

Chapter 2

Preliminaries

In the following chapter the reader is introduced to core concepts and algorithms used in the thesis. The chapter begins with an introduction to fractional calculus tools, including fractional operator definitions and properties. Next, tools related to modeling and analysis of dynamic systems are discussed. Then, fractional-order controllers are introduced. Finally, a brief overview of numerical optimization methods used in this work is provided. Particular attention is given to the treatment of bounded and constrained optimization problems.

2.1 Mathematical Basis

Fractional calculus is a generalization of integration and differentiation to non-integer order operator ${}_a\mathcal{D}_t^\alpha$, where a and t denote the limits of the operation and α denotes the fractional order such that

$${}_a\mathcal{D}_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha} & \Re(\alpha) > 0, \\ 1 & \Re(\alpha) = 0, \\ \int_a^t (d\tau)^{-\alpha} & \Re(\alpha) < 0, \end{cases} \quad (2.1)$$

where generally it is assumed that $\alpha \in \mathbb{R}$, but it may also be a complex number [3]. In this work we consider only the former case.

There exist multiple definitions of the fractional operator [16]. We consider the Grünwald-Letnikov definition, which is used throughout this work for the purpose of numerical solutions to fractional-order differential equations.

Definition 2.1 (*Grünwald-Letnikov*)

$${}_a\mathcal{D}_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{k=0}^{\left[\frac{t-a}{h}\right]} (-1)^k \binom{\alpha}{k} f(t - kh), \quad (2.2)$$

where $[\cdot]$ means the integer part, h is the step size.

Fractional-order differentiation has the following properties [16, 27]:

1. If $f(t)$ is an analytic function, then the fractional-order differentiation ${}_0\mathcal{D}_t^\alpha f(t)$ is also analytic with respect to t .
2. If $\alpha = n$ and $n \in \mathbb{Z}_+$, then the operator ${}_0\mathcal{D}_t^\alpha$ can be understood as the usual operator d^n/dt^n .
3. Operator of order $\alpha = 0$ is the identity operator: ${}_0\mathcal{D}_t^0 f(t) = f(t)$.
4. Fractional-order differentiation is linear; if a, b are constants, then

$${}_0\mathcal{D}_t^\alpha [af(t) + bg(t)] = a {}_0\mathcal{D}_t^\alpha f(t) + b {}_0\mathcal{D}_t^\alpha g(t). \quad (2.3)$$

5. For the fractional-order operators with $\Re(\alpha) > 0$, $\Re(\beta) > 0$, and under reasonable constraints on the function $f(t)$ it holds the additive law of exponents:

$${}_0\mathcal{D}_t^\alpha \left[{}_0\mathcal{D}_t^\beta f(t) \right] = {}_0\mathcal{D}_t^\beta \left[{}_0\mathcal{D}_t^\alpha f(t) \right] = {}_0\mathcal{D}_t^{\alpha+\beta} f(t) \quad (2.4)$$

6. The fractional-order derivative commutes with integer-order derivative

$$\frac{d^n}{dt^n} ({}_a\mathcal{D}_t^\alpha f(t)) = {}_a\mathcal{D}_t^\alpha \left(\frac{d^n f(t)}{dt^n} \right) = {}_a\mathcal{D}_t^{\alpha+n} f(t), \quad (2.5)$$

under the condition $t = a$ we have $f^{(k)}(a) = 0$, $(k = 0, 1, 2, \dots, n - 1)$.

The Laplace integral transform is an essential tool in dynamic system and control engineering. A function $F(s)$ of the complex variable s is called the Laplace transform of the original function $f(t)$ and defined as

$$F(s) = \mathcal{L}[f(t)] = \int_0^\infty e^{-st} f(t) dt \quad (2.6)$$

The original function $f(t)$ can be recovered from the Laplace transform $F(s)$ by applying the reverse Laplace transform defined as

$$f(t) = \mathcal{L}^{-1}[F(s)] = \frac{1}{j2\pi} \int_{c-j\infty}^{c+j\infty} e^{st} F(s) ds, \quad (2.7)$$

where c is greater than the real part of all the poles of function $F(s)$ [16].

Assuming zero initial conditions, the Laplace transform of (2.2) is defined as follows.

Definition 2.2 (*Laplace transform of the Grünwald-Letnikov fractional operator*)

$$\mathcal{L}[\mathcal{D}^\alpha f(t)] = s^\alpha F(s). \quad (2.8)$$

2.2 Fractional-Order Models

A fractional-order continuous-time dynamic system can be expressed by a fractional differential equation of the following form [16]:

$$\begin{aligned} a_n \mathcal{D}^{\alpha_n} y(t) + a_{n-1} \mathcal{D}^{\alpha_{n-1}} y(t) + \cdots + a_0 \mathcal{D}^{\alpha_0} y(t) = \\ b_m \mathcal{D}^{\beta_m} u(t) + b_{m-1} \mathcal{D}^{\beta_{m-1}} u(t) + \cdots + b_0 \mathcal{D}^{\beta_0} u(t), \end{aligned} \quad (2.9)$$

where $(a_i, b_j) \in \mathbb{R}^2$ and $(\alpha_i, \beta_j) \in \mathbb{R}_+^2$. The system is said to be of *commensurate-order* if in (2.9) all the orders of derivation are integer multiples of a base order q such that $\alpha_k, \beta_k = kq, q \in \mathbb{R}^+$. The system can then be expressed as

$$\sum_{k=0}^n a_k \mathcal{D}^{kq} y(t) = \sum_{k=0}^m b_k \mathcal{D}^{kq} u(t). \quad (2.10)$$

If in (2.10) the order is $q = 1/r, r \in \mathbb{Z}_+$, the system will be of rational order. The diagram with linear time-invariant (LTI) system classification is given in Fig. 2.1.

Applying the Laplace transform to (2.9) with zero initial conditions the input-output representation of the fractional-order system can be obtained in the form of a transfer function.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_0 s^{\alpha_0}}. \quad (2.11)$$

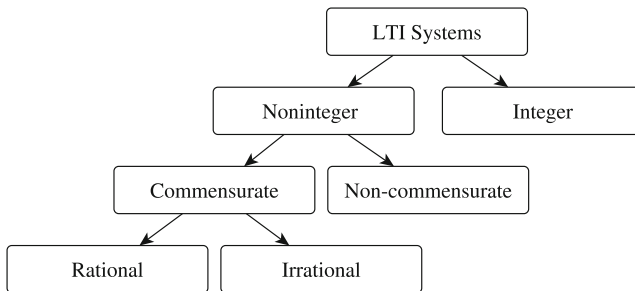


Fig. 2.1 Classification of LTI systems

We shall call the number of fractional poles in (2.11) the *pseudo-order* of the system. In the case of a system with commensurate order q , we may take $\sigma = s^q$ and consider the continuous-time pseudo-rational transfer function

$$H(\lambda) = \frac{\sum_{k=0}^m b_k \sigma^k}{\sum_{k=0}^n a_k \sigma^k}. \quad (2.12)$$

2.2.1 Process Models

In the context of this work we deal with problems of process control. The fractional-order transfer function representation of a process model consists of (2.11) and an input delay term given in the time domain as $u(t) = u_d(t - L)$. The general form is thus

$$G(s) = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \dots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \dots + a_0 s^{\alpha_0}} e^{-Ls}, \quad (2.13)$$

where it is usual to take $\beta_0 = \alpha_0 = 0$ so that the static gain of the system is given by $K = b_0/a_0$, and $L \in \mathbb{R}_+$.

One particular model of this type is the fractional-order first-order plus delay time model [6, 11, 23]:

$$G_{dt}(s) = \frac{K}{1 + Ts^\alpha} e^{-Ls}, \quad (2.14)$$

where K is the static gain, $L \geq 0$ is the delay, $T > 0$ is the time constant, and $\alpha \in (0, 2)$ is the fractional power of the operator. Its conventional counterpart with $\alpha = 1$ has become the basis for numerous tuning rules [21]. Since the generalized version of the process model allows to capture the dynamics of the process under study more accurately, it is expected to be useful in the design of fractional-order controllers [24].

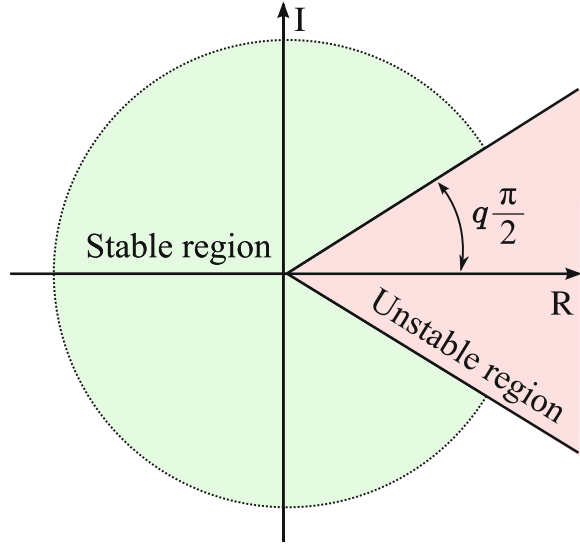
2.2.2 Stability Analysis

In order to determine stability of a fractional system given by (2.10) we consider the following theorem [3, 13].

Theorem 2.1 (Matignon's stability theorem) *The fractional transfer function $G(s) = Z(s)/P(s)$ is stable if and only if the following condition is satisfied in σ -plane:*

$$|\arg(\sigma)| > q \frac{\pi}{2}, \quad \forall \sigma \in C, \quad P(\sigma) = 0, \quad (2.15)$$

Fig. 2.2 LTI
fractional-order system
stability region for
 $0 < q \leq 1$



where $0 < q \leq 1$ and $\sigma := s^q$. When $\sigma = 0$ is a single root of $P(s)$, the system cannot be stable. For $q = 1$, this is the classical theorem of pole location in the complex plane: no pole is in the closed right plane of the first Riemann sheet.

The algorithm for checking the stability of the system in (2.11) can be summarized as follows:

1. Find the commensurate order q of $P(s)$, find $a_1, a_2, \dots, a_n$ in (2.12);
2. Solve for σ the equation $\sum_{k=0}^n a_k \sigma^k = 0$.
3. If all obtained roots satisfy the condition in Theorem 2.1, the system is stable.

Stability regions of a fractional-order system are shown in Fig. 2.2.

Note that there are currently no polynomial techniques, either Routh or Jury type, to analyze the stability of fractional-order systems [16].

2.2.3 Time Domain Analysis

Another solution involves numerical computation of fractional-order derivatives which is carried out by means of a revised Grünwald-Letnikov definition (2.2) rewritten as

$${}_a \mathcal{D}_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{\lceil \frac{t-a}{h} \rceil} w_j^{(\alpha)} f(t - jh), \quad (2.16)$$

where h is the computation step-size and $w_j^{(\alpha)} = (-1)^j \binom{\alpha}{j}$ can be evaluated recursively from

$$w_0^{(\alpha)} = 1, w_j^{(\alpha)} = \left(1 - \frac{\alpha + 1}{j}\right) w_{j-1}^{(\alpha)}, j = 1, 2, \dots \quad (2.17)$$

To obtain a numerical solution for the equation in (2.9) the signal $\hat{u}(t)$ should be obtained first, using the algorithm in (2.16), where

$$\hat{u}(t) = b_m \mathcal{D}^{\beta_m} u(t) + b_{m-1} \mathcal{D}^{\beta_{m-1}} u(t) + \dots + b_0 \mathcal{D}^{\beta_0} u(t). \quad (2.18)$$

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

$$y(t) = \frac{1}{\sum_{i=0}^n \frac{a_i}{h^{\alpha_i}}} \left[\hat{u}(t) - \sum_{i=0}^n \frac{a_i}{h^{\alpha_i}} \sum_{j=1}^{\left[\frac{t-a}{h}\right]} w_j^{(\alpha)} y(t - jh) \right]. \quad (2.19)$$

The presented method is a fixed step method. The accuracy of simulation therefore may depend on the step size [3, 16, 31].

If the system (2.13) has a input-output delay L , the resulting delayed response $y_d(t)$ with $y_d(0) = 0$ is obtained such that

$$y_d(t) = \begin{cases} y(t - L), & t > L \\ 0, & \text{otherwise.} \end{cases} \quad (2.20)$$

2.2.4 Frequency Domain Analysis

Frequency domain response may be obtained by substituting $s = j\omega$ in (2.13). The complex response for a frequency $\omega \in (0; \infty)$ can then be computed as follows:

$$R(j\omega) = \frac{b_m(j\omega)^{\beta_m} + b_{m-1}(j\omega)^{\beta_{m-1}} + \dots + b_0(j\omega)^{\beta_0}}{a_n(j\omega)^{\alpha_n} + a_{n-1}(j\omega)^{\alpha_{n-1}} + \dots + a_0(j\omega)^{\alpha_0}} e^{-L(j\omega)}, \quad (2.21)$$

where j is the imaginary unit.

In addition, consider the following useful relation for the noninteger power $\alpha \in \mathbb{R}$ of the imaginary unit

$$j^\alpha = \cos\left(\frac{\alpha\pi}{2}\right) + j \sin\left(\frac{\alpha\pi}{2}\right). \quad (2.22)$$

2.3 Approximation of Fractional-Order Operators

The Oustaloup recursive filter, proposed in [22] and discussed in [16, 29], gives a very good approximation of fractional operators in a specified frequency range. It is a well-established method and is often used for practical implementation of fractional-order systems and controllers. It is summarized next.

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

$$s^\alpha \approx K \prod_{k=1}^N \frac{s + \omega'_k}{s + \omega_k}, \quad (2.23)$$

where

$$\omega'_k = \omega_b \cdot \omega_u^{(2k-1-\alpha)/N}, \quad (2.24)$$

$$\omega_k = \omega_b \cdot \omega_u^{(2k-1+\alpha)/N}, \quad (2.25)$$

$$K = \omega_h^\alpha, \quad \omega_u = \sqrt{\omega_h/\omega_b}, \quad (2.26)$$

and N is the order of approximation in the valid frequency range $(\omega_b; \omega_h)$.

Due to property in (2.5) for fractional orders $\alpha \geq 1$ it holds

$$s^\alpha = s^n s^\gamma, \quad (2.27)$$

where $n = \alpha - \gamma$ denotes the integer part of α and s^γ is obtained by the Oustaloup approximation by using (2.23). Thus, every operator in (2.13) may be approximated using (2.27) and substituted by the obtained approximation, yielding a conventional integer-order transfer function. For digital implementations, the obtained approximation may be converted to its discrete-time equivalent using a suitable method.

2.4 Fractional-Order Controllers

The notion of a fractional PID controller was introduced by Podlubny in [25, 26]. This generalized controller is called the $PI^\lambda D^\mu$ controller, and has an integrator with an order λ and a differentiator of order μ . Podlubny demonstrated that the fractional-order controller offers superior performance compared to an integer-order one when used in a control loop with a fractional-order plant. In more recent researches [10, 28] it has been confirmed that the fractional controller outperforms the integer-order PID controller.

In the Laplace domain the parallel form of the FOPID controller is given by

$$C_{FOPID}(s) = K_p + K_i s^{-\lambda} + K_d s^\mu. \quad (2.28)$$

Obviously, when taking $\lambda = \mu = 1$ the result is the classical integer-order PID controller.

Since we are dealing with band-limited approximations throughout this work, it is important to implement the fractional-order integrator component in (2.28) as

$$G_I(s) = \frac{1}{s^\lambda} = \frac{s^{1-\lambda}}{s} \quad (2.29)$$

for $0 < \lambda < 1$ since this ensures the effect of an integer-order integrator at low frequencies thereby resulting in faster convergence of the controlled output to its final value [16].

The fractional lead-lag compensator has the general form [14–16]

$$C_{LLC}(s) = K \left(\frac{1 + bs}{1 + as} \right)^\nu, \quad (2.30)$$

where K , b , a , and ν are design parameters. Assuming $a < b$, for $\nu > 0$ a lead compensator is obtained, otherwise for $\nu < 0$ a lag compensator is obtained.

In this work we also consider fractional-order inversion model based control (FOINVM). In particular, for a system described by (2.14) with a lag $L = 0$ we consider the following compensator:

$$C_{INVM}(s) = \frac{1 + Ts^\alpha}{K(1 + T_f s)}, \quad (2.31)$$

where parameters K , T , and α correspond to those in the FFOPDT model, and T_f is a time constant of a low-pass filter. The latter makes the controller realizable.

Throughout this work we typically assume that the control system is represented by a negative unity feedback of the form

$$G_c(s) = \frac{C(s)G_p(s)}{1 + C(s)G_p(s)}, \quad (2.32)$$

where $C(s)$ is the controller and $G_p(s)$ is the plant under control.

Further we briefly summarize the effects of extending the integral and derivative control actions to the fractional case [16]. The effects of fractional-order integrator and differentiator are shown in Fig. 2.3a and b under square and trapezoidal input signals, respectively.

The following can be achieved in the frequency domain by varying the power $\gamma \in [-1, 1]$:

- A constant change in the slope of the magnitude curve that varies between -20 and 20 dB/dec.
- A constant delay in the phase plot that varies between $-\pi/2$ and $\pi/2$ rad.

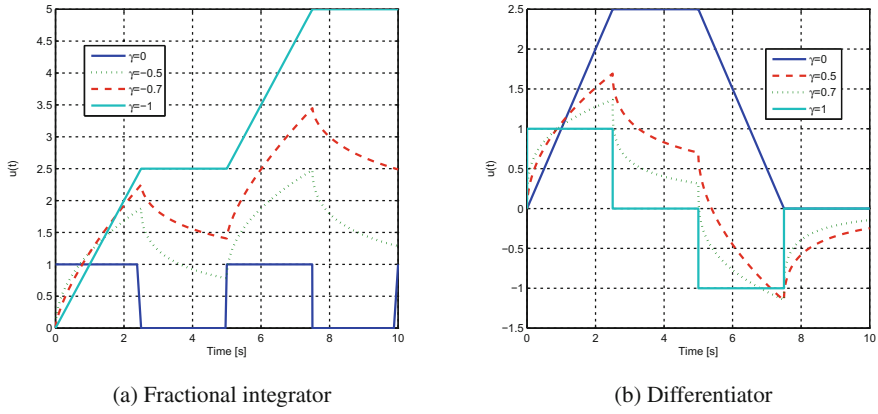


Fig. 2.3 Control actions in the time domain corresponding to s^γ

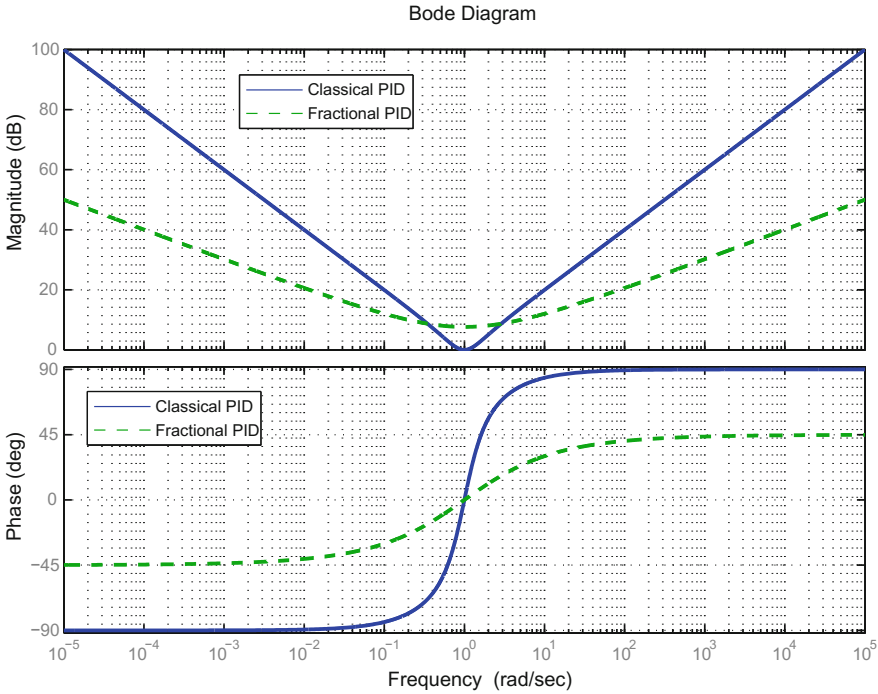


Fig. 2.4 Bode diagram of a frequency response of a classical PID controller with $K_p = K_i = K_d = 1$ and a fractional PID controller with $K_p = K_i = K_d = 1$, $\lambda = \mu = 0.5$

Consider a comparison between a classical PID controller with unity gains and a fractional one with unity gains and with fractional powers $\lambda = \mu = 0.5$ in the frequency domain given in Fig. 2.4.

It can be seen that introducing fractional powers for the integral and differential components of a suitable controller has clear benefits due to additional flexibility in tuning of such controllers to meet particular design specifications.

2.5 Optimization Methods

Application of numerical optimization methods form an important part of the present work. Therefore, we provide here the methods employed. The reason for the choice of particular methods is partially based on the reviews in [9, 20] and is detailed in relevant chapters of the present thesis.

2.5.1 Newton-Raphson Method

The Newton-Raphson method is an iterative process which belongs to a class of numerical methods for solving nonlinear equations [1, 7, 8]. We consider the problem in the form

$$F(x) = 0, \quad (2.33)$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Starting from some initial estimate x_0 the next estimate x^+ is obtained by means of

$$x^+ = x + \Delta x, \quad (2.34)$$

where Δx is the solution of

$$J \Delta x = -F, \quad (2.35)$$

and $J = dF/dx$ is the Jacobian matrix. In case of an univariate function $f(x)$ this process collapses to the well-known iterative formula

$$x^+ = x - \frac{f(x)}{f'(x)}. \quad (2.36)$$

The Newton-Raphson method is also used for solving subproblems in more sophisticated optimization algorithms.

2.5.2 Nonlinear Least-Squares Estimation Methods

The problem is to obtain a model of a certain system by means of minimization of the sum of squares (residual norm)

$$F = \sum_{i=1}^n \varepsilon_i^2 = \|\varepsilon\|_2^2, \quad (2.37)$$

where $\varepsilon_i = y_i - \hat{y}_i$ is the residual (simulation error), y_i is the true system output and $\hat{y}_i$ is the predicted output for collected samples $i = 1, 2, \dots, N$.

First, we consider a Trust Region Reflective method for handling large-scale bounded problems [4, 17]. Given a trust region Δ_k at every k th iteration the following steps are carried out [2]:

1. Compute F_k , g_k (gradient of F_k), D_k (positive diagonal matrix), H_k and C_k (scaling matrices), define the quadratic model

$$\psi_k(s) = g_k^T s + \frac{1}{2} s^T (H_k + C_k) s. \quad (2.38)$$

2. Compute a step s_k , with $x_k + s_k \in \text{int}(\mathcal{F})$, where $\mathcal{F}$ is the feasible region for search variable values, by solving the subproblem

$$\min_s \{ \psi_k(s) : \|D_k s\| < \Delta_k, s \in S_k \}, \quad (2.39)$$

where S_k is a small-dimensional subspace in $\mathbb{R}^n$.

3. If $F(x_k + s_k) < F(x_k)$, then $x_{k+1} = x_k + s_k$, otherwise x_k remains unchanged for the next iteration.
4. Adjust the trust region Δ_k .

In case of the least-squares problem the subspace S_k may be determined by taking into account

$$\min_s \{ \|J s + F\|_2^2 \}, \quad (2.40)$$

where J is the Jacobian of F .

Second, we consider the Levenberg-Marquardt algorithm [12, 18]. The search direction p_k of the Levenberg-Marquardt method is defined by the solution of equations at iteration step k

$$(J_k^T J_k + \lambda_k I) p_k = -J_k^T F_k, \quad (2.41)$$

where J_k is the Jacobian matrix, λ_k is a non-negative scalar, and I is the identity matrix [5].

2.5.3 Nelder-Mead Method

The Nelder-Mead simplex method is used for solving unconstrained optimization problems of the form

$$\min_x F(x), x \in \mathbb{R}^n. \quad (2.42)$$

It is a direct search method, and is therefore well-suited to optimize a function whose derivatives are unknown or non-existent [30]. In the following, we summarize the method described in [9].

First, an initial simplex is constructed by determining $n + 1$ vertices along with corresponding values of F . The k th iteration then consists of the following steps:

1. **Order.** Order the $n + 1$ vertices, so that $F(x_1) \leq F(x_2) \leq \dots \leq F(x_{n+1})$ is satisfied. Apply tie-breaking rules when necessary.
2. **Reflect.** Compute the reflection point

$$x_r = \bar{x} + \rho(\bar{x} - x_{n+1}), \quad (2.43)$$

where

$$\bar{x} = \sum_{i=1}^n x_i / n \quad (2.44)$$

is the centroid of the n best vertices. Evaluate $F_r = F(x_r)$. If $F_1 \leq F_r < F_n$, set $x_{n+1} = x_r$ and terminate the iteration.

3. **Expand.** If $F_r < F_1$, calculate the expansion point

$$x_e = \bar{x} + \chi(x_r - \bar{x}), \quad (2.45)$$

and evaluate $F_e = F(x_e)$. If $F_e < F_r$, set $x_{n+1} = x_e$ and terminate the iteration. Otherwise, set $x_{n+1} = x_r$ and terminate the iteration.

4. **Contract.** If $F_r \geq F_n$, perform a contraction between $\bar{x}$ and the better of x_{n+1} and x_r :

- (a) **Contract outside.** If $F_n \leq F_r < F_{n+1}$, calculate

$$x_c = \bar{x} + \gamma(x_r - \bar{x}) \quad (2.46)$$

and evaluate $F_c = F(x_c)$. If $F_c \leq F_r$, set $x_{n+1} = x_c$ and terminate the operation. Otherwise, go to Step 5 (perform a shrink).

- (b) **Contract inside.** If $F_r \geq F_{n+1}$, perform an inside contraction: calculate

$$x'_c = \bar{x} - \gamma(\bar{x} - x_{n+1}) \quad (2.47)$$

and evaluate $F'_c = F(x'_c)$. If $F'_c < F_{n+1}$ set $x_{n+1} = x'_c$ and terminate the iteration. Otherwise, go to Step 5 (perform a shrink).

5. **Shrink.** Define n new vertices from

$$x_i = x_1 + \sigma(x_i - x_1), \quad i = 2, \dots, n + 1, \quad (2.48)$$

and evaluate F at these points.

In the algorithm described above four scalar coefficients are used, i.e., the coefficients of reflection, expansion, contraction, and shrinkage, denoted by ρ , χ , γ , and σ , respectively. According to the original paper [19], these coefficients should satisfy

$$\rho > 0, \quad \chi > 1, \quad 0 < \gamma < 1, \quad 0 < \sigma < 1. \quad (2.49)$$

2.5.4 Optimization Problems with Bounds and Constraints

While certain optimization algorithms, such as the Levenberg-Marquardt and Nelder-Mead methods, are designed to solve unconstrained problems unbounded in the search space, it is possible to introduce both variable search space bounds and constraints [5] in terms of coordinate transformations and penalty functions.

Let x denote the vector of search variables of size $N \times 1$. For bound constraints, a coordinate transformation may be applied to each individual search variable. Let x_i^L and x_i^U denote the lower bound and upper bound on the i th search parameter, respectively, and denote by z the new search variable vector. Let $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ denote a coordinate transformation function, such that $x = \varphi(z)$ and $z = \varphi^{-1}(x)$.

In this work, we consider quadratic and trigonometric transformations. For problems with lower bounds we have

$$x_i = x_i^L + z_i^2, \quad (2.50)$$

from which it follows that $x_i \geq x_i^L$, since $z_i^2 \geq 0$. Initial estimates $z_{i,0}$ are obtained from original initial estimates $x_{i,0} \geq x_i^L$ as

$$z_{i,0} = \sqrt{x_{i,0} - x_i^L}. \quad (2.51)$$

For problems with upper bounds we have

$$x_i = x_i^U - z_i^2, \quad (2.52)$$

from which it follows that $x_i \leq x_i^U$, since $-z_i^2 \leq 0$. Initial estimates $z_{i,0}$ are obtained from original initial estimates $x_{i,0} \leq x_i^U$ as

$$z_{i,0} = \sqrt{x_i^U - x_{i,0}}. \quad (2.53)$$

Finally, if the problem has both lower and upper bounds, we have

$$x_i = x_i^L + (x_i^U - x_i^L) \frac{\sin(z_i) + 1}{2}, \quad (2.54)$$

from which it follows that $x_i^L \leq x_i \leq x_i^U$, since the values of the function $f(z_i) = (\sin(z_i) + 1)/2$ are always inside of the interval $[0, 1]$ and x_i is therefore bounded by $f(z_i) = 0 \Rightarrow x_i = x_i^L$, $f(z_i) = 1 \Rightarrow x_i = x_i^U$. Initial estimates $z_{i,0}$ are obtained from original initial estimates $x_i^L \leq x_{i,0} \leq x_i^U$ as

$$z_{i,0} = \Re \left(\arcsin \left(\frac{-2x_{i,0} + x_i^L + x_i^U}{x_i^L - x_i^U} \right) \right), \quad (2.55)$$

where $\Re(\cdot)$ denotes the real part.

Next, we consider constrained problems. To introduce constraints, a modification of the cost function

$$\kappa(\cdot) = \kappa^*,$$

where κ^* denotes the cost for the original optimization problem, is necessary. Let us define a function $f_{nz} : \mathbb{R} \rightarrow \mathbb{R}_+$ such that

$$f_{nz}(x) := \begin{cases} x, & x > 0 \\ 0, & x \leq 0. \end{cases} \quad (2.56)$$

Let us also define a *penalty function* $\rho : \mathbb{R} \rightarrow \mathbb{R}$ such that

$$\rho(x) := \begin{cases} e^\gamma - 1 + x, & x > \gamma \\ e^x - 1, & x \leq \gamma, \end{cases} \quad (2.57)$$

where $\gamma > 0$ is some predefined constant.

First, for general nonlinear inequality constraints of the form $c^{ni}(\cdot) \leq 0$, where $c^{ni} : \mathbb{R}^{q \times r} \rightarrow \mathbb{R}^{N_{ni} \times M_{ni}}$, we define the following penalty function $\kappa^{ni} : \mathbb{R}^{N_{ni} \times M_{ni}} \rightarrow \mathbb{R}$ as

$$\kappa^{ni}(c^{ni}(\cdot)) := \rho(c_\Sigma^{ni}(\cdot)), \quad (2.58)$$

where

$$c_\Sigma^{ni}(\cdot) = \sum_{i=1}^{N_{ni}} \sum_{k=1}^{M_{ni}} f_{nz}(c_{i,k}^{ni}(\cdot)). \quad (2.59)$$

Next, for general nonlinear equality constraints of the form $c^{ne}(\cdot) \leq 0$, where $c^{ne} : \mathbb{R}^{q \times r} \rightarrow \mathbb{R}^{N_{ne} \times M_{ne}}$, we define the penalty function $\kappa^{ne} : \mathbb{R}^{N_{ne} \times M_{ne}} \rightarrow \mathbb{R}$ as

$$\kappa^{ne}(c^{ne}(\cdot)) := \rho(c_\Sigma^{ne}(\cdot)), \quad (2.60)$$

where

$$c_{\Sigma}^{ne}(\cdot) = \sum_{i=1}^{N_{ne}} \sum_{k=1}^{M_{ne}} f_{nz}(|c_{i,k}^{ne}(\cdot)|), \quad (2.61)$$

where $|\cdot|$ denotes the absolute value.

The complete cost function κ for the constrained optimization problem thereby has the form

$$\kappa = \kappa^* + \kappa^{ni} + \kappa^{ne}. \quad (2.62)$$

References

1. K. Atkinson, *An Introduction to Numerical Analysis*, 2nd edn. (Wiley, 1989)
2. M.A. Branch, T.F. Coleman, Y. Li, A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM J. Sci. Comput.* **21**(1), 1–23 (1999)
3. Y.Q. Chen, I. Petráš, D. Xue, Fractional order control—a tutorial, in *Proceedings of ACC '09, American Control Conference*, (2009) pp. 1397–1411
4. T.F. Coleman, Y. Li, *An interior trust region approach for nonlinear minimization subject to bounds*. (Department of Computer Science, Cornell University, Ithaca, NY, USA, Tech. Rep., 1993)
5. P.E. Gill, W. Murray, M.H. Wright, *Practical optimization*. (London, Academic Press Inc, Harcourt Brace Jovanovich Publishers, 1981)
6. S.E. Hamamci, An algorithm for stabilization of fractional-order time delay systems using fractional-order PID controllers. *IEEE Trans. Automat. Contr.* **52**(10), 1964–1969 (2007)
7. F. Hildebrand, *Introduction to Numerical Analysis*. International Series in Pure and Applied Mathematics (McGraw-Hill, 1956)
8. C. Kelley, *Solving Nonlinear Equations with Newton's Method*. Fundamentals of Algorithms. (Society for Industrial and Applied Mathematics, 2003)
9. J.C. Lagarias, J.A. Reeds, M.H. Wright, P.E. Wright, Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM J. Optim.* **9**(1), 112–147 (1998)
10. Y. Luo, Y. Chen, Fractional-order [proportional derivative] controller for robust motion control: Tuning procedure and validation, in *Proceedings ACC '09, American Control Conference*, (2009), pp. 1412–1417
11. H. Malek, Y. Luo, Y.Q. Chen, Tuning fractional order proportional integral controllers for time delayed systems with a fractional pole, in *Proceedings of the ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 3 (2011), pp. 1–11
12. D.W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.* **11**(2), 431–441 (1963)
13. D. Matignon, Generalized fractional differential and difference equations: stability properties and modeling issues, in *Proceedings of Mathematical Theory of Networks and Systems Symposium*, (1998), pp. 503–506
14. C.A. Monje, B.M. Vinagre, A.J. Calderon, V. Feliu, Y.Q. Chen, Auto-tuning of fractional lead-lag compensators, in *Proceedings of the 16th IFAC World Congress*, (2005), pp. 319–324
15. C.A. Monje, B.M. Vinagre, V. Feliu, Y.Q. Chen, Tuning and auto-tuning of fractional order controllers for industry applications. *Control Eng. Pract.* **16**(7), 798–812 (2008)
16. C.A. Monje, Y.Q. Chen, B.M. Vinagre, D. Xue, V. Feliu, *Fractional-order Systems and Controls: Fundamentals and Applications, series*. Advances in Industrial Control (Springer, 2010)

17. J.J. Moré, D.C. Sorensen, Computing a trust region step. *SIAM J. Sci. Stat. Comput.* **4**, 553–572 (1983)
18. J. J. Moré, The Levenberg-Marquardt algorithm: implementation and theory, in *Numerical Analysis*, ed. by G. Watson. Lecture Notes in Mathematics, vol. 630 (Springer, Heidelberg, 1978), pp. 105–116
19. J.A. Nelder, R. Mead, A simplex method for function minimization. *Comput. J.* **7**(4), 308–313 (1965)
20. J. Nocedal, Theory of algorithms for unconstrained optimization. *Acta Numer.* **1**, 199–242 (1992)
21. A. O’Dwyer, *Handbook of PI and PID Controller Tuning Rules*, 3rd edn. (Imperial College Press, 2009)
22. A. Oustaloup, F. Levron, B. Mathieu, F.M. Nanot, Frequency-band complex noninteger differentiator: characterization and synthesis, vol. 47, no. 1 (2000), pp. 25–39
23. F. Padula, R. Vilanova, A. Visioli, H_∞ model matching PID design for fractional FOPDT systems. *Am. Control Conf. (ACC)* **2012**, 5513–5518 (2012)
24. F. Padula, A. Visioli, *Advances in Robust Fractional Control*, ser (Bücher. Springer International Publishing, SpringerLink, 2014)
25. I. Podlubny, Fractional-order systems and $PI^\lambda D^\mu$ -controllers, vol. 44, no. 1 (1999), pp. 208–214
26. I. Podlubny, L. Dorčák, I. Kostial, On fractional derivatives, fractional-order dynamic systems and $PI^\lambda D^\mu$ -controllers, in *Proceedings of 36th IEEE Conference Decision and Control*, vol. 5 (1997), pp. 4985–4990
27. I. Podlubny, *Fractional Differential Equations* (Academic Press, Mathematics in science and engineering, 1999)
28. M. Čech, M. Schlegel, The fractional-order PID controller outperforms the classical one, in *Process control 2006*. Pardubice Technical University, (2006), pp. 1–6
29. B.M. Vinagre, I. Podlubny, A. Hernández, V. Feliu, Some approximations of fractional order operators used in control theory and applications. *Fract. Calc. Appl. Anal.* **3**, 945–950 (2000)
30. M.H. Wright, Direct search methods: once scorned, now respectable, in *Numerical Analysis 1995 (Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis)* ed. by D.F. Griffiths, G.A. Watson. Pitman Research Notes in Mathematics, vol. 344 (Boca Raton, Florida: CRC Press, 1996), pp. 191–208
31. D. Xue, Y.Q. Chen, D.P. Atherton, *Linear Feedback Control: Analysis and Design with MATLAB (Advances in Design and Control)*, 1st edn. (Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008)

Fractional-order Modeling and Control of Dynamic
Systems

Tepljakov, A.

2017, XIX, 173 p. 79 illus., 52 illus. in color., Hardcover

ISBN: 978-3-319-52949-3