley-Interscience, 1 ed., 05 1993.

[21] C. A. Monje, Y. Chen, B. M. Vinagre, D. Xue, and V. Feliu, *Fractional-order Systems and Controls*, Advances in Industrial Control, Springer London, 2010.

[22] Y. N. Rabotnov, *Creep problem in structural members*, (1969).

[23] P. Nutting, *A new general law of deformation*, Journal of the Franklin Institute, 191 (1921), pp. 679 – 685.

[24] ——, *A general stress-strain-time formula*, Journal of the Franklin Institute, 235 (1943), pp. 513 – 524.

[25] K. B. Oldham, *The Fractional Calculus: Theory And Applications of Differentiation And Integration to Arbitrary Order*, Dover Pubn, 04 2006.

[26] W. Olmstead and R. Handelsman, *Diffusion in a semi-infinite region with nonlinear surface dissipation*, SIAM Review, 18 (1976), pp. 275–291.

[27] A. Oustaloup, F. Levron, B. Mathieu, and F. Nanot, *Frequency-band complex noninteger differentiator: Characterization and synthesis*, Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, 47 (2000), pp. 25 – 39.

[28] A. Oustaloup, P. Melchior, P. Lanusse, O. Cois, and F. Dancla, *The crone toolbox for matlab*, 02 2000, pp. 190 – 195.

[29] A. Pavlov, N. van de Wouw, and H. Nijmeijer, *Frequency response functions and bode plots for nonlinear convergent systems*, 01 2007, pp. 3765 – 3770.

[30] F. Riesz, *Vorlesungen über Funktionalanalysis*, Hochschulbücher für Mathematik, VEB Deutscher Verlag der Wissenschaften, 1973.

[31] K. Åström and R. Murray, *Feedback systems: An introduction for scientists and engineers*, Feedback Systems: An Introduction for Scientists and Engineers, (2008).

[32] A. Tepljakov, *Fractional-order Modeling and Control of Dynamic Systems*, Springer Theses, Springer International Publishing, 2017.

[33] A. Tepljakov, E. Petlenkov, and J. Belikov, *Fomcon: a matlab toolbox for fractional-order system identification and control*, International Journal of Microelectronics and Computer Science, 2 (2011), pp. 51–62.

[34] D. Valério and J. Sá Da Costa, *Ninteger: a non-integer control toolbox for matlab*, Proc. of the First IFAC Workshop on Fractional Differentiation and Aplications, (2004).

[35] S. Victor, R. Malti, H. Garnier, and A. Oustaloup, *Parameter and differentiation order estimation in fractional models*, Automatica, 49 (2013), pp. 926 – 935.

[36] J. H. Williamson, *Lebesgue Integration*, vol. 13, Cambridge University Press, 1963.

[37] D. Xue, *Fractional-Order Control Systems: Fundamentals and Numerical Implementations*, De Gruyter, 07 2017.