Fabrizio Padula · Antonio Visioli

Advances in Robust Fractional Control

Springer

Advances in Robust Fractional Control

Fabrizio Padula · Antonio Visioli

Advances in Robust Fractional Control

Springer

Fabrizio Padula
Antonio Visioli
Department of Mechanical and Industrial
Engineering
University of Brescia
Brescia
Italy

ISBN 978-3-319-10929-9 ISBN 978-3-319-10930-5 (eBook)
DOI 10.1007/978-3-319-10930-5

Library of Congress Control Number: 2014947700

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To Martina (F. P.)
To Silvia (A. V.)

Preface

Fractional systems and fractional control have received great attention recently, both from an academic and an industrial viewpoint, because of their increased flexibility (with respect to integer-order systems) which allows a more accurate modelling of complex systems and the achievement of more challenging control requirements. In this framework, the aim of this book is to present design methodologies for fractional control systems. Here, fractional control system means that the controller, the process or both can be fractional. Different approaches are described but the common framework is that the robustness of the control system is considered explicitly in the design.

Accordingly, the first part of the book is more industrial oriented and is focused on the design of fractional controllers for integer processes, aiming at evaluating the difference between fractional and integer control. In particular, fractional-order proportional-integral-derivative controllers are considered, since integer-order proportional-integral-derivative regulators are, undoubtedly, the most employed controllers in industry.

The second part of the book deals with a more general approach to fractional control systems. Well-known and effective techniques for integer-order systems, such as $\mathscr{H}_\infty$ optimal control and optimal input–output-inversion-based control are extended to fractional systems.

Actually, apart from the specific topic developed in each single chapter, the leading thread of this book is a widespread effort to generalize to the fractional case methodologies, techniques and theoretical results that are among the reasons for the success of classical (integer) control.

As already mentioned, the performance/robustness trade-off is always explicitly considered in the controller design. It is well known that designing a controller without considering robustness issues is not sensible. Here, different measures of both performance (*e.g.*, integrated absolute error, settling time, closed-loop bandwidth) and robustness (*e.g.*, maximum sensitivity, uncertainty bounds, control effort constraints) are considered, but the common concept remains a control system that explicitly takes into account the fundamental trade-off.

The book can serve as a reference for postgraduate students and academic researchers. It can be also exploited by industrial practitioners to evaluate the suitability of fractional controllers to solve a control problem in a given application. Readers are assumed to know the fundamentals of linear control systems, which are typically taught in a basic automatic control course at the university level.

The authors would like to acknowledge all the people who have contributed to this book in one way or another, in particular their colleagues Giovanna Finzi, Marialuisa Volta, Claudio Carnevale and Manuel Beschi. The authors are also very grateful to Ramon Vilanova and Salvador Alcantara for their contribution in developing the main ideas contained in Chaps. 5 and 6. Special thanks go to the Editor Oliver Jackson and to the Editorial Assistant Charlotte Cross for their help during the preparation of the manuscript. Finally, the authors would like to thank their families and friends for the support during the time spent in writing the book.

Brescia, July 2014

Fabrizio Padula
Antonio Visioli

Contents

Chapter 1
Introduction to Fractional Calculus

1.1 Fractional Calculus

Fractional calculus is the generalization of the classical operation of derivation and integration to orders other than integer.

The first note about the idea of noninteger differentiation dates back to 1695, in a letter that L'Hôpital wrote to Leibnitz, wondering about the concept of differentiation for noninteger numbers.

From then on, many famous mathematicians such as Euler, Laplace, Fourier, Abel, and Laurent have been working on the idea of fractional differential operators. Nevertheless, it was only later in the nineteenth century and mainly due to the contributions of Liouville, Grünwald, Letnikov and Riemann that a complete theory suitable for modern mathematical developments has been formalized.

Nowadays, fractional calculus is a well-established theory with strong mathematical bases [56, 68, 80, 85, 87, 108, 115, 127]. The main reason for the diffusion of fractional calculus is that it actually provides a more accurate tool to describe several physical systems.

For instance, phenomena such as heat conduction through a semi-infinite solid [14, 15, 40], water flowing through a porous dyke [88, 90, 91] or infinite lossy transmission lines [157] are indeed fractional.

In many industrial and research fields, fractional calculus can be conveniently used. Among these, relevant research topics are electrical circuits [48, 158], chemical processes [80], signal processing [83, 144], bioengineering [54, 61], viscoelasticity [24], chaos theory [105], and obviously control systems [91, 106, 116, 118, 146, 148] which are the subject of this book.

It is a common (and meaningful) idea that operations such as integration or differentiation can be repeated a certain (integer) number of times. So, aside from formal details, the concept of higher (integer) order integration or differentiation is rather intuitive.

The idea of fractional calculus is to generalize these operations to any possible real order. It is clear that depending on the derivative/integral order, in a fractional framework, it is possible to continuously change from derivation to integration.

F. Padula and A. Visioli, *Advances in Robust Fractional Control*,
DOI 10.1007/978-3-319-10930-5_1

Accordingly, it is convenient to define a unique operator, hereafter addressed as *differintegrator* that, depending on the sign of the fractional differential order, can be either a differentiator, an integrator, or the identity operator:

$$ {}_aD_t^p = \begin{cases} \frac{d^p}{dt^p} & p > 0 \\ 1 & p = 0 \\ \int_a^t (d\xi)^{-p} & p < 0, \end{cases} \tag{1.1} $$

where $p \in \mathbb{R}$ is the differintegral order and a and t are called, respectively, lower and upper terminal. In the following sections, this concept will be further investigated and the mathematical definitions and properties necessary for the understanding of this book will be stated.

1.1.1 Grünwald–Letnikov Fractional Differintegral

Before going into the mathematical definitions of the fractional derivative and integral, some properties of classical (integer) differential calculus will be briefly summarized. These will be used later on as starting points to define the fractional differintegral and in order to better understand that fractional calculus is the natural generalization of integer-order one.

1.1.1.1 Integer-Order Calculus

Consider the integer derivative of order $n \in \mathbb{N}$ of a given function $f(t)$, it holds that

$$ D^n f(t) = \lim_{h \to 0} \frac{1}{h^n} \sum_{r=0}^{n} (-1)^r \binom{n}{r} f(t - rh), \tag{1.2} $$

where

$$ \binom{n}{r} = \frac{n(n-1)\cdots(n-r+1)}{r!} \tag{1.3} $$

are the well-known binomial coefficients.

An analogous approach can be developed for the p order integral, $p \in \mathbb{N}$ [115], leading to

$$ {}_aD_t^{-p} f(t) = \lim_{\substack{h \to 0 \\ nh = t-a}} h^p \sum_{r=0}^{n} \left[\begin{matrix} p \\ r \end{matrix} \right] f(t - rh) = \int_0^x \frac{(t-\xi)^{p-1}}{(p-1)!} f(\xi) d\xi, \tag{1.4} $$

where the last term is the Cauchy formula for the p-order integral and

$$\left[\begin{matrix} p \\ r \end{matrix}\right] = (-1)^r \binom{-p}{r} = \frac{p(p+1)\cdots(p+r-1)}{r!}. \tag{1.5}$$

It can be shown [115] that (1.2) and (1.4) have a unique representation

$${}_a D_t^p f(t) = \lim_{\substack{h \to 0 \\ nh = t-a}} h^{-p} \sum_{r=0}^{n} (-1)^r \binom{p}{r} f(t - rh), \tag{1.6}$$

that, depending on the sign of (integer) p can be both the p-order derivative or integral.

1.1.1.2 Generalization to the Fractional Case

Starting from (1.6), it is natural to wonder what happens if p is allowed to span over the *real* numbers. First, the binomial coefficients must be expressed in a general way. For this purpose, the Euler *gamma function* $\Gamma(\cdot)$ can be conveniently used. This function is well-known in the literature (see [56, 65, 108, 115] for a detailed treatise) and is defined as follows

$$\Gamma(x) = \int_0^\infty e^{-\xi} \xi^{x-1} d\xi. \tag{1.7}$$

It can be shown that the following property holds:

$$\Gamma(x+1) = x\Gamma(x), \tag{1.8}$$

from which it can immediately be seen that

$$\Gamma(n+1) = n!, \ n \in \mathbb{N}. \tag{1.9}$$

By means of the previous equation, (1.3) can be expressed as

$$\binom{p}{r} = \frac{\Gamma(p+1)}{\Gamma(r+1)\Gamma(r-p+1)}, \tag{1.10}$$

opening the door for a generalization of (1.6).

Up to now, p has been considered to be integer. Now, allowing p to range over $\mathbb{R}$, the Grünwald–Letnikov (GL) fractional differintegral is obtained:

$$ {}_aD_t^p f(t) = \lim_{\substack{h \to 0 \\ nh = t-a}} h^{-p} \sum_{r=0}^{n} \binom{p}{r} f(t - rh), \quad p \in \mathbb{R}. \tag{1.11} $$

It can be shown [115] that, for $p < 0$ the GL differintegral generalizes the Cauchy formula, indeed it holds that

$$ {}_aD_t^p f(t) = \frac{1}{\Gamma(p)} \int_a^t (t-\xi)^{p-1} f(\xi) d\xi, \quad p < 0. \tag{1.12} $$

Restricting the class of allowable functions $f(\cdot)$ to those whose derivatives are continuous in $[a, t]$ till the $(m+1)$th order, being m the greater integer smaller than $|p|$, the following property holds:

$$ {}_aD_t^p f(t) = \sum_{k=0}^{m} \frac{f^{(k)}(a)(t-a)^{k-p}}{\Gamma(-p+1+1)} + \frac{1}{\Gamma(-p+m+1)} \int_a^t (t-\xi)^{m-p} f^{(m+1)}(\xi) d\xi, $$
$$ p > 0. \tag{1.13} $$

Note that, unlike the integer differentiation, fractional differentiation is a nonlocal operation that is defined over an interval $[a, t]$. Accordingly, in the differintegral symbol ${}_aD_t$, both the upper and the lower terminals (t and a, respectively) are explicitly expressed.

1.1.2 Riemann–Liouville Fractional Differintegral

The Riemann–Liouville (RL) fractional differintegral is the most common definition in the literature. The RL fractional integrator is defined as

$$ {}_aD_t^p f(t) = \frac{1}{\Gamma(-p)} \int_a^t (t-\xi)^{-p-1} f(\xi) \mathrm{d}\xi, \quad p < 0, \tag{1.14} $$

whereas the fractional derivative is defined as

$$ {}_aD_t^p f(t) = \frac{1}{\Gamma(k-p)} \frac{\mathrm{d}^k}{\mathrm{d}t^k} \int_a^t (t-\xi)^{k-p-1} f(\xi) \mathrm{d}\xi, \quad p > 0, \tag{1.15} $$

which is the mth (integer) derivative of the $(k-p)$th integral of $f(\cdot)$, being k the smaller integer bigger than p. The RL fractional derivative admits weaker conditions

of existence than the GL one, indeed it is just required the integrability of $f(t)$ over $[a, t]$ and the existence of the derivatives of the integral up to the kth order [115]. Nevertheless, it can be proven that, under the condition of existence of both the GL and RL fractional derivatives, they are equivalent and (1.15) leads to (1.13).

Indeed, since the GL and RL fractional derivatives are equivalent, (1.13) also applies to the RL derivative showing that the GL and RL fractional derivatives of a constant are not null.

1.1.3 Caputo Fractional Differintegral

The Caputo fractional derivative [22, 23] provides an alternative to the GL and the RL ones. As will be clear later on, it is a useful tool to describe physical phenomena. Indeed, one of the main drawbacks of the RL fractional derivatives is that they lead to differential equations whose initial conditions are expressed in terms of fractional derivatives. Fractional initial conditions have no clear physical interpretation, even though this problem has been partially solved in [50].

Unlike GL and RL ones, the Caputo fractional derivatives lead to differential equations whose initial conditions are expressed as integer derivatives (thus with a clear physical meaning) of the function $f(\cdot)$.

The Caputo integral is exactly the same of RL, while the Caputo fractional derivative is defined as follows:

$$ {}_aD_t^p f(t) = \frac{1}{\Gamma(k-p)} \int_a^t (t-\xi)^{k-p-1} f^{(k)}(\xi) \mathrm{d}\xi \tag{1.16} $$

where k is again the smaller integer bigger than p.

1.1.4 Main Properties

Undoubtedly, the most important property of fractional operators is that they are linear.

Besides this, looking at (1.13), it is clear that the convolution integral appearing in the right hand side of the equation is actually the Caputo fractional derivative. Thus, it is immediate to see that, under the conditions of existence for all the three derivatives, the RL, GL, and Caputo fractional derivative coincides provided that [115]

$$ f(a)^{(k)} = 0, \quad k = 0, \ldots, m, \quad m < p < m+1, \quad m \in \mathbb{N}. \tag{1.17} $$

Another interesting characteristic is that semigroup property (often addressed as additive index property):

$$ {}_aD_t^{\alpha}\, {}_aD_t^{\nu} = {}_aD_t^{\nu}\, {}_aD_t^{\alpha} = {}_aD_t^{\alpha+\nu} \tag{1.18} $$

holds provided that (1.17) is satisfied. Indeed, it can be proven that (1.17) is a sufficient condition to make fractional differentiation become a commutative operation. Note that, sometimes, depending on the adopted fractional derivative, commutative and semigroup properties hold under weaker conditions [115].

Beside these properties, fractional operators have other important characteristics [80, 108]:

- if $f(t)$ is an analytic function, so does ${}_aD_t^{\alpha} f(t)$ with respect to t and α;
- if $\alpha \in \mathbb{N}$, then ${}_aD_t^{\alpha} f(t)$ gives the same results of the classical differentiation (independently from the lower terminal);
- the Leibnitz rule for the fractional derivatives:

$$ {}_aD_t^{\alpha}(\phi(t) f(t)) = \sum_{k=0}^{\infty} \binom{\alpha}{k} \phi^{(k)}(t) {}_aD_t^{\alpha-k} f(t). \tag{1.19} $$

1.2 Geometrical Meaning of Fractional Integration and Differentiation

The aim of this section is to give a clear geometrical interpretation of fractional differential calculus. An interesting geometrical interpretation is proposed in [117] based on the RL fractional integral (note that the fractional derivative is actually an integer derivative of a fractional integral). Since GL, RL, and Caputo are the same provided (1.17) to hold, eventually, the proposed interpretation is rather general. Finally, note that a fractional differintegral of order $p < -1$ can be treated as the mth integer integral of ${}_aD_t^{p+m}$, being m the biggest integer smaller than $|p|$, so that it is of interest to study fractional integral for $0 < p < 1$. This is always the case when dealing with the fractional derivatives.

The RL integral (1.14) can be first represented, for the reader convenience, with respect to a positive real α leading to

$$ {}_aD_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\xi)^{\alpha-1} f(\xi) \mathrm{d}\xi. \tag{1.20} $$

The previous expression can be rewritten as

$$ {}_aD_t^{-\alpha} f(t) = \int_a^t f(\xi) dg_t(\xi), \tag{1.21} $$

where

$$g_t(x) = \frac{1}{\Gamma(\alpha-1)}\left(t^\alpha - (t-\xi)^\alpha\right). \tag{1.22}$$

From the previous equations, it is possible to derive a clear interpretation of the fractional integral, indeed (1.21) appears as a normal (integer) integral whose differential $\mathrm{d}g_t(\xi)$ depends both on the actual point ξ along the integration axis and on the actual point t of the integral function (the upper terminal). So, for a given point t, the differential $\mathrm{d}g_t(\xi)$ decreases moving back into the past along the integration axis. In practice, it is like considering a new integration axis for the integral (1.21) that is not homogeneous, on the contrary, it is "stressed" close to the actual point t and "compressed" back in to the past.

Up to now, the integration time axis has been described for a given point t of the integration upper terminal. Now, it will be considered what happens when t changes. The whole integration axis changes its shape because (1.22) also depends on t. The way t affects $g_t(\xi)$ has an interesting self-similarity property. Indeed, given $k > 0$, it holds that

$$g_{kt}(k\xi) = k^\alpha g_t(\xi). \tag{1.23}$$

The previous considerations allow for an interesting geometrical interpretation of the fractional integral due to Podlubny [117]. Consider a 3-dimensional space $\xi - g_t - f$. First, the curve $g_t(\xi)$ for $0 < \xi < t$ is plotted on the $\xi - g_t$ plane and also the 3-dimensional curve $(g_t(\xi), f(\xi))$ for $0 < \xi < t$ is plotted. Note that the two curves create a "fence" in the 3-dimensional space (see Fig. 1.1). Vertical lines are plotted

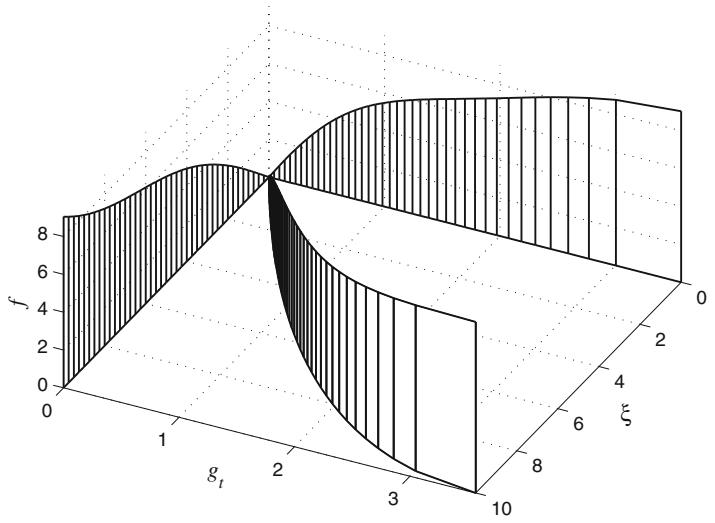

Fig. 1.1 The fence in the $\xi - g_t - f$ space for $\alpha = 0.5$ and $t = 10$

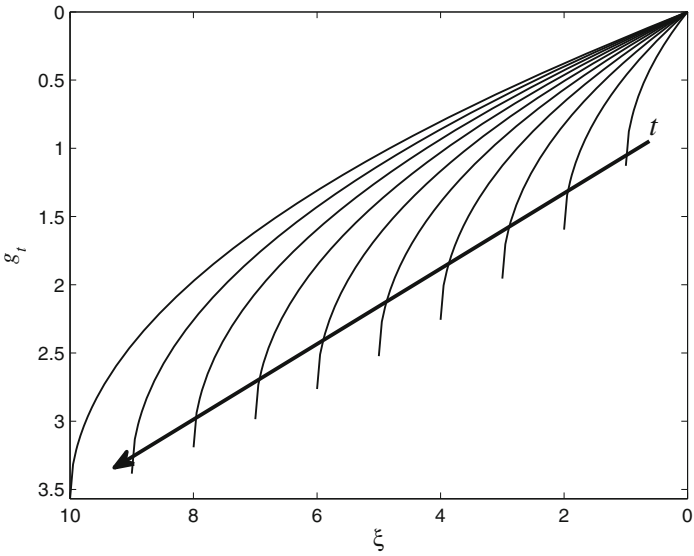

Fig. 1.2 The fences projected onto the $\xi - g_t$ plane for $\alpha = 0.5$ and t varying from 1 to 10

along the fence at uniformly spaced intervals in ξ, then the fence is projected both onto the $\xi - f$ and the $g_t - f$ planes:

- the area of the projection onto the $g_t - f$ plane corresponds to the value of (1.20);
- the projection of the fence onto the $g_t - f$ plane shows how the uniformly spaced steps result stressed when ξ is close to t and compressed elsewhere. This corresponds to the deformed integral axis of (1.21);
- when ξ approaches t, the fence becomes parallel to the g_t axis (the new integral axis) because of the convolution kernel $(t - \xi)^{\alpha-1}$ that tends to infinite and the integral axis becomes (locally) infinitely stressed.

Now, in order to see what happens to the fence, the integration upper terminal t is increased. Figure 1.2 shows how the fence grows as the integration time changes and Fig. 1.3 shows how the shadow of the fence onto the $g_t - f$ plane changes.[1] The consequence of this growth is that the *whole* integral time scale g_t changes as the integration horizon moves ahead. Moreover, Fig. 1.3 also offers an interesting interpretation of the fractional derivative. Indeed, the fractional derivative is the area of the surface between the current shadow and the previous one divided by the time elapsed between the two projections.

[1] A simple movie showing this effect can be found in [92].

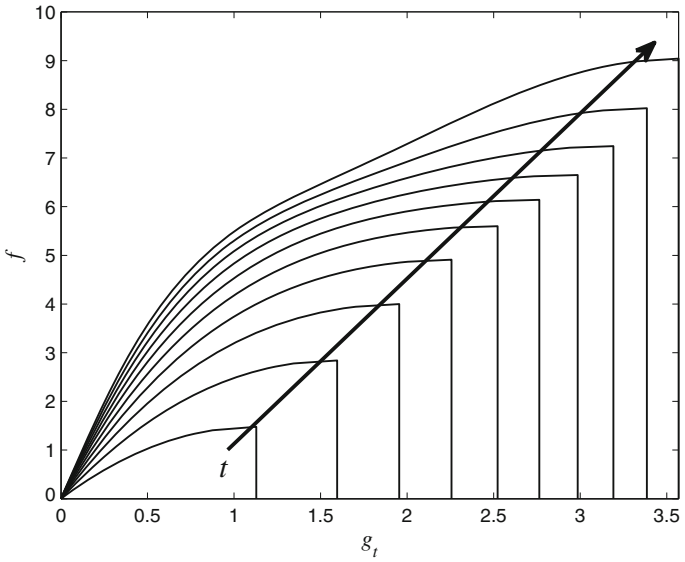

Fig. 1.3 The fences projected onto the $f - g_t$ plane for $\alpha = 0.5$ and t varying from 1 to 10

1.2.1 The Memory Effect

The previous geometrical interpretation shows that fractional integrators have a peculiar memory property. Indeed, the memory of the past history of the system is lost as the time horizon t moves ahead. This behavior is related to the integration timescale that changes its shape continuously. It is geometrically related to the shape of the fence that, at the lower terminal a (0 in Fig. 1.2) is increasingly perpendicular to the g_t axis. Consequently, when ξ is close to the lower terminal, the shadow (projection) onto the $g_t - f$ plane decreases as t increases causing the typical fading memory behavior of fractional integrators. Mathematically, the lossy memory of the operator depends on the convolution kernel appearing in (1.14).

If from one side, the fractional integral has a fading memory behavior, from the other side, the behavior of the integral at the present time t depends on *all* the past history of the system. This behavior can also be interpreted in terms of the fence that dynamically changes its shape as t increases, so that the shadow onto $g_t - f$ changes and the fractional integral must be recomputed at each time instant integrating from the lower terminal.

Analogously, the fractional derivative clearly is a nonlocal operator because, looking at (1.15), it immediately appears that it shows the same memory behavior of the fractional integral.

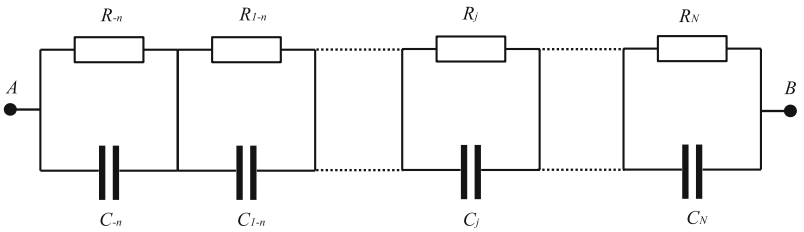

Fig. 1.4 The fractal circuit

1.3 A Physical Interpretation of Fractional Differintegrals

In the literature, there are several physical interpretations of fractional operators, see, for example, [45, 117] and the references therein.

Among the interesting physical properties of fractional operators, a very peculiar one is that they can be conveniently used to describe the dynamic behavior of a class of fractal systems that show certain self-similarity properties. Since this chapter is intended as an introduction to fractional calculus and the whole book is in the field of automatic control, a detailed discussion about these mathematical aspects is out of the scope of the book itself. Nevertheless, an example of physical realization of fractional operators in terms of fractal systems can help the reader to familiarize with the topic.

Consider the linear circuit in Fig. 1.4. It is a chain of resistors and capacitors interconnected at each node. In [81], it has been proven that the relation between the potential difference $E_{AB}(t)$ across the whole circuit and the current $i(t)$ flowing through the circuit is

$$E_{AB}(t) = \int_0^t i(t-\xi) \sum_{j=-n}^{N} \frac{e^{-\xi/R_j C_j}}{C_j} d\xi. \tag{1.24}$$

Now, consider that the capacitors and the resistors are distributed in such a way that

$$R_j = R_0 g^{-j} \quad C_j = C_0 G^{-j}, \tag{1.25}$$

where $g > 1$ and $G > 1$ are, respectively, the geometric progression ratios of resistor and capacitor chains.

Now, making n, N tend to infinity and g, G to one, it can be proven [76] that the inductance of the circuit is *exactly* a fractional operator, namely

$$E_{AB}(t) = K_0 D_t^{\nu-1} i(t), \tag{1.26}$$

where K is a suitable constant and

$$\nu = \frac{\ln(G)}{\ln(gG)}. \tag{1.27}$$

The previous result is actually more general that it could seem at a first sight. Indeed, apart from the electrical application proposed in the section, this idea has been used in several applications that involve the model of a porous dyke [88, 90, 91], rheological models [49], biological models [54, 61], and many others (see [84] and the references therein).

Eventually, it shows that systems described by repeated linear elements geometrically distributed remarkably exhibit a fractional dynamics [27].

1.4 Laplace Transform of Fractional Differintegrals

Right now, fractional operators have been investigated in the time domain. Now, the focus will be on their Laplace transform that opens the door for the frequency domain analysis of fractional operators (note that their are indeed linear operators). Depending on the adopted definition (RL, Caputo, or GL), the Laplace transform of the fractional derivative changes. It is worth stressing that the following results hold provided the lower terminal to be null, i.e., $a = 0$. The following results are quoted without proofs.

Laplace Transform of the RL Fractional Integral and Derivative

The Laplace transform of RL fractional differintegral ${}_0D_t^{\nu} f(t)$ is [108, 115]:

$$\begin{aligned} &\mathscr{L}\left[{}_0D_t^{\nu} f(t)\right] = s^{\nu} F(s), \quad \nu \leq 0; \\ &\mathscr{L}\left[{}_0D_t^{\nu} f(t)\right] = s^{\nu} F(s) - \sum_{k=0}^{n-1} s^k \, {}_0D_t^{\nu-k-1} f(0), \quad n-1 < \nu \leq n \in \mathbb{N}, \ \nu > 0. \end{aligned} \tag{1.28}$$

Laplace Transform of the Caputo Fractional Integral and Derivative

The Laplace transform of Caputo fractional differintegral ${}_0D_t^{\nu} f(t)$ is [108, 115]:

$$\mathscr{L}\left[{}_0D_t^{\nu} f(t)\right] = s^{\nu} F(s), \quad \nu \leq 0;$$

$$\mathscr{L}\left[{}_0D_t^{\nu} f(t)\right] = s^{\nu} F(s) - \sum_{k=0}^{n-1} s^{\nu-k-1} \frac{d^k}{dt^k} f(0), \quad n-1 < \nu \leq n \in \mathbb{N}, \quad \nu > 0. \tag{1.29}$$

Laplace Transform of the GL Fractional Integral and Derivative

The Laplace transform of GL fractional differintegral ${}_0D_t^{\nu} f(t)$ is [115]:

$$\mathscr{L}\left[{}_0D_t^{\nu} f(t)\right] = s^{\nu} F(s). \tag{1.30}$$

The main difference between the RL and Caputo fractional derivatives is that the second one leads to fractional differential equations whose initial conditions are expressed, accordingly to (1.29), in terms of the integer derivatives. Hence, in many practical cases, the Caputo derivatives are preferred because of the clear physical meaning of initial conditions.

It is worth stressing that under null initial conditions, the Laplace transforms of RL, Caputo, and GL fractional differintegral become equivalent. This result is somehow the frequency domain version of the equivalence property presented in Sect. 1.1.4 under condition (1.17). While this is obvious for the Caputo derivative, whose initial conditions are integer, it is not the case of the RL derivative. Nevertheless, in the latter case, it holds that [115]:

$$\left[{}_aD_t^{\nu} f(t)\right]_{t=a} = 0 \Longleftrightarrow f^{(k)} = 0, \quad k = 0, 1, \ldots, n-1, \quad n-1 < \nu < n \in \mathbb{N}. \tag{1.31}$$

The consequence of the previous double implication is that under null initial conditions in the classical (integer) sense, also the fractional derivatives at the lower terminal a are null and the equivalence between the Laplace transforms holds.

As a consequence, the input-output properties of fractional systems (see Chap. 2) are the same, independently from the adopted definition.

Accordingly, null initial conditions will be assumed from now on, unless explicitly specified.

1.5 Frequency Domain Interpretation: The Function s^{ν}

Despite the complexity of fractional operators in the time domain, in the frequency domain, they have a very simple form. Indeed, under null initial conditions, the Laplace transform becomes

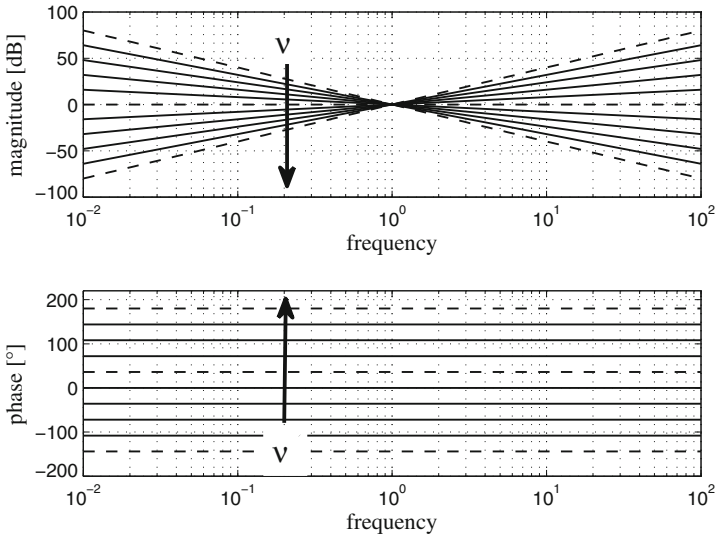

Fig. 1.5 Bode diagrams of s^ν, for ν varying from -2 to 2, *dashed lines* for integer ν

$$\mathscr{L}\left[{}_0D_t^\nu f(t)\right] = s^\nu F(s), \quad \nu \in \mathbb{R}, \tag{1.32}$$

and it is immediate to see that they actually are the natural generalization to an arbitrary order of integer differential operators.

Analyzing the function s^ν by means of its Bode diagram, Fig. 1.5 shows the frequency behavior of the fractional differintegrator depending on the order ν.

In the integer case, the slope of the magnitude Bode plot can only be an integer multiple of 20 dB per decade, depending on the derivative or integral order. Consequently, the phase can only assume values that are integer multiples of 90°.

On the contrary, in the fractional framework, the order ν can continuously vary. Hence, all the intermediate slopes and phases in the Bode diagrams are allowed. This greatly increases the flexibility of fractional control systems.

1.6 The Power Function

The Laplace transform provides a useful tool to further clarify the way fractional operators work. Indeed, it becomes immediate to see how they operate on power functions. The reader should be confident with the classical integrals and derivatives of power functions, so that as a last example, the generalization to the fractional case of the power function derivatives and integrals and the way the simple frequency domain behavior reflects in the time domain creating phenomena such as the memory effect are considered.

First, consider the Laplace transform of the power function, it holds that [87]

$$\mathscr{L}\left[t^{\beta}\right] = \frac{1}{s^{\beta+1}}\Gamma(\beta+1). \tag{1.33}$$

Note that, applying (1.9), when $\beta \in \mathbb{N}$, (1.33) returns exactly the well-known formulae for Laplace transform of step, ramp, parabola and so on.

Now, applying the fractional derivative to the power function and using the Laplace transform properties, it is immediate to write

$${}_0D_t^{\nu} t^{\beta} = \mathscr{L}^{-1}\left[\Gamma(\beta+1)\frac{s^{\nu}}{s^{\beta+1}}\right] \tag{1.34}$$

and applying again (1.33), the left hand side of the previous equation is represented in the time domain as

$${}_0D_t^{\nu} t^{\beta} = \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\nu)} t^{\beta-\nu}. \tag{1.35}$$

Again, note that if $\nu, \beta \in \mathbb{N}$, the previous equation generalizes the well-known formula for the integral and the derivative of power functions.

Some illustrative examples are given. First, consider the fractional derivative of a function with ramps: the results are reported in Fig. 1.6. When the derivative order is lower than one, the fractional derivative is continuous, otherwise it shows an impulse-like behavior. It is interesting to note the long transient of the 0.5th and 1.5th derivatives after the function becomes constant at $t = 10$. This is because of the memory effect. Note that the function in the example has been obtained as the summation of ramp functions, thanks to the linearity of fractional operators. As a second illustrative example, the fractional integrals of a step-like function are proposed in Fig. 1.7. Again, the memory effect is clear, indeed, the integral function does not remain constant when the input becomes zero. Another interesting point is that the fractional integral becomes unstable when $\nu > 1$, whereas if $\nu < 1$, it is stable in the sense that because of the loss of memory, it goes to zero with null input, but is not bounded-input-bounded-output (BIBO) stable. This behavior is peculiar to fractional linear systems.

Finally, consider the fractional differintegration of the square root function: the results are shown in Fig. 1.8. The interesting point here is that the 0.5th derivative of the square root is constant. The square root is not differentiable in the classical (integer) sense, but in the fractional framework, its differentiation becomes feasible. Note that, in general, if the derivative order ν is lower than the order of the power function β, the derivative is smooth, vice versa it shows an impulse-like behavior, accordingly to (1.35). This last example shows how fractional operators that because of their complexity may seem far from classical ones, actually are the natural extension of the latter.

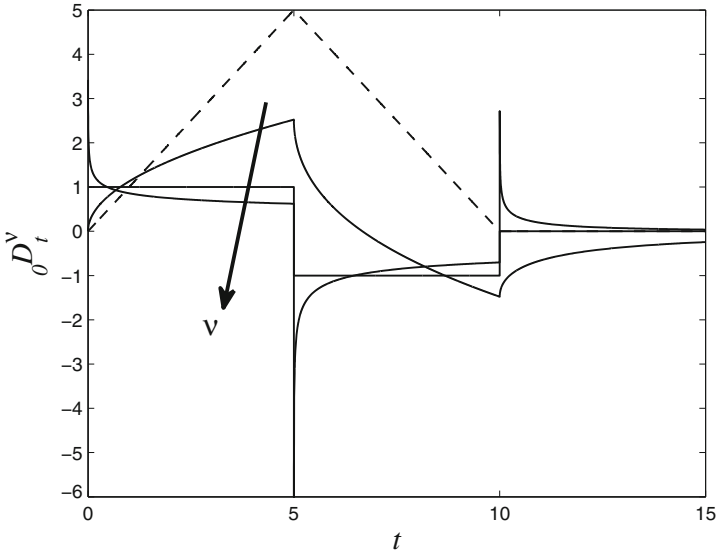

Fig. 1.6 Fractional derivatives (*solid lines*) of a ramp-like function (*dotted line*) for $\nu = 0.5, 1, 1.5$

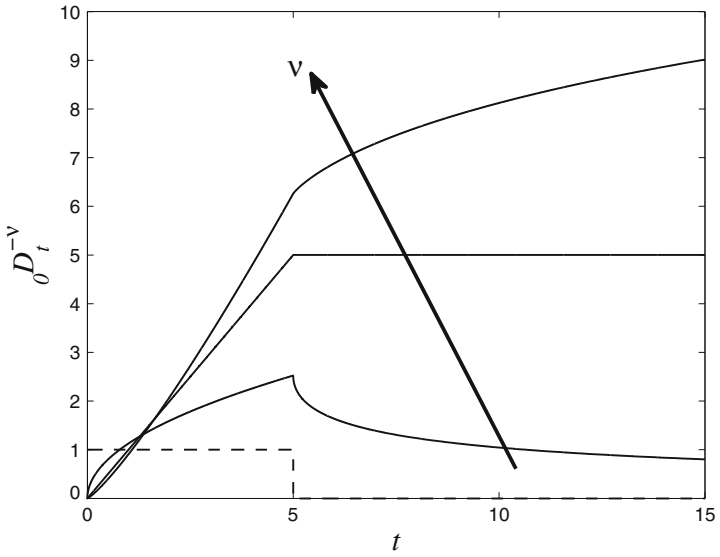

Fig. 1.7 Fractional integrals (*solid lines*) of a step-like function (*dotted line*) for $\nu = 0.5, 1, 1.5$

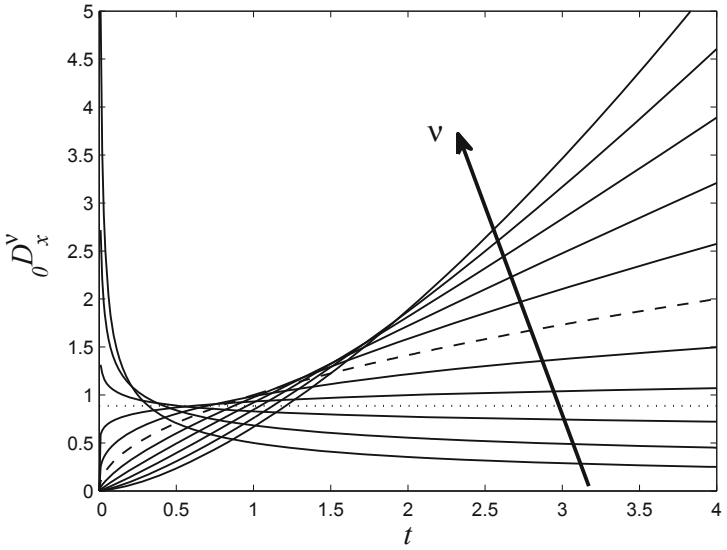

Fig. 1.8 Fractional derivatives (*solid lines*) of the square root function $t^{0.5}$ (*dashed line*) for ν varying from -1 to 1; *dotted line*: $\nu = 0.5$

1.7 Conclusions

A brief introduction to fractional calculus has been provided in this chapter. The main aim has been to give the basic concepts (in both time and frequency domain) which will be used in the rest of the book. A more detailed treatment of fractional calculus can be, however, found in many books [56, 68, 80, 85, 108, 115, 127].

Chapter 2
Fractional Systems for Control

2.1 Fractional Control

Fractional systems have received a great attention recently, both from an academic and an industrial viewpoint [2, 20, 21, 59, 62, 74, 85, 87, 114, 116] because of their increased flexibility (with respect to integer-order systems), which allows a more accurate modeling of complex systems [14, 91, 145], and the achievement of more challenging control requirements [29, 45, 71, 74, 123].

The main idea about fractional control is to design control systems where either the system or the controller, or both of them are fractional dynamic systems [31, 140, 141]. Regarding the design of control systems, many contributions are related to the synthesis of robust control systems (see, for example, [12, 87, 88, 91, 94, 114]). Among these method, it is surely worth mentioning the CRONE control [88]. A great effort has been also focused on the synthesis of fractional-order proportional-integral-derivative controllers (see, for example, [13, 25, 30, 42, 72, 73, 97, 99, 116, 136, 139, 147]). Among the other relevant research topics, it is worth citing the approximation of a fractional system with integer ones [62, 89, 135, 148] and the stability of a fractional (control) systems [2, 107, 124, 125]. All the previous methods share a common mathematical object for the design of fractional control systems: fractional linear time-invariant (LTI) systems. Exactly as in the integer case, also in the fractional one, linear systems are undoubtedly the fundamental tool in control design. Bearing in mind this idea, the rest of this chapter is dedicated to the analysis of fractional LTI systems.

2.2 The Two Parameters Mittag-Leffler Function

Before approaching LTI systems, a fundamental class of functions is introduced: the two parameters Mittag-Leffler functions. These special functions play for fractional LTI systems the same role that exponential functions play for integer ones.

F. Padula and A. Visioli, *Advances in Robust Fractional Control*,
DOI 10.1007/978-3-319-10930-5_2

The Mittag-Leffler function (MLF) is defined as [115]

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

MLF are generalization of the exponential function, indeed it holds that [115]:

$$\begin{aligned} &E_{1,1}(z) = e^z \\ &\vdots \\ &E_{1,m} = \tfrac{1}{z^m}\left(e^z - \sum_{k=0}^{m-2} \tfrac{z^k}{k!}\right). \end{aligned} \tag{2.2}$$

A further interesting generalization is the function introduced by Podlubny in [117] and hereafter addressed as Podlubny function (PF):

$$\varepsilon_k(t, \lambda; \alpha, \beta) = t^{k\alpha+\beta-1} \frac{\mathrm{d}^k}{\mathrm{d}(\lambda t^\alpha)^k} E_{\alpha,\beta}(\lambda t^\alpha), \tag{2.3}$$

being the kth derivative of (2.1) [108]

$$\frac{\mathrm{d}^k}{\mathrm{d}z^k} E_{\alpha,\beta}(z) = \sum_{i=0}^{\infty} \frac{(i+k) z^i}{\Gamma(\alpha(i+k) + \beta)}. \tag{2.4}$$

2.2.1 Laplace Transforms

Now, the Laplace transform of the functions defined in Sect. 2.2 is introduced. The Laplace transform of (2.1) is [108, 115]

$$\mathscr{L}\left[t^{\beta-1} E_{\alpha,\beta}(\lambda t^\alpha)\right] = \frac{s^{\alpha-\beta}}{s^\alpha - \lambda}, \tag{2.5}$$

while the Laplace transform of (2.3) is

$$\mathscr{L}\left[\varepsilon_k(t, \lambda; \alpha, \beta)\right] = \frac{k! s^{\alpha-\beta}}{(s^\alpha - \lambda)^{k+1}}. \tag{2.6}$$

MLFs and PFs are fundamental tools to solve fractional LTI systems. Indeed, the previous equations look like a generalization to the fractional case of simple and multiple poles (and actually they are). This concept will be fully clarified through the rest of the chapter.

2.3 Fractional LTI Systems

In general, a fractional LTI system is represented in the external form by a fractional differential equation with the following structure

$$\sum_{k=0}^{n} a_{k0} D_t^{\beta_k} y(t) = \sum_{k=0}^{m} b_{k0} D_t^{\alpha_k} u(t), \quad \beta_k, \alpha_k \in \mathbb{R}^+, \ a_k, b_k \in \mathbb{R}. \tag{2.7}$$

By means of (1.28)–(1.30), it is immediate to see that, independently from the adopted definition, the transfer function of (2.7) is

$$H(s) = \frac{Y(s)}{U(s)} = \frac{\sum_{k=0}^{m} b_k s^{\alpha_k}}{\sum_{k=0}^{n} a_k s^{\beta_k}}, \quad \beta_k, \alpha_k \in \mathbb{R}^+, \ a_k, b_k \in \mathbb{R}. \tag{2.8}$$

In general, a fractional transfer function is the ratio of two fractional polynomials, i.e., polynomials whose exponents are real numbers.

The following theorem shows the behavior of fractional polynomial at their roots.

Theorem 2.1 *Let $P(s)$ be a fractional polynomial*

$$P(s) = \sum_{i=1}^{n} a_i s^{\gamma_i}, \ a_i \in \mathbb{R}, \ \gamma_i \in \mathbb{R}^+, \tag{2.9}$$

whose zeros are $z_1, \ldots, z_m$. It holds that

$$\lim_{s \to z_i} \frac{P(s)}{(s - z_i)^{k_i}} \tag{2.10}$$

exists finite, for some $k_i \in \mathbb{N}$ for $i = 1, \ldots, m$ provided that $z_i \neq 0$.

Proof By means of the substitution $t = s - z_i$ (2.10) can be rewritten as

$$\lim_{t \to 0} \frac{P(t + z_i)}{(t)^{k_i}}, \ i = 1, \ldots, m. \tag{2.11}$$

By series expansion, $P(s)$ can be represented around z_i as

$$P(t) = \sum_{i=1}^{n} a_i \sum_{m=1}^{l_i} \left. \frac{\mathrm{d}^m P(t + z_i)}{\mathrm{d}(t)^m} \right|_{t=0} t^m + o(t^{l_i}) \text{ for } t \to 0, \ l_i \in \mathbb{N}; \tag{2.12}$$

since the function $P(s)$, evidently, is not constant in the complex plane, for each zero z_i, there exists l_i such as $\sum_{i=1}^{n} a_i \sum_{m=1}^{l_i} \left. \frac{\mathrm{d}^m P(t+z_i)}{\mathrm{d}(t)^m} \right|_{t=0} \neq 0$. Choosing k_i equal to the smallest l_i such as this condition holds, (2.11) exists finite.

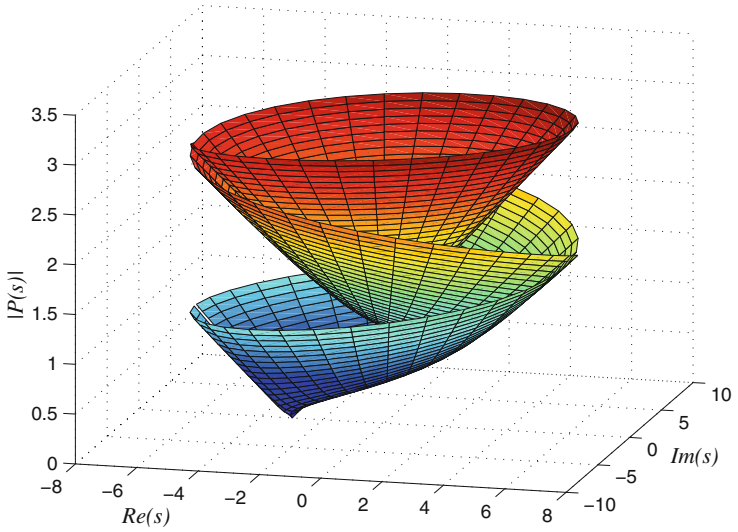

Fig. 2.1 The multivalued function $P(s) = 1 + s^{1/3}$

Definition 2.1 The integer k_i is called order of the zero z_i.

Calculating the zeros of a fractional polynomial is, in general, a complex task and it is beyond the scope of this book. From now on their knowledge is assumed. However, it is worth stressing that when dealing with commensurate-order fractional polynomials (see Sect. 2.4), the computation of the roots becomes trivial [135].

Another interesting characteristic of fractional polynomials (and fractional transfer functions) is that they are multivalued functions [107] as in Fig. 2.1, where a representation of the multivalued behavior of $(1 + s)^{1/3}$ is shown. It is worth stressing that only the first Riemann sheet has a physical meaning [41]; it is defined as [107]

$$\Omega = \{re^{j\phi} : r > 0, -\pi < \phi < \pi\}. \tag{2.13}$$

Hence, from now on, the analysis will be limited to this surface.

2.4 Commensurate Fractional LTI Systems

An important class of fractional LTI systems is the set of commensurate systems.

Definition 2.2 A fractional LTI system (2.8) is said to be commensurate if all the exponents are integer multiples of the same real number ν. The derivative order ν is called commensurate order of the system.

The transfer function of a fractional commensurate LTI system always has the following form:

$$H(s) = \frac{\sum_{k=0}^{m} b_k s^{k\nu}}{s^{n\nu} + \sum_{k=0}^{n-1} a_k s^{k\nu}}, \nu \in \mathbb{R}^+, \ a_k, b_k \in \mathbb{R}, \tag{2.14}$$

It is worth noting that every system whose exponents are rational numbers is commensurate. Moreover, when a system has irrational exponents, they can always be approximated with the desired precision degree as ratios of natural numbers.

For many practical applications, commensurate fractional LTI systems are much more diffused than incommensurate one. First, if the exponents are numerically expressed using a finite number of digits, a fractional LTI system is commensurate, but this is almost always the case, unless the system results to be incommensurate from an analytical procedure. Moreover, they are far more mathematically treatable compared to the incommensurate ones. For this second reason, this chapter, which is intended as an introduction to fractional systems in order to drive the reader through for the rest of the book, is focused on commensurate fractional LTI systems. Furthermore, some theoretical results e.g., stability criteria) are still missing for incommensurate systems. Nevertheless, when a given result also exists for incommensurate systems, references to the literature will always be present. Finally, it has to be stressed that, from a mathematical point of view, incommensurate systems are much more interesting and challenging than commensurate ones.

One of the main advantage of commensurate systems is that they can be factorized via partial fraction expansion exactly as integer systems can. Indeed, it is always possible to associate to a given commensurate system an integer-order one by means of the substitution $s^\nu = w$ into (2.14):

$$\tilde{H}(s) = \frac{\tilde{Y}(w)}{\tilde{U}(w)} = \frac{\tilde{b}(w)}{\tilde{a}(w)} = \frac{\sum_{k=0}^{m} b_k w^k}{w^n + \sum_{k=0}^{n-1} a_k w^k}. \tag{2.15}$$

A function such as (2.14) is, in general, a multivalued function [107] and the substitution operated to obtain the previous equation maps all the Riemann sheets onto the w plane, as Fig. 2.2 shows.

Evidently, it is possible to operate over (2.15) as an integer transfer function and then reobtaining the corresponding fractional one by means of the substitution $w = s^\nu$. Accordingly, a commensurate LTI system can always be factorized via partial fractions expansion leading to:

$$H(s) = \sum_{i=1}^{p} \frac{g_i}{(s^\nu - \lambda_i)^{k_i+1}}, \tag{2.16}$$

where the coefficients λ_i and g_i can be either real or complex (in the latter case they always appear in conjugate pairs).

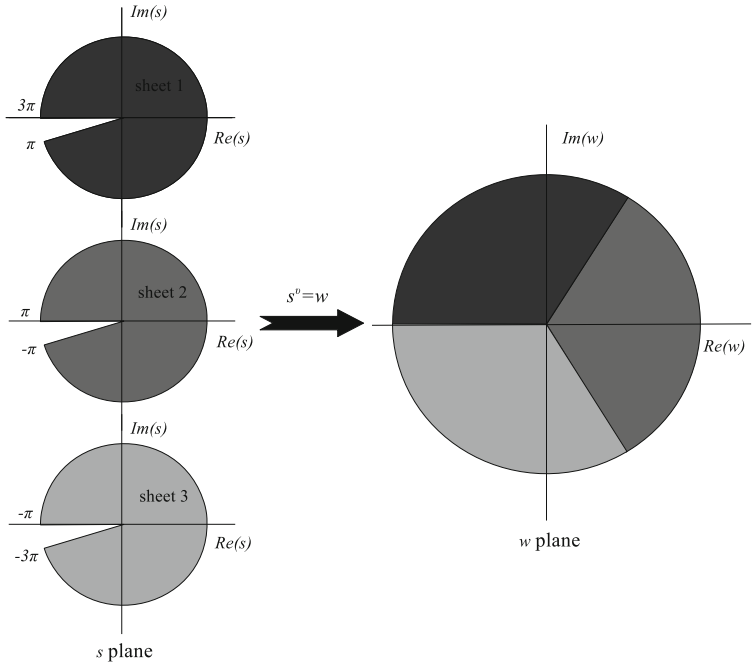

Fig. 2.2 The Riemann sheets for the function $P(s) = 1 + s^{1/3}$ and their map onto the w plane

Now, by means of (2.6), it is immediate to see that the impulse response of (2.16) is

$$\mathscr{L}^{-1}\left[H(s)\right] = \eta(t) = \sum_{i=1}^{p} \frac{g_i}{k_i!} \varepsilon_{k_i}(t, \lambda_i; \nu, \nu). \tag{2.17}$$

Note that, in case of complex coefficients, the PF in (2.16) may be complex valued, but the result of the summation (2.17) is always real valued.

2.5 Modes

The previous results highlight that the basic elements of a commensurate LTI system are a kind of fractional generalization of integer simple fractions, hereafter called fractional-first-order (FFO) system:

$$P(s) = \frac{K}{Ts^{\nu} + 1}. \tag{2.18}$$

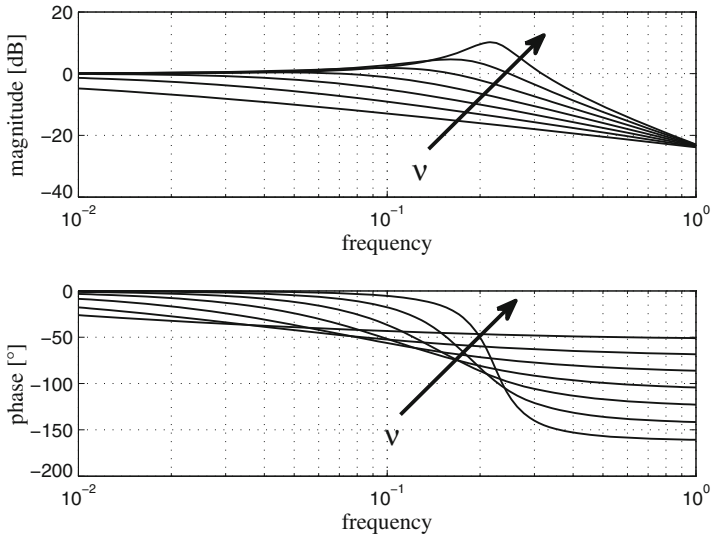

Fig. 2.3 Bode diagram of $P(s) = \frac{1}{1+15s^{\nu}}$ and $\nu \in [0.6, 1.8]$

Figures 2.3 and 2.4 show that FFO systems have the interesting properties of being able to describe both monotonic and nonmonotonic behaviors depending on the exponent ν. Note that both the peak frequency $\bar{\omega}$ and the peak amplitude depend on the derivative order accordingly to [64]

$$\begin{aligned} \bar{\omega} &= -\left(\tfrac{1}{T}\right)^{\frac{1}{\nu}} \cos(\nu\tfrac{\pi}{2}) \\ |P(j\bar{\omega})|_{\mathrm{d}B} &= -20\log\left(\sin\left(\nu\tfrac{\pi}{2}\right)\right), \end{aligned} \tag{2.19}$$

with $\nu > 1$, otherwise (2.18) is not resonant.

It is worth noting that the peak amplitude only depends on the differential order ν, whereas its frequency also depends on the time constant T.

For a given input function $u(t)$, the corresponding function output $y(t)$ can be computed as the convolution integral between (2.17) and $u(t)$ [82, 85, 115], that is:

$$y(t) = \int_0^t \eta(t-\xi)u(\xi)\mathrm{d}\xi. \tag{2.20}$$

When the input function is simple, the solution of the input–output problem for a commensurate LTI system becomes trivial. For fractional integrators, by considering (2.5) with $\lambda = 0$ and $\nu = \alpha = \beta$, it results

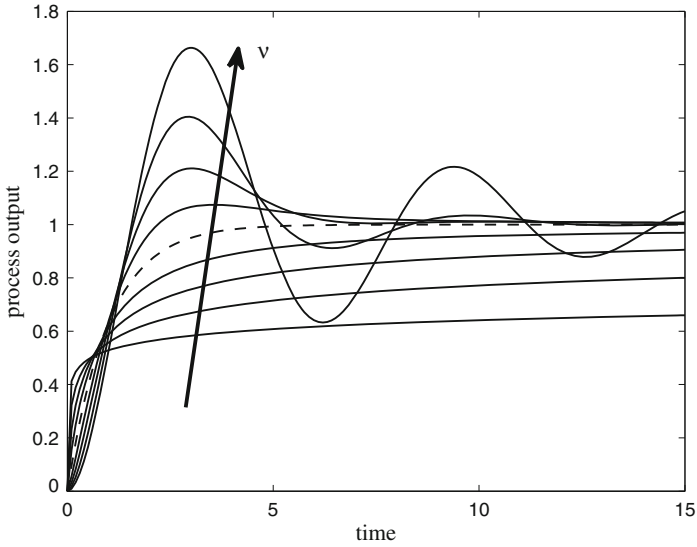

Fig. 2.4 Step responses of $P(s) = \frac{1}{1+s^{\nu}}$ and $\nu \in [0.2, 1.8]$

$$\begin{aligned}
&\mathscr{L}^{-1}\left[\frac{1}{s^{\nu}}\right] = \frac{t^{\nu-1}}{\Gamma(\nu)},\\
&\mathscr{L}^{-1}\left[\frac{1}{s^{\nu}}\mathscr{L}[1(t)]\right] = \frac{t^{\nu}}{\Gamma(\nu+1)},\\
&\vdots\\
&\mathscr{L}^{-1}\left[\frac{1}{s^{\nu}}\mathscr{L}[t^{n}]\right] = \frac{t^{\nu+n}}{\Gamma(\nu+n)},
\end{aligned} \tag{2.21}$$

where $1(t)$ is the Heaviside function.
For simple fractions (FFO systems), the following expressions can be derived (see (2.5) with $\nu = \alpha = \beta$ and $a = \lambda$):

$$\begin{aligned}
&\mathscr{L}^{-1}\left[\frac{1}{(s^{\nu} \mp a)}\right] = t^{\nu-1}E_{\nu,\nu}(\pm a t^{\nu}),\\
&\mathscr{L}^{-1}\left[\frac{1}{(s^{\nu} \mp a)}\mathscr{L}[1(t)]\right] = t^{\nu}E_{\nu,\nu+1}(\pm a t^{\nu}),\\
&\vdots\\
&\mathscr{L}^{-1}\left[\frac{1}{(s^{\nu} \mp a)}\mathscr{L}[t^{n}]\right] = t^{\nu+n}E_{\nu,\nu+n}(\pm a t^{\nu}).
\end{aligned} \tag{2.22}$$

An analogous reasoning can be applied in case of multiple fractions starting from (2.6), yielding

$$\begin{aligned}
&\mathscr{L}^{-1}\left[\frac{1}{(s^{\nu}\mp a)^{k+1}}\right]=\frac{t^{\nu(k+1)-1}}{\Gamma(k+1)}\frac{\mathrm{d}^k E_{\nu,\nu}(\pm a t^{\nu})}{\mathrm{d}(\pm a t^{\alpha})},\\
&\mathscr{L}^{-1}\left[\frac{1}{(s^{\nu}\mp a)^{k+1}}\mathscr{L}[1(t)]\right]=\frac{t^{\nu(k+1)}}{\Gamma(k+1)}\frac{\mathrm{d}^k E_{\nu,\nu+1}(\pm a t^{\nu})}{\mathrm{d}(\pm a t^{\alpha})},\\
&\vdots\\
&\mathscr{L}^{-1}\left[\frac{1}{(s^{\nu}\mp a)^{k+1}}\mathscr{L}[t^{n}]\right]=\frac{t^{\nu(k+1)+n}}{\Gamma(k+1)}\frac{\mathrm{d}^k E_{\nu,\nu+n}(\pm a t^{\nu})}{\mathrm{d}(\pm a t^{\alpha})}.
\end{aligned}\tag{2.23}$$

The previous equations can also be expressed in term of PFs, and together with (2.17) and (2.20), completely address the input–output problem of commensurate LTI systems.

Finally, for the sake of completeness, it has to be said that similar results also exists for incommensurate LTI systems, but in this case, the impulse response (the so-called fractional Green's function) cannot be expressed as a finite summation of PF. For the reasons discussed at the beginning of the section, this solution has been dropped here, the reader may conveniently refer to [115] for a complete treatise.

2.6 Stability

It is well known that stability is a property of main concern for a control systems. Research on stability of fractional linear system is still an ongoing process and many research works have been proposed in the last years [3, 8, 107, 120]. Unlike for integer system, a simple general stability test only exists for commensurate fractional systems. The following theorem generalizes in a very natural way the well-known stability criteria for integer systems [66, 67, 107].

Theorem 2.2 *A fractional-order commensurate fractional systems. The following theorem generalizes in a very natural way the well-known systems described by a commensurate transfer function* (2.14) *with commensurate fractional systems. The following theorem generalizes in a very natural way the well-known order* $\nu < 2$*, is stable if and only if all the poles* $\tilde{p}_i$ *of its integer counterpart* (2.15) *satisfy*

$$|\arg(\tilde{p}_i)| > \nu\frac{\pi}{2}. \tag{2.24}$$

The previous theorem states that the stable region for a fractional commensurate system fractional systems. The following theorem generalizes in a very natural way the well-known is no longer the left half plane (LHP), see Fig. 2.5. This concept is true but sometimes misleading. Indeed (2.24) defines a stable region for the integer

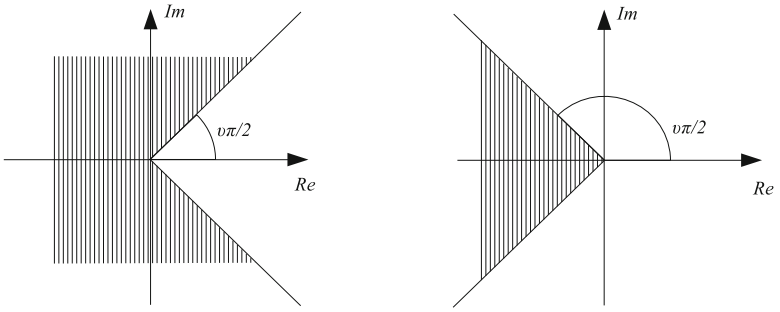

Fig. 2.5 Stable regions with $\nu = 0.5$ (*left*) and $\nu = 1.5$ (*right*)

system associate to a commensurate one. This condition corresponds at having all the singularities of the fractional system (that, in view of Lemma (2.1), may be called poles) in the LHP. Indeed, the first Riemann sheet is mapped onto the region defined by $|\arg(w)| > \nu\pi$ and the right half plane (RHP) is mapped onto the region defined by inequality (2.24).

When the system is incommensurate, stability should be checked computing the roots of the characteristic fractional polynomial (in the first Riemann sheet).

Nevertheless, if it is possible to factorize the system as

$$H(s) = \sum_{i=1}^{N} \sum_{k=1}^{n_k} \frac{g_i, k}{(s^{\beta_i} + \lambda_i)^k}, \quad \beta_i < 2, \tag{2.25}$$

then, it holds that the system is stable if [107]

$$|\arg(\lambda_i)| < \pi \left(1 - \frac{\beta_i}{2}\right), \quad \forall i. \tag{2.26}$$

Again, note that the previous condition is equivalent to saying that $H(s)$ has no poles of the first Riemann sheet in the RHP.

As a final remark, it has to be stressed that only the first Riemann sheet is of interest in the stability analysis because the roots that lie on secondary sheets will give solutions that are always monotonically decreasing [67, 146, 158].

2.7 Conclusions

An introduction to fractional linear system and fractional control has been provided in this chapter. Basic concepts that will be useful in the next chapters have been explained. A more detailed overview on fractional systems and fractional control can be found in many contributions to the literature, see for example [21, 31, 45, 71, 87, 123, 140, 141].

Chapter 3
Fractional-Order Proportional-Integral-Derivative Controllers

3.1 Introduction

Proportional-integral-derivative (PID) controllers are surely the most adopted controllers in industry owing to the cost/benefit ratio they are capable to provide. Indeed, the large numbers of tuning rules [79] for their three parameters and the presence of reliable automatic tuning techniques [11] allow the user to design this kind of controllers with a moderate effort, and to obtain a satisfactory performance for many processes [151]. Hence, it seems that PID controllers are still the preferred control algorithm and object of many interesting researches [16, 98, 121, 152]. Since they are based on an integral and a derivative action, it seems natural to let these action to span over a wider range of dynamics by allowing differentiation and integration of an arbitrary (real) order. Bearing in mind this idea, the application of fractional calculus to PID controllers leads straightforwardly to the main object of this chapter, the fractional-order PID (FOPID) controller.

In the last ten years, the design of FOPID controllers has been the subject of many investigations (see, for example, [13, 17, 20, 25, 26, 30, 34, 42, 43, 63, 70, 72–74, 116, 135, 136, 138, 139, 147, 161]) because of the additional flexibility they are capable to provide with respect to standard (integer-order) PID controllers. Indeed, the presence of five parameters to select makes the achievement of an increased performance virtually possible, but this also implies that the tuning of the controller can be much more complex.

3.2 FOPID Controller Structure

The FOPID controller can be defined both in ideal (noninteracting) or series (interacting) form.

F. Padula and A. Visioli, *Advances in Robust Fractional Control*,
DOI 10.1007/978-3-319-10930-5_3

The first was introduced by Podlubny [116], and is defined as follows:

$$C(s) = K_p \left(1 + \frac{1}{T_i s^{\lambda}} + T_d s^{\mu}\right), \tag{3.1}$$

where λ is the integral order and μ the derivative one.

The interacting form has been proposed in [97], and has the following formulation

$$C(s) = K_p \frac{T_i s^{\lambda} + 1}{T_i s^{\lambda}} (T_d s^{\mu} + 1). \tag{3.2}$$

where, again, λ is the integral order and μ the derivative one.

It can be easily noted that, by selecting $\lambda = \mu = 1$, a standard PID controller is obtained independently from the adopted configuration.

The different FOPID forms have different properties, and they are not equivalent, unless $\lambda = \mu$. Indeed, by expanding Eq. (3.2), the following expression is obtained

$$C(s) = K_p \left(1 + \frac{1}{T_i s^{\lambda}} + T_d s^{\mu} + \frac{T_d}{T_i} s^{\mu-\lambda}\right). \tag{3.3}$$

Comparing the previous equation with Eq. (3.1) it clearly appears that their dynamics differ because of the term

$$K_p \frac{T_d}{T_i} s^{\frac{\mu}{\lambda}}. \tag{3.4}$$

If $T_i \gg T_d$, the term (3.4) can be safely neglected as Figs. 3.1 and 3.2 show (note that this is the case when an ideal PID controller can be represented as a series one).

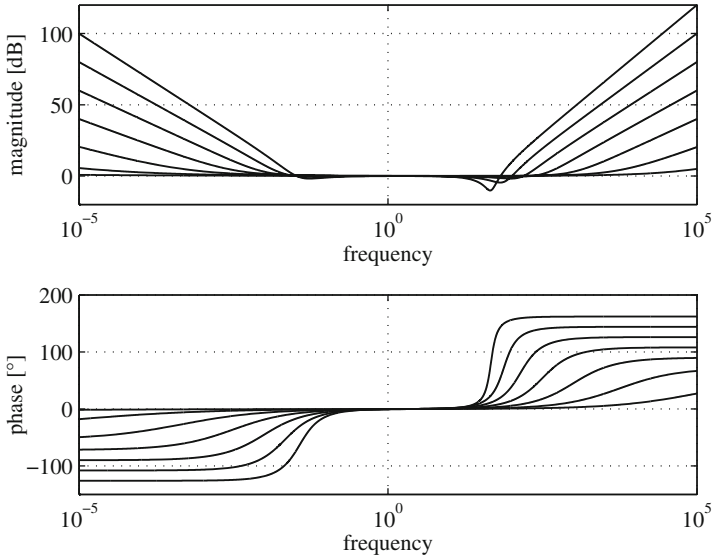

Fig. 3.1 The noninteracting FOPID bode plot, with $T_i = 100$, $T_d = 0.001$ and $\mu = \lambda + 0.4$ swept into the range [0.6 1.8]

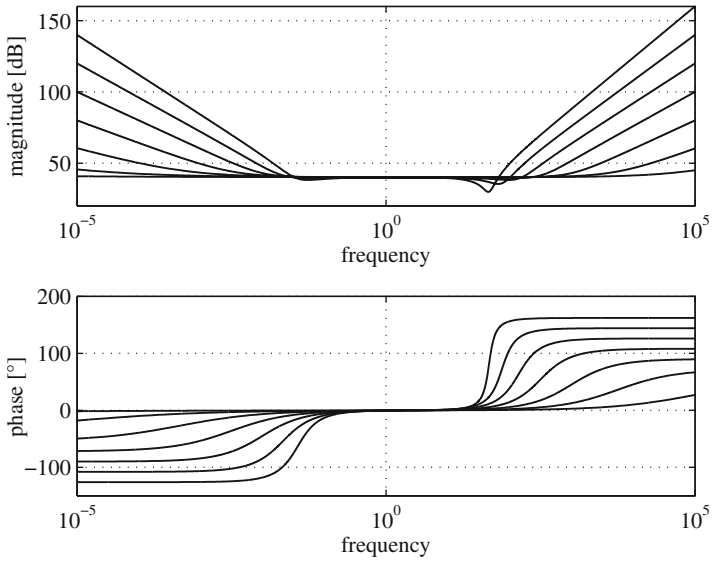

Fig. 3.2 The interacting FOPID bode plot, with $T_i = 100$, $T_d = 0.001$ and $\mu = \lambda + 0.4$ swept into the range [0.6 1.8]

Conversely, when $T_i \ll T_d$ the two controllers have strongly different dynamics, see Figs. 3.3 and 3.4. This is the case when a commutation between parallel and series PID controllers would lead to complex coefficients.

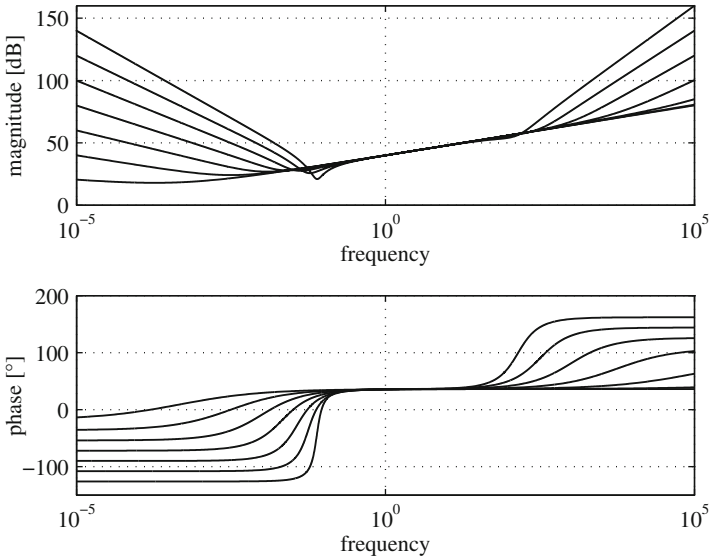

Fig. 3.3 The interacting FOPID bode plot, with $T_i = 0.001$, $T_d = 100$ and $\mu = \lambda + 0.4$ swept into the range [0.6 1.8]

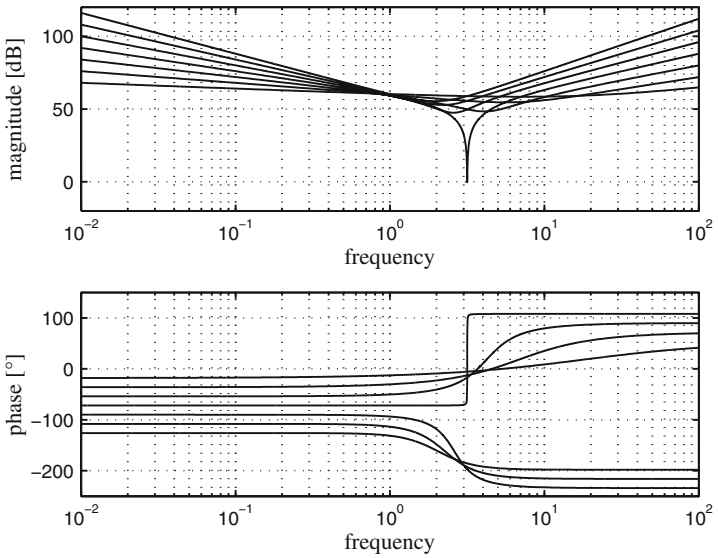

Fig. 3.4 The noninteracting FOPID bode plot, with $T_i = 0.001$, $T_d = 100$, and $\mu = \lambda + 0.4$ swept into the range [0.6 1.8]

3.3 FOPID Tuning

Looking at (3.1) and (3.2) it appears clearly that one of the main extra features that FOPID controllers provide with respect to standard PID controllers is that they allow for a continuous slope regulation of the controller Bode plot both at low and high frequencies, depending, respectively, on λ and μ. This opportunity can be exploited for a more effective loop shaping, but the tuning can be much more complex in practice, in particular because the FOPID transfer function is nonlinear with respect to the coefficients λ and μ.

In order to address this problem, different methods for the design of a FOPID controller have been proposed in the literature. They are based on the use of evolutionary algorithms, where different objective functions are considered [17, 63, 161], or on the solution of a nonlinear constrained optimization problem where, in particular, the iso-damping property (namely, the robustness to variations in the gain of the process) is considered [13, 72, 73].

For example, considering a FOPI controller (namely, avoiding to use the derivative action, i.e., setting $T_d = 0$ in Expressions (3.1) and (3.2)), the three controller parameters K_p, T_i and λ can be selected in order to satisfy three specifications related on the gain crossover frequency ω_{cg}, on the phase margin ϕ_m, and on the flatness of the phase at the gain crossover frequency. Formally, these conditions can be written as

$$\left|C(j\omega_{cg})P(j\omega_{cg})\right|_{\text{dB}} = 0 \text{ dB}, \tag{3.5}$$

$$\arg(C(j\omega_{cg})P(j\omega_{cg})) = -\pi + \phi_m, \tag{3.6}$$

$$\left.\frac{d\arg(C(j\omega)P(j\omega))}{\mathrm{d}\omega}\right|_{\omega=\omega_{cg}} = 0, \tag{3.7}$$

where $P(s)$ is the process transfer function. The application of this method to the process

$$P(s) = \frac{K}{62s+1}e^{-10s}, \tag{3.8}$$

where the nominal value of K is 0.55, is shown in [72]. By setting $\omega_{cg} = 0.02$ rad and $\phi_m = 65°$, and by applying an optimization procedure where Expression (3.5) is minimized subject to constraints (3.6) and (3.7), the following controller is obtained:

$$C(s) = 2.2326\left(1 + \frac{1}{78.4142s^{1.1274}}\right) \tag{3.9}$$

The resulting Bode plot of the loop transfer function $C(s)P(s)$ for $K = 0.55$ is shown in Fig. 3.5 where it appears that the phase is flat around the gain crossover frequency. The robustness of the system with respect to changes in the process gain is shown in Fig. 3.6, where three different set-point step responses are plotted for the values of $K = 0.55$, $K = 0.55 \cdot 2 = 1.10$ and $K = 0.55/2 = 0.275$. It appears the overshoot remains almost the same despite the changes of $\pm 50\,\%$ in the process gain.

It is, however, recognized that for their widespread use in industry, FOPID controllers should possess the same ease of use of standard PID controllers and the

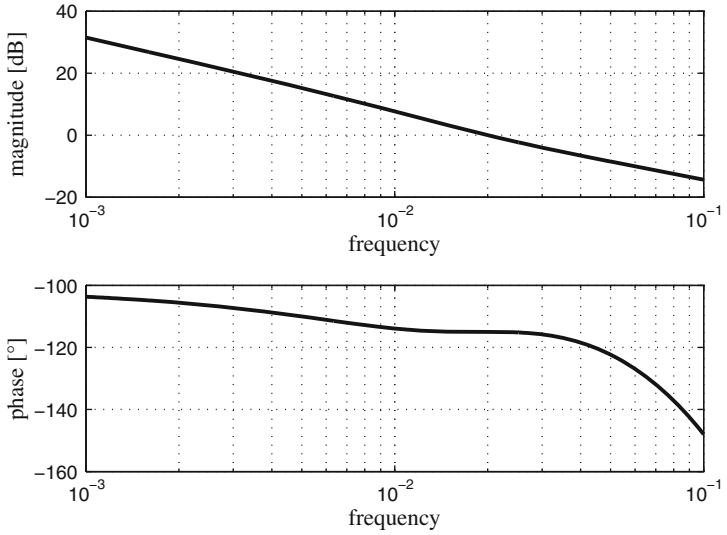

Fig. 3.5 Bode plot of the loop transfer function for the iso-damping tuning

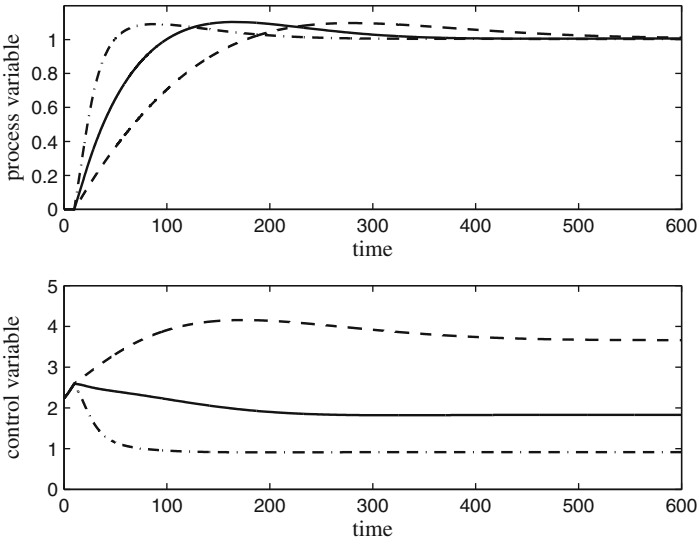

Fig. 3.6 Set-point step responses for different values of the process gain. *Solid line* $K = 0.55$. *Dashed line* $K = 0.275$. *Dash-dot line* $K = 1.10$

improvement in the performance they are capable to provide should be clear. Actually, one of the reasons of the great success of standard PID controllers is the presence of a lot of tuning rules [79] and of the automatic tuning feature [11] that simplifies significantly their design. In [136, 138] tuning rules have been developed in order to achieve the iso-damping property. However, they are not general because they are valid only for first-order-plus-dead-time (FOPDT) processes with specific values of the time constant and of the dead time, and the gain crossover frequency is selected a priori. A more general tuning rule, which is valid for every FOPDT process but just for FOPI controllers, based on the fractional maximum sensitivity constrained integral gain optimization method has been proposed in [30]. Another set of tuning rules has been developed in [42, 43] with the aim of minimizing the effects of low frequency load disturbances.

Indeed, it has to be remarked that effective tuning rules should consider both the set-point following and the load disturbance rejection tasks (the latter is actually often of main concern in the process industry) and different kinds of process dynamics, namely, self-regulating (i.e., asymptotically stable) processes, non-self-regulating (i.e., integral) processes and unstable processes. In fact, this kind of processes is frequently encountered in the process industry. Typical examples of integral processes include tanks, where the level is controlled by manipulating the difference between the input and output flow rates, and batch distillation columns [153], while typical examples of unstable processes are continuous stirred tank reactors, polymerization reactors, and bioreactors [128].

In the next sections, a set of optimization-based tuning rules that address the problem for self-regulating, integral and unstable processes is presented [97, 99].

Robustness is explicitly taken into account by setting a maximum required level of sensitivity.

3.4 Optimal Tuning Rules for Self-Regulating Processes

In this section, a set of tuning rules for PID and FOPID controllers for asymptotically stable processes is given. It is based on the minimization of the integrated absolute error (which is meaningful because this yields, in general, a low overshoot and a low settling time at the same time [130]), subject to a constraint on the maximum sensitivity (as in the well-known Kappa-Tau tuning rules for standard PID controllers [9, 44]) in order to provide a required level of robustness. In fact, the robustness of a control system, which is in general associated to the control effort, is very important in practice. It is shown that, in this context, if the controller is restricted to a PI structure, then the use of a fractional integral action is not useful in improving the performance, and therefore a standard PI controller is the best option. Conversely, the use of a fractional derivative action allows the improvement of the control performance. Both the set-point following and the load disturbance rejection tasks are considered explicitly. An analytical expression of the performance index is also given, and this can be exploited in a performance assessment context. In fact, the performance assessment of a control loop is generally performed by first calculating a performance index based on the available data and then by evaluating the current control performance against a selected benchmark, which represents the desired performance [55, 151].

3.4.1 Problem Formulation

Consider the unity-feedback control scheme of Fig. 3.7 where the process is assumed to have a FOPDT dynamics, namely,

$$P(s) = \frac{K}{Ts+1}e^{-Ls}, \tag{3.10}$$

where, evidently, K is the gain, T is the time constant and L is the dead time. The process dynamics can be conveniently characterized by the normalized dead time

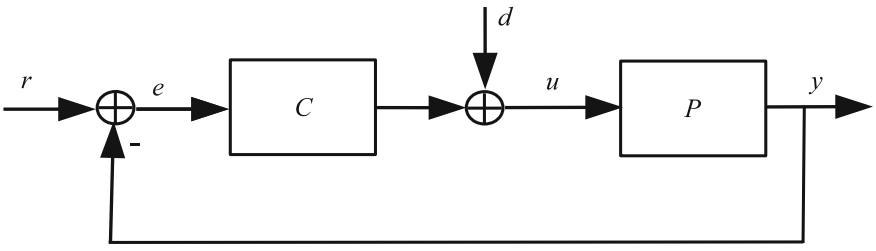

Fig. 3.7 The considered control scheme

defined as L/T or, equivalently, as

$$\tau = \frac{L}{L+T}, \tag{3.11}$$

which represents a measure of the difficulty in controlling the process. Note that, by using (3.11), it results $\tau \in [0, 1]$. Typical values of the normalized dead time are in the range $0.05 \leq \tau \leq 0.8$. Actually, for values of $\tau < 0.05$ the dead time can be virtually neglected and the design of a controller is rather trivial, while for values of $\tau \geq 0.8$ the process is significantly dominated by the dead time, and therefore a dead time compensator should be employed.

The standard PID controller in ideal (noninteracting) form is defined as

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \frac{1}{T_f s + 1} \tag{3.12}$$

where K_p is the proportional gain, T_i is the integral time constant, T_d is the derivative time constant. An output first-order filter, whose time constant is T_f, has been also employed in order to make the controller proper, and to filter the high frequency noise. By generalizing the previous expression, the FOPID controller is defined as

$$C(s) = K_p \left(1 + \frac{1}{T_i s^{\lambda}} + T_d s^{\mu}\right) \frac{1}{T_f s + 1} \tag{3.13}$$

where λ and μ are the non-integer orders of the integral and derivative terms, respectively. The value of the time constant T_f can be conveniently selected as

$$T_f = \min \left\{ \frac{T_i^{\frac{1}{\lambda}}}{10}, \frac{T_d^{\frac{1}{\mu}}}{10} \right\} \tag{3.14}$$

so that the corresponding pole does not influence the controller dynamics significantly, and it does filter the high-frequency noise at the same time (note that this is a typical choice in standard PID controllers [7]). Also note that, even for the FOPID controller, the filter is an integer-order first-order system, because, for the implementation of the fractional-order controller, the Oustaloup continuous integer-order approximation [89] is employed. It consists in using the following approximation based on a recursive distribution of zeros and poles:

$$s^{\nu} \cong k \prod_{n=1}^{N} \frac{1 + \frac{s}{\omega_{z,n}}}{1 + \frac{s}{\omega_{p,n}}}, \quad \nu > 0, \tag{3.15}$$

which is valid in a frequency range $[\omega_l, \omega_h]$ and where the gain k is adjusted so that both sides of (3.15) have the same gain in the mid point of the interval $[\omega_l, \omega_h]$.

Frequencies $\omega_{z,n}$ and $\omega_{p,n}$ are selected as follows. Starting with a zero it is

$$\begin{array}{rll} \omega_{z,1} & = \omega_l \sqrt{\eta} & \\ \omega_{p,n} & = \omega_{z,n}\alpha, & n = 1, \ldots, N \\ \omega_{z,n+1} & = \omega_{p,n}\eta & n = 1, \ldots, N-1, \end{array} \tag{3.16}$$

while starting with a pole it is

$$\begin{array}{rll} \omega_{p,1} & = \omega_l \sqrt{\eta} & \\ \omega_{z,n} & = \omega_{p,n}\alpha, & n = 1, \ldots, N \\ \omega_{p,n+1} & = \omega_{z,n}\eta & n = 1, \ldots, N-1, \end{array} \tag{3.17}$$

where

$$\alpha = \left(\frac{\omega_h}{\omega_l}\right)^{|\nu|/N} \tag{3.18}$$

and

$$\eta = \left(\frac{\omega_h}{\omega_l}\right)^{(1-|\nu|)/N}. \tag{3.19}$$

In order to determine the tuning rules, the value $N = 8$ has been chosen, while ω_l and ω_h have been selected as $0.001\,\omega_c$ and $1{,}000\,\omega_c$ respectively, where ω_c is the gain crossover frequency of the loop transfer function.

The standard PID controller in series (interacting) form is described by the following transfer function:

$$C(s) = K_p \frac{T_i s + 1}{T_i s} \frac{T_d s + 1}{\frac{T_d}{N} s + 1} \tag{3.20}$$

where N is the filter constant. Along the same line, a filter has been added to the fractional-order PID in series form (3.2), leading to

$$C(s) = K_p \frac{T_i s^\lambda + 1}{T_i s^\lambda} \frac{T_d s^\mu + 1}{\frac{T_d}{N} s + 1}. \tag{3.21}$$

Here, $N = 10$ if $T = 1$ has been selected, so that the pole is at frequency $\omega = 10/T_d$, namely, as for the ideal controller, it does not influence the controller dynamics significantly, and it does filter the high-frequency noise at the same time. In all the other cases, the value of N is conveniently modified in order to rigidly shift the Bode magnitude plot along the abscissa axis without changing its shape, (i.e., in order to scale the system step response proportionally to the value of T when the normalized dead time value is kept constant):

$$N = 10T^{(\mu-1)}. \tag{3.22}$$

The specified control requirement is to minimize the (set-point r or load disturbance d) step response integrated absolute error [130]

$$\text{IAE} = \int_0^\infty |e(t)|\mathrm{d}t = \int_0^\infty |r(t) - y(t)|\mathrm{d}t \tag{3.23}$$

because this yields, in general, a low overshoot and a low settling time at the same time in the set-point or load disturbance step response. Obviously, minimizing (3.23) implies that the closed-loop system is stable.

However, aiming at just obtaining the theoretical minimum integrated absolute error that can be achieved for the single-loop system might not be sensible in practical cases because the robustness issue and the control effort have also to be taken into account. For this reason, the devised tuning rules aim at minimizing the integrated absolute errorby constraining at the same time the maximum sensitivity, which is defined as

$$M_s = \max_{\omega\in[0,+\infty)} \frac{1}{1 + C(s)P(s)}, \tag{3.24}$$

and which represents also the inverse of the maximum distance of the Nyquist plot from the critical point $(-1, 0)$. Obviously, the higher the value of M_s is, the less robust is the system to modeling uncertainties.

3.4.2 Optimal Tuning

In order to find the tuning rules for the minimization of the integrated absolute error by constraining the maximum sensitivity value, an approach similar to that employed in [149] has been used. In particular, the set-point following and the load disturbance rejection tasks have been considered separately, and different processes with different values of the normalized dead time have been considered. For each of them, the values of the parameters of both integer and fractional-order PI and PID controllers have been found by means of a genetic algorithm [69] (which are known to provide a global optimum of a problem in a stochastic frame) in order to minimize the integrated absolute error in a step response, by imposing two typical values of the maximum sensitivity, namely, $M_s = 1.4$ and $M_s = 2$ [9]. Note that $M_s = 1.4$ and $M_s = 2.0$ are two different significant cases in the range of suitable values, namely, one where the robustness issue is of primary concern ($M_s = 1.4$) and one where the aggressiveness is more important ($M_s = 2.0$). Finally, for each considered controller, the optimal coefficients found for the different values of L/T or τ, have been interpolated in order to derive suitable tuning rules [9, 11, 129]. In this context, different interpolating functions have been considered [134], by taking into account the aim of providing tuning rules where the resulting performance is scaled by the time constant T. As a result, the following structure for the controller

parameters has been devised for the controllers in ideal form:

$$K_p = \frac{1}{K}\left(a\tau^b + c\right) \tag{3.25}$$

$$T_i = T^{\lambda}\left(a\tau^3 + b\tau^2 + c\tau + d\right) \tag{3.26}$$

$$T_d = T^{\mu}\left(a \cdot \exp(b\tau) + c \cdot \exp(d\tau)\right) \tag{3.27}$$

$$\lambda = 1 \tag{3.28}$$

$$\mu = a\tau^3 + b\tau^2 + c\tau + d \tag{3.29}$$

while the following interpolating functions are used for the PI controller and the controllers in series form:

$$K_p = \frac{1}{K}\left(a\tau^b + c\right), \tag{3.30}$$

$$T_i = T^{\lambda}\left(a\left(\frac{L}{T}\right)^b + c\right), \tag{3.31}$$

$$T_d = T^{\mu}\left(a\left(\frac{L}{T}\right)^b + c\right), \tag{3.32}$$

$$\lambda = 1. \tag{3.33}$$

In this latter case piecewise constant functions have been considered for μ. The different cases are then considered separately in the following subsections.

3.4.3 PI Controller

If a PI controller is considered, (namely, $T_d = 0$), the ideal and series form are equivalent, and therefore just one case has to be considered. As a result of the optimization procedure for fractional-order PI controllers, the value $\lambda = 1$ is obtained for all the cases addressed. This means that, for the problem considered, there is no point in using a fractional-order integrator. The tuning rules obtained by interpolating the results are, therefore, (3.30) and (3.31) with $\lambda = 1$ where the values of the parameters a, b, and c to be employed in the different cases are shown in Tables 3.1 and 3.2. As an example of how the tuning rules have been obtained, the optimal values of K_p for the different values of τ and the corresponding interpolating function has been plotted in Fig. 3.8.

Table 3.1 K_p tuning rule parameters for a PI controller

Control task	$M_s = 1.4$			$M_s = 2.0$		
	a	b	c	a	b	c
Set-point	0.3220	−1.049	−0.2292	0.5261	−1.074	−0.2157
Load disturbance	0.2958	−1.014	−0.2021	0.5327	−1.029	−0.2428

Table 3.2 T_i tuning rule parameters for a PI controller

Control task	$M_s = 1.4$			$M_s = 2.0$		
	a	b	c	a	b	c
Set-point	0.1726	1.156	0.9907	0.4626	0.9252	0.9393
Load disturbance	1.624	0.2269	−0.5556	1.440	0.4825	−0.1019

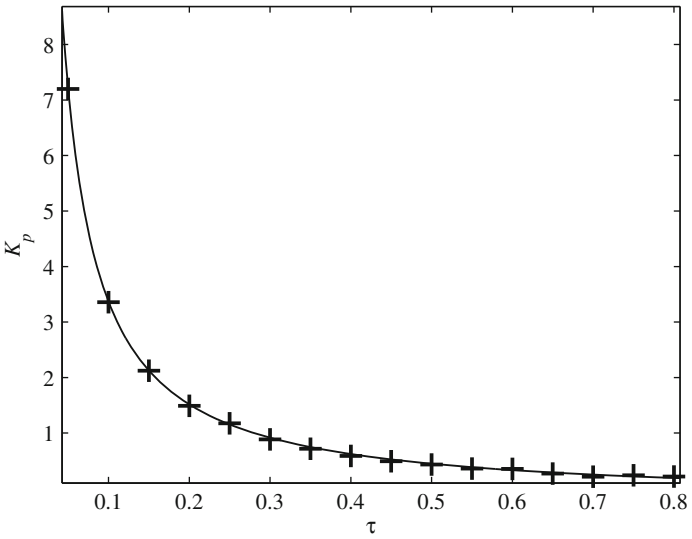

Fig. 3.8 Determination of the K_p tuning rule for a PI controller for set-point following ($M_s = 1.4$). *Plus sign* optimal value of K_p. *Solid line* interpolating function (3.30)

For the purpose of the performance assessment task, it is also useful to determine the optimal value of the integrated absolute error as a function of the process parameters. In fact, the assessment of the performance of a control loop is generally performed by first calculating a performance index based on the available data, and then by evaluating the current control performance against a selected benchmark, which represents the desired performance. The same approach used for the controller parameters have been employed also for the determination of the integrated absolute error function, namely, the optimal IAE values for different values of τ have been interpolated with a suitable function. The desired performance can be,

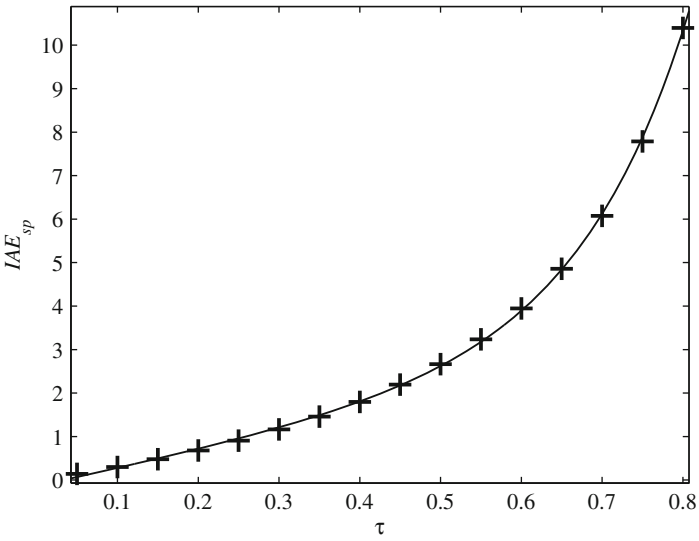

Fig. 3.9 Determination of the performance index function IAE_{sp} for a PI controller for set-point following ($M_s = 1.4$). *Plus sign* optimal value of IAE_{sp}. *Solid line* interpolating function (3.34)

therefore, expressed as

$$\text{IAE}_{sp} = \begin{cases} A_s T \left(0.0020 e^{9.837\tau} + 0.8695 e^{2.270\tau} + 1.158\tau - 0.9188\right) & \text{if } M_s = 1.4; \\ A_s T \left(0.0021 e^{9.046\tau} + 0.5488 e^{2.244\tau} + 1.292\tau - 0.5836\right) & \text{if } M_s = 2.0; \end{cases} \tag{3.34}$$

for the set-point following task, where A_s is the amplitude of the step signal, and

$$\text{IAE}_{ld} = \begin{cases} A_d T \left(0.8220 e^{2.538\tau} + 0.0261 e^{6.453\tau} + 0.0877\tau - 0.9474\right) & \text{if } M_s = 1.4; \\ A_d T \left(0.0234 e^{6.485\tau} + 0.3228 e^{2.474\tau} + 0.5557\tau - 0.4255\right) & \text{if } M_s = 2.0; \end{cases} \tag{3.35}$$

for the load disturbance rejection task, where A_d is the amplitude of the step signal. As an illustrative example, the interpolation for the set-point following case with $M_s = 1.4$ is plotted in Fig. 3.9.

3.4.4 PID Controller

In case the derivative action is employed, the results obtained for integer and fractional-order PID controllers are different. For an integer-order PID controller in series form, the tuning rules obtained for K_p and T_i and T_d are given by expressions

Table 3.3 K_p tuning rule parameters for an integer-order PID controller

Control task	$M_s = 1.4$			$M_s = 2.0$		
	a	b	c	a	b	c
Set-point	0.4290	−1.032	−0.4654	0.6566	−1.076	−0.6705
Load disturbance	0.1724	−1.259	−0.05052	0.2002	−1.414	0.06139

Table 3.4 T_i tuning rule parameters for an integer-order PID controller

Control task	$M_s = 1.4$			$M_s = 2.0$		
	a	b	c	a	b	c
Set-point	0.06949	−0.5366	0.7119	0.1604	−0.1288	0.7814
Load disturbance	0.5968	0.6388	0.07886	0.4460	0.9541	0.1804

Table 3.5 T_d tuning rule parameters for an integer-order PID controller

Control task	$M_s = 1.4$			$M_s = 2.0$		
	a	b	c	a	b	c
Set-point	0.4568	1.153	0.001962	0.4900	1.069	0.001149
Load disturbance	0.5856	0.5004	−0.1109	0.6777	0.4968	−0.1499

(3.30)–(3.32) where, obviously, $\lambda = \mu = 1$ and the values of the parameters a, b, and c for the different cases are shown in Tables 3.3, 3.4 and 3.5. Indeed, as it is always $T_i > 4T_d$, the values of the parameters for a PID controller in ideal form can always be obtained by applying suitable conversion formulae [11]. The optimal values of the integrated absolute error as a function of the process parameters can be expressed as

$$\mathrm{IAE}_{sp} = \begin{cases} A_s T \left(0.0004 e^{12.279\tau} + 0.6107 e^{2.0849\tau} + 1.390\tau - 0.6546\right) & \text{if } M_s = 1.4 \\ A_s T \left(0.0020 e^{9.202\tau} + 0.4256 e^{2.3008\tau} + 0.6196\tau - 0.4405\right) & \text{if } M_s = 2.0 \end{cases} \tag{3.36}$$

for the set-point following task and

$$\mathrm{IAE}_{ld} = \begin{cases} A_d T \left(0.5933 e^{2.587\tau} + 0.0070 e^{8.0967\tau} + 0.0850\tau - 0.6888\right) & \text{if } M_s = 1.4 \\ A_d T \left(0.0059 e^{8.116\tau} + 0.1047 e^{2.8826\tau} + 1.169\tau - 0.1882\right) & \text{if } M_s = 2.0 \end{cases} \tag{3.37}$$

for the load disturbance rejection task.

3.4.5 FOPID Controller

If a fractional-order PID controller (3.13) or (3.21) is considered, the optimal value $\lambda = 1$ is still found (see Sect. 3.4.3), while an optimal value $\mu \neq 1$ is found in

Table 3.6 K_p tuning rule parameters for a FOPID controller in ideal form

Control task	$M_s = 1.4$			$M_s = 2.0$		
	a	b	c	a	b	c
Set-point	0.6189	−0.9695	−0.4648	0.9004	−0.9792	−0.5552
Load disturbance	0.6316	−0.9376	−0.4906	1.024	−0.9978	−0.7035

Table 3.7 T_i tuning rule parameters for a FOPID controller in ideal form

Control task	$M_s = 1.4$				$M_s = 2.0$			
	a	b	c	d	a	b	c	d
Set-point	6.248	−5.930	2.341	0.8237	5.476	−3.913	1.731	0.9057
Load disturbance	6.765	−8.305	4.774	−0.03675	4.138	−3.841	3.631	−0.02886

Table 3.8 T_d tuning rule parameters for a FOPID controller in ideal form

Control task	$M_s = 1.4$				$M_s = 2.0$			
	a	b	c	d	a	b	c	d
Set-point	0.1876	2.927	−0.1867	1.964	0.07963	3.132	−0.08624	−1.608
Load disturbance	0.08677	3.354	−0.08395	0.426	0.05433	3.594	−0.05696	−1.372

Table 3.9 μ tuning rule parameters for a FOPID controller in ideal form

Control task	$M_s = 1.4$				$M_s = 2.0$			
	a	b	c	d	a	b	c	d
Set-point	−0.1360	−0.3489	0.07781	1.199	−0.9618	0.879	−0.2997	1.173
Load disturbance	−0.1285	−0.5261	0.2229	1.124	−0.4019	0.04827	0.02034	1.166

general. In other words, only a fractional derivative action is useful to increase the performance.

For the FOPID controller in ideal form, the structure of the tuning rules is that of (3.25)–(3.29) where the values of the parameters are shown in Tables 3.6, 3.7, 3.8 and 3.9. The optimal IAE values for different values of τ have been also interpolated with a suitable function for the performance assessment purpose. The desired performance can be, therefore, expressed as

$$\text{IAE}_{sp} = \begin{cases} A_s T \left(0.2240 e^{4.205\tau} - 0.2850 e^{-7.940\tau}\right) & \text{if } M_s = 1.4 \\ A_s T \left(0.1747 e^{4.108\tau} - 0.1912 e^{-5.592\tau}\right) & \text{if } M_s = 2.0 \end{cases} \tag{3.38}$$

for the set-point following task and

$$\text{IAE}_{ld} = \begin{cases} A_d K T \left(0.5366 e^{3.430\tau} - 0.5733 e^{1.536\tau}\right) & \text{if } M_s = 1.4 \\ A_d K T \left(919.094 e^{3.25797\tau} - 919.110 e^{3.25750\tau}\right) & \text{if } M_s = 2.0 \end{cases} \tag{3.39}$$

for the load disturbance rejection task.

Table 3.10 K_p tuning rule parameters for a FOPID controller in series form

Control task	$M_s = 1.4$			$M_s = 2.0$		
	a	b	c	a	b	c
Set-point	0.6846	−0.9166	−0.7096	0.9294	−0.9330	−0.9205
Load disturbance	0.2776	−1.095	−0.1426	0.1640	−1.449	0.2108

Table 3.11 T_i tuning rule parameters for a FOPID controller in series form

Control task	$M_s = 1.4$			$M_s = 2.0$		
	a	b	c	a	b	c
Set-point	0.04701	−0.2611	0.9276	−0.001427	−1.003	1.031
Load disturbance	0.6241	0.5573	0.0442	0.6426	0.8069	0.05627

Table 3.12 T_d tuning rule parameters for a FOPID controller in series form

Control task	$M_s = 1.4$			$M_s = 2.0$		
	a	b	c	a	b	c
Set-point	0.3563	1.200	0.0003108	0.4203	1.229	0.01822
Load disturbance	0.4793	0.7469	−0.02393	0.5970	0.5568	−0.09536

Table 3.13 μ tuning rule parameters for a FOPID controller in series form

Control task	$M_s = 1.4$	$M_s = 2.0$
Set-point	1.1 if $\tau < 0.1$	1.0 if $\tau < 0.1$
	1.2 if $\tau \geq 0.1$	1.1 if $0.1 \leq \tau < 0.4$
		1.2 if $\tau \geq 0.4$
Load disturbance	1.0 if $\tau < 0.1$	1.0 if $\tau < 0.2$
	1.1 if $0.1 \leq \tau < 0.4$	1.1 if $0.2 \leq \tau < 0.6$
	1.2 if $\tau \geq 0.4$	1.2 if $\tau \geq 0.6$

Regarding the FOPID controller in series form, the structure (3.30), (3.31) and (3.32), is applied for the tuning rules where the values of a, b, and c are shown in Tables 3.10, 3.11 and 3.12. The tuning rules for μ (note that, once again, it is always $\lambda = 1$) are shown in Table 3.13. For the purpose of performance assessment, the optimal values of the integrated absolute error as a function of the process parameters can be expressed as

$$\text{IAE}_{sp} = \begin{cases} A_s T \left(0.0011 e^{10.414\tau} + 0.6304 e^{2.2226\tau} + 0.7910\tau - 0.6547\right) & \text{if } M_s = 1.4 \\ A_s T \left(0.0017 e^{9.258\tau} + 0.4573 e^{2.2474\tau} + 0.4777\tau - 0.4711\right) & \text{if } M_s = 2.0 \end{cases} \tag{3.40}$$

for the set-point following task and

$$\mathrm{IAE}_{ld} = \begin{cases} A_d T \left(0.6512 e^{2.394\tau} + 0.0031 e^{8.663\tau} + 0.1353\tau - 0.7456\right) & \text{if } M_s = 1.4 \\ A_d T \left(0.0025 e^{8.995\tau} + 0.1188 e^{2.836\tau} + 1.193\tau - 0.2005\right) & \text{if } M_s = 2.0 \end{cases} \tag{3.41}$$

for the load disturbance rejection task.

3.4.6 Comparison

At this point it is worth comparing the performance achievable with optimally tuned FOPID controllers in ideal and series form, with those achievable by using an integer-order PID controller.

The percentage increment/decrement of the IAE values by using an integer-order PID controller and a FOPID controller in series form with respect to a FOPID controller in ideal form are shown in Figs. 3.10 and 3.11 for the set-point following task and in Figs. 3.12 and 3.13 for the load disturbance rejection task, where the cases $M_s = 1.4$ and $M_s = 2.0$ are considered, respectively.

It appears that, in general, there is an advantage in using the FOPID controller in ideal form (with the optimal tuning rules (3.25)–(3.29)), especially for the case of a

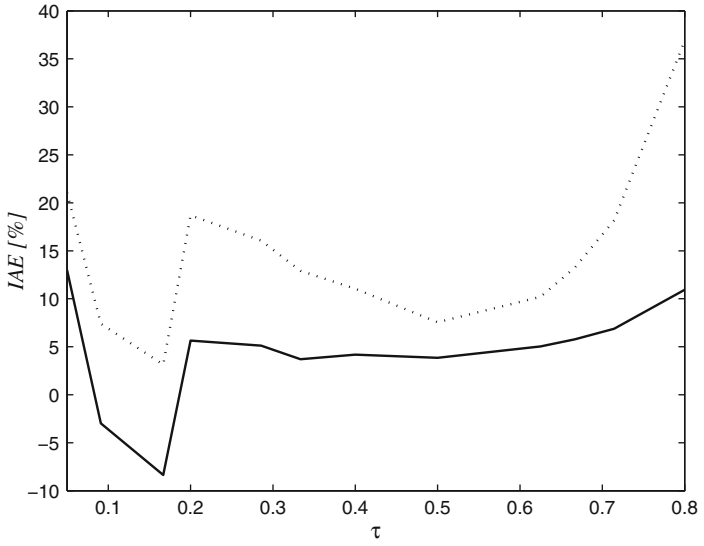

Fig. 3.10 Percentage increment of IAE with respect to ideal FOPID controllers for set-point following tuning with $M_s = 1.4$. *Solid line* series FOPID controllers. *Dotted line* integer-order PID controllers

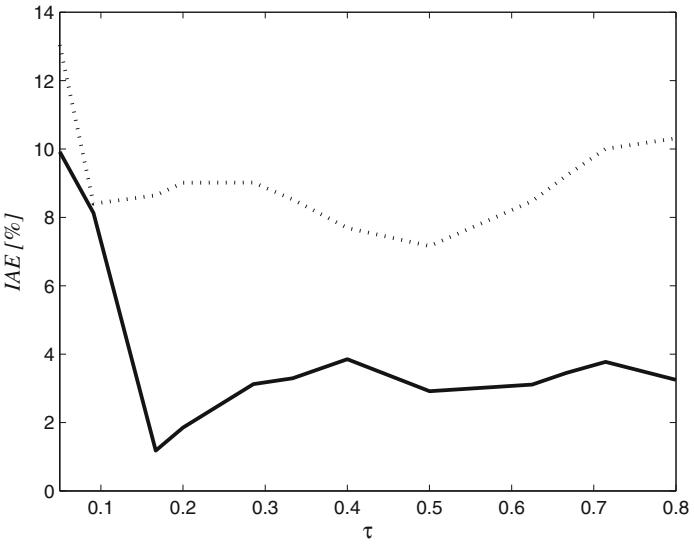

Fig. 3.11 Percentage increment of IAE with respect to ideal FOPID controllers for set-point following tuning with $M_s = 2.0$. *Solid line* series FOPID controllers. *Dotted line* integer-order PID controllers

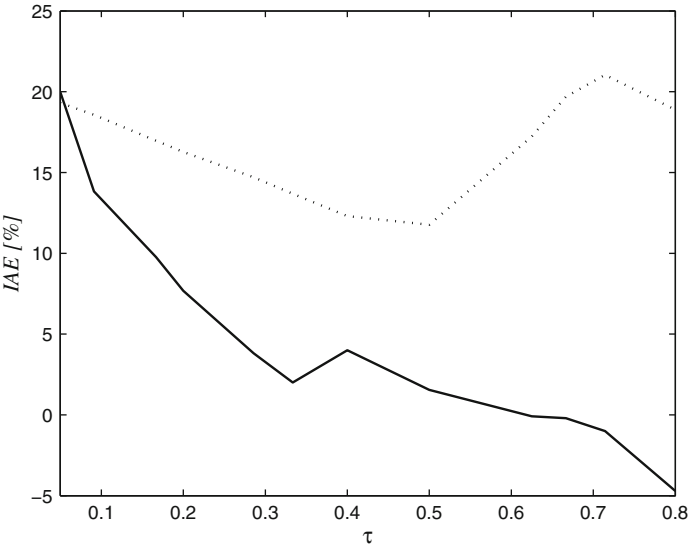

Fig. 3.12 Percentage increment of IAE with respect to ideal FOPID controllers for load disturbance tuning with $M_s = 1.4$. *Solid line* series FOPID controllers. *Dotted line* integer-order PID controllers

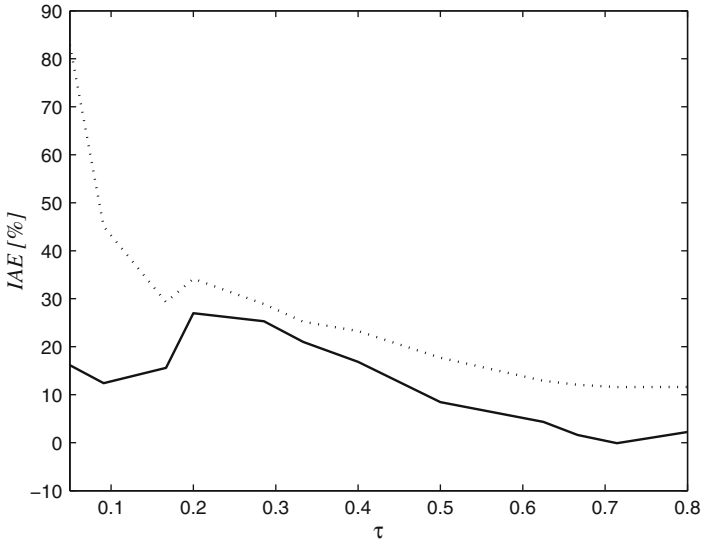

Fig. 3.13 Percentage increment of IAE with respect to ideal FOPID controllers for load disturbance tuning with $M_s = 2.0$. *Solid line* series FOPID controllers. *Dotted line* integer-order PID controllers

load disturbance and $M_s = 2.0$. As it is expected, there is then always a decrement of the IAE by using a FOPID controller with respect to an integer one.

3.4.7 Simulation Results

In order to verify the effectiveness of the proposed tuning rules and to compare the performance achieved by a FOPID controller in ideal or series form with the one achieved by an integer-order PID controller, the following FOPDT process is first considered (note that $\tau = 0.4$):

$$P(s) = \frac{1}{s+1} e^{-0.67s}. \tag{3.42}$$

Simulation results related to both the set-point and load disturbance unit step signals are plotted in Figs. 3.14, 3.15, 3.16 and 3.17, where both cases $M_s = 1.4$ and $M_s = 2.0$ have been considered. The resulting values of the controller parameters, of the integrated absolute errors and the actual maximum sensitivity for the different cases are shown in Table 3.14, where SP is for set-point, LD is for load disturbance, 1.4 and 2.0 denote the required maximum sensitivity and I is for integer-order, FNI is for

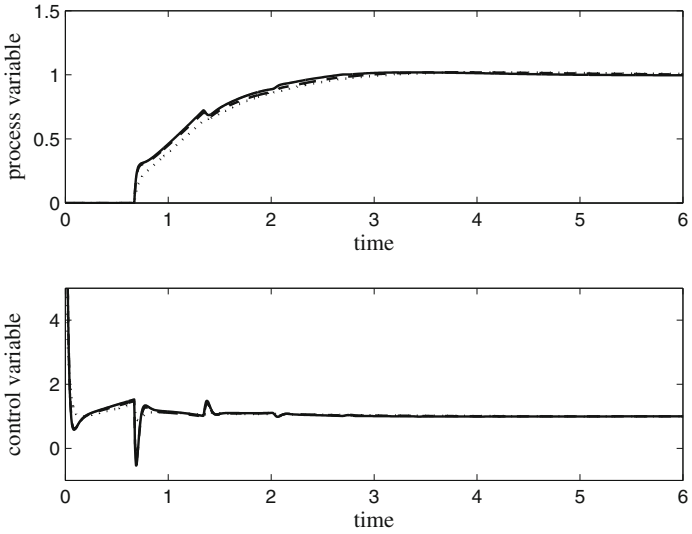

Fig. 3.14 Set-point step response for the FOPDT system and $M_s = 1.4$. *Solid line* FOPID controller in ideal form. *Dashed line* FOPID controller in series form. *Dotted line* integer-order PID controller

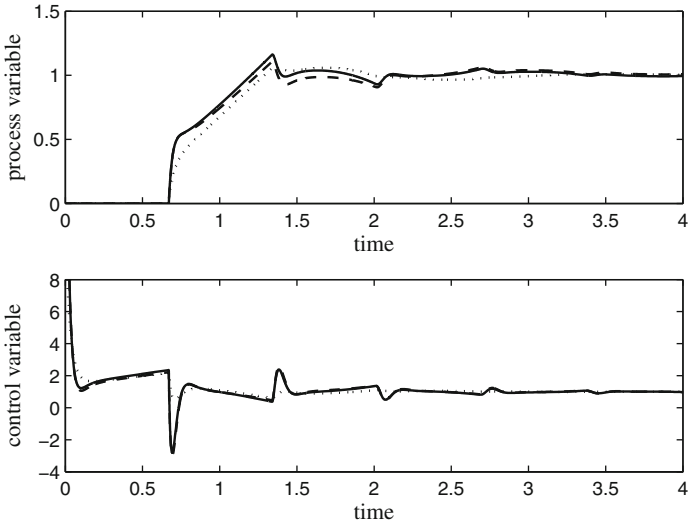

Fig. 3.15 Set-point step response for the FOPDT system and $M_s = 2.0$. *Solid line* FOPID controller in ideal form. *Dashed line* FOPID controller in series form. *Dotted line* integer-order PID controller

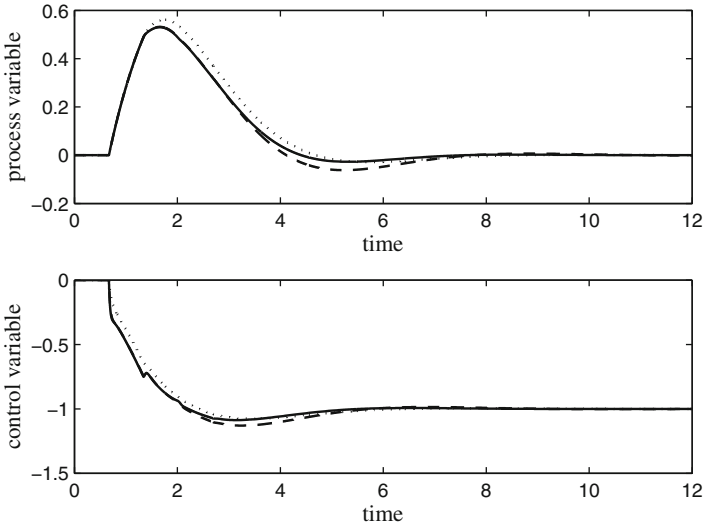

Fig. 3.16 Load disturbance step response for the FOPDT system and $M_s = 1.4$. *Solid line* FOPID controller in ideal form. *Dashed line* FOPID controller in series form. *Dotted line* integer-order PID controller

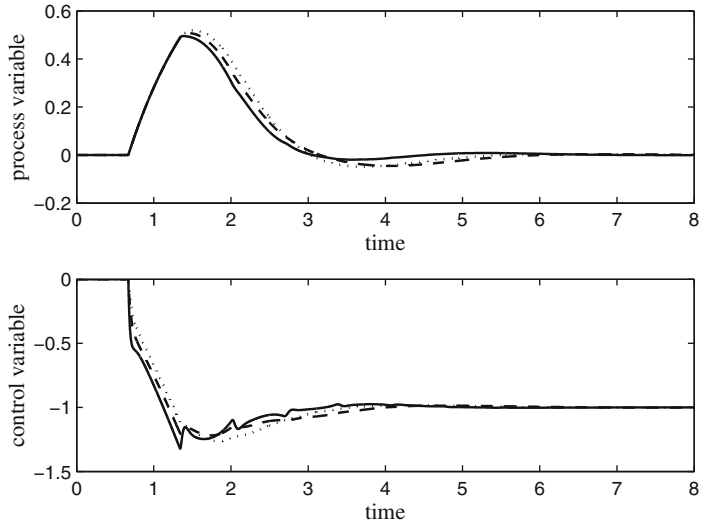

Fig. 3.17 Load disturbance step response for the FOPDT system and $M_s = 2.0$. *Solid line* FOPID controller in ideal form. *Dashed line* FOPID controller in series form. *Dotted line* integer-order PID controller

Table 3.14 Results related to the FOPDT process

Tuning rules	K_p	T_i	λ	T_d	μ	M_s	IAE$_{sp}$	IAE$_{ld}$
SP 1.4 I	0.64	0.80	1	0.29	1	1.42	1.34	1.26
SP 2.0 I	1.09	0.95	1	0.32	1	1.92	0.95	0.87
LD 1.4 I	0.50	0.54	1	0.37	1	1.43	1.42	1.21
LD 2.0 I	0.79	0.48	1	0.40	1	1.98	1.23	0.74
SP 1.4 FNI	1.04	1.21	1	0.20	1.17	1.40	1.22	1.16
SP 2.0 FNI	1.65	1.32	1	0.23	1.13	1.97	0.89	0.80
LD 1.4 FNI	1.00	0.98	1	0.23	1.12	1.40	1.30	1.07
LD 2.0 FNI	1.85	1.07	1	0.20	1.16	2.00	1.07	0.61
SP 1.4 FI	0.83	0.98	1	0.22	1.2	1.42	1.26	1.20
SP 2.0 FI	1.26	1.03	1	0.27	1.2	2.15	0.92	0.83
LD 1.4 FI	0.61	0.54	1	0.33	1.2	1.44	1.38	1.12
LD 2.0 FI	0.91	0.52	1	0.38	1.1	1.95	1.15	0.70

fractional-order in non-interacting form, and FI is for fractional-order in interacting form. It should be noted that the IAE values achieved in the control task (set-point following or load disturbance rejection) different from the one for which the controller has been tuned have also been shown for completeness. However, the controller should be evaluated in relationship with the task for which they have been specifically tuned. It appears that, as it is expected, the FOPID controller in ideal form provides overall the best performance, and the decrement of the IAE value is more significant for the case of load disturbance rejection with $M_s = 2$.

As a second illustrative example, the high-order process

$$P(s) = \frac{1}{(s+1)^8} \tag{3.43}$$

has been considered. The process has been first modeled as a FOPDT process with $K = 1, T = 3.06$, and $L = 4.95$, and then the optimal tuning rules have been applied. Results are shown in Figs. 3.18, 3.19, 3.20 and 3.21 and in Table 3.15. By analyzing them, the same considerations made in the previous example can be derived, thus confirming the effectiveness of the devised tuning rules even in the presence of modeling uncertainties. Note however that the difference between the performance obtained by the two FOPID controllers is not very significant as the high-order dynamics of the process reduces this difference.

More results related to the presented optimal robust tuning techniques can be found in [97].

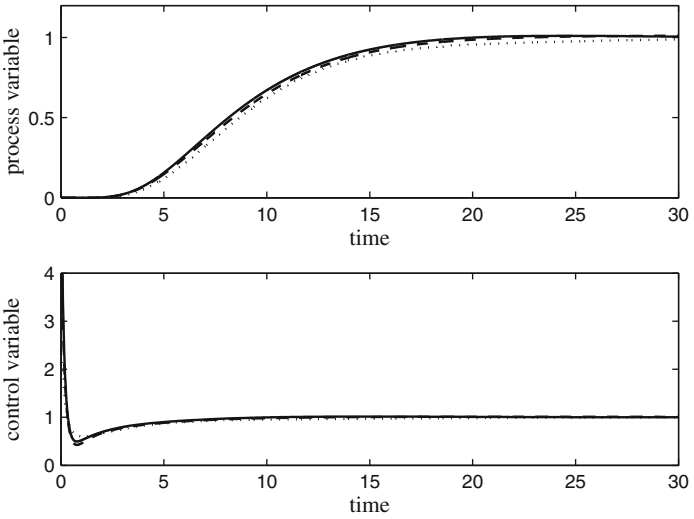

Fig. 3.18 Set-point step response for the high-order system and $M_s = 1.4$. *Solid line* FOPID controller in ideal form. *Dashed line* FOPID controller in series form. *Dotted line* integer-order PID controller

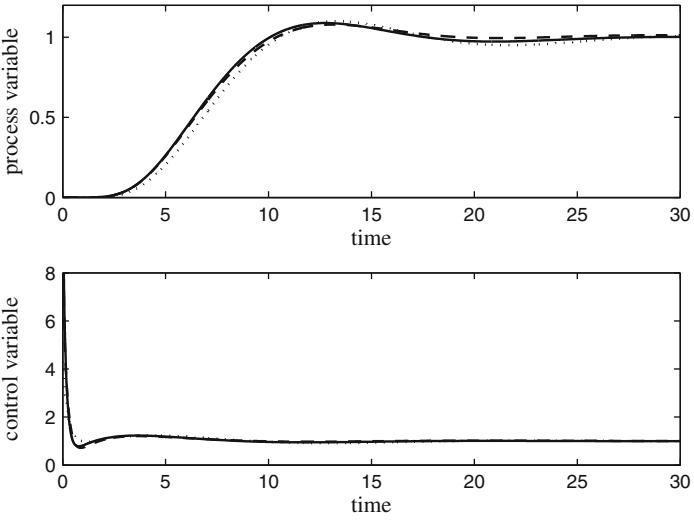

Fig. 3.19 Set-point step response for the high-order system and $M_s = 2.0$. *Solid line* FOPID controller in ideal form. *Dashed line* FOPID controller in series form. *Dotted line* integer-order PID controller

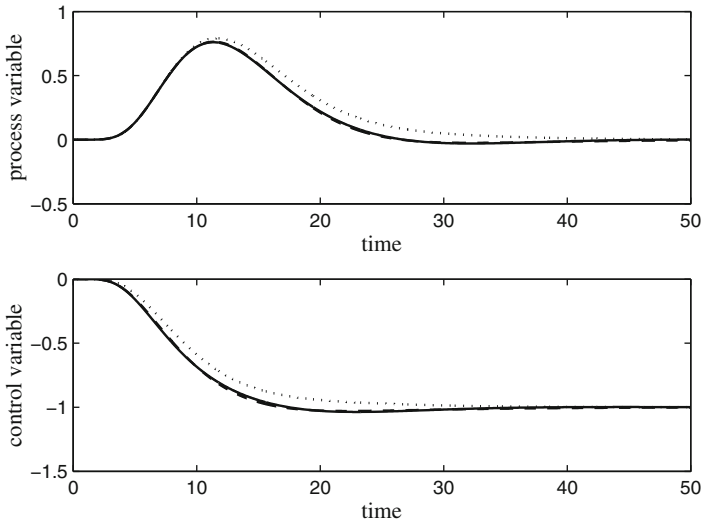

Fig. 3.20 Load disturbance step response for the high-order system and $M_s = 1.4$. *Solid line* FOPID controller in ideal form. *Dashed line* FOPID controller in series form. *Dotted line* integer-order PID controller

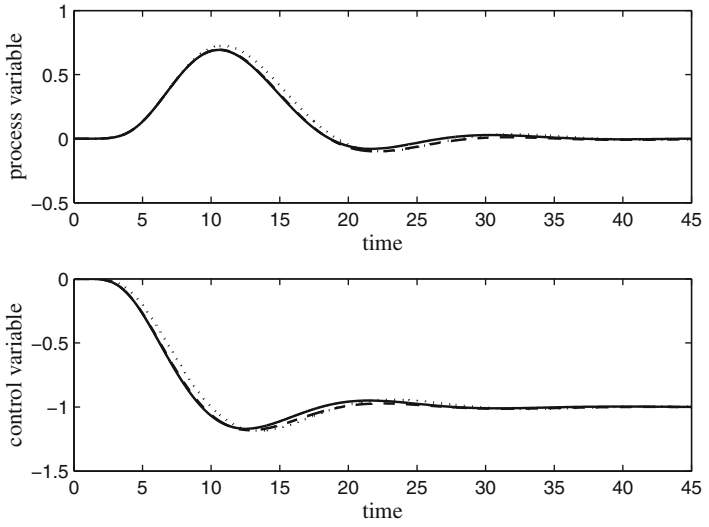

Fig. 3.21 Load disturbance step response for the high-order system and $M_s = 2.0$. *Solid line* FOPID controller in ideal form. *Dashed line* FOPID controller in series form. *Dotted line* integer-order PID controller

Table 3.15 Results related to the high-order process

Tuning rules	K_p	T_i	λ	T_d	μ	M_s	IAE$_{sp}$	IAE$_{ld}$
SP 1.4 I	0.24	2.34	1	2.44	1	1.45	9.77	9.76
SP 2.0 I	0.43	2.85	1	2.51	1	1.94	7.63	6.87
LD 1.4 I	0.27	2.72	1	1.94	1	1.42	10.26	10.25
LD 2.0 I	0.46	2.71	1	2.17	1	2.02	8.25	7.13
SP 1.4 FNI	0.52	1.48	1	0.52	1.08	1.43	8.85	8.82
SP 2.0 FNI	0.89	1.77	1	0.52	1.10	1.76	6.88	6.29
LD 1.4 FNI	0.50	1.34	1	0.58	1.03	1.44	8.92	8.84
LD 2.0 FNI	0.95	1.72	1	0.48	1.10	1.92	7.32	6.34
SP 1.4 FI	0.34	2.96	1	2.43	1.2	1.39	9.25	9.20
SP 2.0 FI	0.54	3.15	1	2.97	1.2	2.08	6.88	6.48
LD 1.4 FI	0.33	2.63	1	2.53	1.2	1.43	8.95	8.81
LD 2.0 FI	0.59	3.07	1	2.62	1.2	2.09	7.28	6.40

3.5 Optimal Tuning Rules for Integral Processes

By following an approach similar to that proposed in the previous section, a new set of tuning rules, based again on the minimization of the integrated absolute error, for PID and FOPID controllers for integrator-plus-dead-time (IPDT) processes is proposed in this section. A peculiar feature of the optimization procedure employed is highlighted. Again, both the set-point following and the load disturbance rejection tasks will be considered explicitly. Further, an analytical expression of the performance index is also given, and this can be exploited in a performance assessment context. It will be shown that, in this context, as for FOPDT self-regulating processes (see Sect. 3.4), in order to improve the performance the use of a fractional integral action is not useful, while it is worth using a fractional derivative action.

3.5.1 Problem Formulation

The unity-feedback control scheme of Fig. 3.7 is considered, where the process is assumed to have an IPDT dynamics, namely,

$$P(s) = \frac{K}{s} e^{-Ls} \tag{3.44}$$

where, evidently, K is the gain and L is the dead time.

As in Sect. 3.4.1, the FOPID controller in ideal form (3.13) and that in series form (3.21) are compared with a standard integer-order PID controller in series form (3.20), which is equivalent to a standard integer-order PID controller in ideal form (3.12) as it always results $T_i > 4T_d$ (analogously to the case of self-regulating processes). The same first-order filters and the same Oustaloup continuous integer-order approximation (3.15) as in Sect. 3.4.1 has been employed. The specified control requirement is again to minimize the (set-point r or load disturbance d) step response integrated absolute error (3.23) under a constraint on the maximum sensitivity (3.24).

3.5.2 Optimal Tuning

Similarly to the case of self-regulating processes, in order to find the tuning rules for the minimization of the integrated absolute error by constraining the maximum sensitivity value, the following approach has been used. First, the set-point following and the load disturbance rejection tasks have been considered separately. Second, the values of the parameters of the FOPID and PID controllers have been found by means of a genetic algorithm [69], which is known to provide a global optimum of a problem in a stochastic frame. The objective function to be minimized is the integrated absolute error in a step response, whereas the maximum sensitivity values $M_s = 1.4$ and $M_s = 2$ are used as constraints [9].

In the context of integral systems, the genetic algorithm can be applied to a normalized process transfer function where $\bar{s} = Ls$, i.e.,

$$\bar{P}(s) = \frac{KL}{\bar{s}} e^{-\bar{s}} \tag{3.45}$$

so that the optimization can be performed just once on the process (3.45) and then the resulting parameters have simply to be scaled by L (note that the gain K can be neglected in the optimization procedure provided that the value of the proportional gain K_p is eventually divided by K).

The tuning rules and the performance indexes obtained in the different cases are reported in the next subsections. It is worth noting that the analytical expressions of the optimal performance indexes can also be used for the assessment of the controller performance.

3.5.2.1 Set-Point Following Task

If only the set-point following task is of concern, the results obtained by applying the optimization procedure show that there is no need of using the (possibly fractional) integral action. Indeed, the pole at the origin of the complex plane in the loop transfer function that ensures a null steady-state error with a constant set-point value is already present in the process, and therefore there is no need to add it in

the controller. The optimal tuning rules obtained by means of the genetic algorithm for the fractional-order PD controller are (note that, if there is no integral action the ideal and series forms of the controller are equivalent)

$$K_p = \frac{a}{KL} \tag{3.46}$$

$$T_d = bL^{\mu} \tag{3.47}$$

and the optimal value of the performance index IAE is

$$\text{IAE}_{opt} = A_s kL \tag{3.48}$$

where the values of the parameters and of μ are shown in Table 3.16 and A_s is the amplitude of the set-point step.

Regarding the integer-order PD tuning rules, the following expressions have been obtained:

$$K_p = \frac{a}{KL} \tag{3.49}$$

$$T_d = bL \tag{3.50}$$

and the optimal value of the performance index IAE is again

$$\text{IAE}_{\text{opt}} = A_s kL \tag{3.51}$$

where the values of the parameters are shown in Table 3.17.

It appears that the presence of the fractional derivative action allows a performance improvement of 17.2 % for $M_s = 1.4$ and of 6.34 % for $M_s = 2.0$.

Table 3.16 Tuning rules and performance index parameters for FOPD controllers for set-point following task for integral processes

M_s	a	b	μ	k
1.4	0.5962	0.3354	1.20	1.80
2.0	0.8699	0.4494	1.15	1.34

Table 3.17 Tuning rules and performance index parameters for PD controllers for set-point following task for integral processes

M_s	a	b	k
1.4	0.4745	0.3300	2.11
2.0	0.7399	0.5061	1.44

3.5.2.2 Load Disturbance Rejection Task

When the (constant) load disturbance rejection task is considered, the (possibly fractional) integral action has to be employed to ensure a null steady-state error. The resulting optimal tuning rules obtained by means of the genetic algorithm for the FOPID controller are

$$K_p = \frac{a}{KL} \tag{3.52}$$

$$T_i = cL^{\lambda} \tag{3.53}$$

$$T_d = bL^{\mu} \tag{3.54}$$

and the optimal value of the performance index IAE is

$$\text{IAE}_{\text{opt}} = A_d k K L^2 \tag{3.55}$$

where the values of the parameters λ and μ are shown in Table 3.18 for the FOPID controller in ideal form and in Table 3.19 for the FOPID controller is series form. Note that in both cases $\lambda = 1$ and $\mu \neq 1$, that is, it is worth using a fractional-order derivative action and an integer-order integral action.

Regarding the integer-order PID tuning rules, the following expressions have been obtained:

$$K_p = \frac{a}{KL} \tag{3.56}$$

$$T_i = cL \tag{3.57}$$

$$T_d = bL \tag{3.58}$$

Table 3.18 Tuning rules and performance index parameters for FOPID controllers in ideal form for load disturbance rejection task for integral processes

M_s	a	b	c	λ	μ	k
1.4	0.5636	0.4170	4.6206	1	1.15	8.75
2.0	1.0357	0.3723	3.1698	1	1.18	3.19

Table 3.19 Tuning rules and performance index parameters for FOPID controllers in series form for load disturbance rejection task for integral processes

M_s	a	b	c	λ	μ	k
1.4	0.5106	0.4489	3.9856	1	1.15	8.79
2.0	0.8015	0.4863	1.9206	1	1.15	3.53

Table 3.20 Tuning rules and performance index parameters for PID controllers in series form for load disturbance rejection task for integral processes

M_s	a	b	c	k
1.4	0.4058	0.5267	3.5035	10.82
2.0	0.6718	0.5099	2.2727	4.13

and the optimal value of the performance index IAE is again

$$\text{IAE}_{\text{opt}} = A_d k K L^2 \tag{3.59}$$

where the values of the parameters are shown in Table 3.20.

In this case the presence of the fractional derivative action allows a performance improvement, with respect to the standard integer-order PID controller, of 19.1 for $M_s = 1.4$ and of 22.7 % for $M_s = 2.0$ for the FOPID controller in ideal form and of 18.8 % for $M_s = 1.4$ and of 14.6 % for $M_s = 2.0$ for the FOPID controller in series form.

3.5.3 Simulation Results

As a first illustrative example, consider the following integral process [149]:

$$P_1(s) = \frac{0.0506}{s} e^{-6s}. \tag{3.60}$$

The tuning rules presented in Sect. 3.5.2 have been applied and the results for the set-point and load disturbance step responses are plotted in Figs. 3.22, 3.23, 3.24 and 3.25 for the different cases. The resulting values of the integrated absolute errors are shown in Table 3.21. Note that the tuning rule employed is described as SP or LD (which means that the set-point following or the load disturbance rejection task is addressed, respectively) followed by the target maximum sensitivity.

It appears that, as expected, the fractional-order PID controller provides a better performance than the integer-order one.

As a second illustrative example, the following high-order integral process is considered:

$$P_2(s) = \frac{1}{s(s+1)^8} e^{-s}. \tag{3.61}$$

In order to apply the tuning rules proposed in Sect. 3.5.2, an IPDT model (3.44) has been estimated with $K = 1$ and $L = 8$. The results obtained for the different cases are summarized in Table 3.22 (note that the obtained value of the maximum

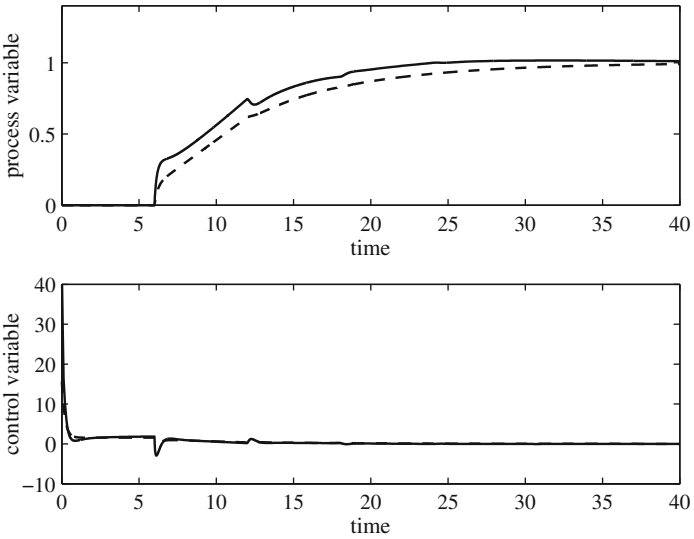

Fig. 3.22 Set-point unit step response for the IPDT process. *Solid line* FOPD with $M_s = 1.4$. *Dashed line* PD with $M_s = 1.4$

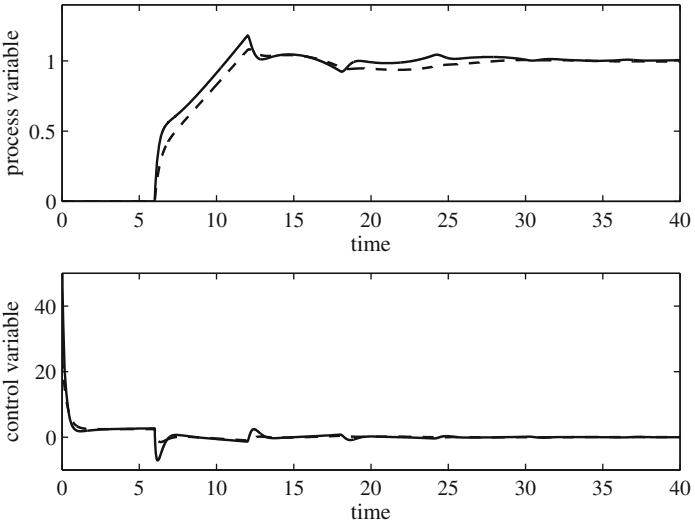

Fig. 3.23 Set-point unit step response for the IPDT process. *Solid line* FOPD with $M_s = 2.0$. *Dashed line* PD with $M_s = 2.0$

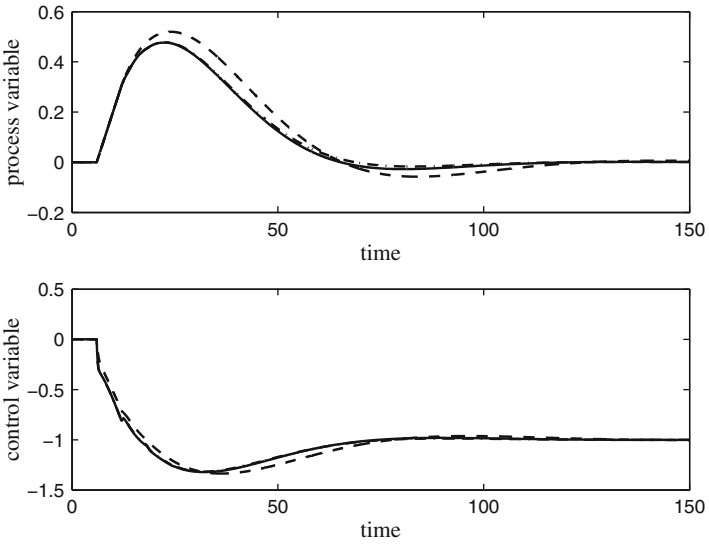

Fig. 3.24 Load disturbance unit step response for the IPDT process. *Solid line* series FOPID with $M_s = 1.4$. *Dash-dot line* ideal FOPID with $M_s = 1.4$ *Dashed line* PID with $M_s = 1.4$

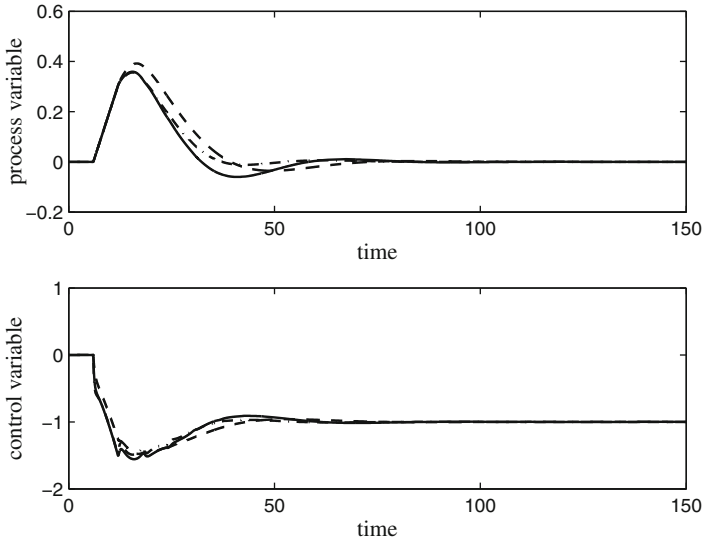

Fig. 3.25 Load disturbance unit step response for the IPDT process. *Solid line* series FOPID with $M_s = 2.0$. *Dash-dot line* ideal FOPID with $M_s = 2.0$. *Dashed line* PID with $M_s = 2.0$

Table 3.21 Results related to the IPDT process

Controller	K_p	T_i	T_d	λ	μ	IAE_{sp}	IAE_{ld}	M_s
PD SP 1.4	1.57	–	1.98	–	–	12.65	∞	1.42
PD SP 2.0	2.44	–	3.04	–	–	8.60	∞	1.98
PID LD 1.4	1.34	21.02	3.16	–	–	23.05	19.68	1.42
PID LD 2.0	2.21	13.64	3.06	–	–	17.06	7.52	2.02
FOPD SP 1.4	1.96	–	2.88	–	1.2	10.74	∞	1.43
FOPD SP 2.0	2.87	–	3.53	–	1.15	8.09	∞	1.99
Series FOPID LD 1.4	1.68	23.91	3.52	1	1.15	20.02	15.92	1.43
Series FOPID LD 2.0	2.64	11.52	3.82	1	1.15	17.06	6.29	2.07
Ideal FOPID LD 1.4	1.86	27.72	3.27	1	1.15	19.61	15.88	1.42
Ideal FOPID LD 2.0	3.41	19.02	3.08	1	1.18	14.72	5.80	2.05

Table 3.22 Results related to the high-order integral process

Controller	K_p	T_i	T_d	λ	μ	IAE_{sp}	IAE_{ld}	M_s
PD SP 1.4	0.06	–	2.64	–	–	16.87	∞	1.42
PD SP 2.0	0.09	–	4.05	–	–	12.26	∞	1.78
PID LD 1.4	0.05	28.03	4.21	–	–	33.36	706.4	1.51
PID LD 2.0	0.08	18.18	4.08	–	–	26.44	286.0	2.19
FOPD SP 1.4	0.07	–	4.07	–	1.2	15.33	∞	1.45
FOPD SP 2.0	0.11	–	4.91	–	1.15	12.06	∞	1.73
Series FOPID LD 1.4	0.06	31.88	4.91	1	1.15	30.10	566.2	1.55
Series FOPID LD 2.0	0.10	15.36	5.31	1	1.15	27.58	261.6	2.49
Ideal FOPID LD 1.4	0.07	36.96	4.56	1	1.15	28.13	560.2	1.54
Ideal FOPID LD 2.0	0.13	25.36	4.33	1	1.18	23.28	229.8	2.32

sensitivity is obviously different from the target one because of the low-order model approximation), while the set-point and load disturbance step responses are shown in Figs. 3.26, 3.27, 3.28, and 3.29 for the different cases.

It appears that the provided tuning rules can address the robustness issue satisfactorily.

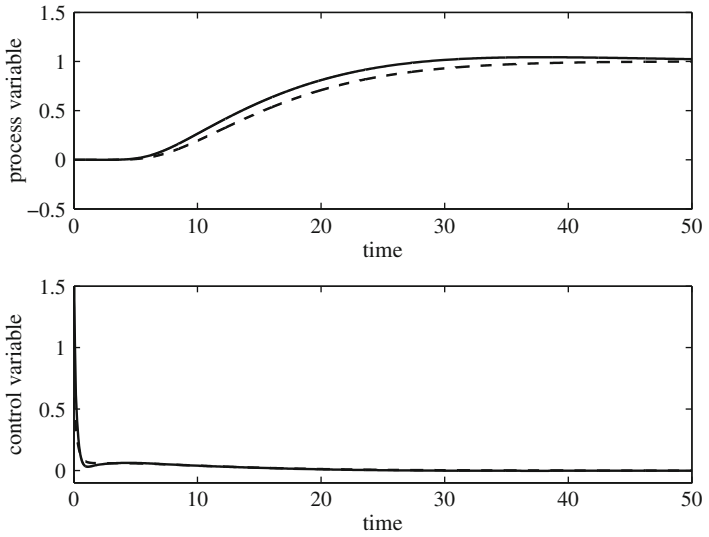

Fig. 3.26 Set-point unit step response for the high-order integral process. *Solid line* FOPD with $M_s = 1.4$. *Dashed line* PD with $M_s = 1.4$

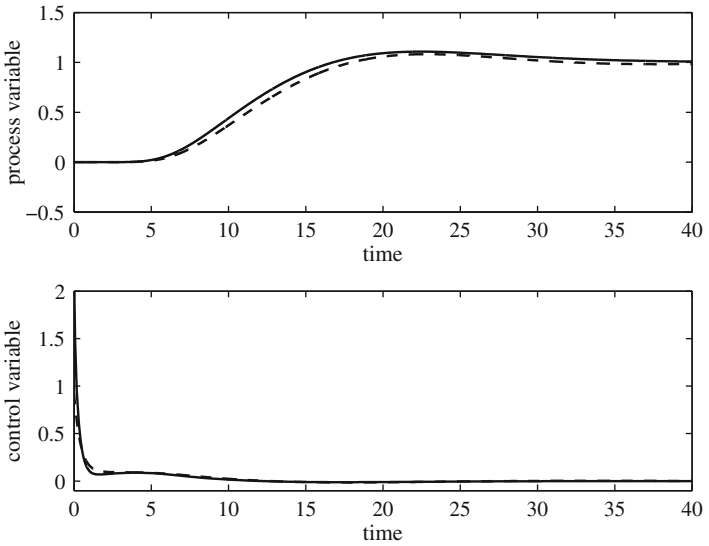

Fig. 3.27 Set-point unit step response for the high-order integral process. *Solid line* FOPD with $M_s = 2.0$. *Dashed line* PD with $M_s = 2.0$

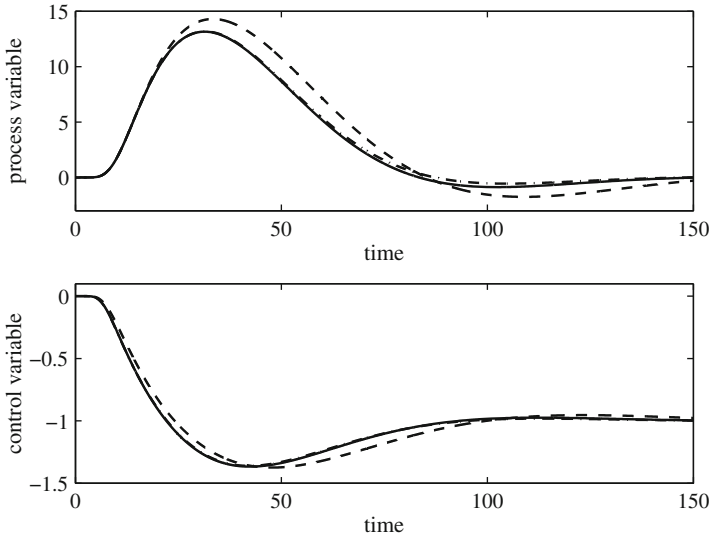

Fig. 3.28 Load disturbance unit step response for the high-order integral process. *Solid line* series FOPID with $M_s = 1.4$. *Dash-dot line* ideal FOPID with $M_s = 1.4$. *Dashed line* PID with $M_s = 1.4$

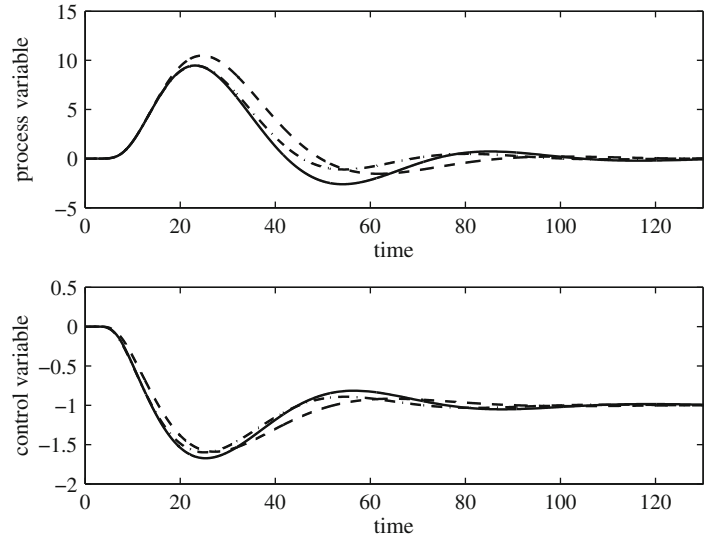

Fig. 3.29 Load disturbance unit step response for the high-order integral process. *Solid line* series FOPID with $M_s = 2.0$. *Dash-dot line* ideal FOPID with $M_s = 2.0$. *Dashed line* PID with $M_s = 2.0$

3.6 Optimal Tuning Rules for Unstable Processes

In this section, the tuning problem is formulated and solved for unstable processes. In particular, the unity-feedback control scheme of Fig. 3.7 is considered, where $P(s)$ is unstable. Once again, optimal tuning rules for both FOPID controllers in ideal and series form as well as for integer-order PID controllers are presented.

3.6.1 Problem Formulation

The process is assumed to have an unstable-first-order-plus-dead-time (UFOPDT) dynamics, namely,

$$P(s) = \frac{K}{Ts - 1} e^{-Ls} \tag{3.62}$$

where K is the gain, T is the time constant and L is the dead time. As for FOPDT models, the process dynamics (3.62) can be conveniently characterized by the normalized dead time defined as L/T. Hereafter unstable processes with $0.05 \leq L/T \leq 1$ are considered. This is a sensible range within the stabilizable range by considering a PID controller [132]. The aim of the tuning rules is always to minimize the integrated absolute error in the set-point and load disturbance step responses (considered separately). However, differently from the previous cases, here a constraint on the maximum sensitivity (3.24) has not been posed. Actually, constraining the maximum sensitivity for an unstable system might prevent the stabilization of the control system and, in this case, minimizing the integrated absolute error yields also a satisfactory robustness.

3.6.2 Optimal Tuning

The optimization procedure employed for unstable processes is more complex than that employed for integral processes. Indeed, in order to find the tuning rules for the minimization of the integrated absolute error an approach similar to that employed in Sect. 3.4 for FOPDT processes has to be used. In particular, the set-point following and the load disturbance rejection tasks have been considered again separately, and different processes with different values of the normalized dead time have been considered. For each of them, the values of the parameters of the FOPID and PID controllers have been found by means of a genetic algorithm and, eventually, for each considered controller, the optimal coefficients found for the different values of L/T have been interpolated in order to derive suitable tuning rules. The obtained performance indexes has been interpolated as well. The tuning rules and the performance indexes obtained in the different cases are reported in the next subsections.

3.6.2.1 Set-Point Following Task

If only the set-point following task is of concern, the optimal tuning rules obtained by means of the genetic algorithm for the FOPID and PID controller are

$$K_p = \frac{1}{K}\left(a\exp\left(-b\frac{L}{T}\right) + c\left(\frac{L}{T}\right)^d\right) \tag{3.63}$$

$$T_i = \left(a\exp\left(b\frac{L}{T}\right) + c\exp\left(d\frac{L}{T}\right)\right)T^\lambda \tag{3.64}$$

$$T_d = \left(a\left(\frac{L}{T}\right)^b + c\right)T^\mu \tag{3.65}$$

where the values of the tuning rule parameters are shown in Tables 3.23 and 3.24 for FOPID controllers in series and ideal form, respectively, and in Table 3.25 for PID controller. Obviously it is $\lambda = \mu = 1$ for the standard integer-order PID controllers. For the FOPID controller, it is

$$\lambda = 1 \tag{3.66}$$

Table 3.23 Tuning rules parameters for FOPID controllers in series form for set-point following task for unstable processes

Parameter	a	b	c	d
K_p	−2.363	0.6377	2.693	−0.6977
T_i	0.4528	3.127	7.359×10^{-16}	25.32
T_d	0.5011	1.303	0.004218	–

Table 3.24 Tuning rules parameters for FOPID controllers in ideal form for set-point following task for unstable processes

Parameter	a	b	c	d
K_p	1.065	0.9063	1.051	−1.088
T_i	0.5659	2.942	−0.6172	−4.655
T_d	0.4884	1.350	0.001938	–

Table 3.25 Tuning rules parameters for PID controllers for set-point following task for unstable processes

Parameter	a	b	c	d
K_p	0.07809	−0.9958	1.035	−0.9305
T_i	6.107×10^{-7}	12.49	0.4247	3.031
T_d	0.5522	1.026	0.006063	–

and

$$\mu = -0.008233\left(\frac{L}{T}\right)^2 - 0.05605\frac{L}{T} + 1.205 \tag{3.67}$$

for the series form, and

$$\mu = 0.1301\left(\frac{L}{T}\right)^2 - 0.1996\frac{L}{T} + 1.216 \tag{3.68}$$

for the ideal form. The optimal value of the performance index IAE can be expressed as

$$\text{IAE}_{\text{opt}} = A_s\left(a\exp\left(b\frac{L}{T}\right) + c\left(\frac{L}{T}\right)^d\right)T \tag{3.69}$$

where the values of the parameters a, b, c and d for FOPID and PID controllers are shown in Table 3.26 and A_s is the amplitude of the set-point step. The unit step response IAE values obtained for different normalized dead times are shown in Fig. 3.30. It appears that the FOPID controller (both in ideal or series form) is capable to provide a better performance than the standard integer-order PID controller. The improvement of the performance for series FOPID is more significant when the normalized dead time of the process increases, as it is shown in Figs. 3.30 and 3.31.

The ideal FOPID controller shows a behavior similar to the series one when the value of the normalized dead time is in the middle of the admissible range. On the contrary, for small values of the normalized dead time it is convenient to use the ideal form, while the opposite is true for big values of the normalized dead time.

It is worth noting that, as for stable FOPDT systems and for IPDT systems the value $\lambda = 1$ results for FOPID controllers, that is, it is just the fractional derivative action that is useful in improving the performance.

3.6.2.2 Load Disturbance Rejection Task

If the load disturbance rejection task is considered, the same expressions (3.63)–(3.65) for the tuning rules can be employed. The related parameters are shown in

Table 3.26 Parameters of the IAE performance index expression for the set-point following task for unstable processes

Controller	a	b	c	d
PID $L/T \leq 0.4$	0.0288	7.1112	3.0542	1.1122
PID $L/T > 0.4$	0.1153	4.6010	2.7518	1.2484
FOPID series $L/T \leq 0.4$	0.0429	5.9277	2.8148	1.1667
FOPID series $L/T > 0.4$	0.3255	3.8420	−5.3195	4.5204
FOPID ideal $L/T \leq 0.4$	0.1137	4.3033	−0.2588	0.2205
FOPID ideal $L/T > 0.4$	4.9722×10^{-6}	13.2717	4.9926	2.7626

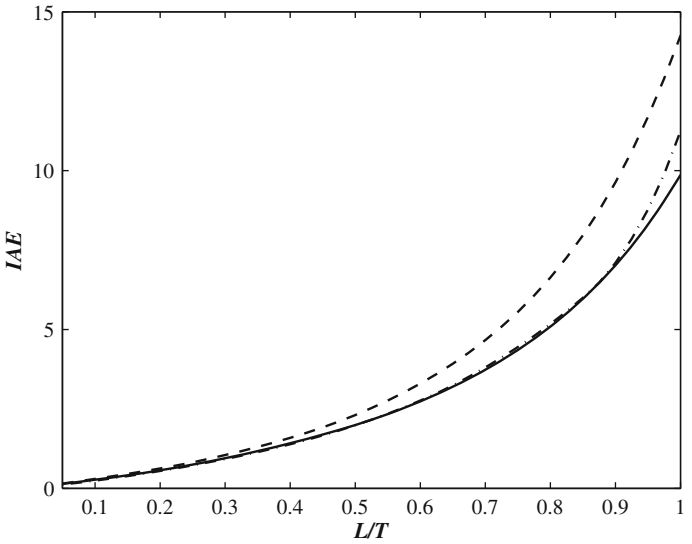

Fig. 3.30 Values of the normalized IAE obtained for the set-point following task for unstable processes. *Solid line* FOPID in series form. *Dash-dot line* FOPID in ideal form. *Dashed line* PID controllers

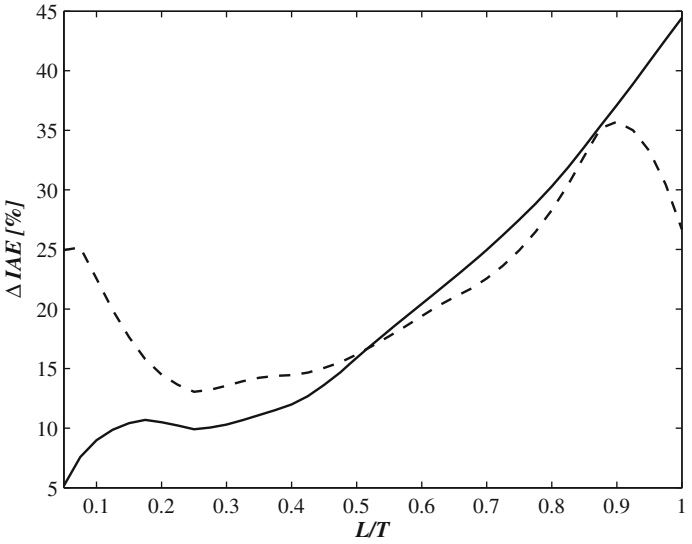

Fig. 3.31 Percentage improvement obtained by the FOPID in series form (*continuous line*) and the FOPID in ideal form (*dashed line*) controllers with respect to the PID controller for the set-point following task for unstable processes

Table 3.27 Tuning rules parameters for FOPID controllers in series form for load disturbance rejection task for unstable processes

Parameter	a	b	c	d
K_p	−0.4346	0	1.947	−0.7747
T_i	0.2736	3.491	−0.5366	−13.33
T_d	0.5038	1.298	0.008506	–

Table 3.28 Tuning rules parameters for FOPID controllers in ideal form for load disturbance rejection task for unstable processes

Parameter	a	b	c	d
K_p	1.602	1.255	1.075	−1.126
T_i	0.5207	2.875	−0.5913	−3.752
T_d	0.4939	1.383	0.002607	–

Table 3.29 Tuning rules parameters for PID controllers in series form for load disturbance rejection task for unstable processes

Parameter	a	b	c	d
K_p	5.418	8.539	1.268	−0.6704
T_i	0.2125	3.758	−0.4673	−16.62
T_d	0.5786	0.9149	0.0006551	–

Tables 3.27, 3.28 and 3.29 for the FOPID and PID controllers respectively. Again, for FOPID controllers it is $\lambda = 1$ while it is useful to employ a fractional derivative action where the derivative order is

$$\mu = -0.04783\left(\frac{L}{T}\right)^2 - 0.01017\frac{L}{T} + 1.218 \tag{3.70}$$

for FOPID controllers in series form, and

$$\mu = 0.06326\left(\frac{L}{T}\right)^2 - 0.1508\frac{L}{T} + 1.245 \tag{3.71}$$

for FOPID controllers in ideal form. The optimal value of the performance index IAE can be expressed as

$$\text{IAE}_{\text{opt}} = A_d\left(a\exp\left(b\frac{L}{T}\right) + c\left(\frac{L}{T}\right)^d\right)TK \tag{3.72}$$

Table 3.30 Parameters of the IAE performance index expression for the load disturbance rejection task for unstable processes

Controller	a	b	c	d
PID $L/T \leq 0.4$	0.0024	11.0222	2.5593	2.0682
PID $L/T > 0.4$	0.0503	6.1840	−14.2533	7.3901
FOPID series $L/T \leq 0.4$	0.0017	11.2262	2.0663	2.0247
FOPID series $L/T > 0.4$	0.0365	6.3840	−15.0042	7.4056
FOPID ideal $L/T \leq 0.4$	7.6922×10^{-4}	10.4503	2.8310	2.2116
FOPID ideal $L/T > 0.4$	8.5461×10^{-6}	12.7409	4.9206	2.7406

where A_d is again the amplitude of the load disturbance step and the values of the parameters a, b, c and d for FOPID and PID controllers are shown in Table 3.30. The unit step response IAE values obtained for different normalized dead times and the improvement of the performance obtained by using the fractional derivative action are plotted in Figs. 3.32 and 3.33, respectively. It appears that in this case the use of the FOPID controller ideal form is more convenient for a large range of the values of the normalized dead time.

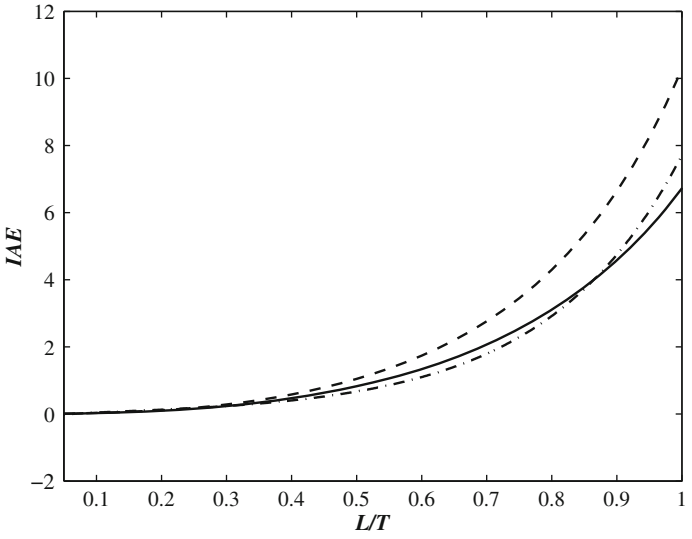

Fig. 3.32 Values of the normalized IAE obtained for the load disturbance rejection task for unstable processes. *Solid line* FOPID in series form. *Dash-dot line* FOPID in ideal form. *Dashed line* PID

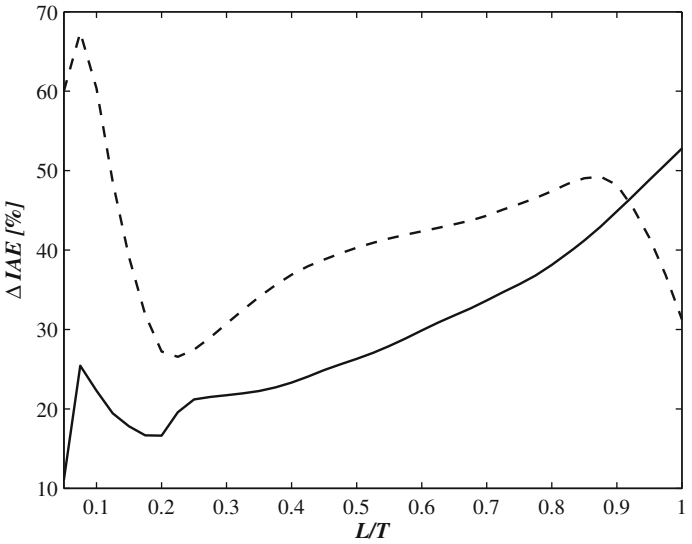

Fig. 3.33 Percentage improvement obtained by the FOPID in series form (*continuous line*) and the FOPID in ideal form (*dashed line*) controllers with respect to the PID controller for the load disturbance rejection task for unstable processes

3.6.3 Simulation Results

As an illustrative example, consider the following unstable process with a small normalized dead time [149]:

$$P(s) = \frac{1}{s-1} e^{-0.2s}. \tag{3.73}$$

The tuning rules presented in Sect. 3.6 have been applied and the results for the different cases are summarized in Table 3.31, while the set-point and load disturbance step responses are shown in Figs. 3.34 and 3.35 respectively, for the different cases.

Table 3.31 Results related to the unstable process with small normalized dead time

Controller	K_p	T_i	T_d	λ	μ	IAE_{sp}	IAE_{ld}
PID SP	4.72	0.78	0.11	–	–	0.63	0.17
PID LD	4.71	0.43	0.13	1	1	0.69	0.11
Series FOPID SP	6.20	0.85	0.07	1	1.19	0.57	0.14
Series FOPID LD	6.34	0.51	0.07	1	1.21	0.70	0.09
Ideal FOPID SP	6.94	0.78	0.06	1	1.18	0.55	0.11
Ideal FOPID LD	7.83	0.65	0.06	1	1.22	0.70	0.09

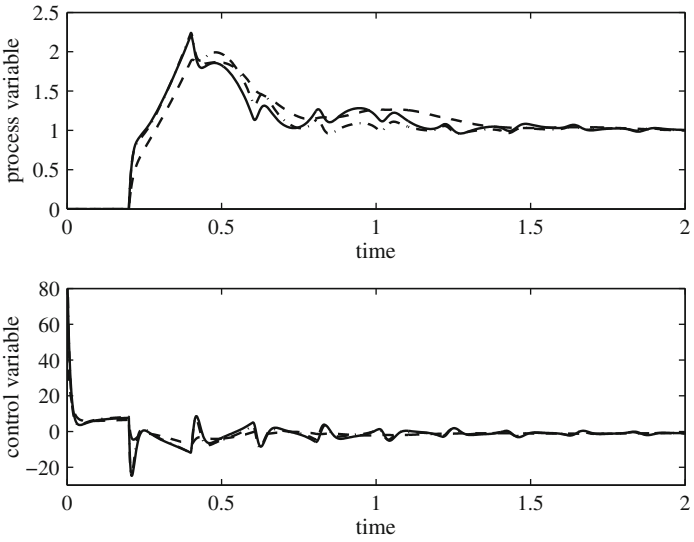

Fig. 3.34 Set-point unit step response for the unstable process with small normalized dead time. *Solid line* FOPID controller in series form. *Dash-dot line* FOPID controller in parallel form. *Dashed line* PID controller

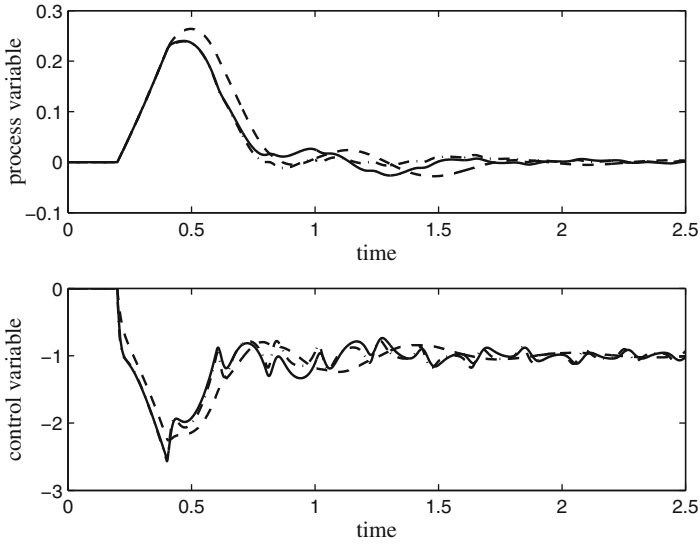

Fig. 3.35 Load disturbance unit step response for the unstable process with small normalized dead time. *Left* set-point unit step response *Right* load disturbance step response. *Solid line* FOPID controller in series form. *Dash-dot line* FOPID controller in parallel form. *Dashed line* PID controller

Table 3.32 Results related to the unstable process with high normalized dead time

Controller	K_p	T_i	T_d	λ	μ	IAE$_{sp}$	IAE$_{ld}$
PID SP	1.25	8.97	0.56	–	–	14.25	11.45
PID LD	1.27	9.11	0.58	–	–	14.05	10.28
Series FOPID SP	1.44	10.32	0.50	1	1.14	9.86	7.34
Series FOPID LD	1.51	8.98	0.51	1	1.16	11.61	6.73
Ideal FOPID SP	1.48	10.72	0.49	1	1.15	11.25	8.64
Ideal FOPID LD	1.53	9.22	0.50	1	1.16	11.27	7.83

It appears that, as expected, the fractional-order PID controllers provides a better performance than the integer-order one.

Finally, as a last illustrative example, consider an unstable process with a greater normalized dead time:

$$P(s) = \frac{1}{s-1}e^{-s}. \tag{3.74}$$

Table 3.32 summarizes the results of the presented tuning methods. The set-point and load disturbance step responses are shown in Figs. 3.36 and 3.37 for the different cases.

Obviously, the peak errors are significant in the different cases because of the large normalized dead time, but in any case the performance provided by the FOPID controllers is much better than that provided by the PID controller.

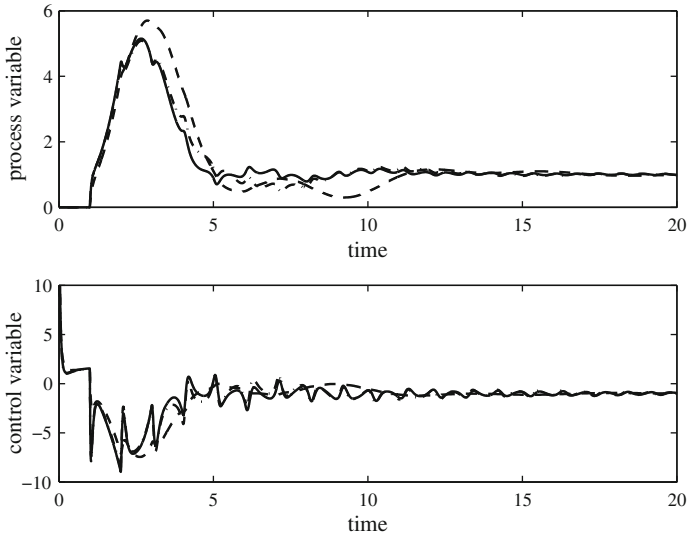

Fig. 3.36 Set-point unit step response for the unstable process with small normalized dead time. *Solid line* FOPID controller in series form. *Dash-dot line* FOPID controller in parallel form. *Dashed line* PID controller

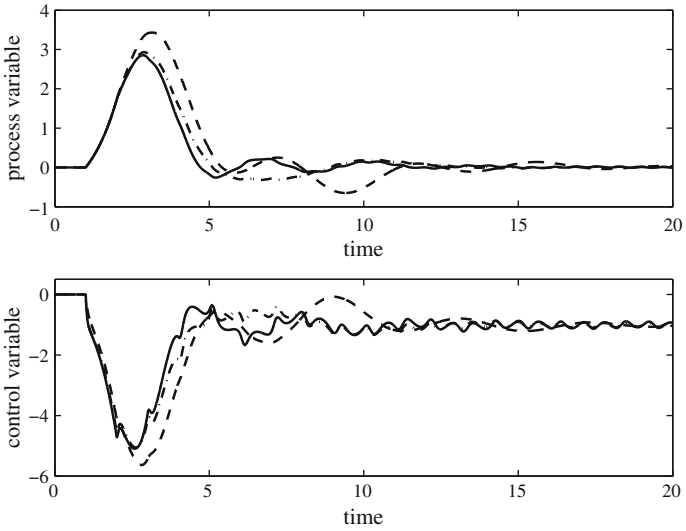

Fig. 3.37 Load disturbance unit step response for the unstable process with small normalized dead time. *Solid line* FOPID controller in series form. *Dash-dot line* FOPID controller in parallel form. *Dashed line* PID controller

3.7 Conclusions

In this chapter a set of tuning rules for fractional-order PID controllers for self-regulating, integral and unstable processes have been presented. In particular, the tuning rules allow the minimization of the integrated absolute error for either the set-point tracking task or the load disturbance rejection task subject to a constraint on the maximum sensitivity (for stable and integral processes).

By comparing the results with those obtained for standard PID controllers (for which tuning rules have been also presented) the improvement of the performance that can be achieved by employing a FOPID controller has been specified quantitatively so that the user can characterize the cost/benefit ratio of such controllers for a given application. Indeed, analytical expressions of the performance index have been provided so that they can be employed effectively for the purpose of performance assessment. More simulation results can be found in [97, 99].

Chapter 4
FOPID Controller Additional Functionalities

4.1 Set-Point Weighting

It has been recognized that, for their widespread use in industry, FOPID controllers should possess the same ease of use and the same capability of dealing with practical issues of standard PID controllers. In particular, they should possess those additional functionalities (anti-windup, feedforward action, set-point weight, derivative filter, etc.) that allow the user to improve the performance in practical cases [151].

In this section, the use of the set-point weighting technique is considered. This is usefully exploited in the context of integer-order PID controllers in order to implement easily a two-degree-of-freedom control scheme, that is, in order to recover the set-point following performance when the controller is tuned in order to compensate the load disturbances as better as possible [11]. It is shown that the method, if properly implemented, is also effective for FOPID controllers and explicit tuning rules are provided for the selection of the optimal set-point weight in order to minimize the integrated absolute error in the set-point step response when the controller parameters are selected, as proposed in the previous chapter, in order to optimize the load disturbance rejection performance [100].

4.1.1 Problem Formulation

As already mentioned in the previous section, set-point weighting is an effective technique for the reduction of the overshoot in the set-point step response when the controller is aggressively tuned (for the purpose of achieving a satisfactory load disturbance rejection performance). In order to introduce the set-point weight for the FOPID controller, it is convenient to extend the procedure which is typically employed for integer-order PID controllers [11, 151].

F. Padula and A. Visioli, *Advances in Robust Fractional Control*,
DOI 10.1007/978-3-319-10930-5_4

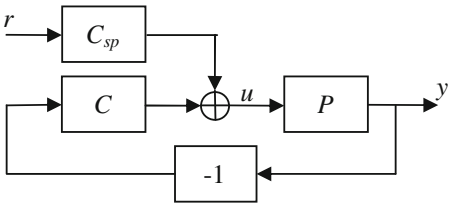

Fig. 4.1 Two-degree-of-freedom FOPID control scheme

In fact, transfer function (3.21) can be expressed as

$$C(s) = K_p \left(1 + \frac{1}{T_i s^{\lambda}} + T_d s^{\mu} + \frac{T_d}{T_i} s^{\mu-\lambda}\right) \frac{1}{T_f s + 1}. \tag{4.1}$$

It can be easily deduced that the term $T_d / T_i s^{\mu-\lambda}$ has to be added to the proportional action only if $\mu = \lambda$. Thus, the use of the set-point weight implies the use of the two-degree-of-freedom control scheme of Fig. 4.1 where $C(s)$ is that of expression (4.1) and

$$C_{sp}(s) = \begin{cases} K_p \left(\beta \left(1 + \frac{T_d}{T_i}\right) + \frac{1}{T_i s^{\lambda}} + T_d s^{\mu}\right) \frac{1}{T_f s+1} \text{ if } \mu = \lambda \\ K_p \left(\beta + \frac{1}{T_i s^{\lambda}} + T_d s^{\mu} + \frac{T_d}{T_i} s^{\mu-\lambda}\right) \frac{1}{T_f s+1} \text{ if } \mu \neq \lambda \end{cases} \tag{4.2}$$

4.1.2 Set-Point Weight Tuning Rules

Tuning rules for the selection of the value of β have been devised by considering the different kind of processes, namely FOPDT, IPDT and UFOPDT, and the corresponding tuning rules presented in Sects. 3.4, 3.5 and 3.6 that minimize the integrated absolute error for the load disturbance step response.

Then, each kind of process (self-regulating, integral, or unstable) has been considered separately. For self-regulating and unstable processes, the value of β that minimizes the integrated absolute error value in the set-point unit step response has been found for different values of the normalized dead time. Eventually, results have been interpolated in order to provide analytical tuning rules. The same procedure has been also applied to the optimally tuned integer-order PID controllers for the sake of comparison.

This procedure is greatly simplified in case of IPDT processes as already highlighted in Sect. 3.5. Indeed, as the optimal controller shape does not depend on L, the set-point weight β does not depend on L as well.

Table 4.1 Parameters of the tuning rules for the set-point weight β

Controller	Process	a	b	c	d
FOPID	FOPDT 1.4	2.3941	1.6256	0.6149	–
	FOPDT 2.0	0.5325	1.9620	0.5116	–
	IPDT 1.4	0.61	–	–	–
	IPDT 2.0	0.41	–	–	–
	UFOPDT	−1.3623	3.0007	−2.1063	0.6876
PID	FOPDT 1.4	2.4767	1.9973	0.7341	–
	FOPDT 2.0	1.1674	4.6515	0.6857	–
	IPDT 1.4	0.57	–	–	–
	IPDT 2.0	0.44	–	–	–
	UFOPDT	−1.129	2.477	−1.874	0.6388

The optimal values of β can be therefore expressed as

$$\beta = a\tau^b + c \tag{4.3}$$

when controlling self-regulating processes, as

$$\beta = a \tag{4.4}$$

when controlling integral processes and as

$$\beta = a\left(\frac{L}{T}\right)^3 + b\left(\frac{L}{T}\right)^2 + c\left(\frac{L}{T}\right) + d \tag{4.5}$$

when controlling unstable processes. The values of a, b, c, and d, depending on the controlled process type and on the required maximum sensitivity (for FOPDT and IPDT processes) are shown in Table 4.1 for both FOPID and PID controllers.Note that, due to the time-scale invariancy property of integrating processes, the optimal set-point weight β in this case is just a number. Also note that, in the case of unstable processes, the optimal set-point weight is parametrized with respect of L/T instead of τ as for FOPDT processes. The reason of this choice lies in the smaller maximum dead time allowable when controlling UFOPDT processes with respect to one allowable when controlling FOPDT processes (with the same time constant). Indeed, the smaller range of admissible values of the dead time can be better (and linearly) parametrized by means of the ratio L/T.

The interpolated function (4.3) for the FOPDT case is plotted in Fig. 4.2, while the decrement of IAE obtained by using the set-point weight in this case is shown in Fig. 4.3 (also for PID controllers). It appears that, for FOPID controllers, with the use of the set-point weight the value of IAE in the step response can be decreased up to 17 % for $M_s = 1.4$ (this decrement is more significant for high values of the normalized dead time) and up to 35 % for $M_s = 2$ (in this case the decrement is

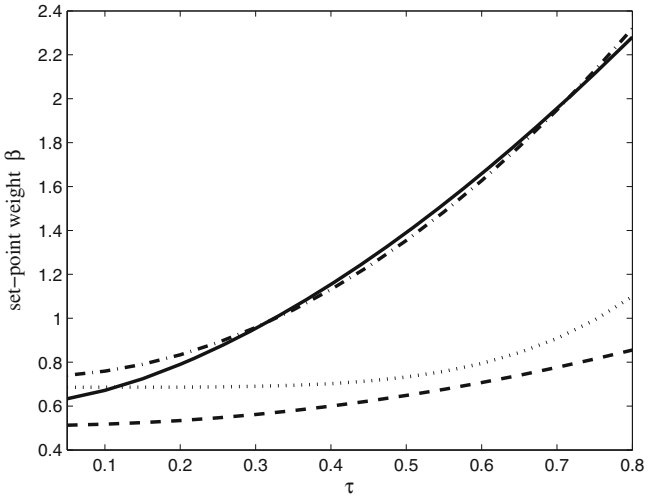

Fig. 4.2 Values of the set-point weight for different values of τ for FOPDT processes. *Solid line* FOPID $M_s = 1.4$. *Dashed line* FOPID $M_s = 2.0$. *Dash-dot line* PID $M_s = 1.4$. *Dotted line* PID $M_s = 2.0$

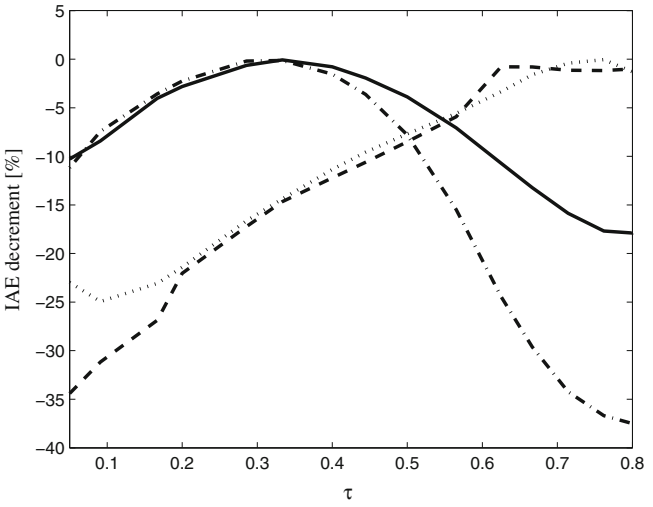

Fig. 4.3 IAE decrement (in percentage) obtained with the set-point weight for FOPDT processes. *Solid line* FOPID $M_s = 1.4$. *Dashed line* FOPID $M_s = 2.0$. *Dash-dot line* PID $M_s = 1.4$. *Dotted line* PID $M_s = 2.0$

Table 4.2 IAE decrement (in percentage) obtained with the set-point weight for IPDT processes

M_s	IAE decrement[%]
F 1.4	21
F 2.0	43
I 1.4	18
I 2.0	38

more significant for low values of the normalized dead time). Similar decrements are obtained in general for standard PID controllers, with the exception of the case of a high normalized dead time for $M_s = 1.4$ where the decrement obtained by PID controllers is higher.

For integral processes, the obtained decrement of IAE is shown in Table 4.2.

Finally, for the case of unstable processes, Figs. 4.4 and 4.5 show again, respectively, the interpolated function (4.5) and the performance improvement in the set-point step response obtained by using the set-point weight. It can be seen that, for FOPID controllers, the value of IAE can be decreased up to 35 % for high values of L/T. A more significant decrement is obtained for PID controllers.

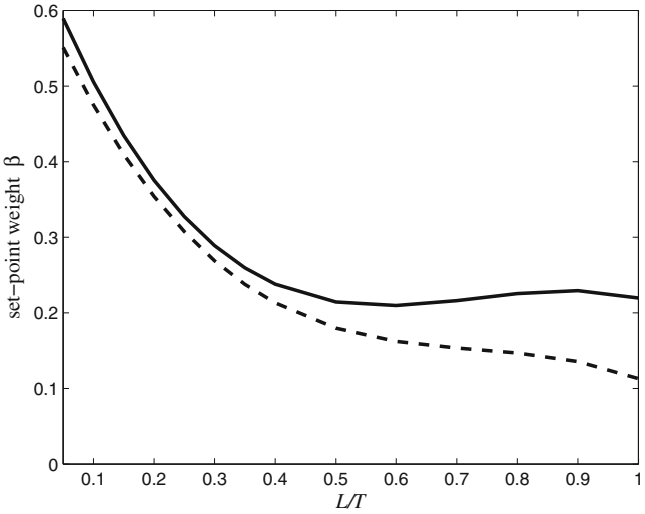

Fig. 4.4 Values of the set-point weight for different values of L/T for UFOPDT processes with FOPID controllers (*continuous line*) and PID controllers (*dashed line*)

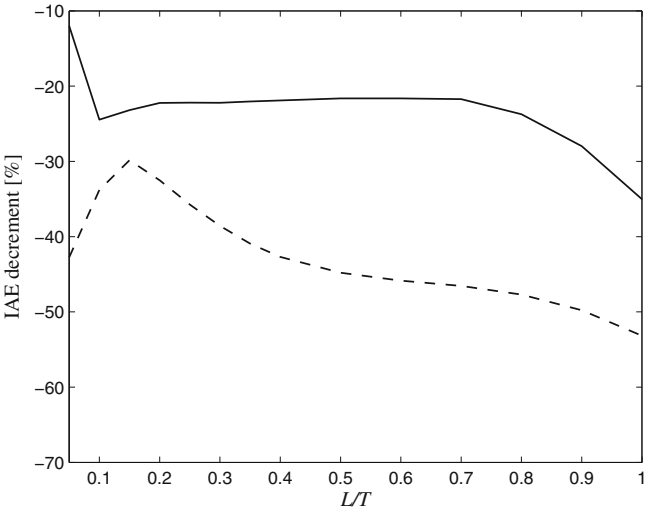

Fig. 4.5 IAE decrement (in percentage) obtained with the set-point weight for UFOPDT processes with FOPID controllers (*continuous line*) and PID controllers (*dashed line*)

4.1.3 Discussion

In order to evaluate the results, it is worth stressing again that the aim of the tuning rules is to minimize the integrated absolute error. In this context, the resulting value of β can be conveniently greater than one for FOPDT processes in order to decrease the rise time at the expense of a larger overshoot [6, 44]. In fact, for $M_s = 1.4$ and for high values of the normalized dead time, the optimal PID parameters provide a somewhat sluggish (robust) response. Obviously, from a practical point of view, if the increment of the set-point weight results in a still too big overshoot (for the given application), the value of β can be easily reduced until the overshoot specification is met.

Conversely, if there are no tight robustness constraints as for $M_s = 2.0$, the minimization of the integrated absolute error in the load disturbance rejection is achieved with an aggressive controller and for this reason the value of the optimal set-point weight is in general less than one as it is essential to reduce the overshoot in the set-point step response in order to minimize the integrated absolute error for this task (with the same controller parameters). By analyzing Fig. 4.3, it can be deduced that the use of the set-point weight when $0.25 \leq \tau \leq 0.4$ for $M_s = 1.4$ and when

$\tau \geq 0.6$ for $M_s = 2.0$ does not provide a significant decrement of the performance because in these cases the trade-off between the overshoot and the rise time in the set-point step response obtained with the optimal (FO)PID parameters and with $\beta = 1$ is such that the resulting integrated absolute error is close to optimality.

Regarding integral and unstable processes, the optimal set-point weight β is always smaller than 1. This happens because large overshoots are likely to occur with this kind of processes.

It has then to be noted that, for FOPDT processes, the performance improvement in the fractional case is similar in general to the integer-order one, and in some cases it is larger for PID controllers than for FOPID controllers. This is due to the fact that FOPID controllers provide already a better performance (without the set-point weight) in the set-point following task case and ,therefore, it is more difficult to obtain an increment in the performance. Conversely, for IPDT processes, despite the fact that FOPID controllers already provide better results without the set-point weight, they are still capable of a more significant increment of the performance. Finally, for UFOPDT processes, the IAE decrement obtained by standard PID controllers is always larger than that obtained by FOPID controllers. This is explained by the fact that for this kind of processes the FOPID (aggressive) tuning obtained to optimize the load disturbance rejection performance is actually more unsuitable for the set-point following task than that employed for PID controllers. In any case the use of the set-point weight for FOPID controllers has been proven to be effective.

4.1.4 Simulation Results

The same processes used in Chap. 3 for the evaluation of the (FO)PID tuning rules are used also here for the evaluation of the set-point weight tuning.

As a first example, consider the self-regulating FOPDT process

$$P(s) = \frac{1}{s+1} e^{-0.67s}. \tag{4.6}$$

By applying the tuning rules for FOPID and PID controllers and for the set-point weight, the results shown in Table 4.3 are obtained, where IAE_β denotes the integrated absolute error obtained by applying the set-point weight. For the different cases, the set-point step responses with and without the use of the set-point weight are plotted in Fig. 4.6. For the case $M_s = 1.4$, the integrated absolute error reduction is not significant (around 1 % for FOPID controller and 2 % for PID controller). This very slight improvement of the performance is expected by considering Fig. 4.3 and by taking into account that the normalized dead time for process (4.6) is $\tau = 0.4$. On the contrary, when $M_s = 2.0$ there is a more sensible IAE reduction of 13 % for the FOPID controller and of 11 % for the PID controller.

As a second example, consider the self-regulating high-order process

Table 4.3 Results obtained with the FOPDT process

Controller	K_p	T_i	λ	T_d	μ	β	IAE_β	IAE
PID 1.4	0.50	0.54	–	0.37	–	1.13	1.39	1.42
PID 2.0	0.79	0.48	–	0.40	–	0.70	1.09	1.23
FOPID 1.4	0.61	0.54	1	0.33	1.2	1.15	1.37	1.38
FOPID 2.0	0.91	0.52	1	0.38	1.1	0.60	1.00	1.15

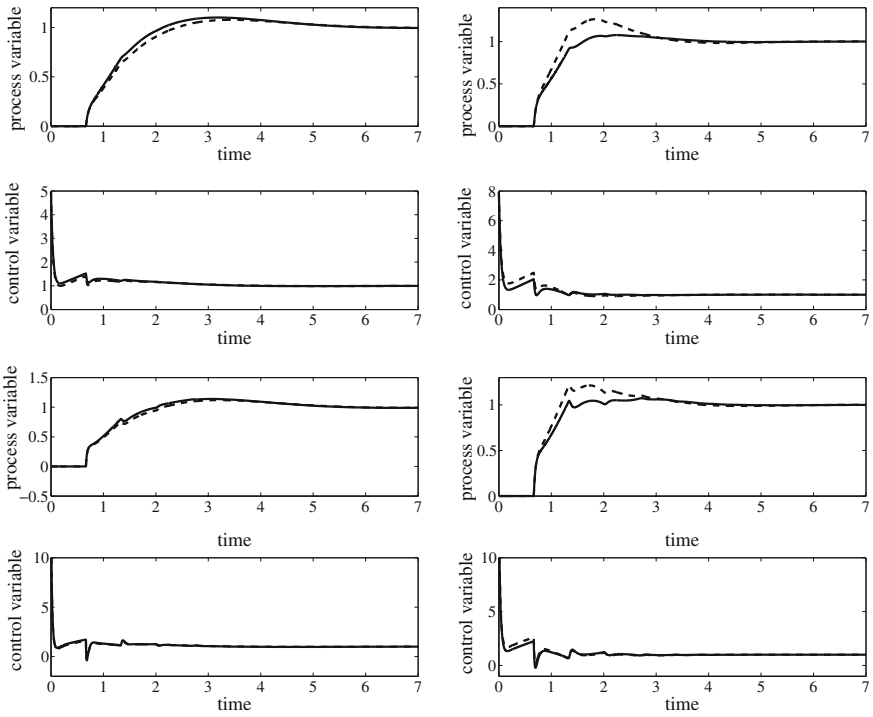

Fig. 4.6 Set-point unit step response for the FOPDT process. *Top-left* PID controller with $M_s = 1.4$. *Top-right* PID controller with $M_s = 2.0$. *Bottom-left* FOPID controller with $M_s = 1.4$. *Bottom-right* FOPID controller with $M_s = 2.0$. *Solid line* with set-point weight. *Dashed line* without set-point weight

$$P(s) = \frac{1}{(s+1)^8}. \tag{4.7}$$

After having approximated it with a FOPDT transfer function with $K = 1, T = 3.06$, and $L = 4.95$ ($\tau = 0.3820$), the devised tuning rules give the results in Table 4.4 for the different controllers and the different values of the desired sensitivity. The set-point step responses with and without the use of the set-point weight are plotted in Fig. 4.7. The integrated absolute error value is decreased of 5.4 % for the FOPID controller and of 20 % for the PID controller with $M_s = 1.4$, while for $M_s = 2.0$, a decrement of 2.3 % for the FOPID controller and of 3.5 % for the PID one is obtained.

Table 4.4 Results obtained with the high-order self-regulating process

Controller	K_p	T_i	λ	T_d	μ	β	IAE_β	IAE
PID 1.4	0.27	2.72	–	1.94	–	1.68	8.20	10.26
PID 2.0	0.46	2.71	–	2.17	–	0.81	7.96	8.25
FOPID 1.4	0.33	2.63	1	2.53	1.2	1.71	8.47	8.95
FOPID 2.0	0.59	3.07	1	2.62	1.2	0.72	7.11	7.28

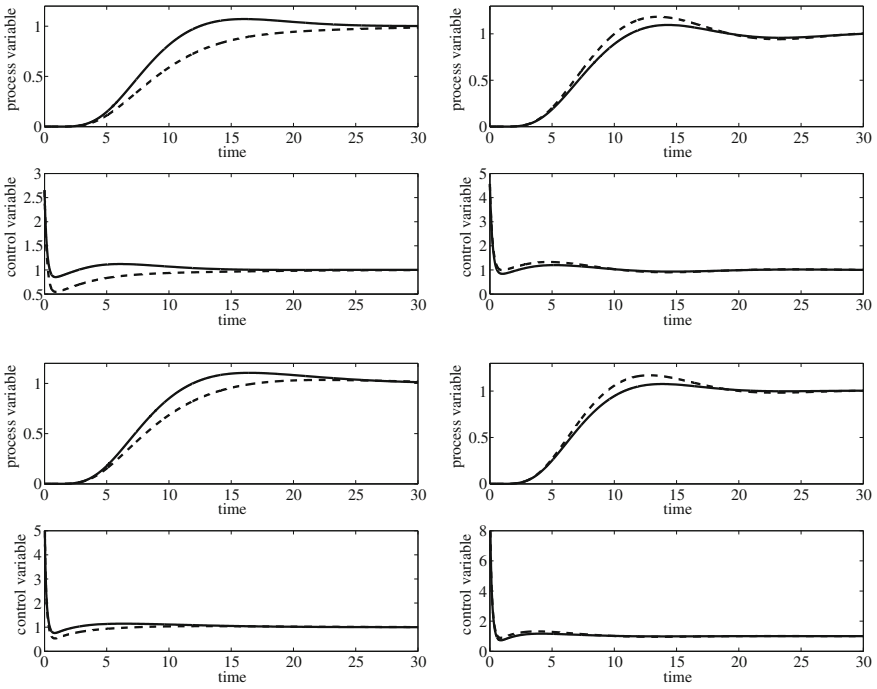

Fig. 4.7 Set-point unit step response for the high-order self-regulating process. *Top-left* PID controller with $M_s = 1.4$. *Top-right* PID controller with $M_s = 2.0$. *Bottom-left* FOPID controller with $M_s = 1.4$. *Bottom-right* FOPID controller with $M_s = 2.0$. *Solid line* with set-point weight. *Dashed line* without set-point weight

It can be seen that the use of the set-point weight is effective also in the presence of modeling uncertainties. Note that the performance improvement is lower when using the FOPID controller because their performance are already satisfactory without the set-point weight.

Table 4.5 Results obtained with the IPDT process

Controller	K_p	T_i	λ	T_d	μ	β	IAE_β	IAE
PID 1.4	1.34	21.02	–	3.16	–	0.57	18.93	23.09
PID 2.0	2.21	13.64	–	3.06	–	0.44	10.42	17.04
FOPID 1.4	1.68	23.91	1	3.52	1.15	0.61	15.76	20.01
FOPID 2.0	2.64	11.52	1	3.81	1.15	0.41	9.65	17.06

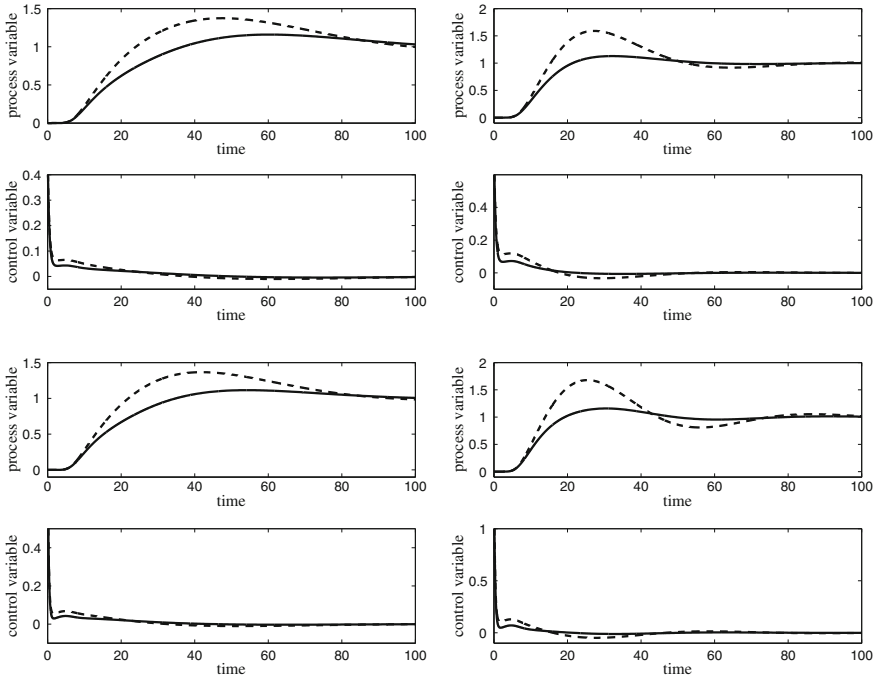

Fig. 4.8 Set-point unit step response for the IPDT process. *Top-left* PID controller with $M_s = 1.4$. *Top-right* PID controller with $M_s = 2.0$. *Bottom-left* FOPID controller with $M_s = 1.4$. *Bottom-right* FOPID controller with $M_s = 2.0$. *Solid line* with set-point weight. *Dashed line* without set-point weight

As a third example, consider the IPDT process [153]

$$P(s) = \frac{0.0506}{s} e^{-6s}. \tag{4.8}$$

In this case, the proposed tuning rules give the results shown in Table 4.5. The set-point step responses with and without the use of the set-point weight are plotted in Fig. 4.8. As expected (see Table 4.2), by using the set-point weight the integrated absolute error value is decreased of 21 % for the FOPID case with $M_s = 1.4$, of 43 %

Table 4.6 Results obtained with the high-order integral process

Controller	K_p	T_i	λ	T_d	μ	β	IAE_β	IAE
PID 1.4	0.05	28.03	–	4.21	–	0.57	26.19	32.93
PID 2.0	0.08	18.18	–	4.08	–	0.44	15.86	25.51
FOPID 1.4	0.06	31.88	1	4.91	1.15	0.61	21.78	28.55
FOPID 2.0	0.10	15.36	1	5.31	1.15	0.41	15.46	27.52

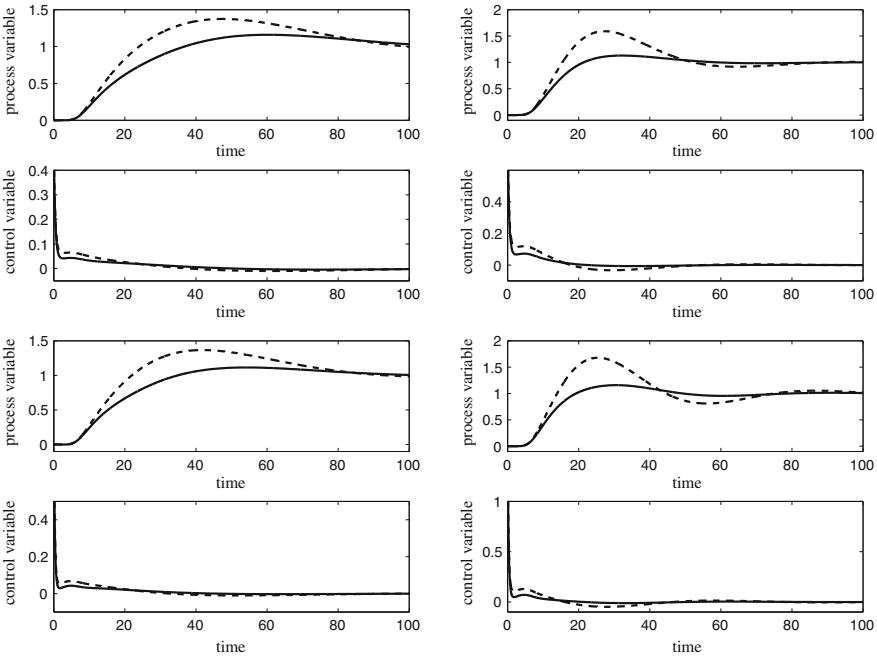

Fig. 4.9 Set-point unit step response for the high-order integral process. *Top-left* PID controller with $M_s = 1.4$. *Top-right* PID controller with $M_s = 2.0$. *Bottom-left* FOPID controller with $M_s = 1.4$. *Bottom-right* FOPID controller with $M_s = 2.0$. *Solid line* with set-point weight. *Dashed line* without set-point weight

for the FOPID case with $M_s = 2.0$, of 18 % for the PID case with $M_s = 1.4$ and of 38 % for the PID case with $M_s = 2.0$.

As a fourth example, consider the high-order integral process

$$P(s) = \frac{1}{s(s+1)^8} e^{-s}, \tag{4.9}$$

which can be approximated with an IPDT transfer function with $K = 1$ and $L = 8$. With these process parameters, the FOPID and PID controller parameters are determined and the corresponding results are shown in Table 4.6. The set-point step responses are plotted in Fig. 4.9. The values of the integrated absolute errors for

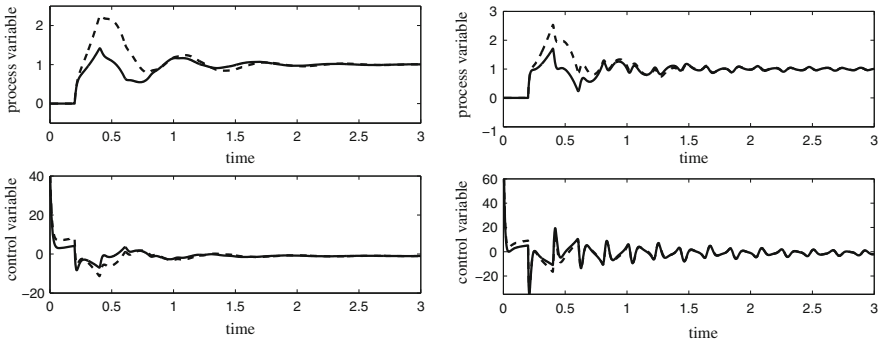

Fig. 4.10 Set-point unit step response for the UFOPDT process with small dead time. *Left* PID controller. *Right* FOPID controller. *Solid line* with set-point weight. *Dashed line* without set-point weight

$M_s = 1.4$ decrease of 24 % when a FOPID controller is considered, while a PID controller gives a performance improvement of 20 %. For $M_s = 2.0$ the performance improvements are of 44 % for the FOPID controller and of 38 % for the PID one. It turns out that, for integral processes, the use of the set-point weight allows the user to increase the set-point following performance significantly also in the presence of modeling uncertainties.

As a fifth example, consider the UFOPDT process with small (normalized) dead time

$$P(s) = \frac{1}{s-1} e^{-0.2s}. \tag{4.10}$$

In this case, the values of the FOPID parameters given by the tuning rules (3.63)–(3.70) are $K_p = 6.34$, $T_i = 0.51$, $\lambda = 1$, $T_d = 0.07$ and $\mu = 1.21$ while the PID parameters are $K_p = 4.71$, $T_i = 0.43$, $T_d = 0.13$. The set-point weight determined by means of (4.5) is $\beta = 0.38$ for the FOPID controller and $\beta = 0.35$ for the PID one. The set-point step responses with and without the use of the set-point weight are plotted in Fig. 4.10. The integrated absolute error value decreases from 0.69 to 0.54, that is, of 22 %when using the FOPID controller and from 0.69 to 0.46 (33 %) when using the PID one, as it could have been derived from Fig. 4.5 (note that $L/T = 0.2$). It is worth noting that, because of the higher order of the fractional derivative, the control effort is increased when using a FOPID controller.

As a last example, consider the following UFOPDT process, with a significant dead time ($L/T = 1$):

$$P(s) = \frac{1}{s-1} e^{-s}. \tag{4.11}$$

In this case, the parameter values of $K_p = 1.51$, $T_i = 8.98$, $\lambda = 1$, $T_d = 0.51$, $\mu = 1.16$, and $\beta = 0.22$ for the FOPID controller and of $K_p = 1.27$, $T_i = 9.11$, and

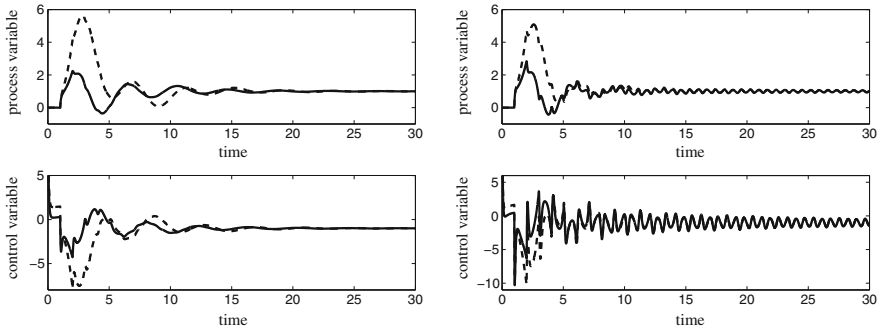

Fig. 4.11 Set-point unit step response for the UFOPDT process with high dead time. *Left* PID controller. *Right* FOPID controller. *Solid line* with set-point weight. *Dashed line* without set-point weight

$T_d = 0.58$ and $\beta = 0.11$ for the PID one are obtained. The set-point step responses with and without the set-point weight are shown in Fig. 4.11. The integrated absolute error decreases from 11.62 to 7.55 (that is, of 35 %) for the FOPID controller and from 14.06 to 6.58 (that is, of 53 %) for the PID controller. Note that the improvement in the performance is more significant, as expected, with respect to the previous example as the normalized dead time is bigger for this example (see, Fig. 4.5).

4.2 Anti-Windup Strategies

Among the additional functionalities that a FOPID controller should possess, the anti-windup ones play a major role. In fact, it is well known that the performance of a PID controller can be severely limited in practical cases by the presence of saturation of the actuators, which causes the integrator windup [51, 52].

To deal with this problem, it should be necessary, from a theoretical point of view, to design the controller explicitly taking into account the actuator constraints from the first stage, e.g., referring to the nonlinear systems framework. However, the overall design becomes much more complicated and ,therefore, quite inappropriate in the typical industrial control context, where the ease of implementation has to be preserved as a major feature. Therefore, the typical method to deal with the integrator windup problem is to tune the controller by ignoring the actuator saturation and subsequently to add an anti-windup compensator to prevent the degradation of the performance.

In this context, several techniques have been devised to design the compensator for a standard PID controller [18, 103]. Basically, they belong to two different approaches, namely, the conditional integration (in which the value of the integrator is frozen when certain conditions are verified) and the back-calculation (in which the difference between the controller output and the actual process input is fed back to

the integral term) [11]. Note that the latter case is comprehensive also of the conditioning technique [47, 155] and that a unified framework of the linear time-invariant anti-windup schemes (including the use of an observer to estimate the correct state of the controller [10, 122]) has been presented in [150]. Indeed, a technique which combines the use of conditional integration and back-calculation has been presented in [150]. Further, a simple anti-windup scheme can be implemented also when the PI controller part is implemented in the automatic reset configuration.

However, despite the remarkable effort for the design of anti-windup schemes for standard PID controllers, this issue has been overlooked for FOPID controllers. In this section, the use of different (standard) anti-windup schemes with FOPID controllers is discuss and compared.

It is worth noting that, although many sophisticated methods have been proposed to solve the general anti-windup problem (see, for example, [160]), this section focuses on those traditionally applied to PID controllers in the industrial framework.

4.2.1 Problem Formulation

A non-interacting FOPID controller (3.1) is considered here. Nevertheless, with respect to the formulation (3.1), the fractional derivative action is applied directly to the process variable rather than to the error variable, so that, in the presence of a set-point step signal, the so-called derivative kick phenomenon is avoided (see, for example, [11]). In this way the saturation of the actuator is not determined by the derivative action in the time instant when the step signal is applied to the set-point. Moreover, a first-order filter whose time constant is ten time faster than the derivative one is applied on the derivative action to obtain a proper transfer function.

4.2.2 Standard Approaches

In this section, anti-windup techniques that are typically employed for integer-order PID controllers are briefly reviewed and applied to fractional-order controllers in order to compare them. In particular, the performance with respect to different values of the fractional order of the integral action will be evaluated.

Consider the control scheme of Fig. 4.12, where u is the controller output, u' is the actual process input, y is the process output, r is the set-point reference value, and

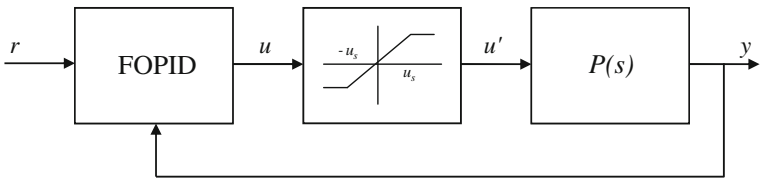

Fig. 4.12 The considered control scheme

$e = r - y$ is the control error. It is worth noting that the integrator windup mainly occurs when a step is applied to the reference set-point signal rather than to the manipulated variable (i.e., in the presence of a load disturbance) [154]. Furthermore, the most significant effects of the integrator windup take place when the process is of low order. For these reasons, hereafter, the analysis will be restricted to the set-point step response of FOPDT systems.

4.2.2.1 Conditional Integration

In the conditional integration approach (also called integrator clamping) the integral term is modified only when certain conditions are satisfied, otherwise it is kept constant. In particular, different strategies can be implemented:

1. the integral term is limited to a selected value;
2. the error for the integral action is set to zero when the system error is large, em i.e., when $|e| > \bar{e}$ where $\bar{e}$ is a selected value;
3. the error for the integral action is set to zero when the controller saturates, em i.e., when $|u| > u_s$ where u_s is the actuator saturation limit;
4. the error for the integral action is set to zero when the controller saturates and the system error and the manipulated variable have the same sign, i.e., when $|u| > u_s$ and $e \cdot u > 0$.

A slightly different approach (called preloading) consists of giving a prescribed value to the integral term when the controller saturates [131].

All these methods have been already compared in the literature [9, 46] and the method 4 has been found to be the best. For this reason the analysis related to the conditional integration will be limited to this case.

4.2.2.2 Automatic Reset Configuration

The PI part of the controller can be implemented in the so-called automatic reset configuration [11, 151]. In this context, a very simple anti-windup technique is that shown in Fig. 4.13, where a saturation block that expresses the true limits of the actuator is placed in the forward part of the controller (similarly, the saturation block can be placed in the feedback part).

4.2.2.3 Back-Calculation

The back-calculation approach consists of recomputing the integral term once the controller saturates. In particular, the integral value is reduced by feeding back the difference of the saturated and unsaturated control signal, as shown in Fig. 4.14 where T_t is an additional parameter called tracking time constant. Formally, denoting by e_i the (fractional) integrator input, it is:

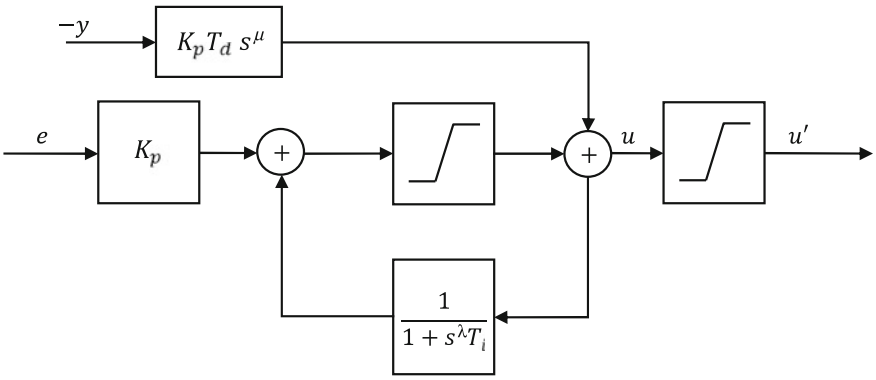

Fig. 4.13 The anti-windup scheme with the PI(D) controller in automatic reset configuration

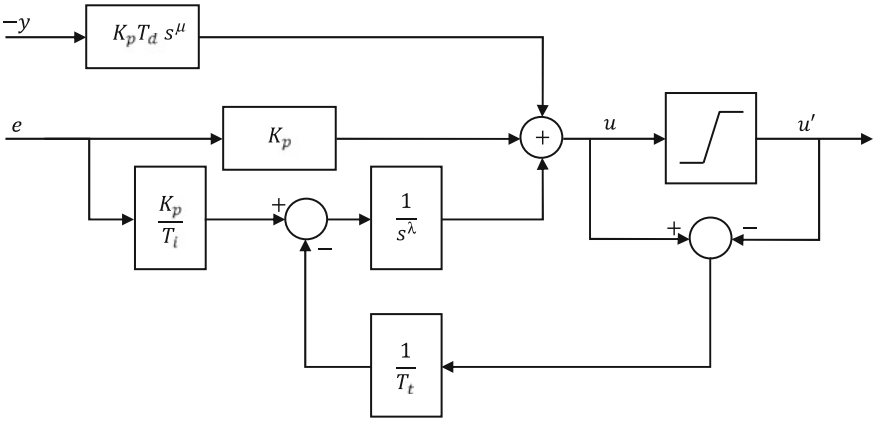

Fig. 4.14 The back-calculation anti-windup scheme

$$e_i = \frac{K_p}{T_i} e + \frac{1}{T_t}(u_s - u). \tag{4.12}$$

The value of T_t determines the rate at which the integral term is reset and its choice determines the performance of the overall control scheme. Some suggestions are to set $T_t = T_i$ [18,103], or $T_t = \sqrt{T_i T_d}$ [9] when the derivative action is employed.

4.2.3 Simulation Results

In order to evaluate and compare the anti-windup techniques for FOPID controllers, a large number of FOPDT processes (see, Sect. 4.2.2) has been considered and a large number of FOPID controllers (em i.e., with different parameters) has been

simulated in order to draw the conclusions about this method. However, for the sake of clarity, hereafter the process with $K = 1$, $T = 1$, and $L = 1$ is considered as a benchmark system, as it is representative of the typical behavior of the different anti-windup techniques. The tuning rules described in the previous chapter (which aims at obtaining a minimum integrated absolute error subject to a maximum sensitivity constrained to the value of $M_s = 2$) have been applied initially, resulting in $K_p = 0.8539$, $T_i = 1.0296$, $T_d = 0.4385$, $\lambda = 1$, and $\mu = 1.2$ (the tracking time constant for the back-calculation technique has been selected as $T_t = \sqrt{T_i T_d} = 0.6719$). A simulation has been performed by considering a unit step signal applied to the set-point at time $t = 0$ and a saturation limit for the control variable $u_s = 1.2$. The resulting process variables, control variables, and integral actions for the different anti-windup schemes are shown in Figs. 4.15, 4.16 and 4.17, respectively.

Then, the order λ of the integral part (which obviously plays a key role in the windup effect) has been increased to $\lambda = 1.3$ (note that reducing the value of λ implies that the windup effect is less significant). The corresponding results are shown in Figs. 4.18, 4.19 and 4.20.

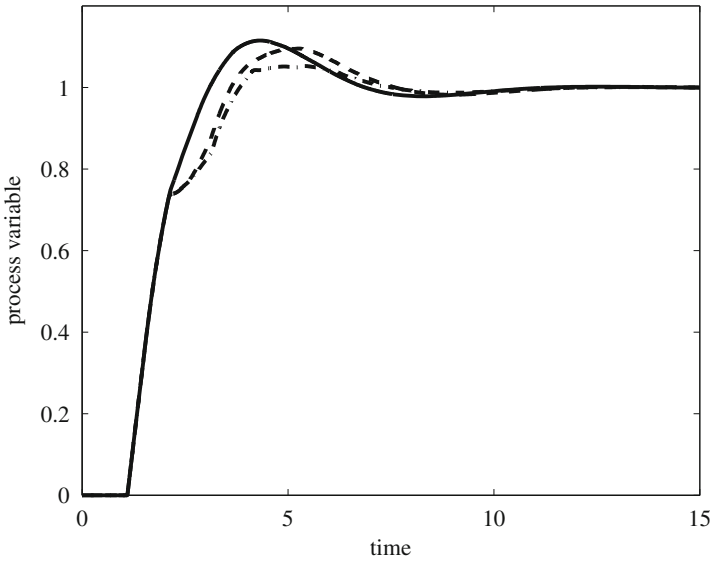

Fig. 4.15 Process variable obtained with $\lambda = 1.0$. *Solid line* back-calculation. *Dashed line* automatic reset. D*ash-dot line* conditional integration

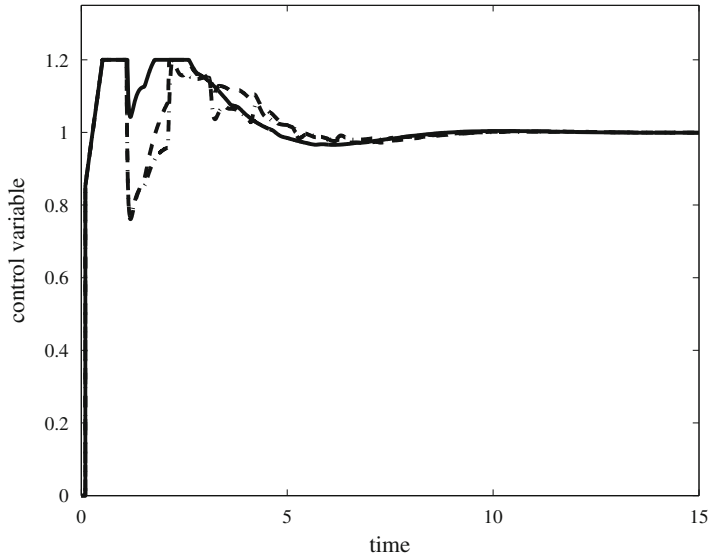

Fig. 4.16 Control variable obtained with $\lambda = 1.0$. *Solid line* back-calculation. *Dashed line* automatic reset. *Dash-dot line* conditional integration

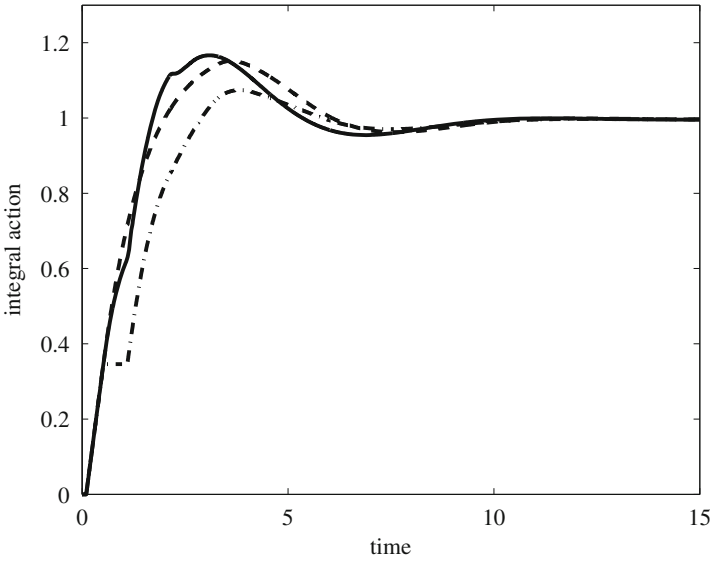

Fig. 4.17 Integral action obtained with $\lambda = 1.0$. *Solid line* back-calculation. *Dashed line* automatic reset. *Dash-dot line* conditional integration

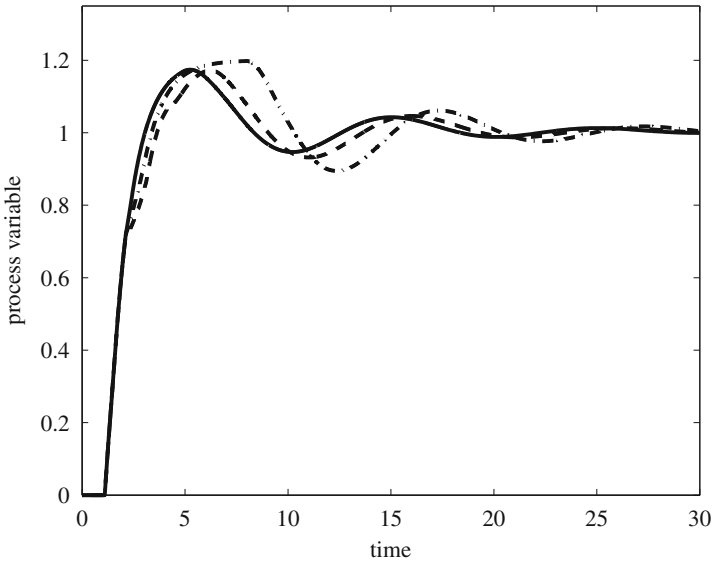

Fig. 4.18 Process variable obtained with $\lambda = 1.3$. *Solid line* back-calculation. *Dashed line* automatic reset. *Dash-dot line* conditional integration

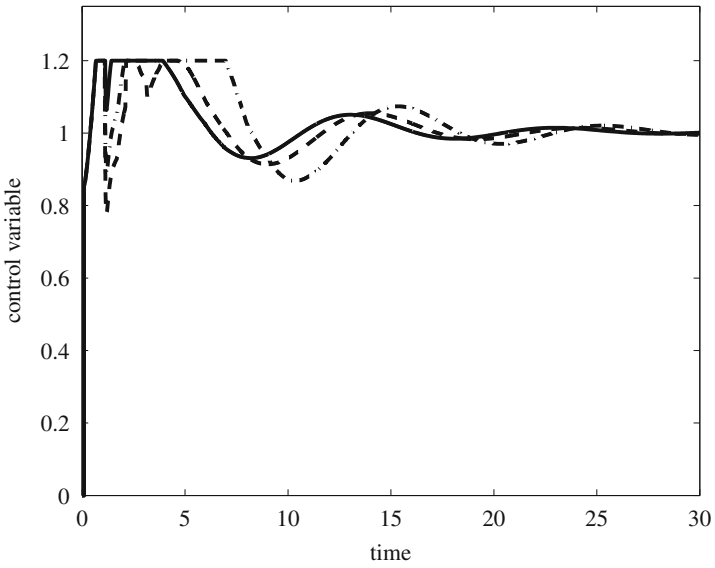

Fig. 4.19 Control variable obtained with $\lambda = 1.3$. *Solid line* back-calculation. *Dashed line* automatic reset. *Dash-dot line* conditional integration

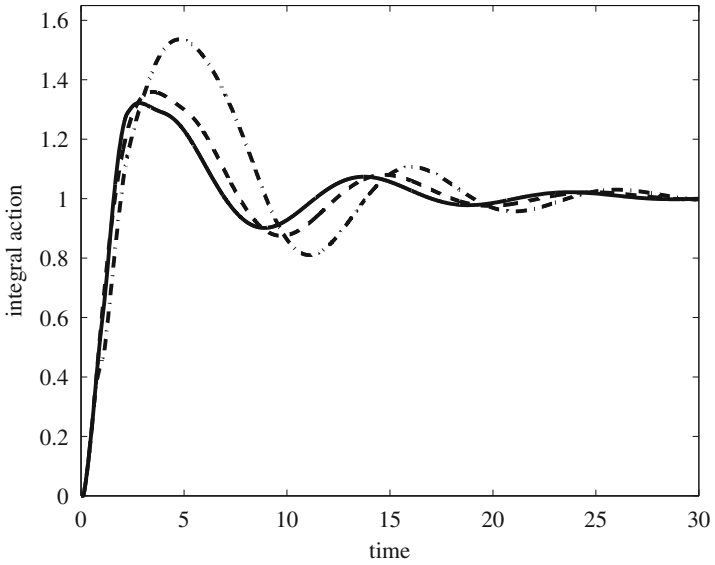

Fig. 4.20 Integral action obtained with $\lambda = 1.3$. *Solid line* back-calculation. *Dashed line* automatic reset. *Dash-dot line* conditional integration

4.2.4 Discussion

From the results obtained it can be deduced that the conditional integration technique is the less appropriate for FOPID controllers. In fact, because of the fractional characteristic of the integrator, the integral action is not frozen when the input error is set to zero (as it is for a standard integer-order integrator). This means that, depending on the order of the integrator, the integral action of the controller can achieve high values during the transient in spite of the anti-windup mechanism. This might therefore yield large overshoots, as it appears in Fig. 4.20.

The best performance is achieved by the back-calculation approach, as the integrator winds down in the same way as it winds up, and, therefore, there are no significant differences in this context between the use of integer-order or fractional-order PID controllers.

An acceptable performance is also achieved by the anti-windup technique for the PI controller part in the automatic reset configuration. Indeed, in that case the integral action is limited to the saturation level of the actuator disregarding the order of the integrator.

Summarizing, it appears that the use of a FOPID controller requires that the anti-windup technique is chosen suitably. In particular, the conditional integration technique should be avoided and it is advisable to use the back-calculation approach.

It is worth noting that, due the infinite memory properties of fractional operators, it is not possible to impose constant integral action by forcing the fractional integrator input to be null. On the other hand, computing a signal such as the integrator output

remains constant requires complex mathematical techniques. Along the same line, it is not possible to easily force a constant integrator output when the actuator saturates and to restart the integration from a suitable initial condition when the actuator no more saturates. Indeed, this would mean that the memory of the previously integrated signal is lost when restarting integration, (i.e., the fractional behavior of the integrator is partially lost).

4.3 Conclusions

In this chapter the use of techniques such as set-point weighting and anti-windup for FOPID controllers has been analyzed.

The use of the set-point weight for FOPID controllers tuned for optimal disturbance rejection performance has been first evaluated. It has been shown that the use of set-point weight allows the user to improve the set-point following performance by conveniently slowing down or speeding up the step response, depending on the achieved level of robustness of the control system and on the process dynamics. The increment of the performance that is achieved has been quantified for the different cases. A comparison with the integer-order PID controllers has been performed. In this context it is worth noting again that, for stable processes, the IAE decrement can be more significant in the integer case, because the FOPID controller, even if tuned by considering just the load disturbance rejection task, provides already a better performance (without the set-point weight) in the set-point following task case and, therefore, it is more difficult to obtain a significant increment in the performance. The quantitative evaluation of the achievable performance allows the user to select the best controller depending on the specific control task, namely, depending on the relevance given, for a particular application, to the set-point following task with respect to the load disturbance rejection one and vice versa.

In the second part of the chapter anti-windup schemes for fractional-order PID controllers have been analyzed. It has been shown that the fractional characteristic of the controller has to be taken into account in the choice of the anti-windup technique. In particular, conditional integration should be avoided while back-calculation seems to provide the best performance.

Chapter 5
$\mathscr{H}_\infty$ Control of Fractional Systems

5.1 Introduction

Despite the large number of results that can be found in the literature, [2, 36, 48, 59, 66, 74, 83, 87, 91, 116, 146] a general analytic solution of the $\mathscr{H}_\infty$ optimization problems involving fractional systems has been missing for a long time, as pointed out in [135, 137], even though methodologies for the computation of the ∞-norm exist [37, 77, 126] and the $\mathscr{H}_\infty$ concept has been recently exploited for the design of fractional proportional-derivative controllers [156]. Only recently, fractional control has been treated analytically under the point of view of $\mathscr{H}_\infty$ optimal control [93].

Linear $\mathscr{H}_\infty$ control became popular in the eighties because of its ability to account for (worst case) model uncertainty, opening the door to the robust control generation. As explained in [35, 38, 58], $\mathscr{H}_\infty$ controllers are usually found by rearranging the problem into a generalized form, sometimes called standard $\mathscr{H}_\infty$ control problem. Within this setup, the aim is to minimize the energy gain of the generalized system, which remarkably allows for a simultaneous robustness/performance optimization. For the purpose of the optimal controller synthesis, the well-known Youla–Kučera parametrization [159] plays an important role, showing that any closed-loop transfer function can be expressed as an affine function of the controller. The final implication is that, ultimately, the $\mathscr{H}_\infty$ synthesis problem turns out to be a model-matching problem, which can be solved in practice. This is the classical route followed in [35, 38]. However, although the $\mathscr{H}_\infty$ standard problem is a well-established one for integer linear systems, this is not the case for fractional systems.

In this chapter, the problem of $\mathscr{H}_\infty$ optimal control for fractional single-input-single-output (SISO) continuous time system is solved. The classical approach of [38, 39] is generalized to the fractional SISO case, i.e.,, to the standard (possibly fractional) $\mathscr{H}_\infty$ problems that result in scalar model-matching problems.

Based on the properties of Taylor series expansion of holomorphic functions (and fractional polynomials), it is shown that it is always possible to factorize the fractional transfer function in such a way that it can be represented as the product of an unstable integer transfer function by a stable fractional one.

F. Padula and A. Visioli, *Advances in Robust Fractional Control*,
DOI 10.1007/978-3-319-10930-5_5

The former result allows the generalization of the so-called Youla parametrization of all stabilizing controllers to the fractional case. Then, following the classical route for integer transfer functions, the $\mathscr{H}_\infty$ optimal control problem is recast, via Youla parametrization, into a model-matching one, which is finally solved via Nehari's theorem. It is remarkable that the optimal model-matching error obtained for the fractional problem is actually an integer transfer function. As a consequence, the optimal stable Youla parameter Q is a fractional linear system. It generalizes to the fractional case the well-known result [38, 39] on the solution of the standard problem for the integer SISO setup. Besides, it is also shown that the model-matching problem, exactly as for the integer case, can be seen as a Nevanlinna–Pick optimal interpolation one.

It is worth stressing that the obtained results apply both to commensurate and incommensurate fractional systems, even if the commensurability property greatly simplify the calculus.

5.2 Factorization of Fractional Transfer Functions

This section is devoted to the development of some tools to factorize and decompose fractional transfer functions. They are needed to solve the main problem proposed in the next sections. For the sake of simplicity, the dependency of transfer functions on s from now on will be dropped most of the times.

First, the concepts of FRH_∞ and FRL_∞ function spaces are introduced. These are well-established concepts for integer transfer functions (RH_∞ and RL_∞ spaces), and their fractional generalizations will be widely used from now on. FRH_∞ is the space of all fractional real-rational functions analytic in the right half plane (RHP). FRL_∞ is the space of all the fractional real-rational functions bounded on the imaginary axes. Note that $RH_\infty \subset \mathrm{FRH}_\infty \subset \mathscr{H}_\infty$ and $RL_\infty \subset \mathrm{FRL}_\infty \subset \mathscr{L}_\infty$.

The next lemmas solve the problems of two important factorizations of fractional transfer functions.

The first one provides a result about inner-outer factorization of FRH_∞ functions. First, the concept of inner and outer functions is introduced: a function $f(s) \in \mathrm{FRH}_\infty$ is *inner* if $f(s)f(-s) = 1$. The zeros of this function all lie in the open RHP, the number of its zeros is called its *degree*. It is *outer* if it has no zeros in the outer RHP. This concept, considering Theorem 2.1, straightforwardly applies to the fractional case.

Lemma 5.1 *Every scalar-valued function f in FRH_∞ has a factorization $f = f_i f_o$, with f_i inner and f_o outer. If f has no zeros on the imaginary axis then $f_o^{-1} \in \mathrm{FRH}_\infty$*

Proof Let f_i be defined as the following product

$$f_i = \prod_i \frac{z_i - s}{\bar{z}_i + s}, \tag{5.1}$$

where z_i ranges over all the RHP zeros of f, counting multiplicities in the sense of Theorem 2.1. Define $f_o = \frac{f}{f_i}$, then f_i is inner by construction and f_o is outer since, in view of Theorem 2.1, it has no zeros in the RHP. If f is not strictly proper and has no zeros on the imaginary axis, so is f_o, then $f_o^{-1} \in \mathrm{FRH}_\infty$.

The next lemma provides another decomposition of fractional transfer functions.

Lemma 5.2 *Every $f \in \mathrm{FRL}_\infty$ admits a unique decomposition of the form*

$$f = f_1 + f_2, \tag{5.2}$$

with f_1 integer, strictly proper and analytic in left half plane, and $f_2 \in \mathrm{FRH}_\infty$.

Proof Via partial fraction expansion compute r_i such that

$$f = \sum_{i=1}^{m} \frac{r_i}{(s - p_i)^{k_i}} + R, \tag{5.3}$$

where m and k_i are suitable integer numbers such that all the RHP poles are taken into account with their multiplicities, in the sense of Theorem 2.1. Define $f_1 = \sum_{i=1}^{m} \frac{r_i}{(z-p_i)^{k_i}}$ and $f_2 = R = f - f_1$. The first function is evidently integer, strictly proper and analytic in the LHP while, since f is proper, so do f_2 considering that f_1 is strictly proper. Moreover, f_2, by construction, cannot have singularities in the RHP and is the summation of FRL_∞ functions, thus it is in FRH_∞, and this completes the proof.

5.3 Stabilizing Controllers

In this section, the problem of parametrizing the family of stabilizing controllers is addressed. A stabilizing controller is a controller that internally stabilizes the negative feedback loop of Fig. 5.1. Internal stability is obtained when all the four transfer function in Fig. 5.1 are stable [32, 133].

This condition can be formalized as $H(K, P) \in H_\infty$, being

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{1+KP} & \frac{-K}{1+KP} \\ \frac{P}{1+KP} & \frac{1}{1+KP} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = H(K, P) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{5.4}$$

In general, not every plant is stabilizable (that is, admits a stabilizing controller). However, when a plant admits a stabilizing controller, it is always possible to find infinitely many others. Moreover, as shown in [119], the parametrization is generally affine in two parameters. In particular, when a plant admits coprime factorization (see below), it is possible to simplify the resulting parametrization by using just a single affine parameter (this is the well-known Youla parametrization). In [19] it has

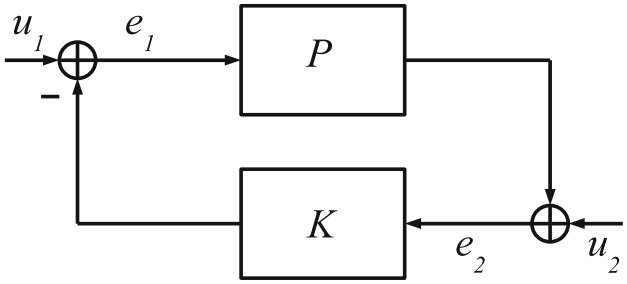

Fig. 5.1 Feedback interconnection of the plant with the controller

been shown how to build coprime factorizations for a class of fractional systems wider than that represented by fractional real-rational transfer functions.

Although these results apply to the systems object of this work, an *ad hoc* approach for fractional linear time-invariant SISO systems has been preferred. After proving the existence of a stabilizing controller, it is shown that it is sufficient to know a stabilizing controller to parametrize *all* the stabilizing controllers. Indeed, finding a stabilizing controller for a SISO system is in general a relatively easy task (in many practical cases it is already known), and the parametrization of all stabilizing controllers is done in order to optimize some criteria guaranteeing stability at the same time. Thus, this approach appears to be simpler in practice.

The next theorem considers the stabilizability of fractional linear systems.

Theorem 5.1 *For any given fractional linear system P, without zeros at the origin of the complex plane, there always exists a stabilizing controller.*

Proof First suppose that P has not a (possibly fractional) pole at the origin of the complex plane. Using the result of Lemma 5.1, P can always be factorized into an unstable nonminimum-phase integer part P_i and into an outer and FRH_∞ fractional part $P_o = \frac{P}{P_i}$.

Now define the following controller

$$\bar{K}(s) = \frac{P_o^{-1}\tilde{K}}{(s+1)^k}, \tag{5.5}$$

for some $k \in \mathbb{N}$, such as $\frac{P_o^{-1}}{(s+1)^k}$ and $\frac{P_i}{(s+1)^k}$ are proper. $\tilde{K}$ is a proper controller that stabilizes $\frac{P_i}{(s+1)^k}$. Note that $\tilde{K}$ always exists because $\frac{P_i}{(s+1)^k}$ is integer. Moreover, because of the properness of $\tilde{K}$, the choice of k that guarantees the properness of $\bar{K}$ does not depend on $\tilde{K}$. Evidently $\bar{K}$ stabilizes the system because, by hypothesis, no (possibly fractional) zero is present at the origin of the complex plane and ${P_o}^{-1}$ is thus stable. Now, suppose that P has also a (possibly fractional) integrator $\frac{1}{s^\nu}$. First define $P_\tau = P\frac{\tau s^\nu}{1+\tau s^\nu}$, P_τ is proper as long as P is and can be stabilized. Let K_τ be

a stabilizing controller for P_τ, it is always possible to increase τ until the system P is stabilized.

Note that the proof of Theorem 4 is constructive, that is, in general it can conveniently be used to find a stabilizing controller. However, if the plant is of commensurate order, a stabilizing controller can be determined more simply by pole placement techniques.

Note also that, only if the system is commensurate it is possible to explicitly use zero-pole cancellation to compute P_o as the ratio of two fractional polynomials without zeros in the RHP.

Now, the concept of coprime factorization over RH_∞ is introduced. Let P be a proper integer transfer function, a couple $M, N \in RH_\infty$ is said to be a coprime factorization of P over RH_∞ if $P = MN^{-1}$ and there exists a couple $X, Y \in RH_\infty$ such that the following condition, called Bezout identity, is satisfied:

$$NX + MY = 1. \tag{5.6}$$

Indeed, once a representation of P as the ratio of coprime elements in RH_∞ is obtained, a couple of transfer function satisfying (5.6) can be found, for instance, by means of Euclid's algorithm [142].

The following theorem shows that the Bezout equation is solvable also in FRH_∞. This allows the generalization of the coprime factorization approach to the fractional case.

Theorem 5.2 *Given a proper SISO fractional linear system P, if a stabilizing controller K_{stb} exists, then there exist $M, N, X, Y \in \mathrm{FRH}_\infty$ such that $P = NM^{-1}$ and the Bezout identity* (5.6) *holds.*

Proof Let K_{stb} a stabilizing controller, and $N_K, M_K \in \mathrm{FRH}_\infty$ a couple of coprime elements in FRH_∞ such that $K_{\mathrm{stb}} = M_K N_K^{-1}$. Define

$$V = NN_K + MM_K. \tag{5.7}$$

Note that, since K_{stb} stabilizes the control loop, V is invertible in FRH_∞ [35].

Now define

$$\begin{aligned} X &= N_K V^{-1} \\ Y &= M_K V^{-1}; \end{aligned} \tag{5.8}$$

since $V^{-1} \in \mathrm{FRH}_\infty$, also X and Y are in FRH_∞ and the Bezout identity holds by construction.

Finally, based on the previous results, the following theorem provides a tool to parametrize all the family of stabilizing controllers.

Theorem 5.3 *The set C of all controllers K that stabilize the feedback loop of Fig.* 5.1 *is:*

$$\mathscr{C} = \left\{ \frac{X + MQ}{Y - NQ} \ : \ Q \in \mathscr{H}_\infty \right\}, \tag{5.9}$$

where $P = MN^{-1}$ *and* M, N, X, Y *satisfy* (5.6).

Proof The proofs in [35, 142] for RH_∞ extends straightforwardly to the FRH_∞ case.

The previous theorem allows a parametrization of all stabilizing controllers with respect to Q. The only requirement to stabilize P is that $Q \in \mathscr{H}_\infty$. By means of Theorems 5.1–5.3 it is always possible to characterize the whole set of stabilizing controllers for a given fractional system.

5.4 The Standard $\mathscr{H}_\infty$ Control Problem

Thanks to the results of the previous section, it is now possible to formalize the standard $\mathscr{H}_\infty$ control problem.

The standard scalar $\mathscr{H}_\infty$ problem control scheme is represented in Fig. 5.2, where

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \tag{5.10}$$

is a two-input-two-output proper fractional linear system.

Let $\mathscr{C}$ be the set of stabilizing controllers for the feedback loop in Fig. 5.2. The standard problem consists of minimizing the norm from the exogenous input w to the output z making use of an stabilizing controller. Formally, this can be stated as follows:

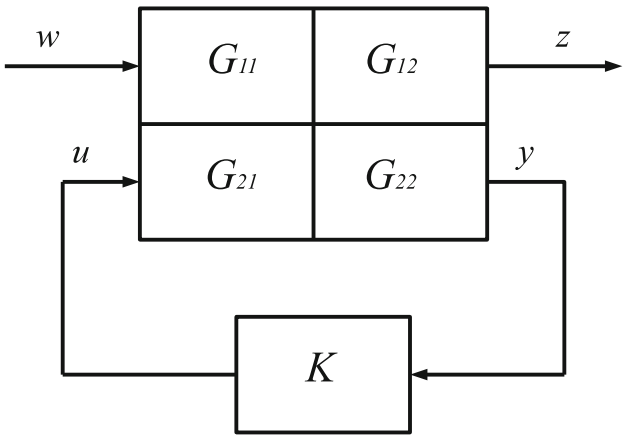

Fig. 5.2 Standard configuration block scheme

Problem 5.1

$$\min_{K \in \mathscr{C}} \|T_{zw}\|_\infty \tag{5.11}$$

where

$$T_{zw} = G_{11} + G_{12}K(1 - G_{22}K)^{-1}G_{21}. \tag{5.12}$$

Problem 5.1 is well-posed only if $1 - G_{22}K$ is invertible. Accordingly, hereafter the condition for well-posedness is assumed.

Note that the strict properness of G_{22} is a sufficient condition for the well-posedness of the problem [38].

Even if the problem is well-posed, the constrained minimization has not always a solution. Indeed the system G has to be stabilizable. In the integer case, using matrix coprime factorization it is possible to test the stabilizability of G. This means that G and G_{22} share the same unstable modes.

In the fractional case, a general tool to test stabilizability is still missing; thus, it has to be assumed that the system G is stabilizable.

It is from this assumption that next section will prepare the way to the main result of this chapter, the model-matching problem for fractional systems. By means of a characterization of stabilizing controllers for fractional systems, Problem 5.1 can be indeed rewritten as a model-matching problem.

5.5 The Model-Matching Problem

In view of the results of the previous sections, developed specifically for fractional systems, it is possible to recast Problem 5.1 into a simpler one and to generalize to fractional systems the model-matching theory developed for integer ones.

Indeed, by defining the controller K according to (5.9) (changing the sign), it is easy to see that (5.12) can be rewritten as

$$z = (T_1 - QT_2)w; \tag{5.13}$$

where

$$\begin{aligned} T_1 &= G_{11} - G_{12}XMG_{21} \\ T_2 &= -G_{12}M^2G_{21}, \end{aligned} \tag{5.14}$$

and where, by construction, both T_1 and T_2 are in FRH_∞.

Note that the requirement that K stabilizes G is automatically satisfied choosing $Q \in \mathscr{H}_\infty$. Moreover, as Q ranges on $\mathscr{H}_\infty$, all stabilizing controllers K are automatically considered. Finally, the problem is affine in Q, thus it is easier to be solved with respect to Q than with respect to K. Problem 5.1 can now be rewritten as a scalar model-matching problem formalized as follows.

Problem 5.2 Find $Q \in \mathscr{H}_\infty$ such that the model-matching error $\|T_1 - QT_2\|_\infty$ is minimized, where both T_1 and T_2 are in FRH_∞.

Note that the hypothesis on T_1 and T_2 are automatically satisfied. Indeed, the transfer functions involved in the model-matching problem are obtained recasting the standard $\mathscr{H}_\infty$ problem into a model-matching one via coprime factorization over FRH_∞, thus they are always in FRH_∞.

Summarizing, a scalar $\mathscr{H}_\infty$ control problem aimed at minimizing the ∞-norm of a linear fractional SISO system is recast into a scalar model-matching problem via parametrization of the family of stabilizing controllers.

Now, some important preliminary results necessary in order to solve the model-matching problem are introduced.

5.5.1 Nehari s Theorem

The following result, known as Nehari's theorem [78] is of main concern to solve Problem 5.2.

Theorem 5.4 [38, 39] *There exists a closest $\mathscr{H}_\infty$-matrix X to a given $\mathscr{L}_\infty$-matrix R, and $\|R - X\| = \|\Gamma_R\|$, where Γ_R is the Hankel operator with symbol R.*

A detailed discussion about the Hankel operator can be found in [101, 102].

Among the properties of the Hankel operator an important one is stated by the following theorem.

Theorem 5.5 [38] *If $f \in RL_\infty$ then Γ_f has finite rank.*

Moreover, the Hankel operator of an $\mathscr{H}_\infty$ matrix is null.

5.5.2 Model-Matching Problem Solution

The model matching problem can be solved by following the strategy proposed in [38, 39].

$$\begin{aligned} \|T_1 - T_2 Q\|_\infty &= \|T_1 - T_{2i} T_{2o} Q\|_\infty \\ &= \|T_{2i}(T_{2i}^{-1} T_1 - T_{2o} Q)\|_\infty \\ &= \|T_{2i}^{-1} T_1 - T_{2o} Q\|_\infty \\ &= \|R - X\|_\infty, \end{aligned} \tag{5.15}$$

where

$$\begin{aligned} R &= T_{2i}^{-1} T_1 \\ X &= T_{2o} Q \end{aligned} \tag{5.16}$$

Equality (5.15) holds because T_{2i} is inner, thus norm preserving and it is always possible to make an inner-outer factorization of T_2. Moreover, considering the hypothesis

of the model matching Problem 5.2 together with Lemma 5.1, it is clear that X and its inverse X^{-1} are in FRH_∞ while R is in FRL_∞.

Define α the infimal model-matching error

$$\alpha = \inf \{\|T_1 - QT_2\|_\infty \ : Q \in \mathscr{H}_\infty\}. \tag{5.17}$$

Using (5.15), it is possible to rewrite (5.17) as

$$\alpha = \inf \{\|R - X\|_\infty \ : X \in \mathscr{H}_\infty\}, \tag{5.18}$$

but this is exactly the problem solved by the Nehari's theorem, thus

$$\alpha = \|\Gamma_R\|. \tag{5.19}$$

Moreover, considering Lemma 5.2, R can be factorized as $R = R_1 + R_2$ with R_1 unstable and analytic in the left half plane (antistable) and $R_2 \in \mathscr{H}_\infty$ and it holds that $\Gamma_R = \Gamma_{R_1}$. But R_1 is RL_∞ (integer) thus the Hankel operator, in view of Theorem 5.5 has finite rank.

Let the functions f and g be a Schmidt pair [38] of the operator Γ_R, the following theorem holds.

Theorem 5.6 [38] *The infimal model-matching error α equals $\|\Gamma_R\|$, the unique optimal X equals $R - \alpha\frac{f}{g}$, and, for the optimal Q, the optimal model-matching error $T_1 - T_2Q$ is all-pass.*

It is remarkable that the optimal error obtained for the fractional problem is actually an integer transfer function. As a consequence, the optimal stable Youla parameter $Q \in \mathscr{H}_\infty$ is actually in FRH_∞, since the optimal model-matching error is all-pass.

Taking into account Theorem 2.1, the model-matching problem can be conveniently recast into an optimal interpolation one [35, 162]. Let $z_1, \ldots z_n$ be the n RHP zeros of T_2, a necessary and sufficient condition for a proper Q to be in FRH_∞ is that it must have no singularities in the RHP. This condition can be used to define the interpolation constraints:

$$\begin{aligned} &E^o(z_i) = T_1(z_i) \ i = 1, \ldots, n \\ &\left.\frac{\mathrm{d}^k E^o(s)}{\mathrm{d}s^k}\right|_{s=z_i} = \left.\frac{\mathrm{d}^k T_1(s)}{\mathrm{d}s^k}\right|_{s=z_i} \ k = 1, \ldots, m_i - 1; \ i = 1, \ldots, n, \end{aligned} \tag{5.20}$$

being m_i the multiplicity of the ith RHP zero of T_2 in the sense of Theorem 2.1 and E^o the optimal model-matching error.

The following theorem gives an alternative way to solve the model-matching problem.

Theorem 5.7 *Consider the model-matching Problem 5.2, the optimal model matching error is an all-pass in RH_∞ whose coefficients are completely determined by the interpolation constraints* (5.20).

Proof Theorem 5.6 states that the optimal interpolation error is an all-pass transfer function (obviously in RH_∞) and, in order to be the optimal one, this function has to meet the interpolation constraints (5.20). But this problem, also known as Nevanlinna–Pick optimal interpolation problem [1, 35] has a unique optimal solution, completely determined by the interpolation constraints, that is, the optimal model-matching error is an all-pass function in RH_∞ completely determined by (5.20).

There are different ways to obtain the model-matching (or optimal interpolation) error, mainly the Nevanlinna–Pick algorithm [35] and the state-space approach [38, 39]. Since nothing changes with respect to the integer case (indeed, R is factorized so that it is the product of an antistable integer part by a fractional stable one) the reader may conveniently refer to these works for the computational procedure.

Once the optimal model-matching error E^o has been computed, the optimal Q can be recovered via

$$Q^o = \frac{(E^o - T_1)}{T_2}. \tag{5.21}$$

Finally, the optimal feedback controller is obtained via (5.9).

Actually, as in any other case where fractional operators are considered, in order to practically implement the controller an approximation using an integer transfer function is required. However, the approximated controller can be arbitrarily close to the optimal fractional one (see, for example, Expression (3.15)).

Note that the exact solution of the fractional $\mathscr{H}_\infty$ problem is obtained instead of approximating the fractional system with an integer one, and then finding the solution of the approximated problem. In this latter case, it would not be possible to evaluate how far the true optimal solution is from the solution obtained, and if a high integer system approximation is employed, the solution of the approximated problem becomes computationally very demanding.

5.6 Illustrative Example

As a worked example, in order to illustrate the presented design procedure, consider the following unstable nonminimum-phase fractional system P whose transfer function is:

$$P(s) = \frac{(s^{0.5} + 3)(s^{0.5} - 1)(s + 7)}{s^{0.5}(s^{1.5} - 8)}. \tag{5.22}$$

The problem considered is that of stabilizing P while at the same time minimizing the weighted sensitivity, that is:

$$\min_{K \in \mathscr{C}} \|WS\| \tag{5.23}$$

where $S = \frac{1}{1+KP}$ and W is the weighting function which is a design parameter, typically chosen as a low-pass filter transfer function. In this case, the following structure is adopted:

$$W_{k,\bar{\omega}} = \frac{(0.01\bar{\omega}^{-1}s^{0.5}+1)^k}{(0.1\bar{\omega}^{-1}s^{0.5}+1)^{k-1}s^{0.5}} \tag{5.24}$$

where $k \in \mathbb{N}$ and $\bar{\omega}$ are parameters to be selected. The larger the value of k, the less emphasis is placed on minimizing the sensitivity for frequencies above $\bar{\omega}$. A fractional integrator in the weighting function is required here because, otherwise, an optimal Q does not exist in FRH_∞. Indeed, an all-pass optimal interpolation error would imply that Q is an integrator since the system itself is an integrator.

First, the control problem is recast into standard $\mathscr{H}_\infty$ problem (Fig. 5.2). The system G becomes [5]

$$G = \begin{bmatrix} W & -WP \\ 1 & -P \end{bmatrix} \tag{5.25}$$

and applying (5.12) it is immediate to see that choosing G in this way $T_{zw} = WS$. Now, in order to parametrize all stabilizing controllers a stabilizing one is required. It is easy to see that the following controller stabilizes the system:

$$K_{\mathrm{stb}} = \frac{0.14(s^{0.5}+100)(s^{0.5}-0.07143)(s+2s^{0.5}+4)}{(s^{0.5}-9)(s^{0.5}+3)(s+7)}. \tag{5.26}$$

Indeed, (5.26) is obtained by first inverting the outer and FRH_∞ part of P. Then, the poles of the equivalent integer system feedback loop (obtained by substituting $s^{0.5}$ by s) are placed in the RHP (note that this is a sufficient, but not necessary, condition for the stability of a fractional system [66]).

Now, both $P = NM^{-1}$ and $K_{\mathrm{stb}} = N_K M_K^{-1}$ are described in terms of ratios of coprime transfer functions over FRH_∞

$$\begin{aligned} N &= \frac{s^{0.5}-1}{s^{0.5}+1}, \\ M &= \frac{s^{0.5}(s^{1.5}-8)}{(s^{0.5}+3)(s^{0.5}+1)(s+7)}, \\ N_K &= \frac{14(s^{0.5}-0.07143)(s+2s^{0.5}+4)}{(s^{0.5}+3)(s+7)}, \\ M_K &= \frac{100(s^{0.5}-9)}{(s^{0.5}+100)}. \end{aligned} \tag{5.27}$$

Via Theorem 5.2, the Bezout parameters X and Y are computed:

$$\begin{aligned} X &= \frac{0.12281(s^{0.5}+100)(s^{0.5}+1)(s^{0.5}-0.07143)}{(s^{0.5}+0.569)(s+1.931s^{0.5}+1.542)}, \\ Y &= \frac{0.87719(s^{0.5}-9)(s^{0.5}+3)(s^{0.5}+1)(s+7)}{(s^{0.5}+0.569)(s+1.931s^{0.5}+1.542)(s+2s^{0.5}+4)}. \end{aligned} \tag{5.28}$$

Now, using (5.14) (paying attention to the signs because here G_{22} is $-P$) the following equations, for $\bar{\omega} = 0.5$ and $k = 3$, are obtained:

$$\begin{aligned} T_1 &= \frac{0.00017544(s^{0.5}+50)^3(s^{0.5}-9)(s^{0.5}-2)}{(s^{0.5}+5)^2(s^{0.5}+0.569)(s+1.931s^{0.5}+1.542)}, \\ T_2 &= \frac{0.0002(s^{0.5}+50)^3(s^{0.5}-2)(s^{0.5}-1)(s+2s^{0.5}+4)}{(s^{0.5}+3)(s^{0.5}+5)^2(s^{0.5}+1)^2(s+7)}. \end{aligned} \tag{5.29}$$

Now, once the inner-outer factorization of T_1 is done, only the inner part T_{2i} is really needed:

$$T_{2i} = \frac{(s-4)(s-1)}{(s+4)(s+1)}. \tag{5.30}$$

Note that, according to Lemma 5.1, the inner part is integer.

Now everything is ready for the computation of R using (5.16):

$$R = \frac{0.00017544(s^{0.5}+50)^3(s^{0.5}-9)(s^{0.5}+1)(s+4)}{(s+5)^2(s^{0.5}+2)(s^{0.5}+0.569)(s-1)(s^2+1.931s+1.542)}. \tag{5.31}$$

Again, note that the unstable part of R is integer. Via partial fraction expansion, according to Lemma 5.2, the antistable part R_1 of R is computed. As expected, it is integer:

$$R_1 = \frac{-2.456}{s-1}. \tag{5.32}$$

At this point, according to Theorem 5.6 and to the procedure of [38, 39], the function

$$\alpha\frac{f}{g} = \frac{1.2282(s+1)}{(s-1)} \tag{5.33}$$

is obtained as well as the optimal model-matching error $E^o = \alpha\frac{f}{g}T_{2i}$:

$$E^o = \frac{-1.2282(s-4)}{(s+4)}. \tag{5.34}$$

Note that the optimal model matching error is all-pass, and it is immediate to check that it satisfies the interpolation constraints (5.20). The optimal Q is obtained via (5.21):

$$\begin{aligned} Q^o = {} & \frac{6142.1272(s^{0.5}+7.546)(s^{0.5}+3)(s^{0.5}+1)^2}{(s^{0.5}+0.569)(s+1.931s^{0.5}+1.542)} \\ & \times \frac{(s+0.756s^{0.5}+3.14)(s+7)(s+7.216s^{0.5}+25.27)}{(s^{0.5}+50)^3(s+2s^{0.5}+4)(s+4)} \end{aligned} \tag{5.35}$$

and the optimal controller K^o via (5.9):

$$K^o = \frac{-(s^{0.5} + 6.565)(s^{0.5} - 1.172)(s + 5.63s^{0.5} + 10.58)(s + 2s^{0.5} + 4)}{(s^{0.5} + 3)(s^{0.5} + 2)(s^{0.5} + 5)^2(s + 7)}. \tag{5.36}$$

Not surprisingly, the optimal controller inverts the LHP part of the system. The previous procedure has been repeated also for different values of k and $\bar{\omega}$, to show the effects of the weighting function (a high frequency pole ($\omega = 0.01$) has been added to the closed-loop system to avoid numerical problems in the simulation). Moreover, in order to investigate how the choice of the weighting function affects the system in the time domain, the set-point step response of a unity feedback loop whose system is (5.22), controlled using (5.36) has been simulated (by approximating each fractional term with an integer system of 14th order obtained by means of the Oustaloup technique [89]).

As initial choice, the following parameter have been selected: $\bar{\omega} = 0.05$ and $k = 1$; it produces a strong interpolation error (Fig. 5.3) and, despite the overshoot, a sluggish step response (Fig. 5.4) because of the lower complementary sensitivity function at low frequencies (Fig. 5.5).

In order to speed up the system response $\bar{\omega}$ has been increased ten times, up to 0.5. The new controller yields a bigger overshoot because, as Fig. 5.6 shows, the sensitivity is higher at high frequencies because of the lower weighting functions at

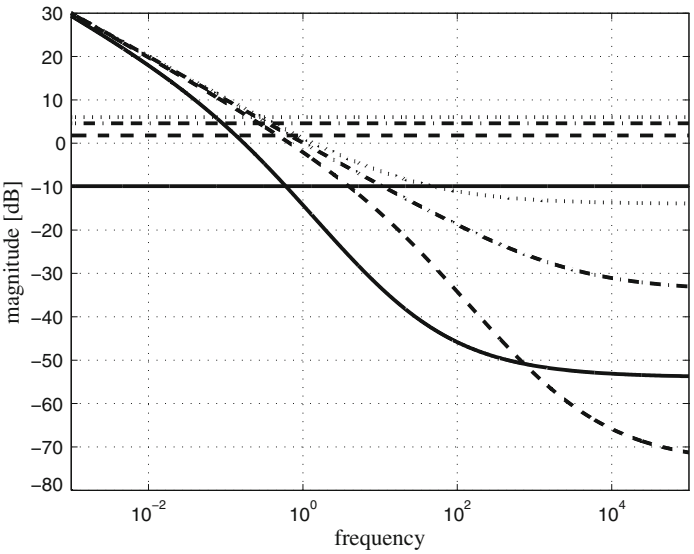

Fig. 5.3 Weighted sensitivity (or model-matching error) and the corresponding weighting function. *Dotted line* $\bar{\omega} = 0.05$ and $k = 1$. *Dash-dot line* $\bar{\omega} = 0.5$ and $k = 1$. *Dashed line* $\bar{\omega} = 0.5$ and $k = 3$. *Solid line* $\bar{\omega} = 0.05$ and $k = 3$

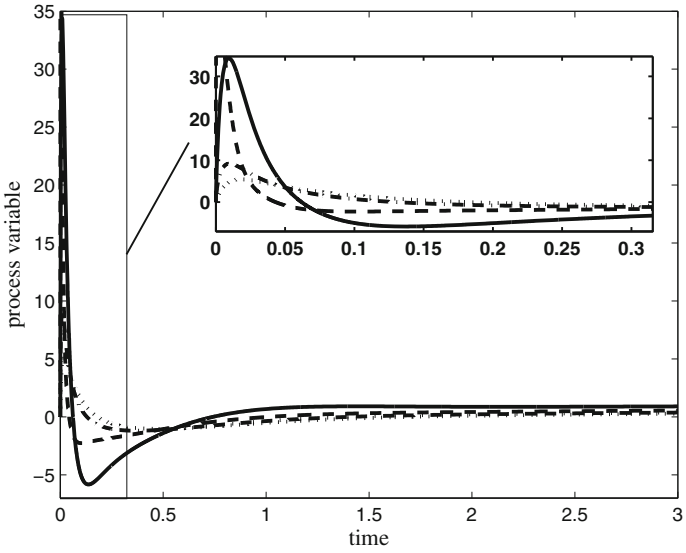

Fig. 5.4 Step response for different values of $\bar{\omega}$ and k. *Dotted line* $\bar{\omega} = 0.05$ and $k = 1$. *Dash-dot line* $\bar{\omega} = 0.5$ and $k = 1$. *Dashed line* $\bar{\omega} = 0.5$ and $k = 3$. *Solid line* $\bar{\omega} = 0.05$ and $k = 3$

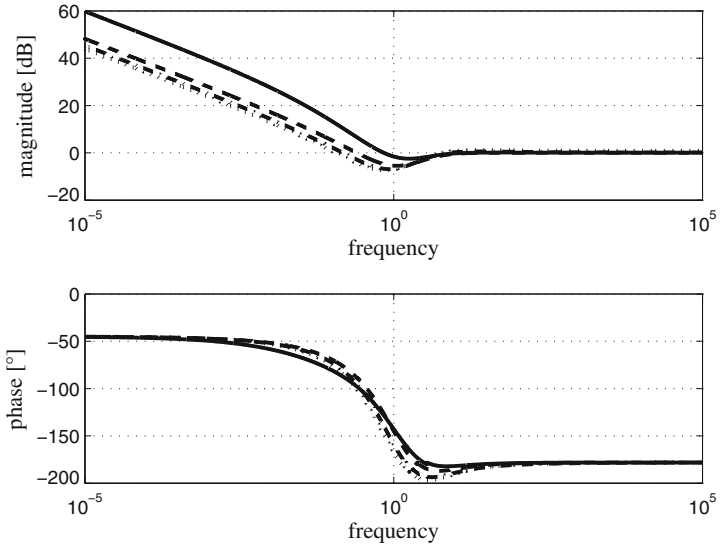

Fig. 5.5 Complementary sensitivity bode plot for different values of $\bar{\omega}$ and k. *Dotted line* $\bar{\omega} = 0.05$ and $k = 1$. *Dash-dot line* $\bar{\omega} = 0.5$ and $k = 1$. *Dashed line* $\bar{\omega} = 0.5$ and $k = 3$. *Solid line* $\bar{\omega} = 0.05$ and $k = 3$

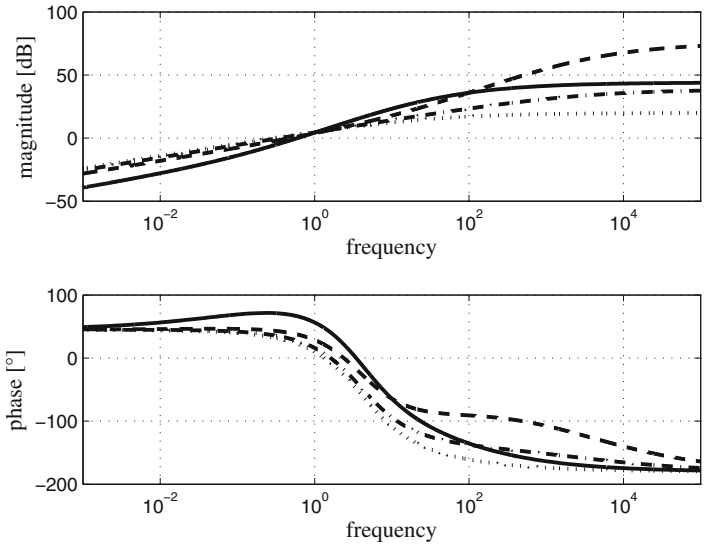

Fig. 5.6 Sensitivity Bode plot for different values of $\bar{\omega}$ and k. *Dotted line* $\bar{\omega} = 0.05$ and $k = 1$. *Dash-dot line* $\bar{\omega} = 0.5$ and $k = 1$. *Dashed line* $\bar{\omega} = 0.5$ and $k = 3$. *Solid line* $\bar{\omega} = 0.05$ and $k = 3$

those frequencies (see Fig. 5.3). Moreover, since the sensitivity is being reduced in a wider frequency range including a RHP zero, the undershoot is increased. Finally, for the same reason, the sensitivity function reduction is smaller at low frequencies with a consequent sluggish behavior, after the initial overshoot and undershoot.

To solve the problem k has been increased up to 3, in order to make the weighting function roll-off more severe (it weights less the higher frequencies). For the same reason discussed before, the overshoot is further increased, as well as the undershoot.

At this point, it appears meaningful to reduce the desired bandwidth by reducing w to 0.05, while keeping k fixed. In this way, all the weight is concentrated into a smaller frequency range (see Fig. 5.3). The system response is sensibly improved and the sluggish behavior disappears, even if the overshoot remains high (see Fig. 5.4). Indeed, since the weighting function bandwidth is reduced, the system has a bigger gain at low frequencies (see Fig. 5.5). The price to pay is a bigger undershoot. In fact, because of the reduced weighting function bandwidth, in this band the sensitivity is strongly reduced, also close to the RHP zeros, as Fig. 5.6 shows.

5.7 Conclusions

In this chapter, the solution of the standard $\mathscr{H}_\infty$ control problem for fractional continuous-time SISO systems has been presented. Using a generalized version of the Youla parametrization for fractional systems, the optimal control problem has

been recast into a model-matching one, solvable via Nehari's theorem. After having shown that it is always possible to factorize a fractional transfer function in such a way that it can be represented as the product of an unstable integer transfer function by a stable fractional one, this property has been used to compute the Hankel norm of the system to obtain the optimal model-matching error. Exactly as in the integer case, the optimal error turns out to be an integer all-pass transfer function, which can be completely determined from the RHP zeros of the system (Theorem 5.7). Consequently, if the system is fractional, the optimal controller is also fractional. The obtained results apply both to commensurate and incommensurate fractional systems.

Chapter 6
$\mathscr{H}_\infty$ Optimization-Based FOPID Design

6.1 Introduction

In this chapter a fractional-order proportional-integral-derivative (FOPID) controller design is presented. It is based on the solution of a $\mathscr{H}_\infty$ model-matching problem for fractional first-order-plus-dead-time (FFOPDT) processes [95]. Starting from the analytical solution of the problem, it is shown that a FOPID suboptimal controller can be obtained. In particular, starting from an Internal Model Control (IMC) representation of the feedback loop, the optimization problem is formulated, solved and, once the optimal IMC controller is obtained. Then, an equivalent feedback controller is derived [75]. Eventually, a FOPID controller is determined by approximating the optimal one. This method extends to fractional-order systems the method proposed in [143] for integer-order ones.

The stability and robustness issues are also addressed and the tuning of the parameters is discussed from a practical point of view by giving suitable guidelines in order to achieve the desired performance. In fact, as said before, dealing with fractional-order dynamics means that more flexibility is introduced at design stage. While this implies that a better performance can be achieved, from another point of view the design can be more difficult in practice. It is therefore essential that the physical meaning of all the design parameters is clearly explained and that a tuning procedure is devised in order to achieve the desired nominal performance while guaranteeing the robust stability at the same time.

It is also highlighted, by means of simulation examples, that, on the contrary to the solution of the model-matching problem for integer-order systems, fractional systems allow the user to treat processes with different dynamics (namely, overdamped or underdamped processes) in a unified framework, in addition to the capability of dealing explicitly with fractional-order processes. This is a clear advantage of the proposed methodology with respect to the integer-order case.

F. Padula and A. Visioli, *Advances in Robust Fractional Control*,
DOI 10.1007/978-3-319-10930-5_6

6.2 Problem Formulation

This section is devoted to the statement of the control problem. First, the process model is introduced, and then the optimization problem is formalized.

6.2.1 Process Model

It is well known that a great number of industrial processes can be described by means of a FOPDT model. Among the others, one of the reasons of the success of this kind of models is that the estimation of the model parameters is relatively simple.

In order to provide a more general result in the fractional-order systems framework, the family of the considered models is generalized by taking into account a FFOPDT system, namely:

$$G_{n_t}(s) = \frac{K}{1 + Ts^\alpha} \mathrm{e}^{-Ls}, \tag{6.1}$$

where K is the process gain, T is the time constant, L is the dead time and α is the fractional derivative order. Note that, when using $\alpha = 1$, an integer-order FOPDT process is obtained.

In order to apply some major results from systems theory, the delay term requires to be approximated by means of a series expansion. Here, the first-order Taylor expansion is employed:

$$\mathrm{e}^{-Ls} \approx 1 - Ls. \tag{6.2}$$

This approximation allows the determination of a suboptimal FOPID controller from the optimal one. Thus, by using Eq. (6.2), the nominal plant for the purpose of $\mathscr{H}_\infty$ optimization is redefined as:

$$G_n(s) = K \frac{1 - Ls}{1 + Ts^\alpha}. \tag{6.3}$$

6.2.2 Optimization Problem

Consider, a unity feedback control system (see, Fig. 6.1), where the feedback controller is denoted as $C(s)$. By assuming $G_n(s) = G_{n_t}(s)$, the nominal complementary sensitivity function of the closed-loop system [defined as $T(s)$] is:

$$T(s) = \frac{C(s)G_n(s)}{1 + C(s)G_n(s)}. \tag{6.4}$$

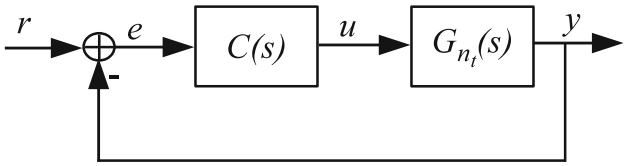

Fig. 6.1 The unity feedback control scheme considered

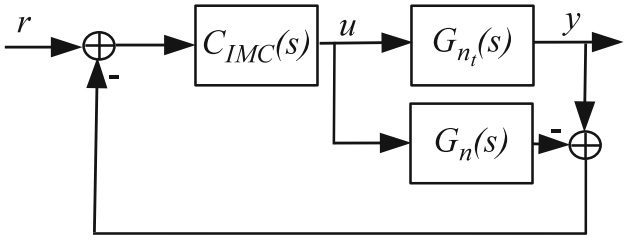

Fig. 6.2 The IMC control scheme considered

It can be shown (see Fig. 6.2) that, using the IMC paradigm [75], $T(s)$ can be expressed by means of the equivalent IMC controller and it leads to a very simple equation:

$$T(s) = G_n(s)C_{\text{IMC}}(s). \tag{6.5}$$

Once the synthesis of a stable $C_{\text{IMC}}(s)$ has been done, the feedback controller can be easily obtained by means of:

$$C(s) = \frac{C_{\text{IMC}}(s)}{1 - C_{\text{IMC}}(s)G_n(s)}. \tag{6.6}$$

Usually, when using the IMC tuning method, the desired closed-loop time constant is treated as a tuning parameter. Thus, through the choice of this time constant the trade-off between robustness and aggressiveness of the control loop is selected. Using the same approach proposed in [143] for integer-order transfer functions, the IMC formulation can be also employed in order to set up a min–max model-matching optimization problem on which the controller architecture is based.

The min–max optimization problem is formulated as finding an IMC controller $C^{\text{o}}_{\text{IMC}}(s)$ that solves the following optimization problem:

$$\min_{C_{\text{IMC}}(s)} \|W(s)(M(s) - C_{\text{IMC}}(s)G_n(s))\|_\infty, \tag{6.7}$$

where $M(s)$ is the reference model for the closed-loop system transfer function, while $W(s)$ is a weighting function. Then, by means of (6.6), the equivalent feedback controller transfer function is obtained.

The reference model is chosen to be a fractional first-order process:

$$M(s) = \frac{1}{1 + T_m s^{\lambda}}, \tag{6.8}$$

whose time constant T_m plays the same role of the time constant of the classical IMC control strategy, i.e., it specifies the desired closed-loop bandwidth.

The weighting function plays a double role: it gives to the user the opportunity to add degrees of freedom to the controller design and it allows the inclusion of the integral action into the equivalent feedback controller. Thus, it guarantees, at least for step signals, zero steady-state set-point following error and complete load disturbance rejection.

Here, a fractional-order weighting function is employed:

$$W(s) = \gamma \frac{1 + z s^{\mu}}{s^{\mu}} \tag{6.9}$$

It is worth noting that a fractional-order integral action $\frac{1}{s^{\mu}}$, with $\mu < 1$, as well as an integer-order one, guarantees zero steady-state error [70].

6.3 Solution of the Optimization Problem

In this section the solution to the min–max optimization problem stated in (6.7) is obtained. In order to solve the problem, the class of allowable fractional-order weights and fractional-order reference models must be slightly reduced to the ones whose exponents can be represented as the ratio of two natural numbers:

$$W(s) = \gamma \frac{1 + z s^{\frac{m}{n}}}{s^{\frac{m}{n}}}, \quad n, m \in \mathbb{N}, \tag{6.10}$$

$$M(s) = \frac{1}{1 + T_m s^{\frac{l}{n}}}, \quad l, m \in \mathbb{N}. \tag{6.11}$$

Note that, with an appropriate choice of n, m, and l, the corresponding real numbers μ and λ can be approximated with arbitrary precision. Therefore, the exponents μ and λ, from now on, are assumed to be rational numbers.

The min–max optimization problem (6.7) can be reformulated as follows.

Problem 6.1 Find a stable transfer function $C_{\mathrm{IMC}}(s)$ such that the cost function

$$\mathscr{J}_\infty = \|E(s)\|_\infty = \|W(s)(M(s) - C_{\mathrm{IMC}}(s)G_n(s))\|_\infty \tag{6.12}$$

is minimized, where $G_n(s)$ is defined in (6.3), $W(s)$ is defined in (6.10), and $M(s)$ in (6.11).

Because of the stability constraint for the free parameter $C_{\text{IMC}}(s)$ (which is the stable Youla parameter), this problem can be interpreted as an optimal interpolation problem [38, 162]. All the transfer functions involved in the optimization problem are stable and minimum phase, except $G_n(s)$ that has a right half-plane (RHP) zero at $z_g = 1/L$. This implies that the optimal IMC controller cannot have an unstable pole at z_g, because it would cause internal instability. Thus, as shown in Chap. 5, the RHP zero introduces an interpolation constraint since, in order to avoid internal instability, the model matching error has to satisfy:

$$E^{\text{o}}(z_g) = W(z_g)M(z_g), \tag{6.13}$$

where $E^{\text{o}}(s)$ is the optimal interpolation error:

$$\mathscr{J}_\infty^{\text{o}} = \|E^{\text{o}}(s)\|_\infty = \min \|E(s)\|_\infty. \tag{6.14}$$

In general, each RHP zero z_i generates a constraints for the optimal interpolation error. The constraints, together with a theoretical result that states that $E^{\text{o}}(s)$ is an all-pass transfer function (see, [28, 38, 93, 162]), lead to a general solution to the $\mathscr{H}_\infty$ interpolation problem.

Nevertheless, an ad hoc approach can be pursued for the particular (but useful to describe plenty of applications) problem stated here. By imposing the constraints (6.13) it is easy to see that the following equation holds:

$$\delta = |E^{\text{o}}(z_g)| = \gamma \frac{L^{2\mu} + zL^{\mu}}{L^{\mu} + T_m L^{\mu-\lambda}}. \tag{6.15}$$

The constraint imposes a lower bound to the ∞-norm of the optimal interpolation error, thus, any stable controller $C_{\text{IMC}}(s)$ such that:

$$\|E(s)\|_\infty = \delta \tag{6.16}$$

is optimal. Note that, at this point, it is not yet possible state the opposite, indeed an optimal controller could make the optimal interpolation error ∞-norm to be larger than δ. The next theorem provides a solution to this problem. First, the following lemma, useful to prove the main theorem of this section is proven:

Lemma 6.1 *A proper fractional-order system whose transfer function is:*

$$P(s) = \frac{\sum_{i=0}^{k} b_i s^{\varphi_i}}{\sum_{j=0}^{m} a_j s^{\psi_j}}, \quad \psi_i < 1, \ a_j \geq 0, \quad j = 1, \ldots, m, \quad a_0 > 0 \tag{6.17}$$

is always stable.

Proof In order to be stable, denominator of the transfer function (6.17) must have no roots in the closed RHP in the first Riemann sheet which, as already said in Chap. 2, is defined as [see, (2.13)] $\Omega = re^{j\phi} : r > 0, -\pi < \phi < \pi$ [107]. Note that the term a_0 lies on the positive real axis. In order to have a pole (everywhere in the complex plane), it means that it should exist some $c \in \Omega$ such that $\sum_{j=1}^{m} a_j c^{\psi_j} = -a_0$. Now, it is shown that there is no c such as $\text{Im}\left(\sum_{j=1}^{m} a_j c^{\psi_j}\right) = 0$. It is easy to see that $|\arg\left(\sum_{j=1}^{m} a_j c_j^{\psi}\right)| < |\arg(c)| \ \forall c \in \Omega$, thus for each point belonging to the first Riemann sheet the problem does not admit any solution. But if $\arg(c) = \pm\pi$, evidently, $\sum_{j=1}^{m} a_j c_j^{\psi}$ cannot be a real number and this completes the proof.

At this point the main theorem can be stated.

Theorem 6.1 *The optimal controller $C_{\text{IMC}}^{\text{o}}(s)$ that solves the min˘max problem stated in (6.7) obeys to the form*

$$C_{\text{IMC}}^{\text{o}}(s) = (W(s)G_n(s))^{-1}(W(s)M(s) - \delta) \tag{6.18}$$

provided that $\mu = \frac{m}{n} < 2, \ \lambda = \frac{l}{n} < 2, \ l, n, m \in \mathbb{R}$.

Proof First note that, if the controller defined in (6.18) is stable, then $E(s) = \delta$. This comes directly from the definition (6.18) of $C_{\text{IMC}}^{\text{o}}(s)$ together with (6.7) and it is a sufficient condition for the optimality, provided $C_{\text{IMC}}^{\text{o}}(s)$ to be stable.

Now, to conclude the proof, the stability of $C_{\text{IMC}}^{\text{o}}(s)$ is shown. It is a fractional transfer function whose expression with respect to the parameters of the transfer functions involved in the optimization process is:

$$C_{\text{IMC}}^{\text{o}}(s) = \frac{1}{K}\frac{1 + Ts^{\alpha}}{(1 + zs^{\mu})(1 + T_m s^{\lambda})}\chi(s), \tag{6.19}$$

where

$$\chi(s) = \frac{1 + zs^{\mu} - s^{\mu}(1 + T_m s^{\lambda})\frac{\delta}{\gamma}}{1 - Ls} \tag{6.20}$$

is a commensurate transfer function. If $\mu < 2$ and $\lambda < 2$, the first part of $C_{\text{IMC}}^{\text{o}}(s)$ is stable, it is a direct consequence of the stability theorem stated in [66].
Now the stability of $\chi(s)$ has to be proven. Define its integer counterpart as

$$\bar{\chi}(s) = \frac{1 + zs^{m} - s^{m}(1 + T_m s^{l})\frac{\delta}{\gamma}}{1 - Ls^{n}}. \tag{6.21}$$

The denominator $d_{\bar{\chi}}(s)$ of $\bar{\chi}(s)$ can be easily factorized:

$$d_{\bar{\chi}}(s) = \left(\sum_{k=0}^{n-1} L^{\frac{k}{n}} s^{k}\right)\left(1 - L^{\frac{1}{n}} s\right) \tag{6.22}$$

and going back to its fractional counterpart leads to:

$$d_\chi(s) = \left(\sum_{k=0}^{n-1} L^{\frac{k}{n}} s^{\frac{k}{n}}\right)\left(1 - L^{\frac{1}{n}} s^{\frac{1}{n}}\right). \tag{6.23}$$

Now, using Lemma 6.1, it is immediate to see that the first multiplicative term of $d_\chi(s)$ is stable, while the second one is clearly unstable (it can be proven by means of [66]).

Focusing on the numerator $n_{\bar{\chi}}(s)$, of $\bar{\chi}(s)$, it can be seen that if it is divisible by $(1 - L^{\frac{1}{n}} s)$ the optimal controller $C^{\mathrm{o}}_{\mathrm{IMC}}(s)$ is stable. By polynomial division, it is easy to see they divide each other and their ratio $Q(s)$ is:

$$Q(s) = \sum_{k=0}^{m-1} - \left(z - \frac{\delta}{\gamma} - \frac{\delta}{\gamma}\frac{T_m}{L^{\frac{l}{n}}}\right)\frac{1}{L^{\frac{m-k}{n}}} s^k + \sum_{k=m}^{m+l-1} \frac{\delta}{\gamma}\frac{T_m}{L^{\frac{m+l-k}{n}}} s^k \tag{6.24}$$

and, going back once again to the fractional counterpart of $n_{\bar{\chi}}(s)$, the optimal controller can therefore be expressed as:

$$\begin{aligned} C^{\mathrm{o}}_{\mathrm{IMC}}(s) = {} & \frac{1}{K}\frac{1 + Ts^\alpha}{(1 + zs^\mu)(1 + T_m s^\lambda)} \\ & \times \frac{\sum_{k=0}^{m-1} -(z - \frac{\delta}{\gamma} - \frac{\delta}{\gamma}\frac{T_m}{L^\lambda})\frac{1}{L^{\frac{m-k}{n}}} s^{\frac{k}{n}} + \sum_{k=m}^{m+l-1} \frac{\delta}{\gamma}\frac{T_m}{L^{\frac{m+l-k}{n}}} s^{\frac{k}{n}}}{\sum_{k=0}^{n-1} L^{\frac{k}{n}} s^{\frac{k}{n}}}. \end{aligned} \tag{6.25}$$

If the conditions of the theorem are satisfied, the previous controller is stable and the theorem is proven.

It is worth stressing that the optimal controller does not depend on the coefficient γ, see (6.10), because it always appears in the ratio $\frac{\delta}{\gamma}$, where δ is defined in (6.15). Actually, the weight function role is to formalize how much a certain frequency with respect to the others has to be considered. Indeed, a multiplicative coefficient that uniformly increases (or decreases) the weighting function magnitude does not change the result.

Then, note that the previous theorem could also be proven by means of the stability results stated in [66], indeed it is easy to see that $d_{\bar{\chi}(s)}$ has a pole on the positive real axis and the other ones are always outside the unstable region, i.e., $|\arg(p)| > \frac{1}{n}\frac{\pi}{2}$, being p a generic pole of $d_{\bar{\chi}(s)}$ except the one lying on the positive real axis.

Note also that $d_{\chi(s)}$ has no singularities in the first Riemann sheet, except the unstable one on the positive real axis. Indeed, mapping back the stable (in the sense of [66]) poles of $d_{\bar{\chi}(s)}$ into the singularities of $d_{\chi(s)}$, they go out the first Riemann sheet. It is a consequence of Lemma 6.1.

Finally, it can be deduced that the polynomial division rest $R(s)$ of $N_{\bar{\chi}(s)}$ by $(1 - L^{\frac{1}{n}}s)$ is:

$$R(s) = 1 + \left(z - \frac{\delta}{\gamma} - \frac{\delta}{\gamma}\frac{T_m}{L^{\lambda}}\right)\frac{1}{L^{\frac{m}{n}}} \tag{6.26}$$

and it goes to zero (i.e., the optimal $C^{\mathrm{o}}_{\mathrm{IMC}}(s)$ is stable) if δ is calculated by means of (6.15).

6.4 Analysis of the Optimal Interpolation Error

The interpolation constraint is needed in order to avoid internal instability (i.e., pole-zero cancelation in the RHP). It could be seen that avoiding to cancel the non-minimum-phase part of (6.2), both in a fractional stability framework or in an integer stability one, give rise to the same interpolation constraint for the stated problem. It is worth pointing out that the optimal interpolation error is related to the parameters μ, z, T_m, and λ. Note that speaking about optimal interpolation error or about the ∞-norm of the optimal interpolation error does not make any difference in this case because $E^{\mathrm{o}}(s) = \delta$ is constant. Without loss of generality (as the optimal controller does not depend on γ) the value $\gamma = 1$ is assumed. In order to study how δ varies with respect to the tuning parameter, its gradient is calculated:

$$\nabla\delta_{\mu,z,T_m,\lambda} = \left[\ln(L)\frac{L^{2\mu}}{L^{\mu} + T_m L^{\mu-\lambda}}, \quad \frac{L^{\mu}}{L^{\mu} + T_m L^{mu-\lambda}}, \right.$$
$$\left. -\frac{L^{2\mu} + zL^{\mu}}{(L^{\mu} + T_m L^{\mu-\lambda})^2}L^{\mu-\lambda}, \quad T_m L^{\mu-\lambda}\ln(L)\frac{L^{2\mu} + zL^{\mu}}{(L^{\mu} + T_m L^{\mu-\lambda})^2}\right]^{\mathrm{T}}. \tag{6.27}$$

Looking at (6.27) it is possible to deduce some interesting properties:

1. the interpolation error δ monotonously increases as μ increases if $L < 1$, it is flat with respect to μ if $L = 1$ and decreases if $L > 1$;
2. the interpolation error always increases with respect to z;
3. the interpolation error always decreases with respect to T_m;
4. the interpolation error increases as λ increases if $L < 1$, it is flat with respect to λ if $L = 1$ and decreases if $L > 1$.

Moreover, it can be noted that the interpolation error only depends on the chosen model delay L, while it does not depend on the model dynamics, namely, on T and α from (6.3). Indeed, neglecting the delay, the model dynamics is stable and minimum phase, and it is canceled by the optimal IMC controller.

6.5 Equivalent Feedback Controller

In this section, the equivalent optimal feedback controller $C^{\mathrm{o}}(s)$ obtained from $C^{\mathrm{o}}_{\mathrm{IMC}}(s)$ (IMC controller) and $G_n(s)$ by means of (6.6) is analyzed. Applying some calculus it is easy to see that the optimal feedback controller is:

$$C^{\mathrm{o}}(s) = \frac{1}{K}\frac{1+Ts^{\alpha}}{\delta s^{\mu}+T_m s^{\lambda}+T_m(z+\delta)s^{\lambda+\mu}} \times \frac{\sum_{k=0}^{m-1}-(z-\delta-\delta\frac{T_m}{L^{\lambda}})\frac{1}{L^{\frac{m-k}{n}}}s^{\frac{k}{n}}+\sum_{k=m}^{m+l-1}\delta\frac{T_m}{L^{\frac{m+l-k}{n}}}s^{\frac{k}{n}}}{\sum_{k=0}^{n-1}L^{\frac{k}{n}}s^{\frac{k}{n}}}. \tag{6.28}$$

It has the following properties:

1. low frequency behavior:

$$\lim_{s\to 0} C^{\mathrm{o}}(s) = \begin{cases} \frac{1}{K\delta s^{\mu}} & \text{if } \mu < \lambda \\ \frac{1}{K(T_m+\delta)s^{\mu}} & \text{if } \mu = \lambda \\ \frac{1}{KT_m s^{\lambda}} & \text{if } \mu > \lambda \end{cases} \tag{6.29}$$

   is a discontinuous function with respect to $\frac{\mu}{\lambda}$. Moreover the optimal feedback controller, at low frequencies, behaves as an integrator whose (fractional) degree is always equal to the smallest value between μ and λ;
2. controller properness:
   - $C^{\mathrm{o}}(s)$ is a strictly proper controller when $\alpha < 1$, as a consequence of approximating the $G_{n_t}(s)$ process delay [see (6.1)] by means of a first-order Taylor series (6.2);
   - it is a proper controller when $\alpha = 1$;
   - it is an improper controller when $\alpha > 1$.

Hence, the optimal controller has the same asymptotic behavior of a filtered FOPID controller. It is worth stressing that, when using unitary exponents $\mu = 1$, $\lambda = 1$ and $\alpha = 1$, the optimal controller obtained by means of (6.28) is the same one obtained in [143]. Indeed, the fractional-order case could be seen as the natural extension of the integer-order one and, vice versa, the integer-order one is a particular case of a more general solution in the fractional-order systems framework.

6.6 FOPID Controller

6.6.1 Controller Reduction

The optimal controller (6.28) that solves the model-matching problem is very difficult to implement because of its complexity. Thus, it appears to be meaningful to design a

simpler (FOPID) suboptimal controller. In order to do that the following assumptions are done: the fractional degree $\mu = \frac{m}{n}$ of (6.10) is equal to the fractional degree $\lambda = \frac{l}{n}$ of (6.11). In this way, the complexity of the optimal controller is reduced by decreasing the number of tuning parameters, allowing for a simpler controller expression. For this reason from now on only μ will be used.

Consider the optimal feedback controller (6.25). By means of (6.26) and by taking into account the previous assumption, expression (6.28) can be rewritten as:

$$C^{\mathrm{o}}(s) = \frac{1}{K} \frac{(1+Ts^{\alpha})(1+\delta \frac{T_m}{L^{\mu}} s^{\mu})}{(\delta + T_m)s^{\mu} + T_m(z+\delta)s^{2\mu}} \frac{\sum_{k=0}^{m-1} L^{\frac{k}{n}} s^{\frac{k}{n}}}{\sum_{k=0}^{n-1} L^{\frac{k}{n}} s^{\frac{k}{n}}}, \tag{6.30}$$

that, rearranging the equation, becomes:

$$C^{\mathrm{o}}(s) = \frac{1}{K} \frac{(1+Ts^{\alpha})(1+\delta \frac{T_m}{L^{\mu}} s^{\mu})}{(\delta + T_m)s^{\mu} + T_m(z+\delta)s^{2\mu}} \left(1 + \frac{-\sum_{k=m}^{n-1} L^{\frac{k}{n}} s^{\frac{k}{n}} + \sum_{k=n}^{m-1} L^{\frac{k}{n}} s^{\frac{k}{n}}}{\sum_{k=0}^{n-1} L^{\frac{k}{n}} s^{\frac{k}{n}}} \right). \tag{6.31}$$

where each summation is supposed to be null when the final index is smaller than the initial one. Neglecting the summation term in the previous equation, the following controller is obtained:

$$\tilde{C}(s) = \frac{1}{K} \frac{(1+Ts^{\alpha})(1+\delta \frac{T_m}{L^{\mu}} s^{\mu})}{(\delta + T_m)s^{\mu} + T_m(z+\delta)s^{2\mu}}, \tag{6.32}$$

that, once again by means of (6.26), can be rewritten as:

$$\tilde{C}(s) = \frac{1}{K(\delta + T_m)} \frac{(1+Ts^{\alpha})(1+T_m \frac{L^{\mu}+z}{L^{\mu}+T_m} s^{\mu})}{s^{\mu}(1+T_m \frac{\delta+z}{\delta+T_m} s^{\mu})}. \tag{6.33}$$

The previous equation is a series form FOPID controller [see, Chap. 3, Expression (3.2)], filtered by means of a fractional pole. It is worth noting that, when using $\mu = 1$, the suboptimal and the optimal solutions coincide, i.e., $\tilde{C}(s) = C^{\mathrm{o}}(s)$. Indeed, when using an integer weight and an integer reference model, the optimal controller is already an interacting FOPID controller. Finally note that, when using $\alpha = \mu$ and $z = T_m$, a FOPI controller is obtained.

It is worth noting that, in any case, the suboptimal FOPID controller has the same low frequencies asymptotic behavior of the optimal one, while, at high frequencies, the true plant (that is a low-pass filter) makes the controllers mismatch less important.

6.6.2 Nominal Stability

When using the nominal process model (6.3), the optimal controller (6.28) guarantees the closed-loop stability for any choice of T_m, z, μ, and λ, provided that the hypothesis of Theorem 6.1 (i.e., $\mu < 2$ and $\lambda < 2$) is satisfied.

Conversely, when considering the actual process model (6.1) and the suboptimal controller (6.33), it is important to verify the control system stability. In order to do that, the dual locus method [4] can be employed. By denoting the open-loop transfer function as $H(s) := \tilde{C}(s)G_{n_t}(s)$, the root locus method results in the following characteristic equation:

$$1 + H(s) = 1 + \frac{(1 + T_m \frac{L^\mu + z}{L^\mu + T_m} s^\mu)}{(\delta + T_m)s^\mu + T_m(\delta + z)s^{2\mu}} \mathrm{e}^{-Ls} = 0. \tag{6.34}$$

By defining

$$H_1(s) := -\frac{(L^{2\mu} + zL^\mu + T_m L^\mu + T_m^2)s}{(L^\mu + T_m) + T_m(L^\mu + z)} - \frac{T_m(L^{2\mu} + 2zL^\mu + zT_m)s^{2\mu}}{(L^\mu + T_m) + T_m(L^\mu + z)} \tag{6.35}$$

and $H_2 := \mathrm{e}^{-Ls}$, Eq. (6.34), taking into account (6.15), can be rewritten as

$$H_1(s) - H_2(s) = 0. \tag{6.36}$$

The dual locus technique is now applied: first, the intersection point has to be determined by solving the equation $|H_1(s)|_{s=j\omega} = 1$. Once the intersection point frequency ω_c has been obtained, the phases $\phi_1(\omega_c)$ and $\phi_2(\omega_c)$ of $H_1(j\omega_c)$ and $H_2(j\omega_c)$, respectively, are obtained. The closed-loop system is stable if $\phi_1 - \phi_2 < 0$. Briefly speaking, the stability condition is satisfied if the locus of $H_1(s)$ reaches the intersection point earlier than the one of $H_2(s)$.

Solving (6.36) analytically could be a complex task, because, in particular when $\mu \neq 1$, it results in complex trigonometric equations. Nevertheless, the dual locus method is useful to study the nominal stability using a worst case approach.

It is worth noting that, as T_m and z decrease, the loop bandwidth increases and the control system robustness decreases. The worst case is thus obtained by setting $T_m = 0$ and $z = 0$. With this choice, (6.36) becomes $-L^\mu s - \mathrm{e}^{-Ls} = 0$, and it is easy to prove that the stability condition is always satisfied, provided that $\mu < 2\frac{\pi - 1}{\pi} \approx 1.36$. Indeed, the actual process model and the suboptimal controller guarantee the closed-loop stability for any choice of the tuning parameter T_m and z provided that $\mu < 1.36$. On the other hand, when $\mu > 1.36$, a numerical solution of the dual locus equation (6.36) can be easily computed. Anyway, it is quite uncommon to find, in practical cases, fractional integrators with order greater than 1.3.

6.6.3 Robust Stability

If the conditions described in Sect. 6.6.2 are satisfied, the closed-loop stability is guaranteed as long as the process model $G_{n_t}(s)$ perfectly describes the process dynamics. However, as mismatches between the actual process and the model used are unavoidable, it is important to be able to stabilize a family of plants around the nominal one.

Assume that the process belongs to a family $\mathfrak{F}$ defined as:

$$\mathfrak{F} = \left\{ G(s) = G_{n_t}(s)(1 + \Delta_m(s)) : |\Delta_m(j\omega)| < |\Gamma(j\omega)| \right\}, \tag{6.37}$$

where

$$\Delta_m(s) = \frac{G(s) - G_{n_t}(s)}{G_{n_t}(s)} \tag{6.38}$$

is the uncertainty description and $\Gamma(j\omega)$ is a weight that defines the plants family by upper bounding the modeling error. It is well known that a controller that stabilizes a control system in the nominal case, also stabilizes the family $\mathfrak{F}$ of the control systems such that:

$$\|\Gamma(s)T_n(s)\|_\infty < 1, \tag{6.39}$$

where $T_n(s)$ is the stable nominal closed-loop transfer function, namely:

$$T_n(s) = \frac{\tilde{C}(s)G_{n_t}(s)}{1 + \tilde{C}(s)G_{n_t}(s)}. \tag{6.40}$$

The robust stability condition (6.39) is always satisfied if

$$|T_n(j\omega)| < \left| \frac{1}{\Gamma(j\omega)} \right|. \tag{6.41}$$

The robust stability constraint, rewritten in this form, is easy to compute and the left hand side of this inequality is usually a low-pass transfer function.

It is worth noting that $T_n(s)$ is different from $T(s) = C^{\mathrm{o}}_{\mathrm{IMC}}(s)G_n(s)$ because of the approximation of the process dead time (6.2) and because of the suboptimal FOPID controller (6.33). For this reason, it is no longer possible to use the IMC paradigm to recover the closed-loop transfer function. It would have been possible to use $T(s)$ as the nominal stable transfer function by including into the uncertainty $\Gamma(s)$ also the mismatches between the actual and approximated dead time expressions and between the optimal and suboptimal controllers. However, this choice may lead to a very poor performance in order to achieve the robust stability because it increases the

uncertainty. Conversely, choosing $T_n(s)$ as the nominal closed-loop transfer function, allows the enlargement of the closed-loop bandwidth by considering only the modeling uncertainty.

6.7 FOPID Tuning Guidelines

The methodology proposed in the previous sections has the great advantage that yields a standard FOPID controller, which has a well-known (standard) structure. However, it is also necessary to provide some guideline for the tuning of the parameters in order to allow the user to achieve the required performance without too much effort. In this context, the robustness/aggressiveness trade-off is handled by means of the z parameter. For a fixed desired bandwidth $(T_m)^{\frac{1}{\mu}}$, the user can modify the z parameter until the robust stability constraint (6.41) is satisfied. Of course, it might be possible that the robust stability constraint is never satisfied, in this case the desired closed-loop bandwidth is too large and it is necessary to reduce it by increasing $(T_m)^{\frac{1}{\mu}}$. Conversely, it might be possible that the robust stability constraint is always satisfied, this means that the required performance is easily achievable and it is possible to speed up the system response by decreasing T_m.

The tuning of the μ parameter is less significant and, in several cases, the use of an integer-order integrator is the best solution (see, Chap. 3). Nevertheless, sometimes the system response becomes sluggish when the system output is close to the set-point value. This behavior mainly depends on the model mismatch between the actual process and the process model. In this context, the μ parameter can be slightly modified in order to perform a fine tuning that tries to optimize the transient response close to the set-point value.

Summarizing, the tuning procedure can be outlined as follows:

1. set $\mu = 1$;
2. select the desired closed-loop bandwidth $T_m^{\frac{1}{\mu}}$;
3. compute the process uncertainty and select z to satisfy the robust stability constraints;
4. if such a value of z does not exist then increase $T_m^{\frac{1}{\mu}}$ and go to 3;
5. if the transient response is sluggish when the process output is close to the set point, reduce μ if the process output tends from above to the set-point value, otherwise increase μ;
   else set $\mu = 1$;
6. if $\mu > 1.36$ then verify the nominal stability;
7. if $\mu \neq 1$ go to 3;
8. end.

6.8 Simulation Results

6.8.1 Example 1

As a first illustrative example, consider the process

$$G(s) = G_{n_t,F}(s) = \frac{K}{Ts^\alpha + 1} e^{-Ls} \tag{6.42}$$

and suppose that the nominal process parameters assume their values in the middle points of following uncertainty intervals: $L \in [0.35, 0.65]$, $T \in [0.7, 1.3]$, $K \in [0.7, 1.3]$, and $\alpha \in [0.4, 1.0]$. The frequency response of the corresponding family of plants is shown in Fig. 6.3.

It is worth noting that the considered uncertainty is significant. In particular, the uncertainty on the order α implies that the controller is required to deal with a wide range of different dynamics. For the sake of comparison, the integer model with the nominal parameters and $\alpha = 1$ has been also considered:

$$G_{n_t,I}(s) = \frac{1}{s+1} e^{-0.5s}. \tag{6.43}$$

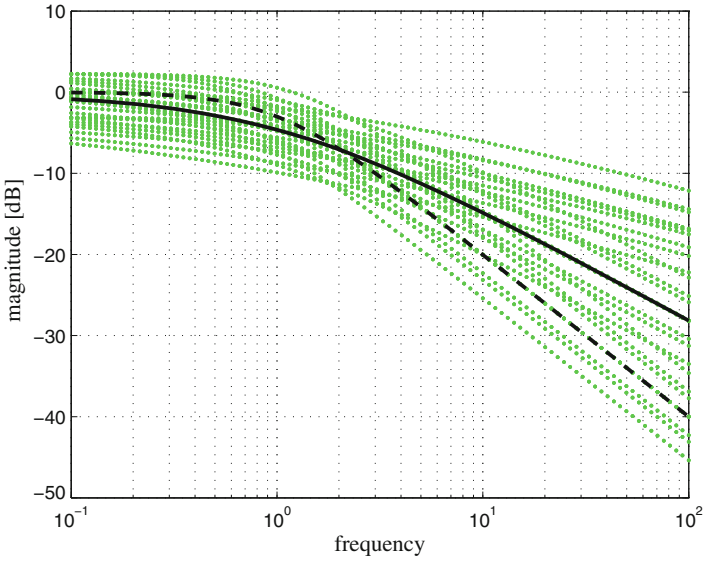

Fig. 6.3 Plants family (*green lines*), fractional model (*solid line*) and integer model (*dotted line*) for Example 1

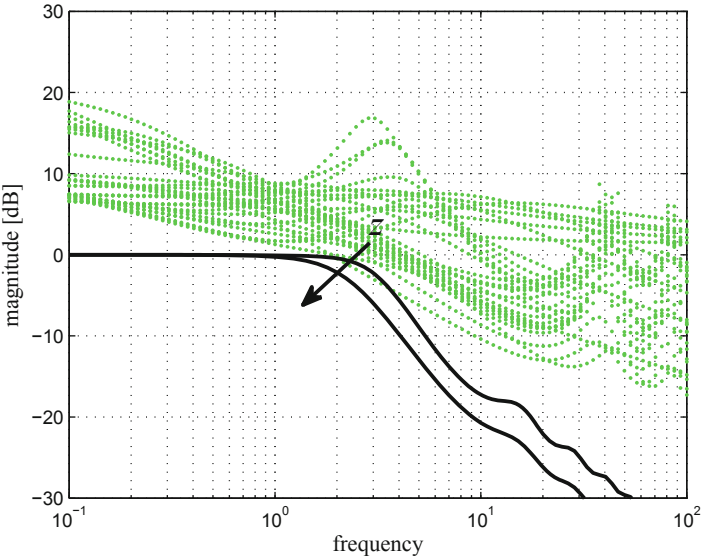

Fig. 6.4 Robust stability boundaries (*green lines*) and nominal closed-loop transfer functions for $z \in \{0.1, 0.8\}$ using the fractional model with $\mu = 1$ and $T_m = (1.2L)^{\mu}$ for Example 1

Note that, in any case, its frequency response is contained in the family of plants (see, Fig. 6.3). The desired closed-loop bandwidth is initially selected as $T_m = (1.2L)^{\mu} = 0.6$. Then, it can be seen that by using $z = 0.1$, both in integer and in the fractional case the robust stability is not achieved (see Figs. 6.4 and 6.5). Using $z = 0.8$ the fractional process model guarantees the robust stability whereas, for the integer one, it is required to use $z = 1.5$.

The step responses plotted in Figs. 6.6 and 6.7 show that the use of a fractional model allows the user to increase the speed of the response without loosing the robust stability with respect to the same family of plants. Finally, in order to improve the response when using the integer model, the integrator degree has been increased to $\mu = 1.1$ (and consequently, to preserve the desired bandwidth, $T_m = (1.2L)^{\mu} = 0.57$), increasing also z to 1.55 in order to satisfy the robust stability constraints (see, Fig. 6.5). Of course, the price to pay in order to control a family of plants different from the nominal one used for controller design is a degradation of the system response that could become oscillatory or sluggish depending on the actual process. It is worth noting that the fractional model allows the user to guarantee the robust stability with a more aggressive tuning (i.e., it improves the robustness/performance trade-off). On the other hand, because of the larger bandwidth, the performance degradation is more significant when the actual process is very different from the nominal model. Nevertheless, the stability is always guaranteed.

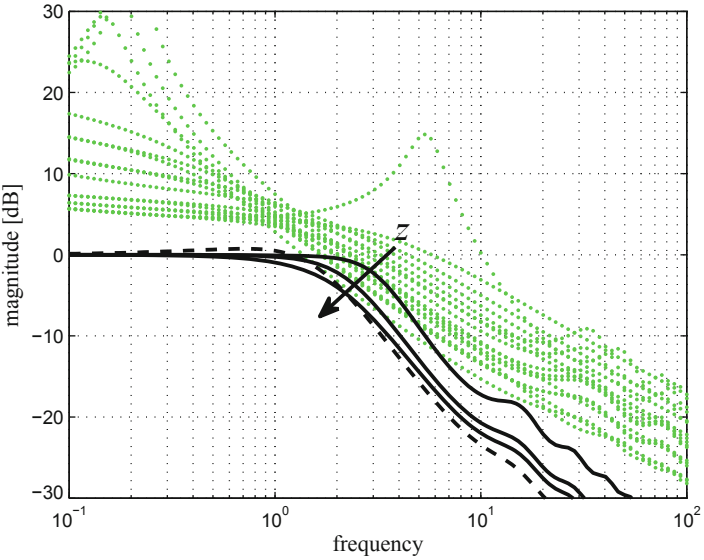

Fig. 6.5 Robust stability boundaries (*green lines*) and nominal closed-loop transfer functions for $z \in \{0.1, 0.8, 1.5\}$ and $\mu = 1$ (*solid lines*) and for $z = 1.55$ and $\mu = 1.1$ (*dashed line*), using the integer model with $T_m = (1.2L)^\mu$ for Example 1

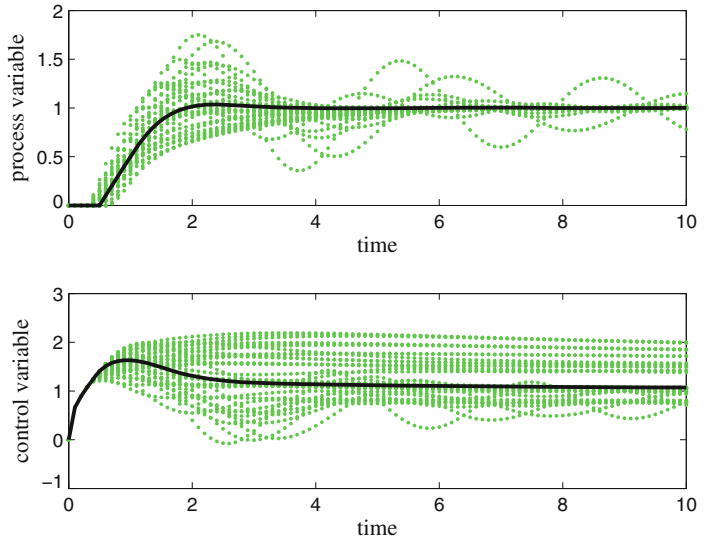

Fig. 6.6 Step responses using the fractional model with $T_m = (1.2L)^\mu$: plant family response with $\mu = 1$ and $z = 0.8$ (*green lines*) and nominal closed-loop step response with $\mu = 1$ and $z = 0.8$ for Example 1

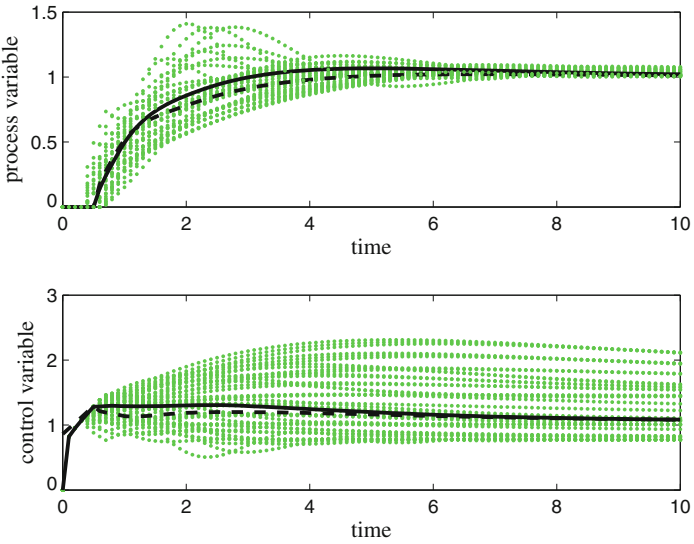

Fig. 6.7 Step responses using the integer model with $T_m = (1.2L)^{\mu}$: plants family response $\mu = 1.1$ and $T_m = 1.55$ (*green lines*), nominal closed-loop response with $\mu = 1.1$ and $T_m = 1.55$ (*solid line*) and with $\mu = 1$ and $T_m = 1.5$ (*dashed line*) for Example 1

6.8.2 Example 2

Consider now the following (underdamped) second-order-plus-dead-time (SOPDT) process:

$$G(s) = \frac{1}{s^2 + 0.4s + 1} \mathrm{e}^{-0.6s}. \tag{6.44}$$

Once again, a FOPDT model of the process has been identified by means of the Matlab function `pem` [60], resulting in:

$$G_{n_t,I}(s) = \frac{1}{0.566s + 1} \mathrm{e}^{-0.9s}. \tag{6.45}$$

Of course, modeling a strongly underdamped process with a FOPDT model is not sensible, but in this case it is required by the tuning method.

Conversely, a FFOPDT model can also capture an oscillatory behavior. The delay has been identified by considering the time interval when the process step response crosses the threshold of 1 % of the steady-state value. Then, considering that the step response of FFOPDT model is described by means of the two parameters Mittag–Leffler function [116], by imposing that that first peak of the model step response coincides with the first peak of the process step response, the FFOPDT model has been obtained as:

$$G_{n_t,F}(s) = \frac{1}{1.05s^{1.702}+1}\mathrm{e}^{-0.74s}. \tag{6.46}$$

The uncertainty has been computed as the difference between the process model and the actual process, normalized with respect to the process model, that is:

$$\Gamma(j\omega) = \frac{G(j\omega) - G_{n_t}(j\omega)}{G_{n_t}(j\omega)}. \tag{6.47}$$

Then, $T_m = 1.5$ has been fixed in order to set the desired bandwidth.

Because of the accurate model, in the fractional case z can be reduced to 0 preserving the robust stability. Thus, the selection of z can be done just with the purpose to speed up or slow down the system response. Conversely, in the integer case it is necessary to set $z = 10$ in order to achieve the robust stability. In Fig. 6.8 the robust stability boundaries and the nominal closed-loop transfer function frequency responses are shown both in the integer and in the fractional case: as expected, the robust stability constraints are much more severe when using an integer model, because of the incapability of a FOPDT model to capture underdamped dynamics. It appears that the capability of treating in a unified framework (that is, using the same process model structure) overdamped and underdamped processes is a great advantage that the fractional model offers.

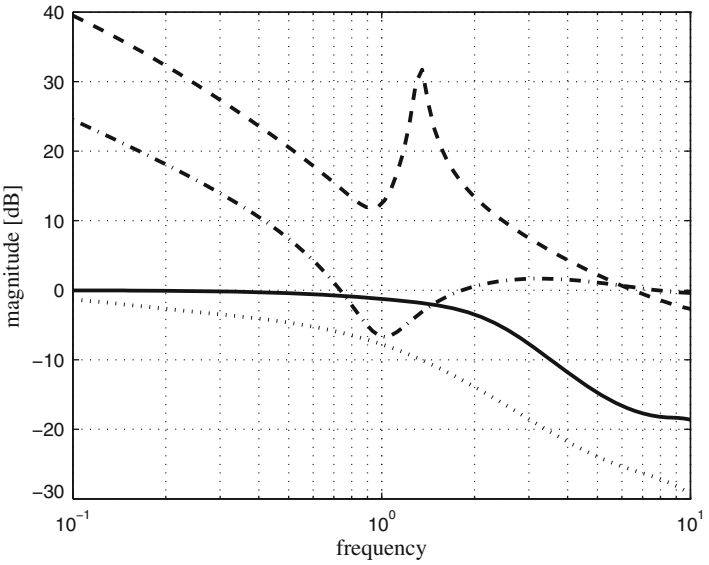

Fig. 6.8 Robust stability boundaries with $T_m = 1.5L$ and $\mu = 1$: fractional case (*dashed line*) and integer case (*dash-dot line*). Nominal closed-loop transfer functions: fractional case $z = 0.1$ (*solid line*) and integer case $z = 10$ (*dotted line*) for Example 2

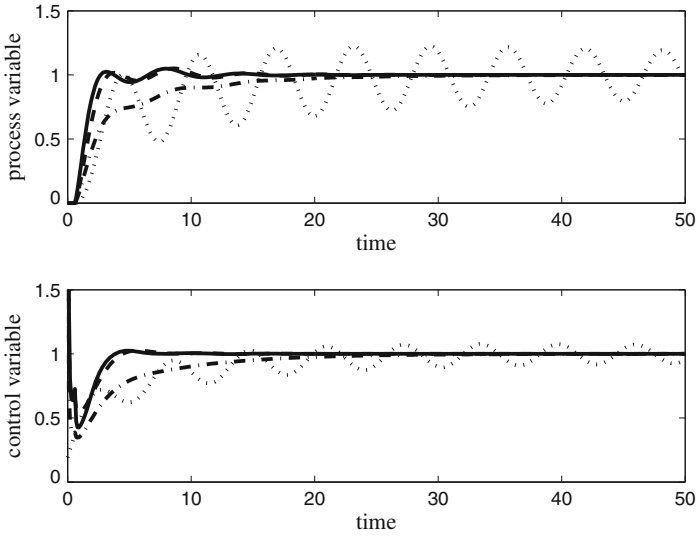

Fig. 6.9 Step responses with $T_m = 1.5L$ and $\mu = 1$: integer model (*dotted line* $z = 10$) and the fractional model for different values of z (*dash-dot line* $z = 10$, *dashed line* $z = 1$ and *solid line* $z = 0.1$) for Example 2

The different step responses shown in Fig. 6.9 for different values of z confirm the role of the parameter z in speeding up or slowing down the transient response and the major improvement obtained by using a fractional model. As the response is satisfactory, there is no need in this case to modify the value of μ.

6.8.3 Example 3

In [116], the following fractional-order process describing the dynamics of an heating furnace has been proposed:

$$G(s) = \frac{1}{14994s^{1.31} + 6009.5s^{0.97} + 1.69}. \tag{6.48}$$

The nominal model has been determined by considering the frequency response. In particular, after having selected $K = G(0)$, the time constant T and the exponent α have been chosen (by gridding the values of the two parameters) in order to minimize the two-norm of the difference between the process frequency response magnitudes and the model ones in the frequency range $[10^{-4}, 10^{-3}]$, around the process cut-off frequency.

Finally, the delay has been chosen such as the model phase is equal to the process one at the frequency of $\omega = 10^{-2}$ where the magnitude is about -40 dB. The resulting

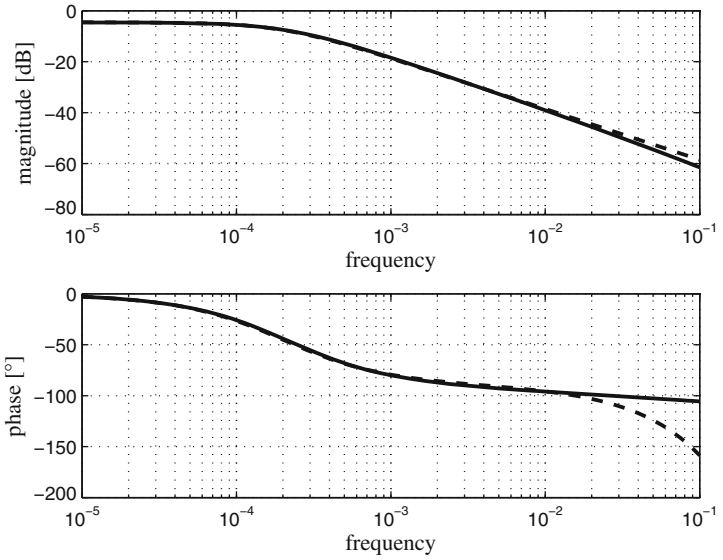

Fig. 6.10 Process transfer function (*solid line*) and process model transfer function (*dashed line*)—Example 3

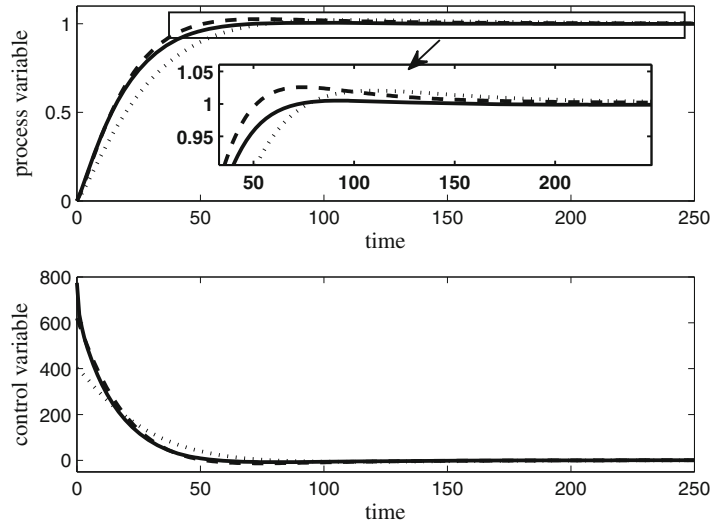

Fig. 6.11 Step responses for $(T_m = 0.5L)^\mu$ using $z = 0.1$ and $\mu = \frac{19}{20}$ (*solid line*), $z = 0.1$ and $\mu = 1$ (*dashed line*) and $z = 1$ and $\mu = 1$ (*dotted line*) for Example 3

(integer-order) process model is therefore

$$G_{n_t}(s) = \frac{0.5917}{5,000s + 1} e^{-12s}. \tag{6.49}$$

Note that it gives an effective description of the actual process frequency behavior up to $\omega = 10^{-2}$, as Fig. 6.10 shows. For higher frequency, the mismatch is indeed acceptable because of the actual process low-pass behavior.

The uncertainty has been computed as in the previous example, and, for the chosen values of z and μ, the control system always respects the robust stability boundaries. After having chosen $\mu = 1$ and $T_m = (0.5L)^\mu = 6$, z has been tuned in order to speed up the step response. However, Fig. 6.11 shows that, even by reducing z, the step response becomes sluggish after the peak in reaching the set-point value. To overcome this problem the exponent μ has been reduced to $\mu = \frac{19}{20}$ and consequently, to keep constant the desired bandwidth, $T_m = (0.5L)^\mu = 5.49$, effectively improving the system response.

6.9 Conclusions

A fractional $\mathscr{H}_\infty$ optimal model-matching methodology has been presented in this chapter. The technique is suitable for FFOPDT processes and allows the designer to use both fractional-order weighting function and fractional-order reference model. A FOPID controller can be derived from the optimal solution as a suboptimal controller. The nominal and robust stability issue has been discussed as well as the tuning of the user-defined parameters. Indeed, the physical meaning of the parameters has been clarified, thus showing that the methodology can be easily applied in practical cases. It has also been shown that the additional flexibility in the design because of dealing with fractional-order systems plays a key role in improving the control system performance. In particular, it has been highlighted that a clear advantage of having extended the model matching design method to fractional-order systems is to have a unique framework for both underdamped and overdamped systems.

Chapter 7
Control Design Based on Input–Output Inversion

7.1 Introduction

Set-point regulation of a system based on input–output inversion design is a powerful design tool that has been exploited for integer-order systems [33, 53, 104, 109, 110, 113, 163] and can be also applied to fractional-order systems [96]. It is based on the idea of solving an output tracking problem by computing a suitable input via input–output inversion.

The input–output inversion procedure aims at obtaining a smooth transition of the system output from a steady-state value to a new one in a predefined transition time τ. Given an arbitrarily smooth desired output signal, the corresponding input signal is computed analytically such that zero tracking error occurs. Both the input and the corresponding output signals are τ-parametrized allowing the user to handle the trade-off between a fast transition time and the attainment of high values of the input function and its derivatives.

Among the possible desired output functions, the transition polynomials proposed in [111] can been chosen because of their nice properties (for example, the optimal output synthesis problem can be addressed) and because they are at the same time computationally simple. Moreover, they allow the user to select the desired smoothness easily by choosing the polynomial order.

Thus, once a τ-parametrized input has been obtained, the problem of finding the minimum constrained transition time can be solved. Indeed, given a set of constraints on the input signal and on the derivatives of the input and output signals, it can be proven that (increasing the transition time), the set of feasible solutions is not empty under very mild and reasonable conditions. A simple bisection algorithm can therefore be employed to solve the optimization problem and to find the minimum allowable transition time.

This methodology can be employed to determine a suitable feedforward control action to be applied to improve the set point tracking performance of a feedback control architecture. In this context, combining the synthesis of the feedback and of the feedforward controller can be very useful in order to handle the system robustness. In particular, a robust control methodology will be presented in order to minimize

F. Padula and A. Visioli, *Advances in Robust Fractional Control*,
DOI 10.1007/978-3-319-10930-5_7

the worst-case settling time by considering a family of plants and constraints on the control and process variables.

7.2 Input–Output Inversion

7.2.1 Problem Formulation

Consider a general minimum-phase stable commensurate fractional linear system Σ whose transfer function is

$$H(s) = \frac{b(s)}{a(s)} = \frac{\sum_{k=0}^{m} b_k s^{k\nu}}{s^{p\nu} + \sum_{k=0}^{p-1} a_k s^{k\nu}}, \tag{7.1}$$

where ν is the commensurate order of the system, and $\rho := (p-m)\nu$ the relative order of the system Σ. The posed problem consists in finding a suitable input function $u(t)$ to obtain a desired output function which allows a transition from an initial steady-state value to a new one in a finite time interval τ, given a set of bounds on the input and output signals and their derivatives. The problem can be formalized as follows:

Problem 7.1 Starting from null initial conditions and given a new steady-state output value y_e, design a "sufficiently smooth" τ-parametrized desired output $\bar{y}(\cdot;\tau)$ such that $\bar{y}(0;\tau) = 0$ and $\bar{y}(t;\tau) = y_e \ \forall t \geq \tau$, and $\bar{y}(\cdot;\tau) \in C^{(k)}$ for some $k \in \mathbb{N}$. Then, find $u(\cdot;\tau)$ such that, for the τ-parametrized couple $(u(\cdot;\tau); \bar{y}(\cdot;\tau))$, it holds that

$$\mathscr{L}[\bar{y}(t;\tau)] = H(s)\mathscr{L}[u(t;\tau)]. \tag{7.2}$$

Moreover, determine the minimum time τ^* such that $u(t;\tau^*)$ and the first $l \in \mathbb{N}_0$ ($\nu \in \mathbb{N}$, respectively) derivatives of $u(t;\tau^*)$ ($\bar{y}(t;\tau^*)$), are bounded:

$$\begin{aligned} &|D^i u(t;\tau^*)| < u_M^i, \ \forall t > 0, \ i = 0, 1, \ldots, l; \\ &|D^i \bar{y}(t;\tau^*)| < y_M^i, \ \forall t > 0, \ i = 1, 2, \ldots, \nu. \end{aligned} \tag{7.3}$$

It is worth stressing that the requirement of null initial conditions is without loss of generality in view of the system linearity.

7.2.2 Output Function Design

Although different function bases could be used to design the output function, the simple and computationally efficient τ-parametrized transition polynomial proposed in [111] is conveniently chosen. It has the nice property of being monotonic, which implies that neither overshoots nor undershoots occur. For the sake of simplicity and

without loss of generality in view of the linearity of the system, the value $y_e = 1$ is considered. The output function is therefore selected as:

$$\bar{y}(t;\tau) := \begin{cases} 0 & \text{if } t < 0 \\ \dfrac{(2n+1)!}{n!\tau^{2n+1}} \times \displaystyle\sum_{r=0}^{n} \frac{(-1)^{n-r}\tau^r t^{2n-r+1}}{r!(n-r)!(2n-r+1)} & \text{if } 0 \le t \le \tau \\ 1 & \text{if } t > \tau. \end{cases} \tag{7.4}$$

Note that $\bar{y}(t;\tau)$ allows an arbitrarily smooth transition between 0 and 1; indeed, it is possible to show that $\bar{y}(t;\tau) \in C^{(n)}$ [111]. Moreover, it can be shown that the following lemma holds.

Lemma 7.1 *Let be given the transition signal* $\bar{y}(\cdot;\tau) \in C^{(n)}$ *defined in* (7.4). *Then there exist constants* $c_i \in \mathbb{R}_+,\ i = 1, \ldots, n+1$ *such that*

$$\max_{t\in[0,\tau]} |D^i \bar{y}(t;\tau)| = \frac{c_i}{\tau^i}. \tag{7.5}$$

Proof This result has been proven in [111] for $i = 1, \ldots, n$. Now consider the case $i = n+1$: it can be easily checked that, if the transition polynomial (7.4) is of order n, $D^n \bar{y}(t;\tau)$ is continuous ($C^{(0)}$), null for $t \le 0$, constant for $t \ge \tau$ and a ramp in $[0, \tau]$. Hence, defining $D^{n+1}\bar{y}(0;\tau) := 0$ and $D^{n+1}\bar{y}(\tau;\tau) := 0$ it is immediately possible to obtain the $(n+1)$th derivative of the transition polynomial as

$$D^{n+1}\bar{y}(\tau;\tau) := \begin{cases} \frac{c_{n+1}}{\tau^{n+1}} & \text{if } \ 0 < t < \tau \\ 0 & \text{elsewhere} \end{cases} \tag{7.6}$$

where c_{n+1} is the slope of the ramp, and this completes the proof.

In fact, the previous lemma means that the transition polynomial becomes flatter when τ increases, that is, it is always possible to increase the value of τ until the second condition of (7.3) is satisfied, provided that $n \ge v - 1$ and $y_M^i > 0$, $i = 1, \ldots, v$.

7.2.3 Input–Output Inversion Procedure

Once the transition polynomial (7.4) has been defined, the corresponding input $u(t;\tau)$ can be computed by inverting the system through Laplace transform, that is

$$U(s;\tau) = H^{-1}(s)\bar{Y}(s;\tau), \tag{7.7}$$

where $\bar{Y}(s;\tau) = \mathscr{L}[\bar{y}(t;\tau)]$ denotes the Laplace transform of the desired output. The integer-order system associated to (7.1) is

$$\tilde{H}(w) = \frac{\tilde{b}(w)}{\tilde{a}(w)} = \frac{\sum_{k=0}^{m} b_k w^k}{w^p + \sum_{k=0}^{p-1} a_k w^k}, \tag{7.8}$$

where $\tilde{a}(s)$ and $\tilde{b}(s)$ are coprime polynomials. By polynomial division on $\tilde{H}(w)$ and by means of the substitution $w = s^\nu$, the following expression is obtained:

$$H^{-1}(s) = \gamma_{n-m} s^\rho + \gamma_{n-m-1} s^{\rho-\nu} + \cdots + \gamma_1 s^\nu + \gamma_0 + H_0(s), \tag{7.9}$$

where $H_0(s)$ is a strictly proper transfer function that describes the zero dynamics of Σ. Note that $H_0(s)$ is always stable in view of the minimum-phase assumption of the system Σ.

Defining $\eta_0(t) := \mathscr{L}^{-1}[H_0(s)]$ and considering the Laplace transform properties, the following proposition can be derived:

Proposition 7.1 *Consider $\bar{y}(t;\tau)$ defined in (7.4). If $n \geq [\rho]$ then there exists a bounded inverting signal defined as follows*

$$\begin{aligned} u(t;\tau) = {} & \gamma_{n-m} D^\rho \bar{y}(t;\tau) + \gamma_{n-m-1} D^{\rho-\nu} \bar{y}(t;\tau) + \cdots \\ & + \gamma_1 D^\nu \bar{y}(t;\tau) + \gamma_0 \bar{y}(t;\tau) + \int_0^t \eta_0(t-\xi)\bar{y}(\xi;\tau) d\xi. \end{aligned} \tag{7.10}$$

Proof The result is immediate by considering (7.9), the properties of linear systems, the existence of the derivatives till the order $n+1$ in view of Lemma 7.1, and the stability of $\eta_0(t)$.

It is worth stressing that, by considering the polynomial nature of the desired output signal and the fractional power law, when the relative order of the plant is of fractional order (i.e., $\rho \in \mathbb{R}\backslash\mathbb{N}$) the obtained inverting signal is always continuous. Indeed, even by selecting the lowest possible order $n = [\rho]$ the obtained signal is not Lipschitz, but continuous.

In view of the minimum-phase assumption, the input (7.10) is bounded. Hence, by means of (7.10), the input–output inversion problem is completely solved.

In order to compute (7.10), the two parameters Mittag-Leffler function

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)} \quad \alpha > 0, \beta > 0 \tag{7.11}$$

and the Podlubny function

$$\varepsilon_k(t, \lambda; \alpha, \beta) = t^{k\alpha+\beta-1} \frac{d^k}{d(\lambda t^\alpha)^k} E_{\alpha,\beta}(\lambda t^\alpha) \tag{7.12}$$

introduced in Sect. 2.2 are considered. By considering (2.6), that is,

$$\mathscr{L}^{-1}\left[\frac{k!s^{\alpha-\beta}}{(s^{\alpha}\pm\lambda)^{k+1}}\right]=\varepsilon_k(t,\mp\lambda;\alpha,\beta) \tag{7.13}$$

and by applying to $H_0(s)$ the same reasoning applied to $H(s)$, it is easy to see that $H_0(s)$ can be factorized via partial fraction expansion and represented as the summation of simple terms:

$$H_0(s)=\sum_{i=1}^{m}\frac{g_i}{(s^{\nu}-\lambda_i)^{k_i+1}}, \tag{7.14}$$

where λ_i and g_i can be either real or complex (in the latter case they always appear in conjugate pairs, so that the time response is always real) and k_i is a nonnegative integer. Moreover, in view of the minimum-phase assumption on Σ, $H_0(s)$ is stable.

Considering (7.13) and (7.14), it can be seen that

$$\eta_0(t)=\sum_{i=1}^{m}\frac{g_i}{k_i!}\varepsilon_{k_i}(t,\lambda_i;\nu,\nu). \tag{7.15}$$

Now, by substituting the previous equation together with (7.4) into the convolution integral appearing in (7.10), after some calculations it results that

$$\begin{aligned}\int_0^t \eta_0(t-\xi)\bar{y}(\xi;\tau)d\xi=&\sum_{i=1}^{m}\frac{g_i}{k_i!}\frac{(2n+1)!}{n!\tau^{2n+1}}\sum_{r=0}^{n}\frac{(-1)^{n-r}\tau^r}{r!(n-r)!(2n-r+1)}\\ &\times\begin{cases}(2n-r+1)!\varepsilon_{k_i}(t,\lambda_i;\nu,2n-r+2+\nu) & \text{if } t\le\tau\\ \displaystyle\int_0^{\tau}\varepsilon_{k_i}(t-\xi,\lambda_i;\nu,\nu)\xi^{2n-r+1}d\xi & \\ \quad+\displaystyle\int_{\tau}^{t}\varepsilon_{k_i}(t-\xi,\lambda_i;\nu,\nu)d\xi & \text{if } t>\tau.\end{cases}\end{aligned} \tag{7.16}$$

In order to obtain the inverting input (7.10), consider that

$$x^n=(x-\tau+\tau)^n=\sum_{j=0}^{n}\binom{n}{j}(x-\tau)^{n-j}\tau^j. \tag{7.17}$$

The transition polynomial can be therefore represented as

$$\bar{y}(t;\tau) = \begin{cases} 0 & \text{if } t < 0 \\ \dfrac{(2n+1)!}{n!\tau^{2n+1}} \displaystyle\sum_{r=0}^{n} \frac{(-1)^{n-r}\tau^r t^{2n-r+1}}{r!(n-r)!(2n-r+1)} & \text{if } 0 \le t \le \tau \\ \dfrac{(2n+1)!}{n!\tau^{2n+1}} \displaystyle\sum_{r=0}^{n} \frac{(-1)^{n-r}\tau^r}{r!(n-r)!(2n-r+1)} & \\ \times \left[t^{2n-r+1} - \displaystyle\sum_{j=0}^{2n-r+1} \binom{2n-r+1}{j} \right. & \\ \left. \times (t-\tau)^{2n-r+1-j}\tau^j \right] + 1(t-\tau) & \text{if } t > \tau, \end{cases} \tag{7.18}$$

where $1(\cdot)$ is the Heaviside function. The previous expression can be further simplified by considering that the transition polynomial is $C^{(n)}$ by construction. Hence, the summation of all the terms that, by differentiating till the order n the transition polynomial, would lead to impulse-like behaviors at $t = \tau$, is null. Thus, the summation over j can be truncated at $n - r$.

Now, consider the fractional differintegral of the transition polynomial. By virtue of the previous reasoning, considering that $D^\alpha x^n = \frac{n!}{\Gamma(n+1-\alpha)} x^{n-\alpha}$, $\alpha \in \mathbb{R}$ and expanding the binomial coefficients in (7.18), the differintegral of the transition polynomial is finally obtained for $-\infty < \alpha \le n+1$:

$$D^\alpha \bar{y}(t;\tau) = \begin{cases} 0 & \text{if } t < 0 \\ \dfrac{(2n+1)!}{n!\tau^{2n+1}} \displaystyle\sum_{r=0}^{n} \frac{(-1)^{n-r}\tau^r (2n-r+1)!}{r!(n-r)!(2n-r+1)\Gamma(2n-r+2-\alpha)} & \\ \times t^{2n-r+1-\alpha} & \text{if } 0 \le t \le \tau \\ \dfrac{(2n+1)!}{n!\tau^{2n+1}} \displaystyle\sum_{r=0}^{n} \frac{(-1)^{n-r}\tau^r (2n-r+1)!}{r!(n-r)!(2n-r+1)} & \\ \times \left(\dfrac{t^{2n-r+1-\alpha}}{\Gamma(2n-r+2-\alpha)} - \displaystyle\sum_{j=0}^{n-r} \frac{\tau^j (t-\tau)^{2n-r+1-j-\alpha}}{j!\Gamma(2n-r+2-j-\alpha)} \right) & \text{if } t > \tau. \end{cases} \tag{7.19}$$

It is worth stressing that the previous equation can also be used for a direct computation of the transition polynomial by selecting $\alpha = 0$.

Finally, the convolution integral (7.16) has to be solved. In $[0, \tau]$, considering that the Laplace transform of the convolution integrals equals the product of the Laplace transforms and that $\mathscr{L}[t^\alpha] = \Gamma(\alpha+1)\frac{1}{s^{\alpha+1}}$, starting from (7.19) and using the same procedure employed to obtain (7.16), the convolution integral (7.16) itself can be derived as an explicit expression in terms of Mittag-Leffler functions (7.12).

For $t > \tau$ a similar result is achievable by considering that the transition polynomial (7.19) can be represented as the summation of a polynomial and a delayed one. Hence, the same reasoning previously applied can be used by considering that $\mathscr{L}[(t-\tau)^\alpha] = \Gamma(\alpha+1)\frac{1}{s^{\alpha+1}} e^{-\tau s}$, that is, the integration of a polynomial function, possibly delayed, that can be solved again in terms of Mittag-Leffler functions,

leading to

$$\int_0^t \eta_0(t-\xi)\bar{y}(\xi;\tau)d\xi = \sum_{i=1}^{m} \frac{g_i}{k_i!} \frac{(2n+1)!}{n!\tau^{2n+1}} \sum_{r=0}^{n} \frac{(-1)^{n-r}\tau^r}{r!(n-r)!(2n-r+1)} (2n-r+1)!$$
$$\times \left[\varepsilon_{k_i}(t, \lambda_i; \nu, 2n-r+2+\nu) \right.$$
$$\left. - \begin{cases} 0 & \text{if } 0 \le t \le \tau \\ \sum_{j=0}^{n-r} \frac{\tau^j}{j!} & \\ \times \varepsilon_{k_i}(t-\tau, \lambda_i; \nu, 2n-r+2-j+\nu) & \text{if } t > \tau \end{cases} \right]. \tag{7.20}$$

It is noteworthy that the computation of (7.10), by means of (7.20) only requires the computation of the Mittag-Leffler function, which is widely treated in the literature (see, for example, [86, 115]). Note that, in the fractional framework, this is a basic requirement since the Mittag-Leffler function plays for fractional systems the same role that the exponential function plays for integer systems.

7.2.4 Minimum-Time Transition

Consider now the constrained minimum-time transition problem (i.e., the second part of Problem 7.1). It has already been shown that the output constraints of (7.3) can always be satisfied by increasing τ. Now, consider the input constraints of (7.3), that is, given the system (7.1), determine the minimum time τ^* such that $u(t;\tau^*)$ and the first $l \in \mathbb{N}$ derivatives of $u(t;\tau^*)$ are bounded (see 7.3), that is ,

$$|D^i u(t;\tau^*)| < u_M^i, \quad \forall t > 0, \quad i = 0, 1, \ldots, l. \tag{7.21}$$

The condition under which the problem admits a solution is determined hereafter. First, consider the following lemma.

Lemma 7.2 *Assume that $n \ge [\rho] + l$. The input–output pair defined by (7.4) and (7.10) satisfies the following limits*

$$\begin{aligned} &\lim_{\tau\to\infty} \|H(0)u(\cdot;\tau) - \bar{y}(\cdot;\tau)\|_\infty = 0; \\ &\lim_{\tau\to\infty} \|D^i u(\cdot;\tau)\|_\infty = 0 \quad i = 1, \ldots, l. \end{aligned} \tag{7.22}$$

Proof Consider the first one of (7.22).

In $[0, \tau]$ the input computed according to (7.16) is a linear combination of terms $\varepsilon_{k_i}(t, \lambda_i; \nu, 2n-r+2+\nu)$.

Suppose that no multiple poles occur in $H_0(s)$ (i.e., $k_i = 0$); then, the following set of equalities, considering (7.11) and (7.12) holds:

$$\begin{aligned}
\varepsilon_0(t,\lambda;\nu,2n-r+2+\nu) &= t^{2n-r+1+\nu}\sum_{j=0}^{\infty}\frac{(\lambda t^{\nu})^j}{\Gamma((j+1)\nu+2n-r+2)}\\
&= t^{2n-r+1}\sum_{j=0}^{\infty}\frac{(\lambda t^{\nu})^{j+1}}{\Gamma((j+1)\nu+2n-r+2)}\\
&= \frac{t^{2n-r+1}}{\lambda}\left(\sum_{j=-1}^{\infty}\frac{(\lambda t^{\nu})^{j+1}}{\Gamma((j+1)\nu+2n-r+2)}-\frac{1}{\Gamma(2n-r+2)}\right)\\
&= \frac{t^{2n-r+1}}{\lambda}\left(\sum_{j=0}^{\infty}\frac{(\lambda t^{\nu})^{j}}{\Gamma(j\nu+2n-r+2)}-\frac{1}{(2n-r+1)!}\right)\\
&= \frac{t^{2n-r+1}}{\lambda}\left(E_{\nu,2n-r+2}(\lambda t^{\nu})-\frac{1}{(2n-r+1)!}\right).
\end{aligned} \tag{7.23}$$

Substituting (7.23) into (7.16) it immediately appears that, if no multiple poles occur, the convolution integral with $t \in [0,\tau]$ is

$$\begin{aligned}
\int_0^t \eta_0(t-\xi)\bar{y}(\xi;\tau)d\xi &= \sum_{i=1}^{m} g_i\frac{(2n+1)!}{n!\tau^{2n+1}}\sum_{r=0}^{n}\frac{(-1)^{n-r}\tau^r}{r!(n-r)!(2n-r+1)}\\
&\quad\times(2n-r+1)!\frac{t^{2n-r+1}}{\lambda_i}\left(E_{\nu,2n-r+2}(\lambda_i t^{\nu})-\frac{1}{(2n-r+1)!}\right)\\
&= H_0(0)\bar{y}(t;\tau)+(2n-r+1)!\frac{\bar{y}(t;\tau)}{\lambda_i}E_{\nu,2n-r+2}(\lambda_i t^{\nu}).
\end{aligned} \tag{7.24}$$

In view of the minimum-phase assumption made on Σ, $E_{\nu,2n-r+2}(\lambda t^{\nu})$ is infinitesimal, moreover $\bar{y}(t;\tau)$ is bounded and monotonically increasing in $[0,\tau]$, thus

$$\lim_{\tau\to\infty}\bar{y}(t;\tau)\left(H_0(0)+\frac{(2n-r+1)!}{\lambda}E_{\nu,2n-r+2}(\lambda t^{\nu})\right) = H_0(0)\bar{y}(t;\tau). \tag{7.25}$$

Now consider the multiple poles case, it is

$$\begin{aligned}
\varepsilon_k(t,\lambda;\nu,2n-r+2+\nu) &= t^{(k+1)\nu+2n-r+1}\frac{d^k}{d(\lambda t^{\nu})^k}\sum_{j=0}^{\infty}\frac{(\lambda t^{\nu})^j}{\Gamma((j+1)\nu+2n-r+2)}\\
&= t^{(k+1)\nu+2n-r+1}\sum_{j=0}^{\infty}\frac{(j+k)!(\lambda t^{\nu})^j}{j!\Gamma((j+k+1)\nu+2n-r+2)}\\
&= t^{(k+1)\nu+2n-r+1}k!\sum_{j=0}^{\infty}\frac{(k+1)_j(\lambda t^{\nu})^j}{j!\Gamma((j+k+1)\nu+2n-r+2)},
\end{aligned} \tag{7.26}$$

where $(\zeta)_n := \zeta(\zeta+1)\cdots(\zeta+n-1)$, $(\zeta)_0 := 1$, $\zeta \neq 0$ is the Pochhammer symbol. The last term of (7.26), except for the term $k!$, is exactly the definition of generalized Mittag-Leffler function according to [65]:

$$E^{\zeta}_{\alpha,\beta}(t) := \sum_{j=0}^{\infty} \frac{(\zeta)_j (t)^j}{j!\Gamma(j\alpha+\beta)} \tag{7.27}$$

and the following property holds [65]

$$t E^{\zeta}_{\alpha,\beta}(t) = E^{\zeta}_{\alpha,\beta-\alpha}(t) - E^{\zeta-1}_{\alpha,\beta-\alpha}(t). \tag{7.28}$$

Via (7.27), it can be determined that

$$\varepsilon_k(t,\lambda;\nu,2n-r+1+\nu) = t^{(k+1)\nu+2n-r+1} k! E^{k+1}_{\nu,2n-r+2+(k+1)\nu}(\lambda t^{\nu}) \tag{7.29}$$

and by recursively applying (7.28) it results

$$\begin{aligned} &t^{(k+1)\nu+2n-r+1} k! E^{k+1}_{\nu,2n-r+2+(k+1)\nu}(\lambda t^{\nu}) \\ &= \frac{t^{\nu+2n-r+1}}{\lambda^k} k! \sum_{i=0}^{k} (-1)^i \binom{k}{i} E^{k+1-i}_{\nu,2n-r+2+\nu}(\lambda t^{\nu}). \end{aligned} \tag{7.30}$$

Now, by applying the same procedure used for the case with only simple poles to the last term of the summation in the right hand side of (7.30) and considering that $E^{1}_{\alpha,\beta}(t) = E_{\alpha,\beta}(t)$, it appears that the convolution integral for $t \in [0,\tau]$ can be rewritten as

$$\begin{aligned} \int_0^t \eta_0(t-\xi)\bar{y}(\xi;\tau)d\xi = &\sum_{i=1}^{m} \frac{g_i}{k_i!} \frac{(2n+1)!}{n!\tau^{2n+1}} \sum_{r=0}^{n} \frac{(-1)^{n-r}\tau^r}{r!(n-r)!(2n-r+1)} \\ &\times (2n-r+1)! k_i! \frac{t^{2n-r+1}}{\lambda_i^{k_i}} \\ &\times \left[(-1)^{k_i} \left(E_{\nu,2n-r+2}(\lambda_i t^{\nu}) - \frac{1}{(2n-r+1)!} \right) \right. \\ &\left. + t^{\nu} \sum_{j=0}^{k_i-1} (-1)^j \binom{k}{j} E^{k_i+1-j}_{\nu,2n-r+2+\nu}(\lambda_i t^{\nu}) \right]. \end{aligned} \tag{7.31}$$

Now, by backward applying (7.26) it is easy to obtain that

$$E^{k+1-j}_{\nu,2n-r+2+\nu}(\lambda t^{\nu}) = \frac{1}{(k-j)!} \frac{d^{(k-j)}}{d(\lambda t^{\nu})^{(k-j)}} E_{\nu,2n-r+2+(1-k)\nu}(\lambda t^{\nu}).$$

Substituting the previous equation in (7.31) and using (7.11)–(7.13), the last term of (7.31) can be rewritten as

$$\frac{(k-1)!}{\Gamma(2n-r+2-k\nu)}\sum_{j=0}^{k-1}(-1)^j\binom{k}{j}\frac{1}{t^{2n-r+1-j\nu}}$$
$$\times\mathscr{L}^{-1}\left[\mathscr{L}\left[t^{2n-r+1-k\nu}\right]\frac{1}{(s^\nu+\lambda)^{k-j+1}}\right] \tag{7.32}$$

that, evidently, is infinitesimal considering that $j < k$ and the minimum-phase assumption (i.e., a stable zero-order dynamics).

Using the previous result in (7.31) it is immediate to check that

$$\lim_{\tau\to\infty}\int_0^t \eta_0(t-\xi)\bar{y}(\xi;\tau)d\xi = H_0(0)\bar{y}(t;\tau) \quad t\in[0,\tau], \tag{7.33}$$

also when multiple poles occur. Using (7.19), it can be seen that the derivatives of the transition polynomial monotonically decrease when τ increases, thus

$$\begin{aligned}\lim_{\tau\to\infty}\ &\gamma_{n-m}D^\rho\bar{y}(t;\tau)+\gamma_{n-m-1}D^{\rho-\nu}\bar{y}(t;\tau)+\cdots\\ &+\gamma_1D^\nu\bar{y}(t;\tau)+\gamma_0\bar{y}(t;\tau)=\gamma_0\bar{y}(t;\tau).\end{aligned} \tag{7.34}$$

The existence of the derivatives is guaranteed by the hypothesis $n \geq [\rho]+l$.

Considering the previous equation together with (7.10) and considering that $H(0)=(H_0(0)+\gamma_0)^{-1}$ the first of (7.22) is proven for in $t\in[0,\tau]$.

Consider now the second expression of (7.22). Again, using (7.19) it is easy to see that the maximum of the derivatives of $\bar{y}(t;\tau)$ is monotonically decreasing as τ increases. Hence, except for the convolution integral (7.16), the first l derivatives of (7.10) are evidently decreasing as τ increases. Thus, it is sufficient to prove that the convolution integral derivatives are infinitesimal with respect to τ. According to (7.4), $|D^i\bar{y}(t;\tau)|$=0 outside $[0,\tau]$. By virtue of the properties of the convolution integral and by taking into account Lemma 7.1 it holds that

$$\left|D\int_0^t \eta_0(t-\xi)\bar{y}(\xi;\tau)d\xi\right| = \left|\int_0^t \eta_0(t-\xi)D\bar{y}(\xi;\tau)d\xi\right| < \frac{c_{ki}}{\tau^i}\int_0^t|\eta_0(t-\xi)|d\xi. \tag{7.35}$$

In view of the minimum-phase assumption, the last integral is bounded, thus

$$\lim_{\tau\to\infty}\frac{c_{ki}}{\tau^i}\int_0^t|\eta_0(t-\xi)|d\xi=0. \tag{7.36}$$

By iteratively applying the reasoning to the higher order derivatives, the second expression of (7.22) is proven.

Finally, the first expression of (7.22) is proven for $t>\tau$. In view of (7.33) and (7.34), for τ sufficiently large, the output of the convolution integral is arbitrarily

close to the steady-state value as well as its derivatives. Considering that the zero-order dynamics is a stable linear system whose input is constant for $t > \tau$ and whose initial condition can be arbitrarily close to the steady-state value at $t = \tau$, it can be seen that, also for $t \geq \tau$ the output of the systems, i.e., the convolution integral value, can be arbitrarily close to the steady state.

Finally, observing again that the maximum of the derivatives of $\bar{y}(t; \tau)$ is monotonically decreasing as τ increases the second of (7.22) is proven also for $t > \tau$ and this completes the proof.

Finally, the following theorem states that the problem admits a solution under very mild conditions.

Theorem 7.1 *The constrained minimum-time transition problem admits a solution provided that*

$$\begin{array}{c} u_M^0 > \frac{1}{H(0)}, \quad u_M^i > 0, \quad i = 1, \ldots, l \\ y_M^i > 0 \quad i = 1, \ldots, v. \end{array} \tag{7.37}$$

Proof The result can be obtained by considering Lemmas 7.1 and 7.2.

Roughly speaking, the previous theorem states that it is just required that the constraints on the maximum input do not prevent the input to keep the system output constant at the desired steady-state value.

Finally, consider both the constraints (7.3). Lemma 7.1 shows that it is sufficient to select $n \geq v - 1$ in (7.4) and to increase the transition time τ in order to decrease the maximum value of the first v derivatives of the transition polynomial. In other words, it is enough to increase τ until the second condition of (7.3) is satisfied.

Analogously, Lemma 7.2 states that the first of (7.3) can be satisfied, provided that $n \geq [\rho] + l$. Thus, in order to satisfy all the constraints (7.3), the desired output function can be selected by choosing

$$\begin{cases} n \geq \max\{v - 1; [\rho + l\} \\ \tau \geq \max\{\tau_i^*; \tau_o^*\}, \end{cases} \tag{7.38}$$

where τ_o^* is the minimum transition time satisfying the output constraints (which can be computed explicitly by following the techniques proposed in [111]), whereas τ_i^* is minimum transition time such that the input constraints are satisfied for each $\tau \geq \tau_i^*$. For this purpose, a simple bisection algorithm can be suitably employed.

7.2.5 Illustrative Example

As an illustrative example, consider the fractional system Σ_1 with the following transfer function:

$$H(s) = \frac{3s^{0.5} + 1}{s^{1.5} + 1}, \tag{7.39}$$

whose commensurate order is evidently 0.5. A smooth transition of the output from 0 to 1 is desired, with both constraints on the input value and on its slew rate (note that this is a common requirement in practical applications). Accordingly, considering that the relative order of the system is $\rho = 1$ and that a constraint is imposed on the input signal first derivative, a transition polynomial of order $n \geq 1$ is needed. Here, the sufficient degree of regularity $n = 3$ has been chosen and the transition polynomial $\bar{y}(t; \tau)$ has been computed via (7.4):

$$\bar{y}(t; \tau) = -\frac{20}{\tau^7}t^7 + \frac{70}{\tau^6}t^6 - \frac{84}{\tau^5}t^5 + \frac{35}{\tau^4}t^4. \tag{7.40}$$

Then, the input–output inversion technique has been applied. The zero dynamics of Σ_1 has been first computed

$$H_0(s) = \frac{0.9630}{3s^{0.5} + 1} \tag{7.41}$$

and its time-domain representation

$$\eta_0(t) = \frac{0.9630}{3}\varepsilon_{k_i}\left(t, \frac{1}{3}; 0.5, 0.5\right). \tag{7.42}$$

Then, the input–output-inversion-based input function has been computed via (7.10), (7.19) and (7.20):

$$u(t; \tau) = 0.3333D^1\bar{y}(t; \tau) - 0.1111D^{0.5}\bar{y}(t; \tau) + 0.0370 + \int_0^t \eta_0(t - \xi)\bar{y}(\xi; \tau)d\xi. \tag{7.43}$$

Note that, being $\gamma_0 = 0.0370$ and $H_0(0) = 0.9630$, as expected $\gamma_0 + H_0(0) = H(0)^{-1} = 1$.

Now, consider the following set of constraints on the input variable:

$$u_M^0 \leq 1.5, \quad u_M^1 \leq 1 \tag{7.44}$$

A bisection algorithm [111] has been applied resulting in the minimum value of the transition time $\tau^* = 1.55$. Numerically computing $u(t; \tau)$ with the obtained transition time τ^* and simulating the system response by using the Oustaloup approximation [89], the result shown in Fig. 7.1 has been obtained.

As expected, the system response is smooth and monotonic and the input variable is smooth and satisfies the constraints (7.44). In this case, the input slew rate constraints is the tightest one and imposes severe restrictions on the transition time.

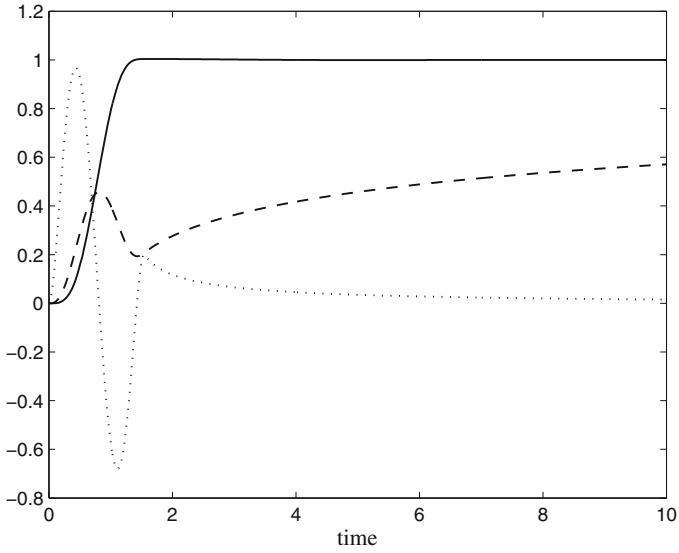

Fig. 7.1 Results for the system Σ_1 and the set of constraints (7.44). *Solid line* input–output inversion response. *Dashed line* input variable. *Dotted line* input variable slew rate

In order to speed up the system response, the constraints on the input slew rate have been relaxed and the one on the maximum input has been tightened in order to be very close to the minimum allowable value (that is $H(0)^{-1}$) resulting in

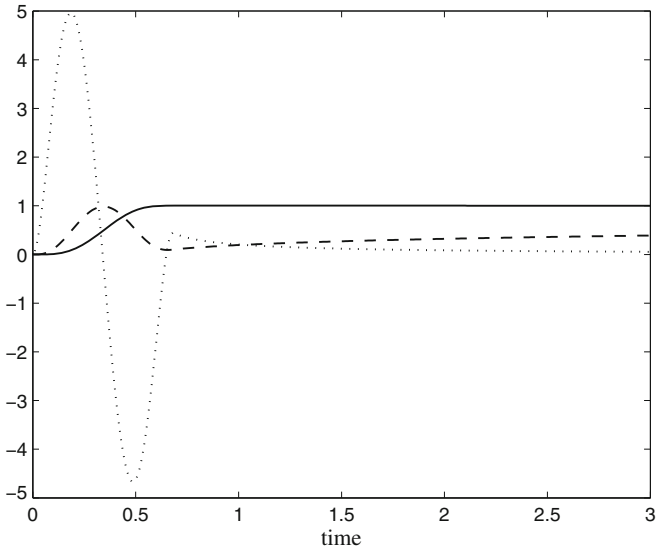

Fig. 7.2 Results for the system Σ_1 and the set of constraints (7.45). *Solid line* input–output inversion response. *Dashed line* input variable. *Dotted line* input variable slew rate

$$u_M^0 \leq 1.001, \quad u_M^1 \leq 6 \tag{7.45}$$

The bisection algorithm results in a value of the transition time $\tau^* = 0.55$. A new simulation has been performed recomputing the input signal, the result is shown in Fig. 7.2.

Now, as expected, the tightest constraint is the one imposed on the maximum control variable. Note that, despite the sensible reduction of the transition time, the system response remains smooth and monotonic.

Finally, for the sake of comparison, in Fig. 7.3, the system response to a ramp is shown. The ramp is defined in such a way that satisfies the second set of input constraints (7.45). The response is evidently worse than the one obtained via input–output inversion (note that, obviously, the step response would present an even larger overshoot and settling time).

It is worth noting the occurrence of a sluggish response of the system, which is caused by the memory effect of the fractional operator. In this context, the use if the inversion methodology appears to be a valuable tool to compensate for this effect.

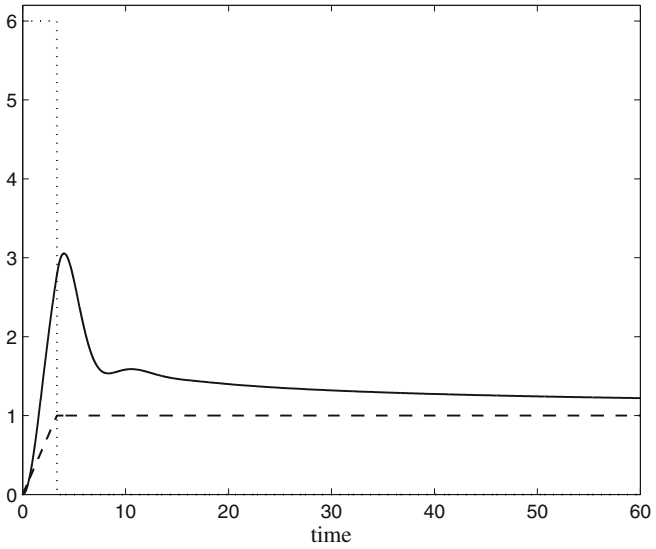

Fig. 7.3 Ramp response for the system Σ_1 and the set of constraints (7.44). *Solid line* process output. *Dashed line* input variable. *Dotted line* input variable slew rate

7.3 Feedforward Control Design

7.3.1 Problem Formulation

The inversion procedure described in the previous section can be effectively used in order to design a feedforward control law that improves the set point regulation of a standard feedback control scheme. In fact, it is well-known that the main purpose of a feedback controller is to compensate for external disturbances and to reduce the effect of modeling uncertainties and the addition of a feedforward control law can improve the set point following performance of a control system [110, 112]. In this context, the problem of achieving a process variable transition from one steady-state value to another one is addressed by assuming that the feedback controller has been already designed. Two different cases are considered: the synthesis of a feedforward signal to be added to the feedback control variable and the synthesis of a command signal to be applied to the closed-loop control system instead of the typical set point step signal.

The problem can be formulated as follows (similarly to the case considered in the previous section). Consider the unity feedback control scheme of Fig. 7.4, where $C(s)$ is a linear time-invariant commensurate fractional controller and

$$G(s) = \bar{G}(s)e^{-Ls} \tag{7.46}$$

is a linear time-invariant strictly proper fractional system $G(s)$, where L is a possible delay term and $\bar{G}(s)$ is minimum phase.

The closed-loop transfer function is

$$T(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} \tag{7.47}$$

and is assumed to be strictly proper. Note that the controller can be also improper from a theoretical point of view in order to make the approach suitable for a wider range of regulators (e.g., PID controllers). In fact, in many cases the controller can

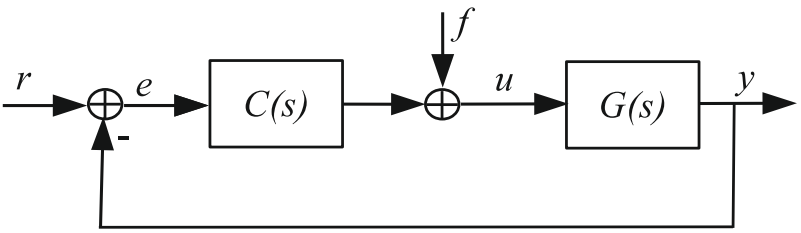

Fig. 7.4 The unity feedback control scheme considered in the feedforward control synthesis problem

be effectively treated and designed as an improper system, even if in practice it has to be made proper in its implementation (possibly by adding high-frequency poles that can be neglected in the design phase). It is also assumed that the controller has been designed in order to make the considered feedback loop internally stable.

The aim of the design method is to find suitable signals for the loop command signal $r(t)$ or for the feedforward signal $f(t)$ in order to obtain a perfect tracking of a desired output which allows a transition from an initial steady-state value to a new one in a finite time interval τ, given a set of bounds on the control and process variables and their derivatives.

Thus, starting from null initial conditions and given a steady-state output value y_e, design a "sufficiently smooth" τ-parametrized desired output $\bar{y}(\cdot;\tau)$ such that $\bar{y}(0;\tau) = 0$ and $\bar{y}(t;\tau) = y_e \ \forall t \geq \tau$, and $\bar{y}(\cdot;\tau) \in C^{(k)}$ for some $k \in \mathbb{N}$. Then, find

1. (feedforward signal generation) $f(\cdot;\tau)$ and $r(\cdot;\tau)$ such that, for the τ-parametrized functions $f(\cdot;\tau)$, $r(\cdot;\tau)$ and $\bar{y}(\cdot;\tau)$, it holds that

$$\mathscr{L}[\bar{y}(t-L;\tau)] = G(s)\mathscr{L}[f(t;\tau))] \tag{7.48}$$

   and $y(t) = \bar{y}(t-L;\tau) \ \forall t \geq 0$; or
2. (command signal generation) $r(\cdot;\tau)$ such that, for the τ-parametrized couple $(r(\cdot;\tau);\bar{y}(\cdot;\tau))$, it holds that

$$\mathscr{L}[\bar{y}(t-L;\tau)] = T(s)\mathscr{L}[r(t;\tau))]. \tag{7.49}$$

Moreover, in both cases, determine the minimum time τ^* such that $u(t;\tau^*)$ and the first $l \in \mathbb{N}_0$ ($v \in \mathbb{N}$, respectively) derivatives of $u(t;\tau^*)$ ($\bar{y}(t;\tau^*)$), are bounded:

$$\begin{aligned} |D^i u(t;\tau^*)| &< u_M^i, \ \forall t > 0, \ i = 0, 1, \ldots, l; \\ |D^i \bar{y}(t;\tau^*)| &< y_M^i, \ \forall t > 0, \ i = 1, 2, \ldots, v. \end{aligned} \tag{7.50}$$

Note that in the first case, since a perfect tracking is obtained, the controller output is null and the feedforward signal $f(t;\tau)$ coincides with the control signal $u(t)$, whereas in the second case the control variable $u(t)$ and the controller output $c(t)$ coincide since no feedforward action is considered.

7.3.2 Feedforward Signal Synthesis

In this section, the problem of feedforward regulation is addressed. Consider the (possibly unstable) process (7.46). First, the delay-free part $\bar{G}(s)$ is considered. Given the transition polynomial $\bar{y}(t;\tau)$ (see 7.4), the corresponding feedforward signal $f(t;\tau)$ can be easily computed via Proposition 7.1 together with (7.19) and (7.20), provided that

$$n \geq [\rho_{\bar{G}}], \tag{7.51}$$

being $\rho_{\bar{G}}$ the relative order of $\bar{G}(s)$.

It is easy to see that (7.48) is satisfied. Indeed, (7.48) can be rewritten as

$$\bar{Y}(s;\tau)e^{-Ls} = G(s)F(s;\tau) \tag{7.52}$$

and, by substituting (7.46) into the previous equation and eliminating the delay terms from both sides the following expression is obtained

$$\bar{Y}(s;\tau) = \bar{G}(s)F(s;\tau). \tag{7.53}$$

The previous equation holds by construction because $f(t;\tau)$ is computed via input–output inversion of $\bar{G}(s)$, thus (7.48) is satisfied.

Note that the inversion-based signal does not excite the system modes, since it is computed as the convolution of the transition polynomial (7.4) with the inverse system $\bar{G}(s)^{-1}$. Thus, it is enough to have null control error to obtain the perfect tracking condition $y(t) = \bar{y}(t-L;\tau)$. Considering that (7.48) holds, it is then enough to use the desired output as the closed-loop reference signal r in order to obtain a null error, that is, the perfect tracking is obtained by imposing:

$$r(t;\tau) = \bar{y}(t-L;\tau). \tag{7.54}$$

Now, the constraints (7.50) can be satisfied by considering (7.38) with $\rho = \rho_{\bar{G}}$, that is,

$$\begin{cases} n \geq \max\{v-1, [\rho_{\bar{G}}]+l\} \\ \tau \geq \max\{\tau_i^*; \tau_o^*\} \end{cases} \tag{7.55}$$

and by increasing the transition time τ until the required conditions are met.

7.3.3 Command Input Synthesis

An alternative approach to the one proposed in the previous section is to compute a suitable command signal $r(t)$ to be applied to the closed-loop system in order to obtain a perfect tracking of the desired output. In this context, a double strategy can be developed, one for the case where no dead time is present in the process dynamics and on for processes with time delay.

Consider first the delay-free case. The closed-loop transfer function $T(s)$, accordingly to (7.47), can be expressed as a fractional transfer function. Hence, the input–output procedure described in Sect. 7.2.3 can be straightforwardly applied to $T(s)$ obtaining a suitable command signal $r(t;\tau)$ that satisfies (7.49), provided that

$$n \geq [\rho_T], \tag{7.56}$$

where ρ_T is the relative order of $T(s)$. It is the product of the relative degree of $\bar{G}(s)$ and the (possibly negative) relative degree ρ_C of the controller $C(s)$. The point here is that the existence of a suitable command signal does not guarantee the existence of a bounded control signal. Indeed, when the controller is not proper, i.e., when $\rho_C < 0$, the following condition for the existence of the control signal must also be considered:

$$n \geq [\rho_{\bar{G}}]. \tag{7.57}$$

Conversely, when the controller is strictly proper, it may happen that the user is forced to use an overregularized control signal $u(t)$ in order to have a feasible command signal $r(t;\tau)$. Note that, thanks to (7.56), the condition (7.57) is always satisfied when $\rho_C \geq 0$. Also note that it is enough to consider the command signal in order to check the existence of the control signal. Indeed, the feedback signal is $y(t) = \bar{y}(t;\tau)$ and its regularity is greater than the one of $r(t;\tau)$ because of the properness of $T(s)$.

Regarding the constraints (7.50), it is sufficient to note that since in view of (7.49) a perfect tracking condition is obtained, this implies that $u(t) = \bar{G}(s)^{-1}\bar{y}(t;\tau)$ and the same reasoning of the previous section can be applied, leading again to (7.55), where $u(t) = u(t;\tau)$ can be computed via input–output inversion of $\bar{G}(s)$. It is worth stressing that constraints satisfaction automatically implies (7.57).

Finally, the case related to processes with dead time is considered. When a delay term is present, the closed-loop system cannot be inverted because of the exponential term appearing in the denominator of the transfer function. In this case, the command signal is calculated as the summation of two different terms. First, consider the open-loop transfer function. Neglecting the delay term, $C(s)\bar{G}(s)$ can be inverted by using the same procedure employed to invert $T(s)$ in the delay-free case, yielding the signal $r_{\text{ol}}(t;\tau)$. Then, a correction term $r_c(t;\tau) = \bar{y}(t-L;\tau)$ must be considered, so that the command signal is

$$r(t;\tau) = r_{\text{ol}}(t;\tau) + r_c(t;\tau). \tag{7.58}$$

It is easy to see that command signal (7.58) satisfies (7.49), indeed

$$Y(s) = T(s)r(t;\tau) = \frac{C(s)\bar{G}(s)e^{-Ls}}{1+C(s)\bar{G}(s)e^{-Ls}}(R_{\text{ol}}(s;\tau) + R_c(s;\tau)), \tag{7.59}$$

that is,

$$Y(s)\left(1 + C(s)\bar{G}(s)e^{-Ls}\right) = C(s)\bar{G}(s)e^{-Ls}\left((C(s)\bar{G}(s))^{-1} + e^{-Ls}\right)\bar{Y}(s;\tau). \tag{7.60}$$

Simplifying and applying the inverse Laplace transform to the previous expression, it is immediately evident that $y(t) = \bar{y}(t-L;\tau)$, that evidently implies (7.49).

Regarding the constraints, considering that the relative order of the open-loop transfer function is the same of the closed-loop one, the reasoning used in the delay-

free case holds straightforwardly and the problem of the command signal synthesis subject to constraints (7.55) and (7.56) is completely solved.

Note that here both the reference signal and the control signal (in order to find τ_i^*) must be computed. Nevertheless, the computational weight increment is minimal because, once a suitable value of n has been chosen, only the control signal must be computed repeatedly. Indeed, the command signal has to be computed just once when τ_i^* and τ_o^* have been already found.

7.3.4 Illustrative Examples

In all the examples presented hereafter, a process variable transition from 0 to 1 is considered. For the purpose of simulation, the fractional-order dynamics has been approximated in the frequency domain by using the Oustaloup approximation [89] as in Sect. 7.2.5.

7.3.4.1 Example 1

As a first illustrative example, consider an unstable fractional system with the following transfer function:

$$C(s) = \frac{3s^{0.5} + 1}{s^{1.5} - 1} e^{-0.1s}, \tag{7.61}$$

whose commensurate order is evidently 0.5. A very simple stabilizing controller can be used, indeed the achievement of the perfect tracking is independent from the chosen controller. Here, a proportional controller $C(s) = 2$ has been chosen. The control requirement is to obtain a smooth transition of the output from 0 to 1 with constraints on both the amplitude and the first derivative (slew rate) of control and process variables (note that these are common requirements in practical applications). Accordingly, considering that the relative order of the system $\bar{G}(s)$ is $\rho_{\bar{G}} = 1$, the choice $n = 3$ has been done and the transition polynomial $\bar{y}(t; \tau)$ has been computed via (7.4):

$$\bar{y}(t; \tau) = -\frac{20}{\tau^7} t^7 + \frac{70}{\tau^6} t^6 - \frac{84}{\tau^5} t^5 + \frac{35}{\tau^4} t^4. \tag{7.62}$$

Then, the technique proposed in Sect. 7.3.2 has been applied. The zero dynamics of $\bar{G}(s)$ has been first obtained as

$$H_0(s) = \frac{-1.0370}{3s^{0.5} + 1} \tag{7.63}$$

and its time-domain representation is

$$\eta_0(t) = \frac{-1.0370}{3}\varepsilon_{k_i}\left(t, \frac{1}{3}; 0.5, 0.5\right). \tag{7.64}$$

Subsequently, the inversion-based feedforward signal $f(t;\tau)$ (note that it coincides with the control variable) has been computed via (7.10):

$$f(t;\tau) = 0.3333D^1\bar{y}(t;\tau) - 0.1111D^{0.5}\bar{y}(t;\tau) + 0.0370 + \int_0^t \eta_0(t-\xi)\bar{y}(\xi;\tau)d\xi. \tag{7.65}$$

Now consider the following set of constraints:

$$\begin{gathered} u_M^0 \leq 1.5, \quad u_M^1 \leq 5 \\ y_M^1 \leq 2 \end{gathered} \tag{7.66}$$

The bisection algorithm results in the minimum value of the transition time $\tau_i^* = 0.72$ while, in order to satisfy the constraint on the output signal, $\tau_o^* = 1.12$ has been obtained. The tightest constraint is the one on the derivative of the output signal, therefore $\tau^* = \tau_o^* = 1.12 > \tau_i^*$. By numerically computing $u(t;\tau)$ with the obtained transition time τ^* and simulating the system response, the result shown in Fig. 7.5 has been obtained. As expected, the system response is smooth and monotonic and the input variable is smooth and satisfies the constraints (7.66). Note that the system

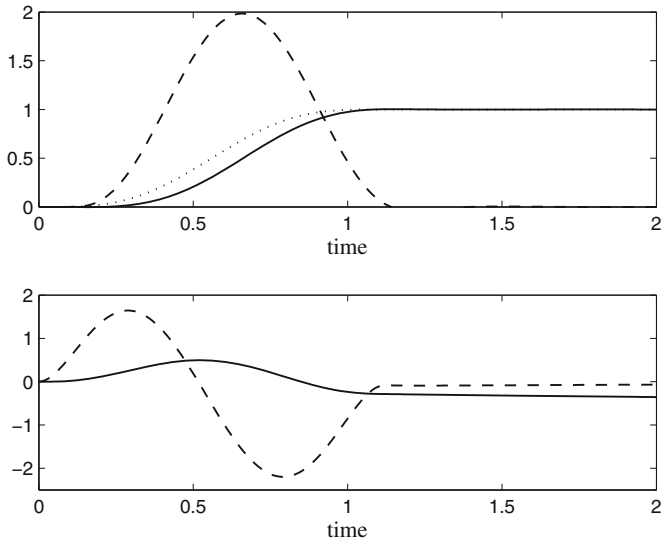

Fig. 7.5 Results for the system (7.61) and the set of constraints (7.66). *Top*, *solid line* process output. *Top*, *dashed line* derivative of the process output. *Top*, *dotted line*: transition polynomial. *Bottom*, *solid line* control variable. *Bottom*, *dotted line* control variable first derivative

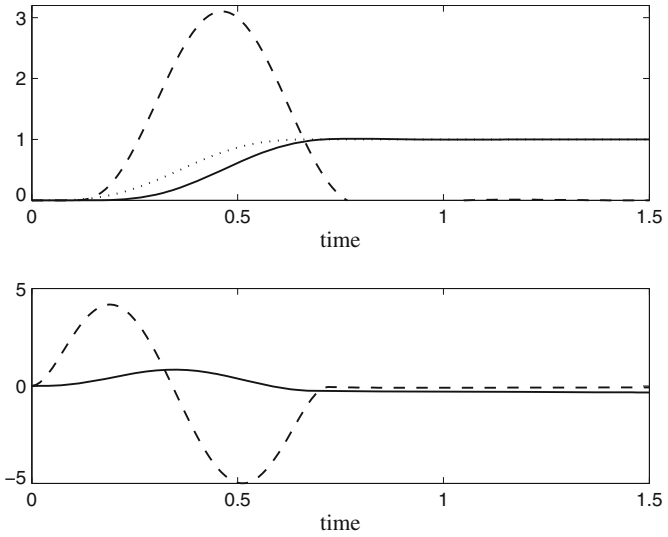

Fig. 7.6 Results for the system (7.61) and the set of constraints (7.67). *Top*, *solid line* process output. *Top*, *dashed line* derivative of the process output. *Top*, *dotted line* transition polynomial. *Bottom*, *solid line* control variable. *Bottom*, *dotted line* control variable slew rate

output and the transition polynomial have that same shape, except for the time shift depending on the process delay.

In order to speed up the system response, the constraints on the output slew rate have been then relaxed by selecting

$$\begin{array}{c} u_M^0 \leq 1.5, \quad u_M^1 \leq 5 \\ y_M^1 \leq 5 \end{array} \tag{7.67}$$

The previously computed value $\tau^* = \tau_i^* = 0.72$ can here be used as in this case the constraint on the output slew rate is satisfied. The results of a new simulation performed by recomputing the input signal are shown in Fig. 7.6. Now, as expected, the tightest constraint is the one imposed on the control variable slew rate.

7.3.4.2 Example 2

As a second example, consider a unity feedback control system where the process and the controller are the ones proposed in [73]. The controlled process has the transfer function (note that it is a typical servomotor transfer function)

$$G(s) = \frac{0.25}{s(s+1)} \tag{7.68}$$

and the proposed controller is a fractional-order PID controller whose transfer function is

$$C(s) = 3.8159 + \frac{2.1199}{s^{0.6264}} + 2.2195 s^{0.809}. \tag{7.69}$$

Aiming at improving the closed-loop response via input–output inversion, first note that the system is already commensurate because the exponents appearing in $C(s)$ have a finite number of digits, thus they are rational number and the proposed algorithm could be directly applied. Nevertheless, if the commensurate order is very small, then the resulting algorithm is computationally demanding, thus it is convenient to approximate the controller by means of a simpler commensurate-order system, that is:

$$\tilde{C}(s) = 3.8159 + \frac{2.1199}{s^{0.6}} + 2.2195 s^{0.8}. \tag{7.70}$$

Hence, the approximated closed-loop system to be inverted is

$$\tilde{T}(s) = \frac{\tilde{C}(s)G(s)}{1 + \tilde{C}(s)G(s)} = \frac{0.5549 s^{1.4} + 0.9540 s^{0.6} + 0.5300}{s^{2.6} + s^{1.6} + 0.5549 s^{1.4} + 0.9540 s^{0.6} + 0.5300}, \tag{7.71}$$

whose commensurate order is $\nu = 0.2$.

Here, only constraints on the maximum control variable have been considered:

$$u_M^0 \leq 10. \tag{7.72}$$

Note that, in the case of a servomotor, this is a common choice that means avoiding saturation of the current loop. In order to select the transition polynomial, it must be considered that relative order of the approximate closed-loop transfer function is $\rho_{\tilde{T}} = 1.2$, while the relative order of the system is $\rho_{\tilde{G}} = 2$. Applying (7.55) and (7.56), the value $n = 2$ can be selected. This choice leads to the following transition polynomial

$$\bar{y}(t;\tau) = \frac{6}{\tau^5} t^5 - \frac{15}{\tau^4} t^4 + \frac{10}{\tau^3} t^3. \tag{7.73}$$

The transition time $\tau^* = \tau_i^*$ can be obtained by applying a bisection algorithm, yielding $\tau^* = 1.73$.

Now, applying to (7.71) the input–output inversion technique, again by polynomial division the zero-order dynamics of the system has been computed:

$$
\begin{aligned}
H_0(s) = & \frac{-1.2369 + 1.4054i}{s^{0.2} + 0.8240 - 0.8901i} + \frac{-1.2369 - 1.4054i}{s^{0.2} + 0.8240 + 0.8901i} \\
& + \frac{0.2838 + 0.3133i}{s^{0.2} - 0.8638 - 0.7011i} + \frac{0.2838 - 0.3133i}{s^{0.2} - 0.8638 + 0.7011i} \\
& + \frac{0.0743 - 0.1112i}{s^{0.2} - 0.3462 - 0.7479i} + \frac{0.0743 + 0.1112i}{s^{0.2} - 0.3462 + 0.7479i} \\
& + \frac{0.0362}{s^{0.2} + 0.7721}.
\end{aligned} \tag{7.74}
$$

Note that the complex coefficients always appear in conjugate pairs. Also, note that $H_0(s)$ is stable, even though some denominator coefficients have real part lower than zero. Indeed, it is worth stressing that in the fractional case, the stability condition is different from the integer case (see 2.24). The time-domain version $\eta_0(t)$ of $H_0(s)$ is not shown for the sake of readability, nevertheless the reader can easily obtain it by means of (2.6). The command signal $r(t;\tau)$ has been computed via (7.10) as

$$
\begin{aligned}
r(t;\tau) = & \, 1.8022 D^{1.2}\bar{y}(t;\tau) - 3.0985 D^{0.4}\bar{y}(t;\tau) \\
& + 1.8022 D^{0.2}\bar{y}(t;\tau) + \bar{y}(t;\tau) + \int_0^t \eta_0(t-\xi)\bar{y}(\xi;\tau)d\xi
\end{aligned} \tag{7.75}
$$

The application of the computed command input to the actual system $T(s)$ has given the result shown in Fig. 7.7, where also the step response is plotted for comparison.

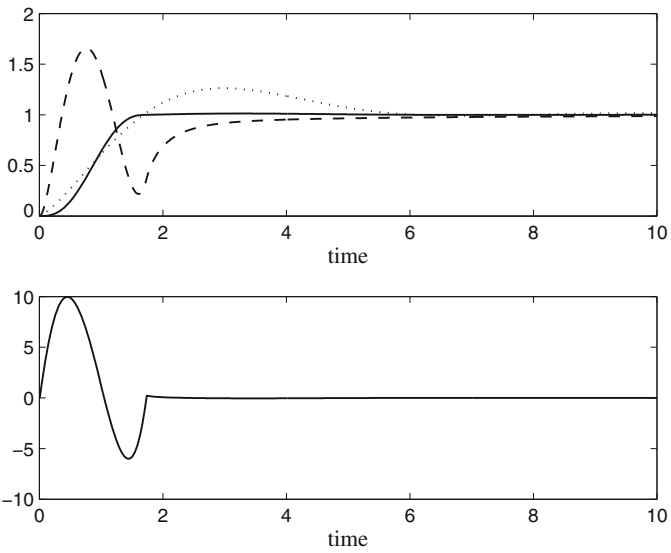

Fig. 7.7 Results for the system (7.71) and the set of constraints (7.72). *Top*, *solid line* process output. *Top*, *dashed line* command input. *Top*, *dotted line* step response. *Bottom*, *solid line* control variable

It is worth noting that the use of the inversion-based command signal, instead of the step one, provides a significant improvement of the performance despite the fractional-order PID controller is already very well tuned. Also, note that the proposed technique gives a continuous control signal, whereas the control signal obtained with the step response in not shown because it shows a very high peak due to the so called "derivative kick" phenomenon [151].

7.4 Combined Robust Feedback/Feedforward Design

7.4.1 Generalities

The inversion-based design for the feedforward action described in Sect. 7.3 can be effectively combined with the design of the feedback controller in order to minimize the worst-case settling time when a family of fractional first-order-plus-dead-time (FFOPDT) systems is considered. The approach consists in designing the feedback controller by following the methodology described in Chap. 6 with a modification of the weighting function in order to guarantee the properness of the controller. Then, the command input is designed by applying the input–output inversion procedure to the closed-loop system and, finally, the transition time τ is selected together with the parameters of the weighting function in order to minimize the worst-case settling time subject to constraints on the maximum overshoot and on the maximum amplitude of the control variable (obviously, also the robust stability condition has to be satisfied).

7.4.2 Feedback Control Design

Considering a nominal FFOPDT system, namely,

$$G(s) = \frac{K}{1 + Ts^{\alpha}} \mathrm{e}^{-Ls}, \quad \alpha \in (0, 2) \tag{7.76}$$

the approach described in Chap. 6 can be effectively employed in order to design the feedback controller. However, differently from (6.11), here an integer-order first-order system is chosen as a reference model:

$$M(s) = \frac{1}{1 + T_m s}, \tag{7.77}$$

Further, a simpler weighting function is used, that is,

$$W(s) = \frac{1 + zs^{\beta}}{s}, \quad \beta = \max\{1, \alpha\}. \tag{7.78}$$

This choice is done for the sake of simplicity and it is motivated by the fact that there is no need in this case to obtain a fractional-order PID controller as a feedback controller. Further, the weighting function (7.77) always guarantees a proper controller and a proper nominal real-rational closed-loop transfer function. Moreover, it automatically includes in the feedback loop an integer-order integral action $1/s$. Indeed, as shown in Chap. 3, in many cases the use of an integer integral action is the best solution.

The solution of the optimization problem (6.7) in this case yields

$$E^o(z_g) = W(z_g)M(z_g) = \delta(T_m, z) = \frac{L^\beta + z}{T_m L^{\beta-2} + L^{\beta-1}}, \tag{7.79}$$

where $E^o(s)$ is the optimal interpolation error.

Since a single RHP zero appears here, the optimal interpolation error is constant all over the complex plane and it holds that:

$$E^0(s) = \delta(T_m, z). \tag{7.80}$$

At this point, the optimal IMC controller (see Figs. 6.1 and 6.2) can be easily obtained as

$$\begin{aligned} C^o_{IMC}(s) &= (M(s) - W^{-1}(s)\delta)G_n^{-1}(s) \\ &= \frac{1}{K}\frac{(1 + zs^\beta - \delta T_m s^2 - \delta s)}{(T_m s + 1)(zs^\beta + 1)}\frac{Ts^\alpha + 1}{1 - Ls} \end{aligned} \tag{7.81}$$

and finally, the equivalent optimal feedback controller is obtained via (6.6) as:

$$C^o(s) = \frac{1}{K}\frac{1}{s}\frac{(1 + zs^\beta - \delta T_m s^2 - \delta s)}{\delta T_m s + zT_m s^\beta + \delta + T_m}\frac{Ts^\alpha 1 + 1}{Ls - 1}. \tag{7.82}$$

Stability is guaranteed by means of the controller $C^o(s)$ as long as the nominal real-rational process model

$$G_n(s) = K\frac{1 - Ls}{1 + Ts^\alpha} \tag{7.83}$$

perfectly describes the process dynamics. However, as mismatches between the actual process and the model used are unavoidable and because of the delay approximation (6.2), it is important to stabilize a family of plants around the nominal real-rational one.

Assume that the process belongs to a family $\mathfrak{F}$ defined as

$$\mathfrak{F} = \{G(s) = G_n(s)(1 + \Delta_m(s)) : |\Delta_m(j\omega)| \leq |\Gamma(j\omega)|\}, \tag{7.84}$$

where

$$\Delta_m(s) = \frac{G(s) - G_n(s)}{G_n(s)} \tag{7.85}$$

is the uncertainty description and $\Gamma(j\omega)$ is a weight that defines the plants family by upper bounding the modeling error. It is well-known that a controller that stabilizes a control system in the nominal case, also stabilizes the family $\mathfrak{F}$ of control systems such that:

$$\|\Gamma(s)T_n(s)\|_\infty < 1, \tag{7.86}$$

where $T_n(s)$ is the stable real-rational nominal closed-loop transfer function, namely:

$$C^o_{IMC}(s)G_n(s) = \frac{1 + zs^\beta - \delta T_m s^2 - \delta s}{(T_m s + 1)(zs^\beta + 1)}. \tag{7.87}$$

The following lemma states a sufficient condition under which robust stability can be obtained.

Lemma 7.3 *There exists a couple of parameters* $(T_m; z)$ *that satisfy the robust stability constraint* (7.86) *if* $|\Gamma(0)| < 1$ *and* $\Gamma(j\omega)$ *has degree of properness* $\rho_\Gamma \geq 1 - \max\{1, \alpha\}$.

Proof It is easy to see that

$$\lim_{T_m \to \infty} \delta(T_m, z) = 0 \tag{7.88}$$

and

$$\lim_{T_m \to \infty} T_m \delta(T_m, z) = L^2 + zL^{2-\beta}. \tag{7.89}$$

Now, consider the nominal real-rational closed-loop transfer function (7.87). Using the previous results, it can be checked that, for T_m sufficiently large, (7.87) becomes arbitrarily close to

$$\frac{1 + zs^\beta - (L^2 + zL^{2-\beta})s^2}{(T_m s + 1)(zs^\beta + 1)}. \tag{7.90}$$

For T_m and T_m/z sufficiently large, the absolute value of the previous expression can be arbitrarily small for all frequencies greater than an arbitrarily chosen value. Considering that the relative order of (7.87) is $\rho_{T_n} = \max\{1, \alpha\} - 1$, the inequality $|T_n(j\omega)| < \frac{1}{|\Gamma(j\omega)|}$ can be satisfied for $\omega \in [\bar{\omega}, \infty)$, where $\bar{\omega}$ is an arbitrarily small frequency.

Finally, considering that $T_n(0) = 1$ and, by hypothesis, $|\Gamma(0) < 1|$, it is clear that it is always possible to find a couple $(T_m; z)$ in order to satisfy (7.86).

7.4.3 Combined Tuning

In this section, the feedback controller design described in the previous subsection and the command input design described in Sect. 7.3.3 are combined to improve both performance (feedforward) and robustness (feedback) for an uncertain system by taking into account both of them simultaneously. Thus, the design of the feedback controller $C(s; T_m, z)$ and of the command signal $r(t; \tau)$ can be addressed by considering the following problem.

Consider a family of FFOPDT systems (for the sake of simplicity the same number l of possible values is considered for each uncertain parameter)

$$\mathfrak{F} = \left\{ \tilde{G}(s) = \frac{\tilde{K}}{\tilde{T}s^{\tilde{\alpha}} + 1} e^{-\tilde{L}s} : \tilde{K} \in \{K_1, \ldots, K_l\}, \right. \tag{7.91}$$
$$\left. \tilde{T} \in \{T_1, \ldots, T_l\}, \tilde{\alpha} \in \{\alpha_1, \ldots, \alpha_l\}, \tilde{L} \in \{L_1, \ldots, L_l\} \right\}; l \in \mathbb{N}$$

and the nominal system (7.76). The uncertainty set $\mathfrak{U}$ is defined as the set of uncertain systems $G_i(s)$ $i = 1, \ldots, l^4$ obtained with any possible combination of the uncertain parameters K_i, T_i, α_i and L_i and the family $\mathfrak{F}$ as

$$\mathfrak{F} = \left\{ \tilde{G}(s) : \tilde{G}(s) \in \mathfrak{U} \right\}. \tag{7.92}$$

The worst-case settling time (at a given percentage) is defined as the maximum settling time obtained by ranging all over the uncertain set $\mathfrak{U}$:

$$t_{s,wc}(\tau, T_m, z) := \max_{i=1,\ldots,l^4} t_{s,i}(\tau, T_m, z), \tag{7.93}$$

where $(\tau; T_m; z)$ is a triple of design parameters to be selected. Thus, the following min-max optimization problem can be posed.

Problem 7.2

$$\min_{\tau, T_m, z} t_{s,wc}(\tau, T_m, z) \tag{7.94}$$

subject to

1. (Robust stability) $\|\Gamma(s)T_n(s)\|_\infty < 1$,

$$\Gamma(j\omega) := \frac{G_i(j\omega) - G_n(j\omega)}{G_n(j\omega)}, \quad i = 1, \ldots, l^4; \tag{7.95}$$

2. (Maximum overshoot) $\max y_i(t; \tau, T_m, z) < y_f(1 + O_{\max})$, $i = 1, \ldots, l^4$;
3. (Maximum control variable) $\max |u_i(t; \tau, T_m, z)| < U_{\max}$, $i = i, \ldots, l^4$;

where $(u_i(\cdot); y_i(\cdot))$ is the input–output couple for the ith uncertain system, y_f is the set-point value, $O_{\max} >$ is the maximum allowable overshoot and $U_{\max} > 0$ the maximum acceptable control variable.

Note that here, since the controller is always proper and only a constraint on the maximum control variable is imposed (i.e., $v = 0$ and $l = 0$ in 7.50), the existence conditions (7.55) and (7.56) for control and command signals reduce to

$$n \geq [\rho_{C\tilde{G}}]. \tag{7.96}$$

The previous problem is quite complex and can only be solved analytically in trivial cases. Nevertheless, it can be approached numerically, provided the existence of an optimal solution. The next theorem states that an optimal solution indeed exists, under very reasonable conditions.

Theorem 7.2 *Problem 7.2 is solvable provided that*

$$\max_{\tilde{K}\in\{K_1,\ldots,K_l\}} \left| \frac{\tilde{K} - K}{K} \right| < 1 \tag{7.97}$$

and

$$U_{\max} > \max_{\tilde{K}\in\{K_1,\ldots,K_l\}} \left| \frac{y_f}{\tilde{K}} \right|. \tag{7.98}$$

Proof First consider that

$$\Gamma(j\omega) = \frac{\tilde{K}(T(j\omega)^{\alpha} + 1)e^{-\tilde{L}j\omega}}{K(\tilde{T}(j\omega)^{\tilde{\alpha}} + 1)(1 - Lj\omega)} - 1, \tag{7.99}$$

where each uncertain parameter, denoted by $\tilde{\cdot}$, respectively belongs to its uncertain set. Evidently, the smaller possible degree of properness is obtained when $\tilde{\alpha} = 0$ or $\tilde{T} = 0$ and its value is $\rho_{\Gamma} = 1 - \max\{1, \alpha\}$. Now consider (7.97): it implies that $|\Gamma(0) < 1|$. Hence, it is immediately evident that the hypotheses of Lemma 7.3 are satisfied and this guarantees the existence of T_m and z that satisfy the robust stability condition.

Now, consider the command signal $r(t;\tau)$ see (7.58) and suppose a unitary final value y_f. For the nominal systems it holds, by construction, that

$$\begin{aligned} R(s;\tau) = T^{-1}(s)\bar{Y}(s;\tau) &= \left(\tfrac{1}{C(s)G(s)e^{-Ls}} + 1\right)\bar{Y}(s;\tau)e^{-Ls} \\ &= (C(s)G(s))^{-1}\bar{Y}(s;\tau) + \bar{Y}(s;\tau)e^{-Ls} \\ &= R_{\text{ol}}(s;\tau) + R_c(s;\tau) \end{aligned} \tag{7.100}$$

Using Lemma 7.2, it turns out that $\lim_{\tau\to\infty} \|R_{\text{ol}}(t;\tau)\|_{\infty} = 0$ and $\lim_{\tau\to\infty} \|R(t;\tau)\|_{\infty} = \bar{y}(t - L;\tau)$. Further, since for the nominal process a perfect tracking is obtained and each input–output pair for a given system is unique, it holds that $\lim_{\tau\to\infty} \|u(t;\tau)\|_{\infty} = \frac{y_f}{G(0)}$.

The same reasoning is then applied to an arbitrary process $\tilde{G}(s) \in \mathfrak{F}$. It is immediately evident that, if the actual process would be known, a perfect track-

ing would be obtained, and because of unitary loop dc-gain, it would hold that $\lim_{\tau\to\infty} \|\tilde{R}(t;\tau)\|_\infty = \bar{y}(t-\tilde{L};\tau)$.

Thus, when $\tau \to \infty$, the perfect command signal for each process of the family $\mathfrak{F}$ is always the same transition polynomial, just differently delayed. Evidently, the output shape does not depend on the delay of the command signal. Hence it holds that

$$\lim_{\tau\to\infty} \|R(t;\tau)\tilde{T}(s) - \bar{y}(t-L;\tau)\|_\infty = 0, \tag{7.101}$$

where $\tilde{T}(s)$ is the closed-loop transfer function of an arbitrary plant $\tilde{G}(s) \in \mathfrak{F}$. Hence, by increasing τ, each response of the family can be made arbitrarily monotonic.

Finally, considering again that the input–output pairs are unique for each plant and that when $\tau \to \infty$ a perfect tracking is obtained all over the family of plants, it can be noted that

$$\lim_{\tau\to\infty} \left\| R(t;\tau)\frac{C(s)}{1+C(s)\tilde{G}(s)} - \frac{y(t)}{\tilde{G}(0)} \right\|_\infty = 0. \tag{7.102}$$

Considering that $\tilde{G}(0) = \tilde{K}$ and that, in view of the linearity of the systems, every signal is simply scaled by y_f, condition (7.98) is obtained and this concludes the proof.

Roughly speaking, the previous theorem states that the min-max optimization problem is solvable provided that the uncertainty affecting the process gain is less than 100 % (that is a very high value) and that the constraints on the control variable does not prevent the controller to keep each process of the family $\mathfrak{F}$ at the required steady-state value y_f.

Note that robust stability is a structural property, hence it only depends on the controller parameters T_m and z and not on the command signal.

Finally, some practical considerations can be made. In order to find an optimal solution, a numerical approach, for example, a genetic algorithm, can be used. It is worth stressing that, for the family $\mathfrak{F}$, the robust boundary $\Gamma(j\omega)$ can be easily obtained via (7.85) by gridding the allowable values of the parameters of $\tilde{G}(s)$. This operation may seem computationally demanding, but it is never involved in the numerical optimization since it must be performed just once before the optimization starts.

7.4.4 Illustrative Example

Consider the following nominal systems

$$G(s) = \frac{1}{s^{1.5}+1}e^{-s} \tag{7.103}$$

and the family of FFOPDT systems

$$\mathfrak{F} = \left\{ \tilde{G}(s) = \frac{\tilde{K}}{\tilde{T}s^{\tilde{\alpha}}+1} e^{-\tilde{L}s} : \tilde{K} \in [K_{\min}, K_{\max}], \right. \\ \left. \tilde{T} \in [T_{\min}, T_{\max}], \tilde{\alpha} \in [\alpha_{\min}, \alpha_{\max}], \tilde{L} \in [L_{\min}, L_{\max}] \right\} \tag{7.104}$$

where an uncertainty of $\pm 10\,\%$ over the process parameters is considered in order to define their minimum and maximum values.

An extremal system $G_i(s)$ is then defined as each system that is obtained with any possible combination of the extremal values of the uncertainty intervals. The extremal has then been defined as

$$\mathfrak{U} := \{G_i(s) \;\; i = 1, \ldots, 16\}. \tag{7.105}$$

Thus, the aim of the method is to solve Problem 7.2 with $\mathfrak{F} = \left\{ \tilde{G}(s) : \tilde{G}(s) \in \mathfrak{U} \right\}$ and the nominal plant $G(s)$ and with the percentage settling time at $2\,\%$, with a unitary set-point value $y_f = 1$, with a constraint on the maximum control variable of $U_{\max} = 1.5$ and that on the maximum acceptable overshoot of $O_{\max} = 0.2$.

The robust stability boundary has been first computed by gridding the process uncertainty intervals and by determining for each process $\tilde{G}(s)$ the uncertainty (7.85) with respect to the nominal real-rational model. function $\Gamma(j\omega)$ has been obtained by upper bounding the computed uncertainties for each frequency, obtaining the results shown in Fig. 7.8. At this point, the value $\beta = \alpha = 1.5$ has been selected as well as a transition polynomial with regularity $n = 3$ in order to satisfy (7.96). Finally, the min-max optimization problem has been solved by means of a genetic

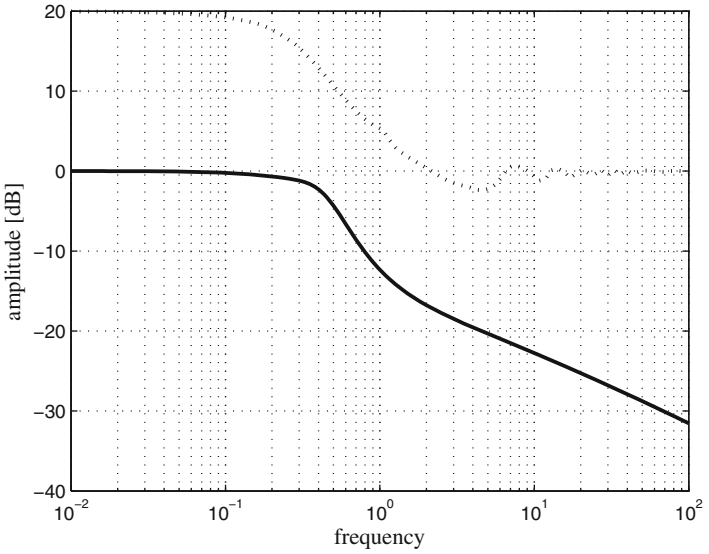

Fig. 7.8 Results for the case with overshoot constraint $O_{\max} = 0.2$. *Solid line* nominal closed-loop transfer function. *Dotted line* robust stability boundary

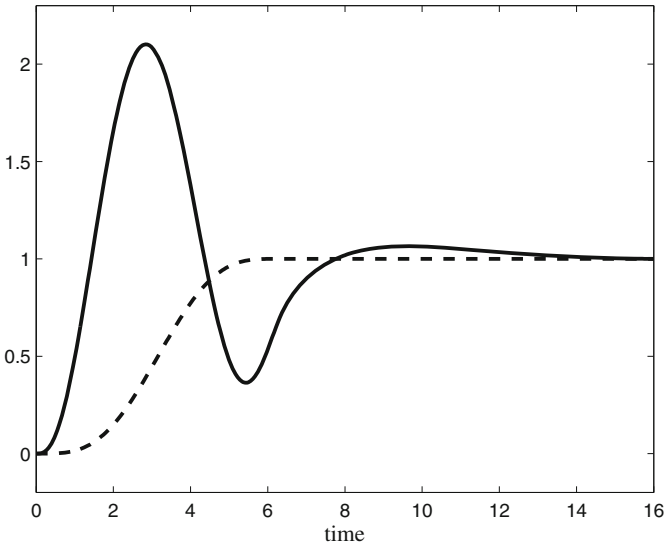

Fig. 7.9 Results for the case with overshoot constraint $O_{\max} = 0.2$. *Solid line* command signal. *Dashed line* transition polynomial

algorithm [69]. The obtained optimal parameters are $T_m = 3.5469$, $z = 3.4155$ and $\tau = 6.3122$, that lead to the optimal worst-case settling time $t_{s,wc} = 13.42$.

The obtained transition polynomial and the corresponding command signal are plotted in Fig. 7.9. As shown in Fig. 7.8, the robust stability constraint is satisfied. Finally, Fig. 7.10 shows that also the constraints on the control variable and on the maximum overshoot are satisfied. Actually, in this case, the most tightening constraint results to be the one imposed on the maximum overshoot. It is interesting to note that, thanks to the proposed technique, the nominal system has a prefect response (see Fig. 7.10), but also all the extremal systems present acceptable responses.

In order to further test the proposed methodology, the maximum overshoot constraint has been tightened to $O_{\max} = 0.05$ and a different transition polynomial with $n = 1$ (note that (7.96) is still satisfied) has been chosen. The obtained results are shown in Figs. 7.11, 7.12 and 7.13. Again, a perfect tracking is obtained in the nominal case and all the extremal systems responses exhibit a satisfactory behavior. The obtained optimal parameters are $T_m = 1.9474$, $z = 0.5158$ and $\tau = 9.6479$, while the optimal worst-case settling time is $t_{s,wc} = 13.55$.

The obtained optimal controller is much more aggressive than the one obtained in the previous case, but the transition time is increased to avoid excessive oscillations. It is interesting to note that, despite the tighter constraint, the worst-case settling exhibits a small increment. Obviously this is paid in term of rise time that is evidently increased (see Figs. 7.10 and 7.13). Also, it is worth analyzing the obtained command signal (see Fig. 7.12). As mentioned in Sect. 7.2.3, even though the smaller acceptable degree of regularity for the transition polynomial has been chosen, the obtained command signal is continuous.

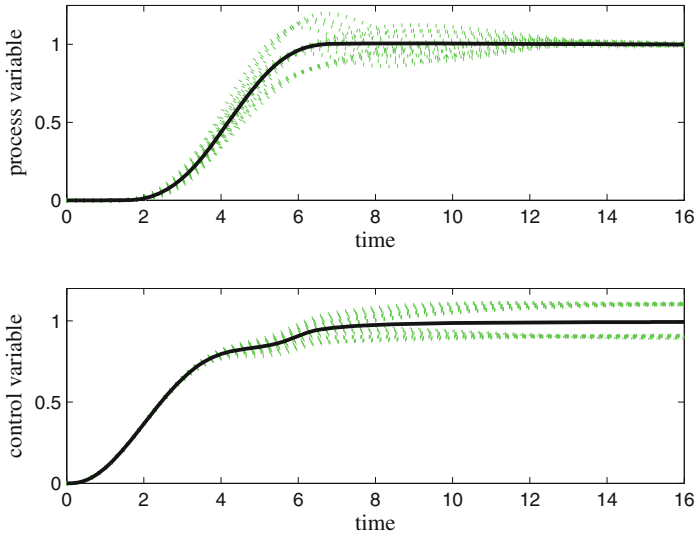

Fig. 7.10 Results for the case with overshoot constraint $O_{\max} = 0.2$. *Solid line* response of the nominal system. *Dotted line* responses of the uncertain system

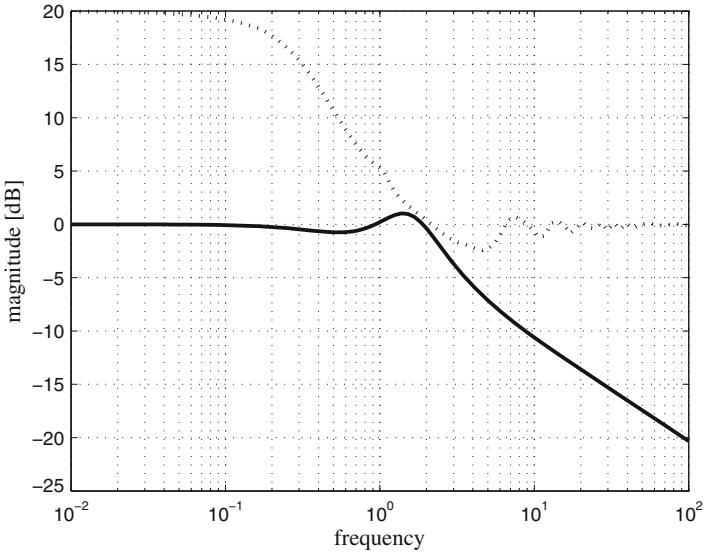

Fig. 7.11 Results for the case with overshoot constraint $O_{\max} = 0.05$. *Solid line* nominal closed-loop transfer function. *Dotted line* robust stability boundary

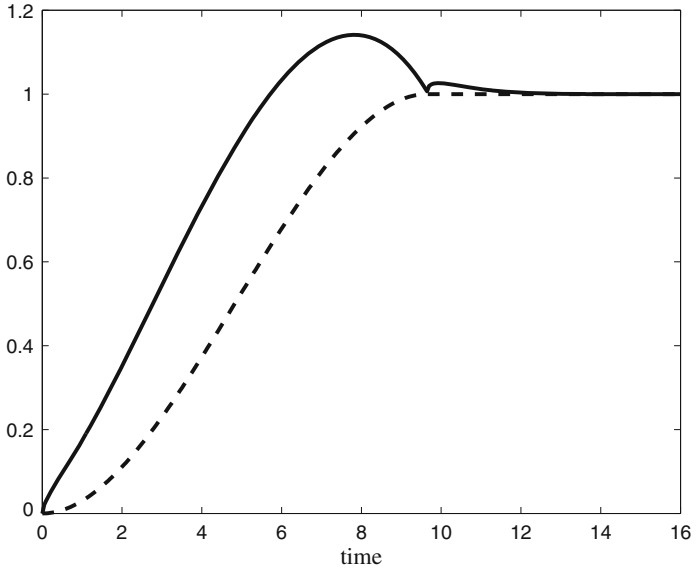

Fig. 7.12 Results for the case with overshoot constraint $O_{\max} = 0.05$. *Solid line* command signal. *Dashed line* transition polynomial

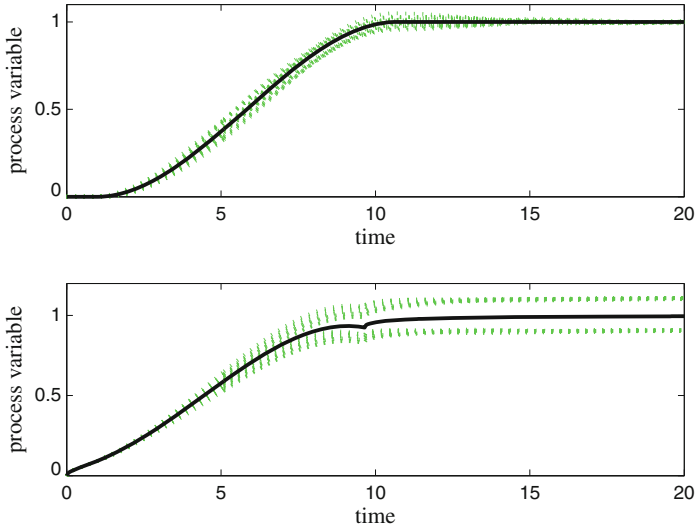

Fig. 7.13 Results for the case with overshoot constraint $O_{\max} = 0.05$. *Solid line* response of the nominal system. *Dotted line* responses of the uncertain system

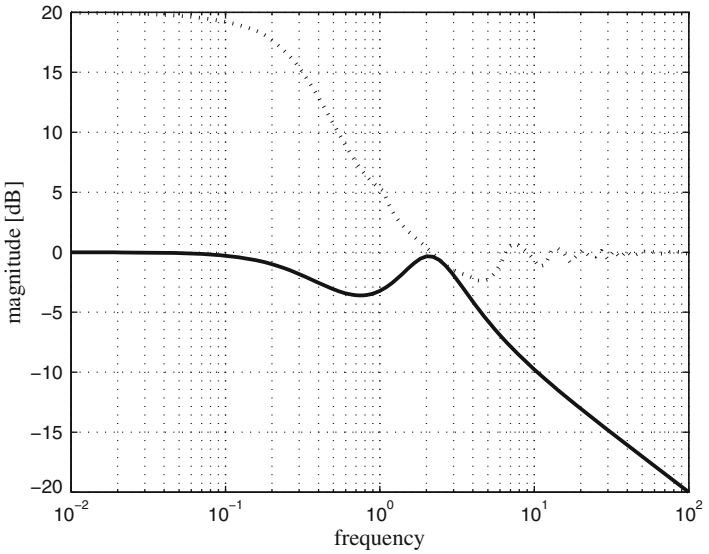

Fig. 7.14 Results obtained using a step command signal. *Solid line* nominal closed-loop transfer function. *Dotted line* robust stability boundary

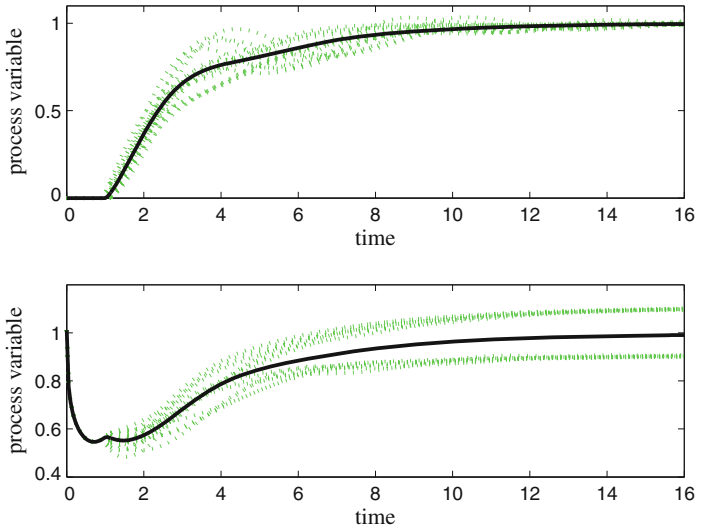

Fig. 7.15 Results obtained using a step command signal. *Solid line* response for the nominal system. *Dotted line* responses for the uncertain systems

Finally, for the sake of comparison, Problem 7.2 has been solved using a step command signal instead of the one obtained via input–output inversion (i.e., only T_m and z have been optimized). The results are shown in Figs. 7.14 and 7.15. In this case, the settling time minimization guarantees an overshoot smaller than 5 %, hence the

obtained controller is optimal independently from the chosen overshoot constraint. The optimal parameters are $T_m = 3.0188$ and $z = 0.3221$ and the optimal worst-case settling time is $t_{s,wc} = 14.82$. The obtained controller is even more aggressive (with a larger closed-loop bandwidth, see Fig. 7.14) than the one obtained in the previous case, but an inflection at $\omega \approx 0.8$ appears. This is because in this way excessive oscillations are avoided and the settling time is reduced, but the price to pay is a bigger rise time and, in any case, a bigger settling time (compared to the previous results). Actually, the great benefit of an input–output-inversion-based command signal lies in its capability to have a higher gain at those frequencies that are needed to speed up the system response and a lower gain at those frequencies that would cause oscillations.

7.5 Conclusions

In this chapter, an approach for the controller designed based on the input–output inversion concept has been presented.

The inversion procedure has been first described for general nonminimum-phase fractional systems and the solution of the minimum transition time problem subject to input and output constraints have been determined. Second, the approach has been applied in order to compute a feedforward action which improves the constrained set point regulation performance of a feedback fractional control system. Finally, a technique where the feedback controller and the feedforward action are designed in a combined way in order to address explicitly the robustness of the system has been presented.

In general, the described methodologies extends to (nonminimum-phase) fractional systems design techniques already developed for integer-order systems.

References

1. Abrahamse MB (1979) The Pick interpolation theorem for finitely connected domain. Mich Math J 26:195–203
2. Ahn H, Chen YQ (2008) Necessary and sufficient stability condition of fractional-order interval linear systems. Automat 44(11):2985–2988
3. Akcay H, Malti R (2008) On the completeness problem for fractional rationals with incommensurable differential orders. In: Proceedings 17th IFAC world congress, Seoul (ROK) pp 15367–15371
4. Alcántara S (2011) Analytical design of feedback compensators based on robustness/performace and servo/regulator trade-offs. Ph.D. thesis, Universitat Autónoma de Barcelona (E)
5. Alcántara S, Zhang WD, Pedret C, Vilanova R, Skogestad S (2011) IMC-like analytical $\mathscr{H}_\infty$ design with S/SP mixed sensitivity consideration: Utility in PID tuning guidance. J Process Control 21:976–985
6. Alfaro VM, Vilanova R, Arrieta O (2010) Maximum-sensitivity based robust tuning for two-degree-of-freedom proportional-integral controllers. Ind Eng Chem Res 49(11):5415–5423
7. Ang KH, Chong G, Li Y (2005) PID control systems analysis, design, and technology. IEEE Trans Control Syst Technol 13:559–576
8. Aoun M, Malti R, Levron F, Oustaloup A (2007) Synthesis of fractional Laguerre basis for system approximation. Autom 43(9):1640–1648
9. Åström K, Hägglund T (1995) PID Controllers: Theory. Design and tuning. ISA Press, Research Triangle Park, USA
10. Åström K, Wittenmark B (1997) Computer controlled system theory design. Prentice Hall, Upper Saddle River
11. Åström KJ, Hägglund T (2006) Advanced PID control. ISA Press,Research Triangle Park
12. Baños A, Cervera J, Lanusse P, Sabatier J (2011) Bode optimal loop shaping with CRONE compensators. J Vib Control 17(13):1964–1974
13. Barbosa RS, Tenreiro Machado JA, Ferreira IM (2004) Tuning of PID controllers based on Bode's ideal transfer function. Nonlinear Dyn 38:305–321
14. Battaglia JL, Cois O, Puigsegur I, Oustaloup A (2001) Solving an inverse heat conduction problem using a non-integer identified model. Heat Mass Transf 44(14):2671–2680
15. Battaglia JL, Le Lay L, Batsale J, Oustaloup A, Cois O (2000) Utilisation de modèles d'identification non entiers pour la résolution de problèmes inverses en conduction. Int J Therm Sci 39(3):374–389
16. Beschi M, Dormido S, Sánchez J, Visioli A (2012) Characterization of symmetric send-on-delta PI controllers. J Process Control 22(10):1930–1945
17. Biswas A, Das S, Abraham A, Dasgupta S (2009) Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution. Eng Appl Artif Intell 22:343–350

F. Padula and A. Visioli, *Advances in Robust Fractional Control*,
DOI 10.1007/978-3-319-10930-5

18. Bohn C, Atherton DP (1995) An analysis package comparing PID anti-windup strategies. IEEE Control Syst Mag 15(2):34–40
19. Bonnet C, Partignon JR (2000) Coprime factorization and stability of fractional differential systems. Syst Control Lett 41:167–174
20. Caponetto R, Dongola G, Fortuna L, Gallo A (2010) New results on the synthesis of FO-PID controllers. Commun Nonlinear Sci Numer Simul 15(4):997–1007
21. Caponetto R, Dongola G, Fortuna L, Petráš I (2010) Fractional order systems: modelling and control applications. World Scientific, Singapore
22. Caputo M (1967) Linear model of dissipation whose Q is almost frequency independent-II. Int Geophys J R Astron Soc 13(5):529–539
23. Caputo M (1969) Elasticità e Dissipazione. Zanichelli, Bologna
24. Carpinteri A, Mainardi F (2006) Fractal and fractional calculus in continuum mechanics. Springer, New York
25. Castillo FJ, Feliu V, Rivas R, Sanchez L (2011) Comparative analysis of stability and robustness between integer and fractional-order PI controllers for first order plus dead time plants. In: Proceedings 18th IFAC World Congress, Milan (I), pp 15019–15024
26. Cech M (2008) Robust controller design for fractional order systems. Ph.D. thesis, University of West Bohemia, Pilsen (CZ)
27. Charef A, Sun HH, Tsao YY, Onaral B (1992) Fractal system as represented by singularity function. IEEE Trans on Autom Control 37(9):1465–1470
28. Chen B (1984) Controller synthesis of optimal sensitivity: multivariable case. IEE Proc Control Theory Appl 131(1):47–51
29. Chen YQ (2006) Ubiquitous fractional order controls? In: Proceedings 2nd IFAC Workshop of fractional differentiation and its applications, Porto (P), pp 481–492
30. Chen YQ, Bhaskaran T, Xue D (2008) Practical tuning rule development for fractional order proportional and integral controllers. ASME J Comput Nonlinear Dyn 3:0214031–0214037
31. Chen YQ, Petráš I, Xue D (2009) Fractional order control a tutorial. In: Proceedings American control conference, St. Louis (USA), pp 1397–1411
32. Dehghani A, Lanzon A, Anderson BDO (2006) $\mathscr{H}_\infty$ design to generalize internal model control. Automatica 42(11):1959–1968
33. Devasia S (2002) Should model-based inverse inputs be used as feedforward under plant uncertainty? ASME J Dyn Syst Meas Contr 47(11):1865–1871
34. Dormido S, Pisoni E, Visioli A (2012) Interactive tools for designing fractional-order PID controllers. Int J Innovative Comput Inf Control 8(7):4579–4590
35. Doyle J, Francis B, Tannenbaum A (1990) Feedback control theory. Macmillan, New York
36. Duarte F, Tenreiro Machado JA (2002) Chaotic phenomena and fractional order dynamics in the trajectory control of redundant manipulators. Nonlinear Dyn 29(1–4):315–342
37. Fadiga L, Farges C, Sabatier J, Moze M (2011) On computation of H_∞ norm for commensurate fractional order systems. In: Proceedings 50th conference on decision and control and European control conference, Orlando (USA), pp 8231–8236
38. Francis B (1987) A Course in H_∞ control theory. Springer Verlag, New York
39. B Francis, J Doyle (1987) Linear control theory with an $\mathscr{H}_\infty$ optimality criterion. SIAM J Control Optim 25(4):815–844
40. Gabano JD, Poinot T (2011) Fractional modeling applied to non destructive thermal characterization. In: Proceedings 18th IFAC world congress, Milan (I), pp 13972–13977
41. Gross B, Braga EP (1961) Singularities of linear system functions. Elsevier, US
42. Gude JJ, Kahoraho E (2009) . New tuning rules for PI and fractional PI controllers. In: Proceedings European control conference, Budapest (HU)
43. Gude JJ, Kahoraho E (2009) Simple tuning rules for fractional PI controllers. In: Proceedings IEEE International conference on emerging technologies and factory automation, Mallorca (E), pp 1—8
44. Gude JJ, Kahoraho E (2012) Kappa-tau type PI tuning rules for specified robust levels. Proceedings IFAC Conference on Advances in PID Control. University of Deusto, Spain, pp 589–594

45. Gutierrez RE, Rosario JM, Tenreiro Machado JA (2010) Fractional order alculus: basic concepts and engineering applications. Math Probl Eng 2010
46. Hansson A, Gruber P, Todtli J (1994) Fuzzy anti-reset windup for PID controllers. Control Eng Pract 2(3):389–396
47. Hanus R, Kinnaert M, Henrotte J-L (1987) Conditioning technique, a general anti-windup and bumpless transfer method. Automatica 23(6):729–739
48. Hartley TT, Lorenzo CF, Qammar HK (1995) Chaos in a fractional order Chua's system. IEEE Trans Circuits Syst I 42(8):485–490
49. Heymans N, Bauwens JC (1994) Fractal rheological models and fractional differential equations for viscoelasticity behavior. Rheological Acta 33(3):210–219
50. Heymans N, Podlubny I (2006) Physical interpretation of initial conditions for fractional differential equations with Riemann-Liouville fractional derivatives. Rheologica Acta 45(5):765–772
51. Hippe P (2006) Windup in control. Springer, London
52. Scottedward Hodel A, Hall CE (2001) Variable structure PID control to prevent integrator windup. IEEE Trans Industr Electron 48(2):442–451
53. Hunt LR, Meyer G, Su R (1996) Non causal inverse for linear systems. IEEE Trans Autom Control 41(4):608–611
54. Ionescu CM (2013) The human respiratory system: an analysis of the interplay between anatomy, structure. Springer, Breathing and Fractal Dynamics, London
55. Jelali M (2006) An overview of control performance assessment technology and industrial applications. Control Eng Pract 14:441–466
56. Kilbas AA, Srivastava KM, Trujillo JJ (2000) Theory and applications of fractional differential equations. Elsevier, Amsterdam
57. Kothare MV, Campo PJ, Morari M, Nett CN (1994) A unified framework for the study of anti-windup design. Automatica 30(12):1869–1883
58. Kwakernaak H (1993) Robust control and $\mathscr{H}_\infty$-optimization - tutorial paper. Automatica 29(2):255–273
59. Li Y, Chen YQ, Podlubny I (2009) Mittag-Leffler stabilitiy of fractional order nonlinear dynamic systems. Automatica 45(8):1965–1969
60. Ljung L (2013) System identification toolbox user's guide. The Mathworks, Natick
61. Magin RL (2006) Fractional calculus in bioengeneering. Begell House Publishers, USA
62. Maione G (2008) Continued fractions approximation of the impulse response of fractional order dynamic systems. IET Control Theory and Appl 2(7):564–572
63. Maiti D, Acharya A, Chakraborty M, Konar A, Janarthanan R (2008) Tuning PID and $PI^{\lambda}D^{\delta}$ controllers using the integral time absolute error criterion. In: Proceedings IEEE International conference on information and automation for austainability, Colombo (CL), pp 457–462
64. Malti R, Moreau X, Khemane F, Oustaloup A (2011) Stability and resonance conditions of elementary fractional transfer functions. Automatica 47(11):2462–2467
65. Mathai AM, Haubold HJ (2008) Special functions for applied scientists. Springer, New York
66. Matignon D (1996) Stability results for fractional differential equation with applications to control processing. In: Proceedings IMACS multiconference on computational engineering in systems applications, Lille (F), pp 963–968
67. Matignon D (1998) Stability properties for generalized fractional differential systems. In: Proceedings of fractional differential systems: models, methods and applications, Paris (F), vol 5. pp 145–158
68. Miller KS, Ross B (1993) An introduction to the fractional calculus and fractional differential equations. Wiley, New York
69. Mitchell M (1998) An introduction to genetic algorithms. MIT press, Cambridge
70. Monje CA (2006) Design methods of fractional order controllers for industrial applications. Ph.D. thesis, University of Extremadura, Badajoz (E)
71. Monje CA, Chen Y, Vinagre BM, Xue D, Feliu V (2010) Fractional-order systems and controls. Springer, Heidelberg

72. Monje CA, Vinagre BM, Calderon AJ, Feliu V, Chen YQ (2004) On fractional PI^{λ} controllers: some tuning rules for robustness to plant uncertainties. Nonlinear Dyn 38:369–381
73. Monje CA, Vinagre BM , Chen YQ, Feliu V, Lanusse P, Sabatier J (2004) Proposals for fractional $PI^{\lambda}D^{\mu}$ tuning. In: Proceedings IFAC workshop on fractional differentiation and its applications, Bordeaux (F), pp 156–1610
74. Monje CA, Vinagre BM, Feliu V, Chen YQ (2008) Tuning and auto-tuning of fractional order controllers for industry applications. Control Eng Pract 16(7):798–812
75. Morari M, Zafirou E (1989) Robust process control. Prentice Hall
76. Moshrefi-Torbati M, Hammond JK (1998) Physical and geometrical interpretation of fractional operators. J Franklin Inst 336(6):1077–1086
77. Moze M, Sabatier J, Oustaloup A (2007) LMI characterization of fractional systems stability. In: Sabatier J, Agrawal OP, Tenreiro Machado JA (eds) Advances in fractional calculus: theoretical developments and spplications in physics and engineering. Springer, New York, pp 419–434
78. Nehari Z (1957) On bounded bilinear forms. Ann Math 65(1):153–162
79. O'Dwyer A (2006) Handbook of PI and PID tuning rules. Imperial College Press, London
80. Oldham KB, Spanier J (1974) The fractional calculus: theory and application of differentiation and integration of arbitrary order. Academic Press, New York
81. Oldham KB, Zoski CG (1983) Analogue instrumentation for processing polarographic data. J Electroanal Chem Interfacial Electrochem 157(1):27–51
82. Ortigueira MD (2000) Introduction to fractional linear systems. part 1: continuous-time case. IEE Proc Vis Image Signal Proc 147(1):62–70
83. Ortigueira MD (2006) A coherent approach to non-integer order derivatives. Signal Process 86(10):2505–2515
84. Ortigueira MD (2008) An introduction to the fractional continuous-time linear systems: the 21st century systems. IEEE Circuits Syst Mag 8(3):19–26
85. Ortigueira MD (2011) Fractional calculus for scientists and engineers. Springer, Dordrecht
86. Ortigueira MD, Coito FJV, Trujillo JJ (2013) A new look into the discrete-time fractional calculus: transform and linear systems. In: Proceedings 6th IFAC workshop on fractional differentiation and its applications, Grenoble (F), pp 630–635
87. Oustaloup A (1991) La commande CRONE: commande robuste d'Ordre non entier. Hermes, Paris
88. Oustaloup A, Lanusse P, Melchior P, Moreau X, Sabatier J (2006) The CRONE approach: theoretical developments and major applications. In Proceedings 2nd IFAC workshop on fractional differentiation and its applications, Porto (P), pp 324–354
89. Oustaloup A, Levron F, Mathieu B, Nanot FM (2000) Frequency-band complex noninteger differentiator: characterization and synthesis. IEEE Trans Circ Syst I: Fundam Theor Appl 47(1):25–39
90. Oustaloup A, Moreau X, Nouillant M (1996) The CRONE suspension. Control Eng Pract 4(8):1101–1108
91. Oustaloup A, Sabatier J, Lanusse P, Malti R, Melchior P, Moreau X, Moze M (2008) An overview of the CRONE approach in system analysis, modeling and identification, observation and control. In: Proceedings 17th IFAC world congress, Seoul (ROK), pp 14254–14265
92. Padula F http://www.ing.unibs.it/fabrizio.padula/fractional
93. Padula F, Alcantara S, Vilanova R, Visioli A (2013) $\mathscr{H}_{\infty}$ control of fractional linear systems. Automatica 49(7):2276–2280
94. Padula F, Vilanova R, Visioli A (2012) $\mathscr{H}_{\infty}$ model matching PID design for fractional FOPDT systems. In: Proceedings American control conference, Montreal (CA), pp 5513–5518
95. Padula F, Vilanova R, Visioli A (2014) $\mathscr{H}_{\infty}$ optimization based fractional-order PID controllers design. Int J Robust Nonlinear Control
96. Padula F, Visioli A (2014) Inversion-based feedforward and reference signal design for fractional constrained control systems. Automatica 50(8):2169–2178
97. Padula F, Visioli A (2011) Tuning rules for optimal PID and fractional-order PID controllers. J Process Control 21(1):69–81

98. Padula F, Visioli A (2012) On the stabilizing PID controllers for integral processes. IEEE Trans Autom Control 5(2):494–499
99. Padula F, Visioli A (2012) Optimal tuning rules for proportional-integral-derivative and fractional-order proportional-integral-derivative controllers for integral and unstable processes. IET Control Theor Appl 6(6):776–786
100. Padula F, Visioli A (2013) Set-point weight tuning rules for fractional-order PID controllers. Asian J Control 15(4):1–13
101. Partignon JR (1988) An introduction to hankel operators. Cambridge University Press, Cambridge
102. Peller VV (2003) Hankel operators and their applications. Springer, New York
103. Peng Y, Vrancic D, Hanus R (1996) Anti-windup, bumpless, and conditioned transfer techniques for PID controllers. IEEE Control Syst Mag 16(4):48–57
104. Perez H, Devasia S (2003) Optimal output-transitions for linear systems. Automatica, 39(4):181–192
105. Petráš I (2012) Chaos in fractional-order population model. Int J Bifurcat Chaos Appl Sci Eng 22(4):28
106. Petráš I (1999) The fractional-order controllers: methods for their synthesis and application. J Electr Eng 50(9–10):284–288
107. Petráš I (2009) Stability of fractional-order systems with rational orders: a survey. Fractional Calc Appl Anal 10(3):269–298
108. Petráš I (2011) Fractional Order Nonlinear Systems. Springer, New York
109. Piazzi A, Visioli A (2005) Using stable input-output inversion for minimum-time feedforward constrained regulation of scalar systems. Automatica 41(2):305–313
110. Piazzi A, Visioli A (2001) Optimal inversion-based control for the set-point regulation of nonminimum-phase uncertain scalar systems. IEEE Trans Autom Control 46(10):1654–1659
111. Piazzi A, Visioli A (2001) Optimal noncausal set-point regulation of scalar systems. Automatica 37(1):121–127
112. Piazzi A, Visioli A (2001) Robust set-point constrained regulation via dynamic inversion. Int J Robust Nonlinear Control 11:1–22
113. Piazzi A, Visioli A (2006) A noncausal approach for PID control. J Process Control 16(8):831–843
114. Pisano A, Rapaić M, Jeličić Z, Usai E (2010) Sliding mode control approaches to the robust regulation of linear multivariable fractional-order dynamics. Int J Robust Nonlinear Control 20(18):2021–2044
115. Podlubny I (1999) Fractional differential equations. Academic Press, San Diego
116. Podlubny I (1999) Fractional-order systems and $PI^{\lambda}D^{\mu}$ controllers. IEEE Trans Autom Control 44(1):208–214
117. Podlubny I (2002) Geometric and physical interpretation of fractional integration and fractional differentiation. Fractional Calc Appl Anal 5(4):367–386
118. Podlubny I, Petráš I, Vinagre BM, O'Leary P, Dorčák L (2002) Analogue realizations of fractional-order controllers. Nonlinear Dyn 29(1):281–296
119. Quadrat A (2003) On a generalization of the Youla-Kučera parametrization. Part I: the fractional ideal approach to SISO systems. Syst Control Lett 50(2):135–148
120. Radwan AG, Soliman AM, Elwakil AS, Sedeek A (2009) On the stability of linear systems with fractional-order elements. Chaos, Solitons Fractals 40(5):2317–2328
121. Romero-Perez JA, Arrieta O, Padula F, Reynoso-Meza G, Garcia-Nieto S, Balaguer P (2012) Estudio comparativo del algoritmos de autoajuste de controladores PID. Revista Iberoamericana de Automatica e Informatica Industrial 9(2):182–193
122. Rönnbäck S, Walgama KS, Sternby J (1992) An extension to the generalized anti-windup compensator. In: Borne P, Tzafestas SG, Radhy NE, (eds) Mathematics of the analysis and design of process control, Elsevier, North Holland, pp 275–285
123. Sabatier J, Agrawal OP, Tenreiro Machado JA (2007) Advances in fractional calculus: theoretical developments and applications in physics and engineering. Springer, New York

124. Sabatier J, Farges C (2012) On stability of commensurate fractional order systems. In J Bifurcat Chaos 22(4):1–8
125. Sabatier J, Moze M, Farges C (2010) LMI stability conditions for fractional order systems. Comput Math Appl 59(5):1594–1609
126. Sabatier J, Moze M, Oustaloup A (2008) On fractional system $\mathscr{H}_\infty$-norm computation. In: Proceedings 44th IEEE international conference on decision and contro and European control conferencel, Seville (E), pp 5758–5763
127. Samko SG, Kilbas AA, Marichev OI (1993) Fractional integrals and derivatives: theory and applications. Gordon and Breach Science, Switzerland
128. Shamsuzzoha M, Jeon J, Lee M (2007) Improved analytical PID controller design for the second order unstable process with time delay. In: Proceedings 17th European symposium on computer aided process engineering, Bucharest (RO), pp 1–6
129. Shen J-C (2002) New tuning method for PID controller. ISA Trans 42:473–484
130. Shinskey FG (1994) Feedback controllers for the process industries. New York, McGraw-Hill
131. Shinskey FG (1996) Process control systems—application, design, and tuning. McGraw-Hill, London
132. Silva GJ, Datta A, Bhattacharyya SP (2002) New results on the synthesis of PID controllers. IEEE Trans Autom Control 47(2):241–252
133. Smith MC (1989) On stabilization and the existence of coprime factorization. IEEE Trans Autom Control 34(9):1005–1007
134. The Mathworks. Curve Fitting Toolbox User's Guide. The Mathworks (2013)
135. Valerio D (2005) Fractional robust system control. Ph.D. thesis, University of Lisbon (P)
136. Valerio D, Sá da Costa J (2006) Tuning of fractional PID controllers with Ziegler-Nichols-type rules. Sign Process 86(10):2771–2784
137. Valerio D, Sá da Costa J (2006) Tuning of fractional controllers minimising H_2 and H_∞ norms. Acta Polytech Hung 3(4):55–70
138. Valerio D, Sá da Costa J (2006) Tuning-rules for fractional PID controllers. In: Preprints 2nd IFAC Workshop on Fractional Differentiation and its Applications , Porto (P) pp 28–33
139. Valerio D, Sá da Costa J (2010) A review of tuning methods for fractional PIDs. In: Preprints 4th IFAC Workshop on fractional differentiation and its applications, Badajoz (E)
140. Valerio D, Sá da Costa J (2011) Introduction to the single-input, single-output fractional control. IET Control Theor Appl 5(8):1033–1057
141. Valerio D, Sá da Costa J (2012) An introduction to fractional control. IET, London
142. Vidyasagar M (1985) Control system synthesis: A factorization approach. MIT Press, Cambridge
143. Vilanova R (2008) IMC based robust PID design: tuning guidelines and automatic tuning. J Process Control 18(1):61–70
144. Vinagre BM, Chen YQ, Petráš I (2003) Two direct Tustin discretization methods for fractional order differentiator/integrator. J Franklin Inst 340(5):349–362
145. Vinagre BM, Feliu V (2002) Modeling and control of dynamic systems using fractional calculus: application to electrochemical processes and flexible structures. In: 41st IEEE conference on decision and control tutorial workshop, Las Vegas, USA
146. Vinagre BM, Feliu V (2007) Optimal fractional controllers for rational order systems: a special case of the Wiener-Hopf spectral factorization method. IEEE Trans Autom Control 52(12):2385–2389
147. Vinagre BM, Monje CA, Calderon AJ, Suarez JI (2007) Fractional PID controllers for industry application. A brief introduction. J Vib Control 13(9–10):1419–1429
148. Vinagre BM, Podlubny I, Hernandez A, Feliu V (2000) Some approximations of fractional order operators used in control theory and applications. Fractional Calc Appl Anal 3(3):945–950
149. Visioli A (2001) Optimal tuning of PID controllers for integral and unstable processes. IEE Proc Control Theor Appl 148(2):180–184
150. Visioli A (2003) Modified anti-windup scheme for PID controllers. IEE Proc Control Theor Appl 150(1):49–54

151. Visioli A (2006) Practical PID control. Springer, London
152. Visioli A, Vilanova R (eds) (2012) PID Control in the third millennium: lessons learned and new approaches. Springer, London
153. Visioli A, Zhong Q-C (2010) Control of integral processes with dead time. Springer, London
154. Vrancic D (1997) Design of anti-windup and bumpless transfer protection. Ph.D. thesis, University of Ljubljana (SLO)
155. Walgama KS, Ronnback S, Sternby J (1991) Generalisation of conditioning technique for anti-windup compensator. IEE Proc Control Theor Appl 139(2):109–118
156. Wang D, Gao X (2012) H_∞ design with fractional-order PD$^\mu$ controllers. Automatica 48(5):974–977
157. Wang JC (1987) Realization of generalized Warburg impedance with RC ladder networks and transmission lines. J Electrochem Soc 134(8):1915–1920
158. Westerlund S, Ekstam L (1994) Capacitor theory. IEEE Trans Dielectr Electr Insul 1(5):826–839
159. Youla DC, Jabr HA, Bongiorno JJ (1976) Modern Wiener-Hopf design of optimal controllers—part II: the multivariable case. IEEE Trans Autom Control 21(3):319–338
160. Zaccarian L, Teel AR (2011) Modern anti-windup synthesis: Control augmentation for actuator saturation. Princeton University Press, Princeton
161. Zamani M, Karimi-Ghartemani M, Sadati N, Parniani M (2009) Design of fractional order PID controller for an AVR using particle swarm optimization. Control Eng Pract 17(12):1380–1387
162. Zames G, Francis B (1983) Feedback, min-max sensitivity and optimal robustness. IEEE Trans Autom Control 28(5):585–601
163. Zou Q, Devasia S (1999) Preview-based stable-inversion for output tracking of linear systems. J Dyn Syst Meas Contr 121:625–630

Index

F. Padula and A. Visioli, *Advances in Robust Fractional Control*,
DOI 10.1007/978-3-319-10930-5