IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 13 November 2022, accepted 5 December 2022, date of publication 8 December 2022, date of current version 14 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3227864

RESEARCH ARTICLE

Fractional Generalized Predictive Control Strategy With Fractional Constraints Handling

MIGUEL ROMERO (Senior Member, IEEE), AND CAROLINA MAÑOSO (Senior Member, IEEE)
Control and Communication Systems Department, Universidad Nacional de Educación a Distancia (UNED), 28040 Madrid, Spain
Corresponding author: Miguel Romero (mromero@scc.uned.es)

This work was supported by Spanish Distance Education University (UNED) under Project 096-034298, from 2021 to 2022.

ABSTRACT The control strategies based on the methodology known as Model–based Predictive Control (MPC) have been developed and widely adopted to control real plants. This is mainly due to their intrinsic ability to handle constrains and their capacity to predict and optimize the future behavior of the process using a dynamical model of the plant. On the other hand, the mathematical tool known as fractional calculus has been currently used for reformulating the predictive control strategies to reach a better performance adding new control parameters. This work extends the use of fractional operators for the constraints in one type of fractional predictive control strategy known as Fractional–order Generalized Predictive Control (FGPC), interpreting and discussing the results. In addition, a new method to soften constraints using fractional operator is proposed and illustrated with examples, even to adjust the final response of the system. A practical tuning of the rest of controller parameters with the help of a well–known mathematical software is also included to make use of the beneficial characteristics of this fractional predictive formulation.

INDEX TERMS Model predictive control, fractional calculus, fractional constraints, optimization.

I. INTRODUCTION

The term MPC (Model Predictive Control) began to be used in the late of 1970s in works as [1] and [2] where this control strategy was used mainly within the petrochemical industry to meet its specialized control needs. These early works were followed by others both industrial and academic environments, covering a wide spectrum of applications as are described in [3], [4], and [5].

Due to its success, nowadays, MPC has become a standard in these environments, where there exist many implementations: DMC (Dynamic Matrix Control), GPC (Generalized Predictive Control), etc. All of them share a common methodology, they rely on a dynamical model that represents the plant to predict its future behavior by means of the minimization of a costs function over a time interval [6], [7], [8]. Obviously, the more similar the model is to the real plant, the better the estimates of its future behavior are. This is not easy to achieve in many real cases, since there are plants with complex behaviors such as the stop–and–go maneuvers in commercial gasoline–propelled vehicles, controlling the throttle pedal which exhibit highly non-linear dynamics at low speed [9]; or the viscoelasticity phenomenon in rheological tissues describing the flow of blood through the arteries [10]. In some of these cases, applying an advanced control strategy could be very beneficial to obtain the expected results compensating unmodeled dynamics and external disturbances. To do so, it might even be necessary to apply a hybrid model as in [11] and [12].

In this sense, some recent implementations use both a fractional model to accurately describe non–integer order dynamics exhibited by some real physical systems and a fractional formulation for the controller to reach a better performance. However, the idea of using fractional operators both for the formulation of the controller and for describing the dynamics of the used model is not exclusive to the MPC. These operators have been used in other control strategies to enhance the system performance. For example, the classical PID control was upgraded using fractional operators to re-formulate its integral and derivative terms [13], [14], [15] or the fractional–order control methodology known as CRONE [16].

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo.

The fractional predictive control strategy known as FGPC combines the valuable characteristics of fractional calculus and predictive control by means of a real–order fractional cost function that is formulated using fractional operators. This is the fractional control strategy that is used in this work. In the literature, we can find more and more examples of authors who have followed this proposal; without the intention of being exhaustive; we can mention the manuscript [17] where the so–called FPFC controller is used in fractional industrial processes, [18] where a neurophysiological process has been modeled by a fractional order system, [19] where the fractional order model of heating furnace is used or [20] where two rods thermal system has been identified and used.

Nowadays, the predictive control strategies have demonstrated to be a valuable option due to their intrinsic characteristics both for compensating un–modelled dynamics and for reaching complex control objectives by means of the advisable setting of constraints. Although the computational cost for large number of variables and constraints of large–scale processes could be troublesome, there are current techniques for mathematical processing using the so–called GPU computing to get an important performance boost [21].

In this sense, the main aim in this paper is to introduce the use of fractional constraints in the predictive fractional control strategy by means of using fractional operators not only in the cost function but also in the mathematical formulation of its constraints. Their effect on system performance is interpreted and discussed, analyzing the usefulness of the proposal to soften constraints and adjust the final response of the system. On the other hand, the optimal tuning method to obtain the controller parameters is revised and illustrated with some examples using two different mathematical solvers.

The remainder of this paper is structured as follows: In Section II the fundamentals of the fractional calculus and the fractional–order definite integral operator are described. Section III summarizes the fundamentals of fractional predictive control strategy and includes the formulation, design and tuning of the FGPC controller using a practical guide with optimization software. Section IV introduces the fractional constraints for FGPC controller, describing mathematically their formulation and discussing its impact on system performance. Furthermore, a new method to soften constraints using fractional operators is also introduced and explained. Finally, Section V draws the main conclusions of this work.

II. FRACTIONAL CALCULUS

Fractional calculus is defined as a generalization of derivatives and integrals to non–integer orders, in this way, it allows to make calculations such as differentiate a function to real, even complex order [22]. This branch of mathematical analysis dates from the 17th century when Leibniz and L'Hôpital discussed about the possibility of non–integer derivatives, specifically that n could be a fraction 1/2 for nth derivative. Although this calculus is an old idea, it was really developed at the 19th century by Liouville, Letnikov, Riemann and other mathematicians [23].

There exist several ways to evaluate fractional operators [24], [25], which are commonly represented by D^{α}, where positive values of α represents derivatives and negative values correspond to integrals. Since MPC formulation is usually described in discrete terms, in this work, it will be only considered the Grünwald–Letnikov (GL) definition which expression is (1).

$$D^{\alpha} f(t)_{t=kh} = \lim_{h\to 0} h^{-\alpha} \sum_{j=0}^{\infty} (-1)^j \binom{\alpha}{j} f(kh - jh), \quad \forall \alpha \in \mathbb{R} \tag{1}$$

It is important to note that expression (1) has infinite discrete terms. Because of this specific feature, fractional calculus is a valuable tool for modeling and describing complex effects associated with infinite memory behaviors such as polymer viscoelasticity.

The dynamic behavior of fractional systems is often described using the Laplace transform. Expression (2) gives the Laplace transform of the GL definition under zero initial conditions.

$$L\left\{D^{\pm\alpha} f(t)\right\} = s^{\pm\alpha} F(s), \quad \forall \alpha \in \mathbb{R} \tag{2}$$

Nevertheless, the discretization of (2) does not lead to a transfer function with a limited number of coefficients in z. Thus, these infinite coefficients can be calculated using the binomial expression (3) or using the well–known recursive algorithm [26].

$$\omega_j = (-1)^j \binom{\alpha}{j}, \quad \forall j \in [0, \infty) \tag{3}$$

The coefficients evolve quickly approaching to 0 in accordance with the so–called short memory principle which is generally used to evaluate fractional operators because of only the recent past plays an important role. Therefore, its application leads to a n–term truncated series, paying a penalty in the form of some inaccuracy [26].

In this paper, the fractional–order definite integral operator ${}^{\alpha}I_a^b(\cdot)$ is used. The concept of integer definite integral can be generalized to real–order for a function $f(x)$ using the expression (4).

$${}^{\alpha}I_a^b f(x) = \int_a^b \left[D^{1-\alpha} f(x)\right] dx \quad \alpha, a, b \in \mathbb{R} \tag{4}$$

Using the previous GL definition and considering that $D^{1-\alpha}[f(x)] \neq 0$, this fractional operator can be discretized with sampling period Δx as:

$${}^{\alpha}I_a^b f(x) = \Delta x^{\alpha} W' f \quad \text{with } w_j = \omega_j - \omega_{j-n}, \quad n = b - a \tag{5}$$

where f is a vector with the values of the function and W is a vector with a weight sequence using the coefficients of binomial expression (3). See [27] and the references within.

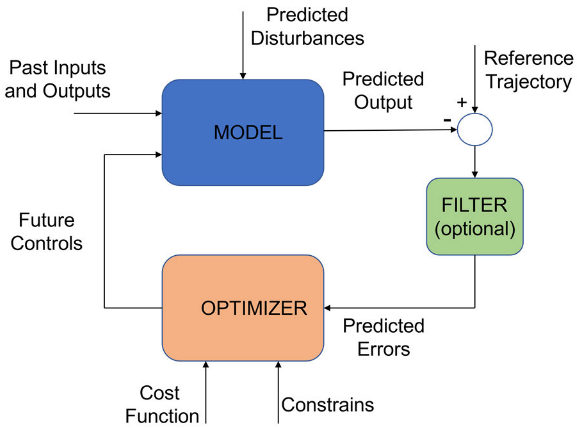

FIGURE 1. FGPC basic diagram.

III. FRACTIONAL PREDICTIVE CONTROL STRATEGY

The Fractional–order Generalized Predictive Controller, FGPC, is defined as the fractional generalization of GPC control strategy [28], [29], that is one of the most representative MPC formulations. FGPC and GPC share characteristics but there exist important differences as they do not define the same set of controllers [30]. FGPC uses the fractional operators to enhance the system performance and it is part of predictive controller family. It follows the concept behind all model based predictive algorithms. This idea has been used by some authors to design their control strategies; without the intention of being exhaustive; such as [31] where a fractional predictive controller is applied to Steam/Water Loop in Large Scale Ships in a distributed scheme, [32] where this fractional control strategy has been used to achieve the optimal frequency control of an islanded microgrid or [33] where the fractional predictive method has been tested with an industrial heating furnace.

The FGPC control strategy can be summarized as follows. At each present instant of time, t, the controller generates a set of future control signals $\Delta u(t+k|t)$ based on the prediction of the future process outputs $y(t+k|t)$ by means of minimization of a defined cost function within a time interval define by $[N_1, N_2]$. However, only the first term of the control signal vector is used as system input $\Delta u(t|t)$, the rest ones are neglected. Next step, $t+1$, the algorithm is repeated to calculate the new system input, $\Delta u(t+1|t+1)$. Therefore, the prediction window has moved forward (receding horizon control). The Figure 1 schematically presents this strategy.

Traditionally in MPC, the cost function is formulated using a quadratic criterion due to its mathematical convenience [34] for its optimization. FGPC also makes use of this formulation. However, some MPC strategies with non–quadratic formulations could be also found in the literature as [35] and [36]. In any case, this cost function has as input the predicted errors which are defined as the difference between the predicted output and the reference trajectory, $e(t+k\,|\,t) = y(t+k\,|\,t) - r(t+k|t)$, filtered or not.

On the other hand, FGPC uses a discrete time linear model to capture the dynamic behavior of the real plant, where the equations that describe its dynamics are formulated using integer–order calculus. However, a fractional linear model could also be used in case of the plant to be controlled had that dynamic, but it is not the subject of this work.

Finally, the possibility of using process constraints in predictive control strategies is one of their most valuable characteristics. FGPC uses constraints that are expressed as a set of linear inequalities:

$$L\Delta u(k) \leq l(k) \tag{6}$$

where L is a matrix and l is a vector with c terms. c is the number of constraints.

Due to the use of a quadratic programming algorithm in FGPC, which allows only to satisfy constraints with general expression as (6), only linearly dependent variables of control signal Δu are used.

A. FORMULATION OF THE FGPC CONTROLLER

FGPC Controllers are characterized by relying on a CARIMA (Controlled Auto–Regressive Integrated Moving Average) model to describe the system dynamics (7).

$$A\left(z^{-1}\right)y(t) = B\left(z^{-1}\right)u(t) + \frac{T\left(z^{-1}\right)}{\Delta}\xi(t) \tag{7}$$

The numerator and denominator of the model transfer function are $B\left(z^{-1}\right)$ and $A\left(z^{-1}\right)$, respectively. Δ is the increment operator, $\xi(t)$ is the uncorrelated zero–mean white noise and $T\left(z^{-1}\right)$ is a (pre)filter to improve the system robustness that rejects disturbances and noise.

Using the model (7), the future outputs $y(t+k|t)$ can be predicted at any time. So, it is common to write these outputs as the sum of two terms: controlled response and free response, $y(t+k\,|\,t) = y_C(t+k\,|\,t) + y_F(t+k|t)$, with $y_C \equiv G \cdot \Delta u$ the part of the future output that depends on the future control Δu; and y_F the part of the future output that does not depend on Δu; G is a matrix with the step response coefficients of the model. For the sake of simplicity in the notation $(\cdot|t)$ is omitted, since all expression from this point are referred to the present time t, unless otherwise stated.

$$G = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ g_N & g_{N-1} & \cdots & g_{N-N_u+1} \end{bmatrix} \tag{8}$$

where $N = N_2 - N_1 + 1$ represents the number of predictions to make.

On the other hand, the FGPC cost function has the expression (9) and it is defined by means of the fractional–order definite integral operators ${}_a^bI^{\alpha}(\cdot)$ discretized with sampling period Δt and evaluated using (4).

$$J_{FGPC}(\Delta u, t) = {}^{\alpha}I_{N_1}^{N_2}\left[e(t)\right]^2 + {}^{\beta}I_1^{N_u}\left[\Delta u(t-1)\right]^2 \tag{9}$$

where N_1 and N_2 are the minimum and maximum costing horizons, respectively, N_u represents the control horizon, and it is assumed that $u(t)$ remains constant from time instant $t+N_u$ $(1 \leq N_u \leq N_2)$. The parameters α, $\beta \epsilon \mathbb{R}$ are the orders of the fractional definite integral operator (4), discretized with sampling period Δt. This fractional operator, under very general assumptions and using the GL definition (1), verifies the so–called fractional–order Barrow's rule (10). See [37] for a more exhaustive explanation.

$$ {}^{\alpha}I_a^b f(x) = F^{\alpha}(b) - F^{\alpha}(a) \tag{10} $$

where $F^{\alpha}(x) \equiv I^{\alpha} f(x)$ is any α-order primitive of $f(x)$.

After using the above operator, it is possible to translate the cost function (9) into its matrix form (11). Γ and Λ represent infinite–dimensional square real weighting matrices.

$$ J_{FGPC}(\Delta u, t) = e' \Gamma(\alpha, \Delta t) e + \Delta u' \Lambda(\beta, \Delta t) \Delta u \tag{11} $$

The discrete expression of the operator (10) can be found in [37] but it is reproduced here for convenience and adapted to the notation of (11).

$$ \Gamma \equiv \Delta t^{\alpha} diag \left(\ldots \quad w_m \quad w_{m-1} \quad \ldots \quad w_1 \quad w_0 \right) \tag{12} $$

with $w_j = \omega_j - \omega_{j-n}$, $n = N_2 - N_1$, $\omega_k = (-1)^k \begin{pmatrix} -\alpha \\ k \end{pmatrix}$

$$ \Lambda \equiv \Delta t^{\beta} \mathrm{diag} \left(\ldots \quad w_{N_u-1} \quad w_{N_u-2} \quad \ldots \quad w_1 \quad w_0 \right) \tag{13} $$

with $w_j = \omega_j - \omega_{j-n}$, $n = N_u - 1$, $\omega_k = (-1)^k \begin{pmatrix} -\beta \\ k \end{pmatrix}$

In order to obtain the FGPC optimal control law and due to the memory characteristic of the fractional operator, the matrix form of cost function is rewritten using a new notation where the symbols $(\rightarrow)$ and $(\leftarrow)$ represent future and past values, respectively [27].

$$ \begin{aligned} J_{FGPC} &= \begin{bmatrix} \overleftarrow{e}' & \overrightarrow{e}' \end{bmatrix} \begin{bmatrix} \overleftarrow{\Gamma} & 0 \\ 0 & \overrightarrow{\Gamma} \end{bmatrix} \begin{bmatrix} \overleftarrow{e} \\ \overrightarrow{e} \end{bmatrix} + \begin{bmatrix} \overleftarrow{\Delta u}' & \overrightarrow{\Delta u}' \end{bmatrix} \\ &\quad \times \begin{bmatrix} \overleftarrow{\Lambda} & 0 \\ 0 & \overrightarrow{\Lambda} \end{bmatrix} \begin{bmatrix} \overleftarrow{\Delta u} \\ \overrightarrow{\Delta u} \end{bmatrix} \\ &= \left(\overrightarrow{e}' \overrightarrow{\Gamma} \overrightarrow{e} + \overrightarrow{\Delta u}' \overrightarrow{\Lambda} \overrightarrow{\Delta u} \right) + \left(\overleftarrow{e}' \overleftarrow{\Gamma} \overleftarrow{e} + \overleftarrow{\Delta u}' \overleftarrow{\Lambda} \overleftarrow{\Delta u} \right) \\ &= \overrightarrow{J_{FGPC}} + \overleftarrow{J_{FGPC}} \end{aligned} \tag{14} $$

The manipulated independent variable is $\overrightarrow{\Delta u}$, the optimal control law is given by the expression $\Delta u^*_{FGPC}(t) =$ arg J_{FGPC}.

Although the cost function (14) has two terms, past and future values of the variables, the minimization only depends on the future ones. On the other hand, predictions are made using the model (7). Therefore

$$ \overrightarrow{J_{FGPC}} = \overrightarrow{\Delta u}' \left(G' \overrightarrow{\Gamma} G + \overrightarrow{\Lambda} \right) \overrightarrow{\Delta u} - 2 \overrightarrow{E_0}' \overrightarrow{\Gamma} G \overrightarrow{\Delta u} + \overrightarrow{E_0}' \overrightarrow{\Gamma} \overrightarrow{E_0} \tag{15} $$

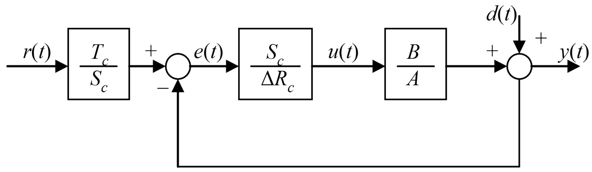

FIGURE 2. Closed–loop equivalent control system schema.

The predictive errors vector has the expression (16).

$$ E_0 \equiv \left(e_0(t+1), \ldots, e_0(t+N_2)\right)' = r(t+j) - y_F(t+j) \tag{16} $$

Finally, the optimal control law has the expression (17) without active constraints.

$$ \begin{aligned} \Delta u^*_{FGPC}(t) &= arg \min_{\Delta u} J_{FGPC} = \left(G' \overrightarrow{\Gamma} G + \overrightarrow{\Lambda} \right)^{-1} G' \overrightarrow{\Gamma} \overrightarrow{E_0} \\ &\equiv K \overrightarrow{E_0} \end{aligned} \tag{17} $$

This optimization problem has a solution if the matrix (18) can be inverted. This square hessian matrix is symmetric, and its eigenvalues determine the closed–loop system response.

$$ H = G' \overrightarrow{\Gamma} G + \overrightarrow{\Lambda} \tag{18} $$

As it is well–known, if matrix (18) is positive–definite, that is, all its eigenvalues are positive, then the solution of optimization problem is unique. Thus, we could have some negative values in the weighting sequences but H would be positive definite.

This optimization without activating the constraints leads to a control law that is linear time invariant (LTI) and can be pre–computed in advance. The equivalent closed loop control schema is shown in Figure 2.

The polynomials R_c and S_c are obtained from the model polynomials A and B. See [38] for an exhaustive explanation.

T_c is considered as a filter mainly due to the incremental constitution of control law. The selection of the polynomial T_c is not an easy task, there exist a general guideline in the manuscript [39] where it is recommended the expression (19).

$$ T_c(z^{-1}) = \left(1 - \rho z^{-1} \right)^{N_1} \tag{19} $$

where ρ is recommended to be closed to the dominant pole of the system model.

Sensitive functions will play an important role in the tuning method explained posteriorly. Although many sensitivity functions can be defined in predictive control [23], we shall only consider two of them, the *classical* S and T functions which expressions are (20) and (21), respectively.

$$ S = \frac{R_c \Delta A}{R_c \Delta A + S_c B} \tag{20} $$

$$ T = \frac{S_c B}{R_c \Delta A + S_c B} \tag{21} $$

B. PRACTICAL TUNING OF FGPC USING MATLAB^{TM} OPTIMIZATION TOOLBOX

The proposed optimal tuning method for FGPC is shown in the Appendix of this manuscript. This method allows us to set horizon parameters together with fractional orders using different mathematical solvers included in numeric computing software as Matlab, GNU Octave… Evidently, this is a critical task because closed–loop system performance depends on this choice.

The optimization toolbox of Matlab provides functions to find parameters that minimize or maximize the objectives subjects to a set of constraints. Specifically, it includes solvers for general optimization problems as the one described above. As the previous set of functions to optimize are continuous, (37)–(41), we have chosen specifically two solvers to resolve it. The solver for nonlinear optimization problems, *fmincon*, and the solver that uses genetic algorithms, *ga*. Both of them have a similar syntaxis:

mincon(@objfun,x0,[],[],[],[],[],[],@confuneq,options)

where:

- objfun: main function to optimize.
- x0: initial values.
- confuneq: constraints.
- options: optimization options.

ga(@objfun,nvars,[],[],[],[],[],[],@confuneq,options)

where:

- objfun: main function to optimize.
- nvars: number of variables.
- confuneq: constraints.
- options: optimization options (in this case, initial values are passed within this vector of parameters).

In order to illustrate the optimization process, two FGPC controllers will be tuned using previous solvers. The dynamics of the plant to be controlled has been discretized with a sampling time equal to 0.1 s. Its transfer function is shown in expression (22).

$$G\left(z^{-1}\right) = \frac{0.0952}{1 - 0.9048z^{-1}} z^{-1} \qquad (22)$$

Both controllers will be obtained using the following parameters:

- $N_1 = 1$
- $N_u = 2$
- $N_2 = 30$
- $T_c\left(z^{-1}\right) = 1 - 0.9z^{-1}$ (Using prefilter.)

and they will be optimized to fulfill the following requirements:

- Maximize the gain margin (no specification is set on the phase margin).
- $|S(j\omega)| \leq -15\, dB$ for $\omega \leq 0.1\, rad/s$
- $|T(j\omega)| \leq -15\, dB$ for $\omega \geq 10\, rad/s$

In order to find the optimal pair α and β, an initial seed is needed to initialize the optimization algorithm. To do so, the gain and phase margin is calculated for different values of the fractional parameters, obtaining the graphic representation that is depicted in Figure 3 and Figure 4.

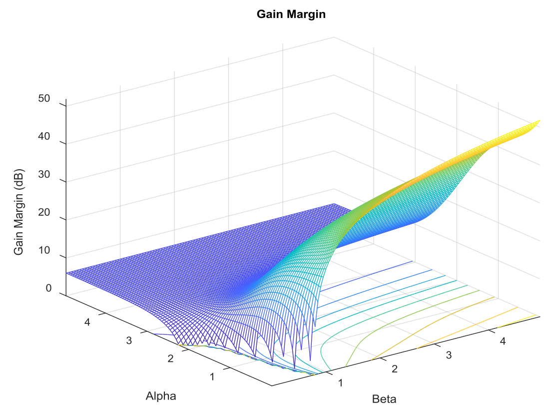

FIGURE 3. Gain margin vs. α and β for obtained optimization seed.

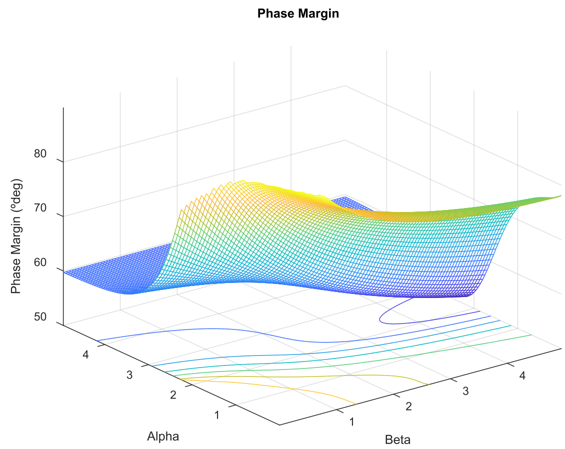

FIGURE 4. Phase margin vs. α and β for obtained optimization seed.

TABLE 1. Results of the optimization process.

Solver	Fractional parameters	Gain and Phase margin
fmincon	$\alpha = 0.6379$, $\beta = 4.8927$	43.99 *dB*, 75.30º
ga	$\alpha = 0.2228$, $\beta = 1.8411$	36.94 *dB*, 82.46º

The values $\alpha_0 = 0.9$ and $\beta_0 = 0.8$ are selected as seed to start the optimization algorithm due to their corresponding initially good gain and phase margins.

The optimization process has been carried out in an interval of 20–30 seconds for *fmincon* solver and 60–100 seconds for *ga* solver using a PC computer with AMD Ryzen^{TM} 1600 processor running MATLAB^{TM} 2019b. The results of the optimization process as well as the parameters of the resulting system are shown in the Table 1.

Both systems are stable and have definite-positive H matrices with the following eigenvalues (0.5179, 3.4051) and (0.5481, 2.7256) for the *fmincon* and *ga* solvers, respectively. They also meet the requirements imposed in terms of sensitivity functions as shown in Figure 5. The gain margin obtained

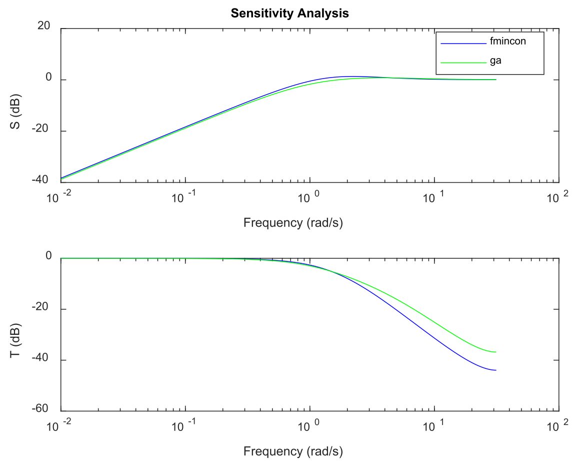

FIGURE 5. Sensitivity functions comparison.

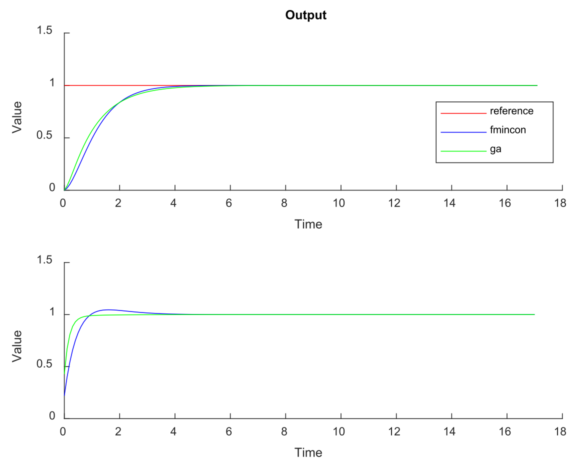

FIGURE 6. Step responses of FGPC controllers.

with the solver *fmincon* is larger, but the system obtained by means of the solver *ga* has better phase margin, although it is not the objective of this optimization.

Figure 6 depicts the responses of both systems. The system response with the controller obtained by *fmincon* solver is represented in blue line and the system response with the controller obtained by *ga* solver is represented in green line. One observes that these responses are quite similar. They follow the reference with no overshoot and fulfilling gain margins, sensitivity functions and robustness specifications in an easy and straightforward way.

In practice, due to the optimization problem and the constraints involved, the procedure described here must be repeated a few times. Normally, in the first attempts the solvers could not converge and we would not obtain the desired results due to feasibility problems. However, the experience gained during one of the runs is used to better adjust of the parameters such as the initial seed in the next run until consistent results are obtained. Moreover, this method discards during the execution of the algorithm those systems that are unstable.

IV. FRACTIONAL CONSTRAINTS

Although any real control system is subject to diverse types of constrains: physical restrictions (actuators have a minimum and a maximum value that limit their operations), security boundaries (outside of them, the system integrity and people safety are not guaranteed), product quality specifications (without them, the product would not meet the established standards), technological requirements, etc. [40]; It is not usual that these constraints have been set explicitly in the design of many control strategies. Therefore, the designed controllers could result in inadequate closed–loop performance or undesirable behavior.

In the case of predictive controllers, the constraints are introduced natively in the cost function. Therefore, they are systematically included during the controller design process, constituting one of the most important advantages of this control strategy in comparison with others. However, as the optimal control sequence must be obtained at each sampling time by minimization of the cost function (14) subject to set of constraints (6), including these constraints leads to an increase of the computational cost necessary to solve the mathematical problem in real time.

Due to the growing presence of systems with fractional operating dynamics such as fuel cells [41], supercapacitors [42] or applications in viscoelasticity [43], the inclusion in the control strategy of constraints that act on the fractional increments of certain variables could be necessary. To this purpose, their formulation will be generalized in the same way as in the formulation of the cost function of the controller. Furthermore, in this paper, they are utilized to define a new method to soften constraints applying fractional operators, which may be of interest even for applications that do not present these fractional dynamics.

It is well known that the presence of hard constraints in predictive control strategies could lead to infeasibility causing instabilities [44]. In order to avoid them, some constraints could be relaxed by treating them as soft constraints. The case of infeasibility is treated by other authors using techniques as introducing slack variables that soften some of the constraints or even temporally removing some of them [45].

Having these ideas in mind, we consider the possibility of introducing constraints with fractional behavior in our fractional predictive control strategy. To do so, we introduce the fractional δ–increment of the future value of the control signal, $u(t + N_u - 1)$, depending on the control horizon, N_u. Expression (23) shows the fractional expansion using the binomial expression (3).

$$D^{\delta} u\,(t + N_u - 1) = \lim_{h \to 0} h^{-\delta} \sum_{j=0}^{\infty} \omega_j u\,(t + N_u - 1 - j) \tag{23}$$

The value of the fractional δ–increment has the expression:

$$\begin{aligned}\Delta^{\delta} u\,(t + N_u - 1) &= \sum_{j=0}^{\infty} \omega_j u\,(t + N_u - 1 - j)\\ &= [\omega_0 u\,(t + N_u - 1) + \omega_1 u\,(t + N_u - 2)\end{aligned}$$

$$+\ldots+\omega_{N_u-1}u\,(t)+\omega_{N_u}u\,(t-1) + \omega_{N_{u+1}}u\,(t-2)+\ldots] = \left[\sum \overleftarrow{u} + \sum \overrightarrow{u}\right] \quad (24)$$

Once the expression of the summation is performed in (24), it is obtained the expression of δ–increment of the control signal, $\Delta^{\delta}u$. Analogously to what happened with the cost function (14), the expression (24) depends on the values of the control signal, $u\,(t+k)$ and presents two terms: one that depends on the future values (when k is higher or equal to 0), $\overrightarrow{u}$, and another ones that depends on the past values of this signal (when k is less than 0), $\overleftarrow{u}$. Assuming:

- a sufficiently high value of N_u, and
- the past terms of increments $\overleftarrow{u}$ are not variables but constants. Therefore, they do not affect to the optimization problem.

We can represent the constraint $\Delta^{\delta}u(t+N_u-1) \leq \Delta^{\delta}u_{max}$ in the form (25).

$$\begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \omega_{N_u-1} & \omega_{N_u-2} & \omega_{N_u-3} & \ldots & \omega_0 \end{bmatrix} \times \begin{bmatrix} u\,(t) \\ u\,(t+1) \\ \ldots \\ u\,(t+N_u-1) \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ \Delta^{\delta}u_{max}\,(t+N_u-1) \end{bmatrix} \quad (25)$$

We also shall consider constraints on a linear combination of the increments of control signal Δu. Manipulating algebraically the expression (25), we can put the constraints in terms of $\Delta u(t)$ using (26).

$$\Delta u\,(t) = u\,(t) - u(t-1) \quad (26)$$

So:

$$\begin{aligned} &\sum_{l=0}^{N_u-1} \omega_l u\,(t+N_u-2-l) \\ &= \sum_{l=0}^{N_u-1} \omega_l\,[(\Delta u\,(t+N_u-2-l))\,(u\,(t+N_u-3-l))] \\ &= \omega_0\,[\Delta u\,(t+N_u-1)+u\,(t+N_u-2)] \\ &\quad +\omega_1\,[\Delta u\,(t+N_u-2)+u\,(t+N_u-3)]+\ldots \\ &\quad +\omega_{N_u-1}\,[\Delta u\,(t)+u\,(t-1)] \\ &= \omega_l\left[\sum \Delta u\,(t)\right]+\Delta^{\delta}u\,(t+N_u-2) \end{aligned} \quad (27)$$

In matrix form:

$$\begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \omega_{N_u-1} & \omega_{N_u-2} & \omega_{N_u-3} & \ldots & \omega_0 \end{bmatrix} \times \begin{bmatrix} \Delta u\,(t) \\ \Delta u\,(t+1) \\ \ldots \\ \Delta u\,(t+N_u-1) \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ \Delta^{\delta}u_{max}\,(t+N_u-1) - \Delta^{\delta}u(t+N_u-2) \end{bmatrix} \quad (28)$$

In a similar way, the constraint for minimum values $\Delta^{\delta}u(t+N_u-1) \geq \Delta^{\delta}u_{min}$ can be obtained IV-A.

$$\begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ -\omega_{N_u-1} & -\omega_{N_u-2} & -\omega_{N_u-3} & \ldots & -\omega_0 \end{bmatrix} \times \begin{bmatrix} \Delta u\,(t) \\ \Delta u\,(t+1) \\ \ldots \\ \Delta u\,(t+N_u-1) \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\Delta^{\delta}u_{min}\,(t+N_u-1) + \Delta^{\delta}u(t+N_u-2) \end{bmatrix} \quad (29)$$

The expressions (28) and (29) show as constraints affect to the future values of the control signal not to the values at the present instant of time. This time displacement allows to relax the fulfillment of the constraint depending on the value of control horizon, N_u. Evidently, a value of $N_u = 1$ implies a "hard" compliance with the constraint. Furthermore, the assigning $\delta = 1$ leads to set the constraint as in the known integer case.

Although we have considered the constraints as hard and they must be satisfied at any cost from a mathematical point of view, increasing the value of N_u allows us to soften them the higher its value is, because of the hard constraints compliance will be further away in time. Therefore, the proposed method softens constraints on the future predicted values of the constraints, not on the values at the present instant of time. In the next subsection, we will illustrate these concepts with some examples.

A. COMPUTING SIMULATION OF FRACTIONAL CONSTRAINTS

In order to illustrate how these fractional constraints work, computational simulations have been performed, where the impact of the constraints on the closed–loop performance is shown when the values of δ and N_u are modified. The plant is the one previously proposed for illustrating the optimization (22). The controller setting parameters have also been maintained.

To do so, we have used again the optimization toolbox of Matlab, specifically, the solver *quadprog* [46], that allows to

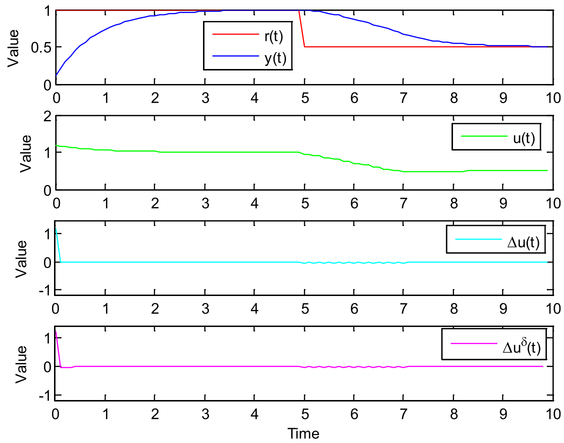

FIGURE 7. Simulation 1 of the constrained controller 1.

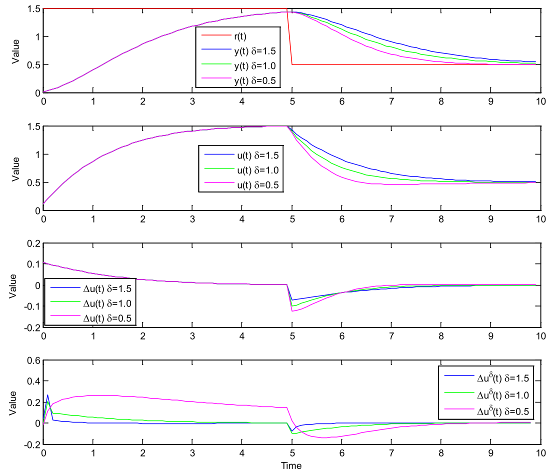

FIGURE 8. Simulation 2 of the constrained controller 1.

solve quadratic programming problems subject to constraints in the form:

$$\min_{x} \frac{1}{2}x'Hx + f'x$$
$$subject\ to\ Ax \leq b \tag{30}$$

From equations (15) and (18) we can obtain the values of H and f with ease. Therefore,

$$\min_{\Delta u} \frac{1}{2}\overrightarrow{\Delta u}'\left(G'\overrightarrow{\Gamma}G + \overrightarrow{\Lambda}\right)\overrightarrow{\Delta u} + \overrightarrow{E_0'}\overrightarrow{\Gamma}G\overrightarrow{\Delta u} \tag{31}$$

The values of matrices A and b are obtained directly from the expressions (28) and IV-A for each case. The solver syntax is:

quadprog(H,f,A,b)

Initially, we set a constraint on the minimum value of the fractional δ–increment as it is shown in expression (32). Figure 7 depicts the simulation for $\delta = 1$ and $N_u = 1$. It can be easily seen that $\Delta u(t)$ and $\Delta^\delta u$ are coincident as expected, $\Delta^\delta u(t + N_u - 1) = \Delta^1 u(t + 1 - 1) = \Delta u(t)$. It is also observable that from $t = 5$ the constraint acts and the output cannot follow the reference as fast as it would be desirable, even a small ripple appears.

$$\Delta^\delta u(t + N_u - 1) \geq -0.05 \tag{32}$$

We now consider a new value for the control horizon, $N_u = 3$, to do a new simulation. Figure 8 depicts this simulation for three values of δ, specifically, $\delta = (0.5, 1.0, 1.5)$. The lower plot shows the values of $\Delta^\delta u(t)$ (just when the step down occurs). The values at this moment for $t = 5$ are:

$$\Delta^{0.5}u(t)_{t=5} = 0.0180$$
$$\Delta^{1.0}u(t)_{t=5} = -0.0992$$
$$\Delta^{1.5}u(t)_{t=5} = -0.0993 \tag{33}$$

It seems that the constraint set by (32) is not fulfilled for $\delta = 1.5$ and $\delta = 1.0$. The values of the δ–increments are below the minimum value set for them. However, the constraint concerns the predicted value for $\Delta^\delta u(t + N_u - 1)$ not the value at the current time $\Delta^\delta u(t)$ that is calculated using the pass values as is indicated by GL definition (1). This results in a mathematical method of applying soft constraints to the system, which can be adjusted using the fractional δ parameter.

On the other hand, the variation of the δ parameter has also affected the temporal response of the system. One observes that both the output of the system, $y(t)$, and the value of the control signal, $u(t)$, are equal for the three values of δ considered within the interval$t = [0, 5)$, before the step down occurs. In this case, it is obvious that the control signal is growing to allow the system to reach the reference, $r(t)$. Therefore, it is not affected by the constraint imposed by (32). Nevertheless, after the step down, within interval $t = [5, 10)$, it is necessary for the control signal to decrease so that the system output also decreases and reaches the new reference. In this case, the output and the control signal are affected by the activation of the constraint (32) and have the shape that is shown in the upper plots of Figure 8, with the values of both signals, y and u, for $\delta = 0.5$ being below the values of both for $\delta = 1.0$. Analogously, the values of both signals for $\delta = 1.5$ are above the values of both for $\delta = 1.0$. Therefore, we have different shapes of the system output depending on the parameter δ.

Figure 9 depicts the simulation for $\delta = 0.01$ and $N_u = 3$. It can be easily seen that the shape of $\Delta^\delta u(t)$ (magenta line) tends to the shape of $u(t)$ (green line) as expected.

In the same way, for illustrative purposes, we set a constraint on the maximum value of the fractional δ–increment as it is shown in expression (34).

$$\Delta^\delta u(t + N_u - 1) \leq 0.1 \tag{34}$$

Figure 10 depicts the simulation for $\delta = 0.8$ and the two values of the control horizon, $N_u = 1$ and $N_u = 3$. In this case, the constraint acts within the interval $t = [0, 5)$ affecting the temporal response of both. However, the one tuned with $N_u = 1$ (blue line) suffers the effect of the hard constraint, appearing a small ripple in $\Delta^\delta u(t)$ as it is shown in lower

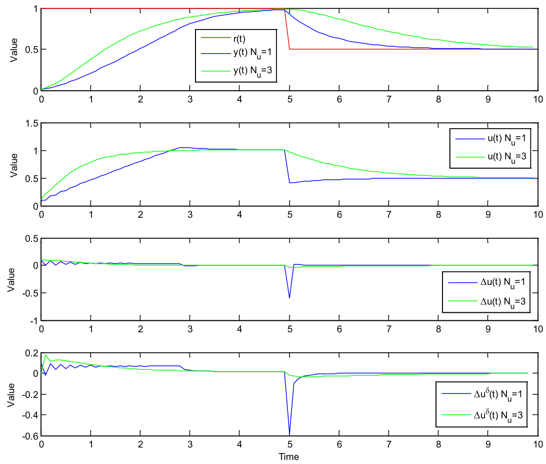

FIGURE 9. Simulation 3 of the constrained controller 1.

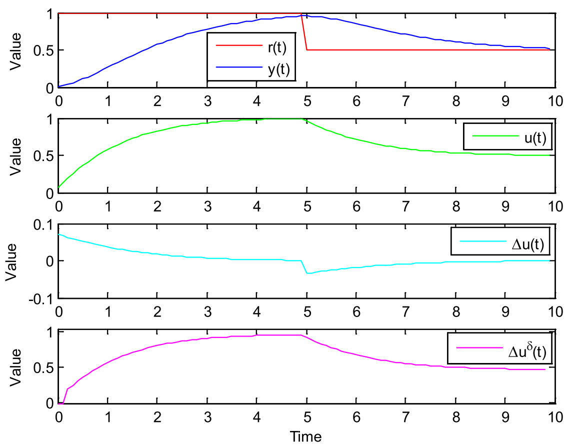

FIGURE 10. Simulation 4 of the constrained controller 1.

plot. The upper plot shows as the system output grows more slowly with $N_u = 1$ than with $N_u = 3$. With $N_u = 3$, the constraint presents a soft behavior due to the constraint concerns a predicted value.

Finally, we shall test the behavior of the proposed method with a model of a real plant. Specifically, it is the identified model of the throttle pedal of a gasoline car used in [38], whose transfer function is shown in expression (35). One observes that the dynamics of this model is more complex than the previous used model (22).

$$G\left(z^{-1}\right) = \frac{5.1850}{1 - 0.7344z^{-1} - 0.2075z^{-2}} z^{-4} \qquad (35)$$

For illustration purposes, a new FGPC controller for (35) will be tuned using the following parameters: $N_1 = 1$, $N_2 = 50$, $N_u = 2$, using prefilter $T_c\left(z^{-1}\right) = 1 - 0.9z^{-1}$ and $\alpha = 0.22$, $\beta = 2.30$. This control system, without activating the constraints, presents a highly oscillating temporal response as it is shown in the upper plots of Figure 11 (blue line). In this case, the oscillations of $\Delta u(t)$ are approximately within the interval $[-0.1, 0.1]$. Obviously, in order to illustrate our

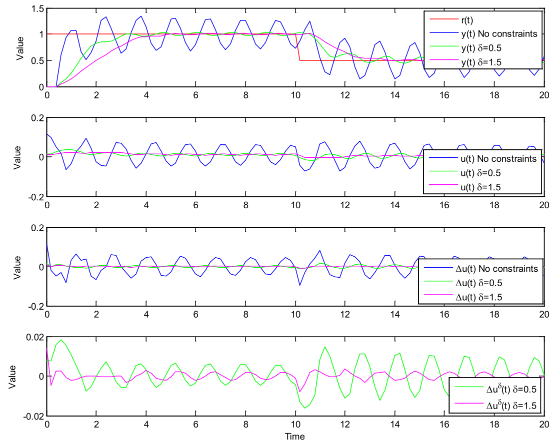

FIGURE 11. Simulation 1 of the constrained controller 2.

example, we have proposed this extreme case, where the controller has not been properly tuned.

In the following, we shall limit that oscillation using the interval $[-0.01, 0.01]$, setting the minimum and maximum values of fractional constraints as it is shown in expression (36).

$$\begin{aligned} \Delta^{\delta} u\left(t+N_u-1\right) &\geq -0.01 \\ \Delta^{\delta} u\left(t+N_u-1\right) &\leq 0.01 \end{aligned} \qquad (36)$$

Figure 11 depicts the system simulations subject to previous constraints (36) for two values of the δ parameter, $\delta = 0.5$ (green line) and $\delta = 1.5$ (magenta line). One observes that the oscillations both in the system output, $y(t)$, and in the value of the control signal, $u(t)$, have been drastically reduced after setting the constraints.

Since the oscillation has been reduced to a small ripple that is larger in the case of $\delta = 0.5$, as it is shown in the lower plot of Figure 11, the system presents a more suitable output, improving its performance with stable, smooth and reasonably good response in comparison with the system response without constraints. In this case, the control signal, $u(t)$, corresponds to the throttle pedal of the gasoline car. Therefore, it is crucial that the oscillation is as small as possible, both for a better comfort of the vehicle's occupants and to keep the engine running and avoid the car stalled.

Moreover, in the same way as with the previous controller (Figure 9), the larger the δ parameter value is, the slower the system response will be. Thus, the system output, $y(t)$, will take longer to reach the reference. Nevertheless, larger values of the δ delta parameter also reduce the amplitude of the oscillations.

V. CONCLUSION

In this paper, the use of fractional constraints in the predictive control strategy has been introduced and explained, discussing the results obtained by means of the fractional δ–increments depending on the δ parameter. Its use has also provided, on the one hand, a new method to soften constraints using fractional operators and, on the other hand, obtaining different shapes of the system output depending on the parameter δ.

Moreover, a practical tuning of a plant with FGPC using Matlab™ Optimization Toolbox FGPC is included in order to make use of the beneficial characteristics of this control strategy. This example has been carried out to illustrate the validity of this method using two solvers, one based on quadratic optimization and the other based on genetic algorithms. Using them and with the help of this well–known mathematical software, the controller settings can be achieved that lead to optimal system performance with ease.

In this way, the design process of the FGPC involves finally two basic steps. In the first step, we obtain the controller parameter N_1, N_2, N_u, α and β to fulfill an appropriated performance using the tuning method based on optimization of some criteria. In the second step, the constraints are set to meet the operating specifications and limitations of the real system using the parameter δ, moreover, for softening constraints as far as possible, even to adjust the final response of the system.

APPENDIX
TUNING FGPC CONTROLLER

Tuning the predictive controller means to set numerical values for the controller parameters to fulfill a series of objectives in terms of robustness and performance criteria. Obviously, the final performance of the loop system will depend on the goodness of this adjustment. To do so, various algorithms could be proposed based on techniques such as optimization, genetic algorithms, expert knowledge, etc.

In our case, the FGPC controller presents a wide set of parameters for tuning: N_1, N_2, N_u, α and β. The proposed method [47] is based on optimization of some criteria to fulfill an appropriated performance:

- Gain margin (GM):

$$M = -20\,log\left|\frac{BS_c}{R_c\Delta A}\right|, \quad \text{where } arg\left(\frac{BS_c}{R_c\Delta A}\right) = -180^\circ \tag{37}$$

- Phase margin (PM):

$$arg\left(\frac{BS_c}{R_c\Delta A}\right) = -(-180^\circ) + PM, \quad \text{where } -20\,log\left|\frac{BS_c}{R_c\Delta A}\right| = 0 \tag{38}$$

- High frequency noise rejection:

$$|T(j\omega)| \le A_t dB \quad \text{for } \omega \ge \omega_t rad/s \tag{39}$$

where T is the complementary sensitivity function and A_t is the desired noise attenuation for frequencies $\omega \ge \omega_t$.

- Good output disturbance:

$$|S(j\omega)| \le B_s dB \quad \text{for } \omega \le \omega_s rad/s \tag{40}$$

where B_s is the desired value of the sensitivity function S for frequencies $\omega \le \omega_s$.

- Robustness to variations in the gain of the plant:

$$\left[\frac{d\left(arg\left(\frac{BS_c}{R_c\Delta A}\right)\right)}{d\omega}\right]_{\omega=\omega_c} = 0 \tag{41}$$

with this condition, the phase of the open–loop system is forced to be flat at ω_C (crossover frequency) and to be almost constant within an interval around this value.

This method proposes that one of the previous expressions (37)–(41) is chosen as the main function to optimize and the rest of them could be taken as constraints. In order to keep the dimension of the optimization problem low, it is assumed that the parameters N_1, N_2, N_u, whose values are integer, are given using, for example, the thumb–rules proposed in [28] and the fractional orders α and β, whose values are real, are used in the optimization process as the parameters to be calculated. Including the horizons parameters in the optimization problem would lead to a mixed quadratic programming problem that would unnecessarily complicate the calculation process for the controller tuning.

REFERENCES

[1] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control—Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978, doi: 10.1016/0005-1098(78)90001-8.

[2] C. R. Cutler and B. L. Remaker, "Dynamic matrix control—A computer control algorithm," in *Proc. AIChE 86th Nat. Meeting*, Houstong, TX, USA, 1979, pp. 585–590.

[3] T. Alvarez, M. Sanzo, and C. De Prada, "Identification and constrained multivariable predictive control of chemical reactors," in *Proc. Int. Conf. Control Appl.*, Albany, NY, USA, 1995, pp. 663–664, doi: 10.1109/CCA.1995.555815.

[4] E. F. Camacho and M. Berenguel, "Application of generalized predictive control to a solar power plant," in *Proc. IEEE Int. Conf. Control Appl. (CCA)*, Glasgow, U.K., 1994, pp. 1657–1662, doi: 10.1109/CCA.1994.381468.

[5] D. A. Linkens and M. Mahfouf, "Generalized predictive control (GPC) in clinical anesthesia," in *Advances in Model-Based Predictive Control*, D. W. Clarke, Ed. Oxford, U.K.: Oxford Univ. Press, 1994, pp. 429–445.

[6] J. M. Maciejowski, *Predictive Control With Constraints*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[7] J. A. Rossiter, *Model Based Predictive Control: A Practical Approach*. Boca Raton, FL, USA, 2003, doi: 10.1201/9781315272610.

[8] E. F. Camacho and C. Bordóns, *Model Predictive Control*, 2nd ed. Cham, Switzerland: Springer, 2004.

[9] V. Milanes, J. Villagra, J. Perez, and C. Gonzalez, "Low-speed longitudinal controllers for mass-produced cars: A comparative study," *IEEE Trans. Ind. Electron.*, vol. 59, no. 1, pp. 620–628, Jan. 2012, doi: 10.1109/TIE.2011.2148673.

[10] D. Craiem and R. L. Armentano, "A fractional derivative model to describe arterial viscoelasticity," *Biorheology*, vol. 44, no. 4, pp. 251–263, Jan. 2007.

[11] M. Romero, A. P. De Madrid, C. Mañoso, and V. Milanés, "Low speed hybrid generalized predictive control of a gasoline-propelled car," *ISA Trans.*, vol. 57, pp. 373–381, Jul. 2015, doi: 10.1016/j.isatra.2015.01.004.

[12] S. Domek, "Mixed logical dynamical modeling of discrete-time hybrid fractional systems," in *Fractional Dynamical Systems: Methods, Algorithms and Applications* (Studies in Systems, Decision and Control), vol. 402, P. Kulczycki, J. Korbicz, and J. Kacprzyk Eds. Berlin, Germany: Springer, 2022, pp. 77–105, doi: 10.1007/978-3-030-89972-1_3.

[13] I. Podlubny, "Fractional-order systems and $PI^{\lambda}D^{\mu}$ controllers," *IEEE Trans. Autom. Controlol.*, vol. 44, no. 1, pp. 208–214, Jan. 1999, doi: 10.1109/9.739144.

[14] I. Petras, "The fractional—Order controllers: Methods for their synthesis and application," *J. Elec. Eng.*, vol. 50, pp. 284–288, Apr. 1999.

[15] F. Meng, S. Liu, and K. Liu, "Design of an optimal fractional order PID for constant tension control system," *IEEE Access*, vol. 8, pp. 58933–58939, 2020, doi: 10.1109/ACCESS.2020.2983059.

[16] A. Oustaloup, B. Mathieu, and P. Lanusse, "The CRONE control of resonant plants: Application to a flexible transmission," *Eur. J. Control*, vol. 1, no. 2, pp. 113–121, 1995, doi: 10.1016/S0947-3580(95)70014-0.

[17] Q. Zou, Q. Jin, and R. Zhang, "Design of fractional order predictive functional control for fractional industrial processes," *Chemometric Intell. Lab. Syst.*, vol. 152, pp. 34–41, Mar. 2016, doi: 10.1016/j.chemolab.2016.01.011.

[18] S. Chatterjee, O. Romero, A. Ashourvan, and S. D. G. M. Pequito, "Fractional-order model predictive control as a framework for electrical neurostimulation in epilepsy," *J. Neural Eng.*, vol. 17, Nov. 2020, Art. no. 066017, doi: 10.1088/1741-2552/abc740.

[19] R. Zhang, Q. Zou, Z. Cao, and F. Gao, "Design of fractional order modeling based extended non-minimal state space MPC for temperature in an industrial electric heating furnace," *J. Process Control*, vol. 56, pp. 13–22, Aug. 2017, doi: 10.1016/j.jprocont.2017.05.003.

[20] N. Bigdeli, "The design of a non-minimal state space fractional-order predictive functional controller for fractional systems of arbitrary order," *J. Process Control*, vol. 29, pp. 45–56, May 2015, doi: 10.1016/j.jprocont.2015.03.004.

[21] M. R. Hortelano, "Applications of GPU computing to control and simulate systems," in *Proc. Actas De Las 39th Jornadas De Automática, Badajoz, De Septiembre De*, Badajoz, Spain, Mar. 2020, pp. 792–798, doi: 10.17979/spudc.9788497497565.0792.

[22] K. B. Oldham and J. Spanier, "The fractional calculus," in *Mathematics in Science and Engineering*, vol. 3. New York, NY, USA: Academic Press, 1974.

[23] D. S. Miller and B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations*. New York, NY, USA: Wiley, 1993.

[24] M. Ortigueira and J. Machado, "Which derivative?" *Fractal Fract.*, vol. 1, p. 3, Jul. 2017, doi: 10.3390/fractalfract1010003.

[25] R. Liko, P. O. Mohammed, A. Kashuri, and Y. S. Hamed, "Reverse Minkowski inequalities pertaining to new weighted generalized fractional integral operators," *Fractal Fractional*, vol. 6, no. 3, p. 131, Feb. 2022, doi: 10.3390/fractalfract6030131.

[26] I. Podlubny, "Fractional differential equations," in *Mathematics in Science and Engineering*, vol. 198. San Diego, CA< USA: Academic Press, 1999.

[27] M. Romero, A. P. De Madrid, C. Manoso, and B. M. Vinagre, "A survey of fractional-order generalized predictive control," in *Proc. 51st IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 6867–6872, doi: 10.1109/CDC.2012.6426244.

[28] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control—Part I. The basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987, doi: 10.1016/0005-1098(87)90087-2.

[29] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control—Part II. Extensions and interpretations," *Automatica*, vol. 23, no. 2, pp. 149–160, 1987, doi: 10.1016/0005-1098(87)90088-4.

[30] M. Romero, C. Manoso, A. P. De Madrid, and B. M. Vinagre, "Fractional-order generalized predictive control: Formulation and some properties," in *Proc. 11th Int. Conf. Control Autom. Robot. Vis.*, Dec. 2010, pp. 1495–1500, doi: 10.1109/ICARCV.2010.5707408.

[31] S. Zhao, R. Cajo, R. D. De Keyser, and C.-M. Ionescu, "The potential of fractional order distributed MPC applied to steam/water loop in large scale ships," *Processes*, vol. 8, no. 4, p. 451, Apr. 2020, doi: 10.3390/pr8040451.

[32] M.-R. Chen, G.-Q. Zeng, Y.-X. Dai, K.-D. Lu, and D.-Q. Bi, "Fractional-order model predictive frequency control of an islanded microgrid," *Energies*, vol. 12, no. 1, p. 84, Dec. 2018, doi: 10.3390/en12010084.

[33] X. Hu, Q. Zou, and H. Zou, "Design and application of fractional order predictive functional control for industrial heating furnace," *IEEE Access*, vol. 6, pp. 66565–66575, 2018, doi: 10.1109/ACCESS.2018.2878554.

[34] C. V. Rao and J. B. Rawling, "Linear programming and model predictive control," *J. Process Control*, vol. 10, nos. 2–3, pp. 283–289, 2000, doi: 10.1016/S0959-1524(99)00034-7.

[35] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989, doi: 10.1016/0005-1098(89)90002-2.

[36] J. Richalet, *Pratique de la Commande Préditive in Traité des Nouvelles Technologies*. (Série Automatique). Paris, France: Hermès Science Publications, 1993.

[37] M. Romero, A. P. De Madrid, and B. M. Vinagre, "Arbitrary real-order cost functions for signals and systems," *Signal Process.*, vol. 91, no. 3, pp. 372–378, Mar. 2011, doi: 10.1016/j.sigpro.2010.03.018.

[38] M. Romero, A. P. De Madrid, C. Mañoso, V. Milanés, and B. M. Vinagre, "Fractional-order generalized predictive control: Application for low-speed control of gasoline-propelled cars," *Math. Problems Eng.*, vol. 2013, pp. 1–10, Jan. 2013, doi: 10.1155/2013/895640.

[39] T.-W. Yoon and D. W. Clarke, "Observer design in receding-horizon predictive control," *Int. J. Control*, vol. 61, no. 1, pp. 171–191, Jan. 1995, doi: 10.1080/00207179508921897.

[40] A. G. Kuznetsov and D. W. Clarke, "The performance of generalized predictive control with interval constraints," *Eur. J. Control*, vol. 2, no. 4, pp. 260–277, 1996, doi: 10.1016/S0947-3580(96)70051-1.

[41] R. Caponetto, F. Matera, E. Murgano, E. Privitera, and M. G. Xibilia, "Fuel cell fractional-order model via electrochemical impedance spectroscopy," *Fractal Fractional*, vol. 5, no. 1, p. 21, Mar. 2021, doi: 10.3390/fractalfract5010021.

[42] T. J. Freeborn, B. Maundy, and A. S. Elwakil, "Measurement of supercapacitor fractional-order model parameters from voltage-excited step response," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 3, no. 3, pp. 367–376, Sep. 2013, doi: 10.1109/JETCAS.2013.2271433.

[43] X.-J. Yang, F. Gao, and Y. Ju, *General Fractional Derivatives With Applications in Viscoelasticity*. Cambridge, MA, USA, Academic Press, 2020.

[44] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Syst. Mag.*, vol. 20, no. 3, pp. 38–52, Jun. 2000.

[45] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, "Review on model predictive control: An engineering perspective," *Int. J. Adv. Manuf. Technol.*, vol. 117, nos. 5–6, pp. 1327–1349, Nov. 2021, doi: 10.1007/s00170-021-07682-3.

[46] *Quadratic Programming: Minimize Quadratic Functions Subject to Constraints*. MATLAB. Accessed: Nov. 10, 2022. [Online]. Available: https://mathworks.com/help/optim/ug/quadprog.html

[47] M. Romero, I. Tejado, A. P. De Madrid, and B. M. Vinagre, "Tuning predictive controllers with optimization: Application to GPC and FGPC," in *Proc. 18th IFAC World Congr.*, 2011, pp. 6867–6872, doi: 10.3182/20110828-6-IT-1002.02060.

MIGUEL ROMERO (Senior Member, IEEE) received the M.Sc. degree in industrial engineering, specialized in automatic control and electronics, from the University of Córdoba University, Spain, in 2003, and the Ph.D. degree in automatic control from the Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain, in 2010.

He is currently an Associate Professor with the Control and Communication Systems Department, UNED. He has developed his research activity working in several research projects, financed in competitive calls. He has also published more than 30 research papers in journals, books, and national and international conferences. His research interests include fractional control, the IoT, robotics, and GPU computing to control and simulate systems. His research has also focused on fractional calculus and predictive control with applications to automotive sector.

CAROLINA MAÑOSO (Senior Member, IEEE) received the M.Sc. degree in physics, specialized in electronics and automatic control, from the University of the Basque Country, Spain, in 1990, and the Ph.D. degree in sciences from the Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain, in 1998.

She is currently an Associate Professor with the Control and Communication Systems Department, UNED. She has developed her research activity working in several research projects, financed in competitive calls. She has also published more than 40 research papers in journals, books, and national and international conferences. Her research interests include robotics, human–computer interfaces, and the study of the implementation and acceptance of new technologies in education. Her research has also focused on constrained optimal control, especially predictive control, from different points of view: robust, fractional–order, and multiparameter.