Control of a grinding mill circuit using fractional order controllers

Norelys Aguila-Camacho[a,b,*], Johan D. Le Roux[c], Manuel A. Duarte-Mermoud[a,b], , Marcos E. Orchard[a,b]

[a] *Department of Electrical Engineering, Faculty of Mathematical and Physical Sciences, University of Chile, Av. Tupper 2007, Santiago de Chile, Chile.*

[b] *Advanced Mining Technology Center, Av. Tupper 2007, Santiago de Chile, Chile.*

[c] *Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa.*

Abstract

This paper presents the design and application of fractional single-input-single-output (SISO) controllers to a grinding mill circuit, which is a multiple-input-multiple-output (MIMO) process. Two kinds of controllers are presented: fractional order proportional-integral (FOPI) controllers, and a combination of FOPI and fractional order model reference adaptive controllers (FOMRAC). The parameters of the controller are tuned using off-line particle swarm optimization. In the presence of disturbances and process noise, the SISO fractional controllers achieve similar or better performance compared to Linear Model Predictive Control (LMPC).

Keywords: Fractional Order Controllers, Model Reference Adaptive Control, Grinding Mill Circuit, Particle Swarm Optimization, Process control, Comminution.

1. Introduction

Grinding mill circuits are difficult nonlinear processes to control because of the coupling and interaction between process variables. In general, multivariate control techniques are used to solve this issue, such as Model Predictive Control (MPC), [1, 2, 3, 4, 5, 6], inverse Nyquist array [7], extended horizon [8, 9], pole placement [8, 9], multivariate model reference adaptive control [8, 9], direct Nyquist array [8, 9], sequential loop closing [8, 9], and predictive multivariate neural control [10]. These multivariate techniques can significantly improve process performance compared to decentralized SISO controllers [4, 11]. However, because of the ease of implementation of SISO techniques in terms of tuning and maintenance, SISO control techniques remain most prevalent in industrial grinding mill circuits [12].

A great percentage of industries still use proportional integral and derivative (PID) controllers (or only PIs) in their milling circuits, usually because of the difficulty of implementing and maintaining advanced process control, and also because there is a lack of sufficient dynamic and fundamental models for mineral processing circuits [13]. On the other hand, PID controllers are simple, relatively easy to tune and can handle many operating conditions relatively well. However, PID controllers usually cannot handle parameter variations, disturbances and noise in a robust manner, compared to multivariable techniques. Thus, obtaining improvements in the behavior of milling circuits under these operating conditions, using SISO controllers, could be a great intermediate step for industries where the implementation of advanced multivariable control strategies is difficult.

The control of grinding processes is further complicated by the presence of unmodeled process dynamics, time-varying parameters, large time delays, and noisy measurements. These issues, which are impediments

[*] Corresponding author. Tel.: +56 2 29784920; fax: +56 2 26720162
E-mail addresses: naguila@ing.uchile.cl (N. Aguila-Camacho), derik.leroux@up.ac.za (J.D. Le Roux), mduartem@ing.uchile.cl (M.A. Duarte-Mermoud), morchard@ing.uchile.cl (M.E. Orchard)

for any control technique, are managed through adaptations in the control techniques [8, 9, 14, 15, 16, 17, 18, 19] to improve the grinding circuit's performance.

It has been reported in some works that the use of fractional operators [20] as part of control strategies can improve system robustness in the presence of disturbances, noisy environments, and time-varying parameters. It has been also reported in Aguila-Camacho and Duarte-Mermoud [21] that their use can improve the management of the control energy. Why these advantages are achievable is a research topic currently under investigation, but it seems that the implicit memory incorporated in the definitions of fractional integral and fractional derivative [20] is the key to answer this question. Several control techniques have been generalized with the use of fractional operators: Fractional Order Proportional Integral control (FOPI) [22], Fractional Order Model Reference Adaptive Control (FOMRAC) [23], and adaptive gain-order fractional order control [24]. Although the application of these fractional control techniques is found mainly in SISO processes [21, 25], some were proposed for MIMO processes [26, 27, 28, 29].

This work investigates the design and application of fractional order SISO controllers to a grinding mill circuit, which is a MIMO process. Thus, the control techniques preserve their simplicity in terms of SISO control, but are also capable of improving the robustness by means of the fractional operators in the presence of disturbances, parameter variations and noisy measurements. As far as the authors are aware, the use of fractional order controllers for grinding mill circuits has not been reported in literature, with the exception of the use of fractional disturbance observers [17] together with classic PI controllers.

Two fractional order SISO control strategies are proposed in this paper for a single-stage grinding mill circuit. The first control strategy, FOPI control, is a non-adaptive control strategy, and the second, FOPI control combined with FOMRAC, is an adaptive control strategy.

The paper is organized as follows: Section 2 describes the grinding mill circuit and the corresponding nonlinear model used for simulation; Section 3 introduces the design of the proposed SISO fractional controllers and Section 4 the tuning of the SISO fractional controller parameters; Section 5 introduces an LMPC, which is used to compare the behavior of the SISO fractional controllers; Section 6 presents the results obtained from simulations; Section 7 presents the conclusions.

2. Grinding mill circuit

2.1. Process Description

A single-stage closed run-of-mine (ROM) ore milling circuit, as shown in Fig. 1, is considered in this study. The circuit consists of a semiautogenous (SAG) mill with an end-discharge grate, a sump and a hydrocyclone. The mill receives four streams as inputs: mined ore (MFO), water (MIW) to assist with material transport, steel balls (MFB) to assist with ore breakage, and underflow from the hydrocyclone.

The fraction of the mill filled with charge is denoted by $LOAD$. The ground ore in the mill mixes with water to form a slurry. The slurry is discharged from the mill into the sump through an end-discharge grate. The end-discharge grate limits the particle size of the discharged slurry. The slurry in the sump is diluted with water (SFW) and is pumped to the hydrocyclone for classification. The total volume of slurry in the sump is denoted by $SVOL$. It is assumed the pump is fitted with a variable speed motor to manipulate the cyclone feed flow-rate (CFF). The cyclone feed density can be adjusted by the sump dilution water as long as the sump does not overflow or run dry.

The hydrocyclone is responsible for the separation of the in-specification and out-of-specification ore discharged from the sump. The lighter, smaller and in-specification particles in the slurry pass to the overflow of the hydrocyclone, while the heavier, larger and out-of-specification particles pass to the underflow. The underflow is passed to the mill for further grinding while the overflow flows to a downstream process. The volumetric flow-rate of solids in the overflow is the throughput of the circuit and is equal to the volumetric feed rate of ore at steady-state operation of the circuit. The quality of the circuit product is indicated by the fraction of particles in the overflow smaller than specification size (PSE). The controlled and manipulated variables mentioned in this section are shown in Table 1.

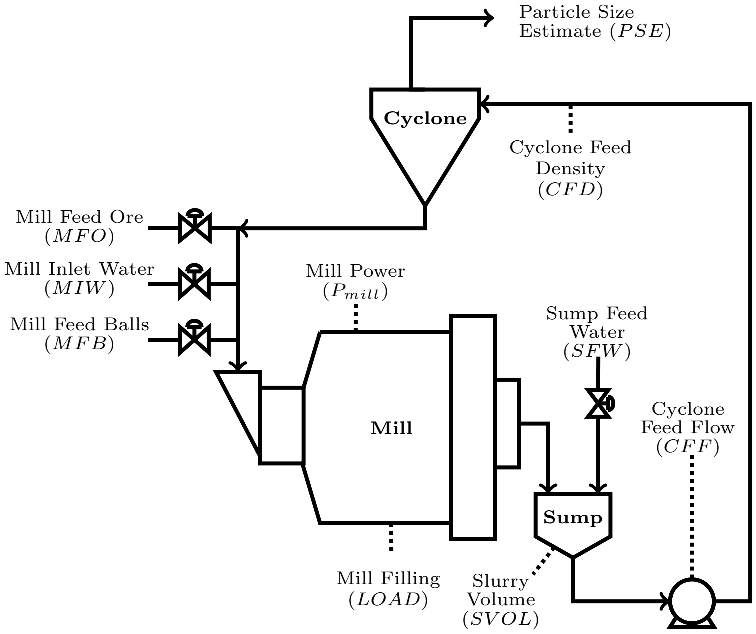

Figure 1: Single-stage closed run-of-mine ore milling circuit.

2.2. Model description

The continuous time dynamic phenomenological nonlinear population balance model validated by Le Roux et al. [30] is used in this study to describe the circuit shown in Fig. 1. Each process unit in the circuit is modelled separately. The model is suitable for control purposes as it uses as few parameters and states as

Table 1: Manipulated, Controlled and State Variables

Variable	Value	Unit
Manipulated Variables		
CFF	374	$[\mathrm{m}^3/\mathrm{h}]$
MFO	65.2	[t/h]
SFW	140.5	$[\mathrm{m}^3/\mathrm{h}]$
MIW	4.64	$[\mathrm{m}^3/\mathrm{h}]$
MFB	5.68	[t/h]
Controlled Variables		
PSE	67	$[\% < 75\mu\mathrm{m}]$
$LOAD$	33	[%]
$SVOL$	11.8	$[\mathrm{m}^3]$
States		
X_{mw}	4.63	$[\mathrm{m}^3]$
X_{ms}	4.65	$[\mathrm{m}^3]$
X_{mf}	0.96	$[\mathrm{m}^3]$
X_{mr}	1.99	$[\mathrm{m}^3]$
X_{mb}	8.23	$[\mathrm{m}^3]$
X_{sw}	8.10	$[\mathrm{m}^3]$
X_{ss}	3.70	$[\mathrm{m}^3]$
X_{sf}	0.76	$[\mathrm{m}^3]$

Table 2: Description of subscripts.

Subscript	Description
$X_{\square -}$	m-mill; s-sump; c-cyclone
$X_{-\square}$	w-water; s-solids; c-coarse; f-fines; r-rocks; b-balls
$V_{--\square}$	i-inflow; o-outflow; u-underflow

possible to produce reasonably accurate model responses.

The model divides the ore into three size classes: rocks, coarse ore and fine ore. Rocks are classified as ore too large to pass through the mill discharge grate. Coarse ore can pass through the mill discharge grate but is larger than the specification size. Fine ore also passes through the mill discharge grate but is within specification size. The sum of coarse and fine ore is defined as solids. Although only three size classes are used to describe the ore in the circuit, they are sufficient for the model to produce qualitatively accurate responses [31].

The model defines fives states to describe the mill charge volumetric hold-ups: water (X_{mw}), solids (X_{ms}), fines (X_{mf}), rocks (X_{mr}), and steel balls (X_{mb}). Because of the mill discharge grate, only three states are necessary to describe the sump slurry volumetric hold-ups: water (X_{sw}), solids (X_{ss}), and fines (X_{sf}).

For the population balance model equations, V denotes a flow-rate in m^3/h and X denotes the states of the model as volumes in m^3. Table 2 provides a description of the subscripts for V and X. The first subscript indicates the process unit considered (mill, sump or cyclone), the second subscript specifies which state is considered (rocks, solids, coarse, fines, balls, or water), and in the case of flow-rates the final subscript indicates an inflow, outflow or underflow. Only a brief overview of the model is presented here. A detailed description of the model is provided in Le Roux et al. [30]. The model nomenclature can be seen in Table 3.

2.2.1. Mill model:

The population volume balance of mill hold-ups - water (X_{mw}), solids (X_{ms}), fines (X_{mf}), rocks (X_{mr}), and steel balls (X_{mb}) - are defined in terms of the inflow, outflow and generation/consumption of each state:

$$\dot{X}_{mw} = V_{mwi} + V_{cwu} - V_{mwo} \tag{1a}$$

$$\dot{X}_{ms} = V_{msi} + V_{csu} - V_{mso} + RC \tag{1b}$$

$$\dot{X}_{mf} = V_{mfi} + V_{cfu} - V_{mfo} + FP \tag{1c}$$

$$\dot{X}_{mr} = V_{mri} - RC \tag{1d}$$

$$\dot{X}_{mb} = V_{mbi} - BC, \tag{1e}$$

where V_{mwi}, V_{msi}, V_{mfi}, V_{mri} and V_{mbi} (m^3/h) are the flow-rates of water, solids, fines, rocks and balls into the mill respectively, V_{mwo}, V_{mso} and V_{mfo} (m^3/h) are the discharge flow-rates of water, solids and fines respectively, RC, BC and FP (m^3/h) are the rock consumption, ball consumption and fines production respectively, and V_{cwu}, V_{csu} and V_{cfu} (m^3/h) are the cyclone water, solids and fines underflow flow-rates respectively.

The flow of material into the mill is defined as:

$$V_{mwi} = MIW \tag{2a}$$

$$V_{msi} = (1 - \alpha_r)MFO/\rho_S \tag{2b}$$

$$V_{mfi} = \alpha_f MFO/\rho_S \tag{2c}$$

$$V_{mri} = \alpha_r MFO/\rho_S \tag{2d}$$

$$V_{mbi} = MFB/\rho_B, \tag{2e}$$

where ρ_S (t/m^3) is the ore density, ρ_B (t/m^3) is the ball density, and parameters α_f and α_r represent the fraction of fines and rocks in MFO respectively. Although MFO in an industrial plant can be controlled

Table 3: Circuit parameter values. (Dimensionless parameters are shown without units.)

Parameter	Value	Description
Mill parameters		
α_f	0.055	Mass fraction of fines in the feed ore
α_r	0.465	Mass fraction of rocks in the feed ore
α_P	1	Fractional power reduction per fractional reduction of critical mill speed
α_{speed}	0.72	Fraction of critical mill speed
α_{φ_f}	0.01	Fractional change in kW/fines produced per change in fractional filling of mill
χ_P	0	Cross-term for maximum power draw
δ_{Ps}	17.46	Power-change parameter for fraction solids in the mill
δ_{Pv}	17.46	Power-change parameter for volume of mill filled
d_0	88.0	Discharge rate [h^{-1}]
ε_{sv}	0.6	Max fraction of solids by volume slurry at zero slurry flow
$\varphi_{P_{max}}$	0.57	Rheology factor for maximum mill power draw
ϕ_b	90	Steel abrasion factor [kWh/t]
ϕ_f	29.5	Power needed per ton of fines produced [kWh/t]
ϕ_r	6.72	Rock abrasion factor [kWh/t]
P_{max}	1670	Maximum mill power draw [kW]
ρ_S	3.2	Density of ore [t/m^3]
ρ_B	7.85	Density of balls [t/m^3]
ρ_W	1	Density of water [t/m^3]
$v_{P_{max}}$	0.34	Fraction of mill volume filled for maximum power draw
v_{mill}	59.12	Mill volume [m^3]
Hydrocyclone parameters		
α_{su}	0.9154	Parameter related to fraction solids in underflow
ε_c	126.93	Parameter related to coarse split [m^3/h]
C_1	0.6	Constant
C_2	0.7	Constant
C_3	4	Constant
C_4	4	Constant
C_5	0.6	Constant

fairly well, variations in feed size distribution (represented by α_r and α_f) upset the equilibrium in a mill considerably and impede maintaining a mill in the optimum operating region.

The mill discharge flow-rates are defined as:

$$V_{mwo} = \varphi d_0 X_{mw} \left(\frac{X_{mw}}{X_{ms} + X_{mw}} \right) \tag{3a}$$

$$V_{mso} = \varphi d_0 X_{mw} \left(\frac{X_{ms}}{X_{ms} + X_{mw}} \right) \tag{3b}$$

$$V_{mfo} = \varphi d_0 X_{mw} \left(\frac{X_{mf}}{X_{ms} + X_{mw}} \right), \tag{3c}$$

where d_0 (1/h) is the discharge rate, and φ is an empirical function called the rheology factor. The rheology factor attempts to incorporate the effect of the fluidity and density of the slurry on the milling circuit's performance and is defined as:

$$\varphi = \left[\max \left(0, 1 - \left(\frac{1}{\varepsilon_{sv}} - 1 \right) \frac{X_{ms}}{X_{mw}} \right) \right]^{0.5}, \tag{4}$$

where ε_{sv} is the maximum fraction of solids by volume of slurry at zero slurry flow. A rheology factor of unity corresponds to $\frac{X_{ms}}{X_{mw}} = 0$ indicating the slurry consists only of water. A rheology factor of zero corresponds to $\frac{X_{ms}}{X_{mw}} = \frac{\varepsilon_{sv}}{1-\varepsilon_{sv}}$ indicating the slurry is a non-flowing mud.

The general formulation of the breakage equations has its parallel in the cumulative breakage rates expressions in Hinde and Kalala [32] and Amestica et al. [33]. The rock consumption (RC), ball consumption (BC) and fines production (FP) are defined as:

$$RC = \frac{\varphi P_{mill} X_{mr}}{\rho_S \phi_r \left(X_{mr} + X_{ms}\right)} \tag{5}$$

$$BC = \frac{\varphi P_{mill} X_{mb}}{\phi_b \left[\rho_S \left(X_{mr} + X_{ms}\right) + \rho_B X_{mb}\right]} \tag{6}$$

$$FP = \frac{P_{mill}}{\rho_S \phi_f \left[1 + \alpha_{\phi_f} \left(LOAD - v_{P_{max}}\right)\right]}, \tag{7}$$

where ϕ_r and ϕ_b (kWh/t) are the abrasion rates of rocks and balls respectively, ϕ_f (kWh/t) is the energy required per tonne of fines produced, $v_{P_{max}}$ is the fraction of the mill filled at maximum power draw, α_{ϕ_f} accounts for the change in ϕ_f per change in mill filling, and P_{mill} is the mill power draw. The equilibrium of the mill is not only upset through variations of the feed size distribution, but also variations in ore hardness. This can be simulated through variation of parameters ϕ_f and ϕ_r.

The fraction of the mill filled with charge ($LOAD$) is defined as:

$$LOAD = \left(X_{mw} + X_{ms} + X_{mr} + X_{mb}\right)/v_{mill}, \tag{8}$$

where v_{mill} (m^3) is the total volume of the mill.

The mill power draw is modelled as a quadratic function depending on the total mill charge and the fluidity and density of the slurry in the mill:

$$P_{mill} = P_{max} \left(\alpha_{speed}\right)^{\alpha_P} \left(1 - \delta_{Pv} Z_x^2 - 2\chi_P \delta_{Pv} \delta_{Ps} Z_x Z_r - \delta_{Ps} Z_r^2\right), \tag{9}$$

where P_{max} (kW) is the maximum mill power draw, α_{speed} is the fraction of critical mill speed, α_P is the fractional power reduction per fractional reduction from critical mill speed, δ_{Pv} is the power change parameter for volume of mill filled, δ_{Ps} is the power change parameter for the fraction of solids in the mill, and χ_P is the cross term for maximum power draw. The parabolic shape of mill power draw as a function of mill load is further discussed in Powell et al. [34].

The effect of the total charge on mill power is modelled by the empirical definition of $Z_x = \frac{LOAD}{v_{P_{max}}} - 1$, and the effect of the solids content on the mill power is modelled by the empirical definition of $Z_r = \frac{\varphi}{\varphi_{P_{max}}} - 1$ where $\varphi_{P_{max}}$ is the rheology factor at maximum mill power draw.

2.2.2. Sump model:

The population volume balance of sump hold-ups - water (X_{sw}), solids (X_{ss}), and fines (X_{sf}) - are defined as:

$$\dot{X}_{sw} = V_{mwo} - V_{swo} + SFW \tag{10a}$$

$$\dot{X}_{ss} = V_{mso} - V_{sso} \tag{10b}$$

$$\dot{X}_{sf} = V_{mfo} - V_{sfo}, \tag{10c}$$

where V_{swo}, V_{sso} and V_{sfo} (m^3/h) are the sump discharge flow-rates of water, solids and fines respectively. It is assumed the slurry in the sump is fully mixed.

The discharge of each state from the sump through the variable speed pump is defined as:

$$V_{swo} = CFF \frac{X_{sw}}{X_{sw} + X_{ss}} \tag{11a}$$

$$V_{sso} = CFF \frac{X_{ss}}{X_{sw} + X_{ss}} \tag{11b}$$

$$V_{sfo} = CFF \frac{X_{sf}}{X_{sw} + X_{ss}}. \tag{11c}$$

The volume of the sump filled with slurry ($SVOL$) (m^3) is:

$$SVOL = X_{ss} + X_{sw}. \tag{12}$$

2.2.3. Hydrocyclone model:

The nonlinear static cyclone model presented here aims to model the product size and density by taking into account the slurry density, slurry viscosity, and the effects of angular velocity of the particle inside the cyclone. The underflow of coarse material (V_{ccu}) (m^3/h) is modelled as:

$$V_{ccu} = (V_{sso} - V_{sfo}) \left(1 - C_1 \exp\left(\frac{-CFF}{\varepsilon_c}\right)\right) \left(1 - \left(\frac{F_i}{C_2}\right)^{C_3}\right) \left(1 - P_i^{C_4}\right), \tag{13}$$

where $F_i = \frac{V_{sso}}{CFF}$ is the fraction solids in the cyclone feed, $P_i = \frac{V_{sfo}}{V_{sso}}$ is the fraction fines in the feed solids, ε_c (m^3/h) relates to the coarse split, C_1 relates to the split at low-flows when the centrifugal force on particles is relatively small, C_2 normalizes the fraction solids in the feed according to the upper limit for the packing fraction of solid particles, and C_3 and C_4 adjusts the sharpness of the dependency on F_i and P_i.

To determine the amount of water and fines accompanying the coarse underflow, the fraction of solids in the underflow (F_u) must be determined. This is modelled as:

$$F_u = C_5 - (C_5 - F_i) \exp\left(-V_{ccu}/(\alpha_{su}\varepsilon_c)\right), \tag{14}$$

where C_5 is the approximate maximum packing fraction, and α_{su} relates to the fraction solids in the underflow.

The cyclone underflow flow-rates (as shown in (1)) are defined as:

$$V_{cwu} = \frac{V_{swo}\left(V_{ccu} - F_u V_{ccu}\right)}{F_u V_{swo} + F_u V_{sfo} - V_{sfo}} \tag{15a}$$

$$V_{cfu} = \frac{V_{sfo}\left(V_{ccu} - F_u V_{ccu}\right)}{F_u V_{swo} + F_u V_{sfo} - V_{sfo}} \tag{15b}$$

$$V_{csu} = V_{ccu} + V_{cfu}. \tag{15c}$$

These equations follow from the assumption that the fines are not influenced by centrifugal forces. This implies the ratio of fines to water in the overflow, underflow and feed is equal, and that the fraction of solids in the underflow can be written as $F_u = \frac{V_{csu}}{V_{csu}+V_{cwu}}$. Consequently, the cyclone water overflow flow-rate (V_{cwo}), solids overflow flow-rate (V_{cso}) and fines overflow flow-rate (V_{cfo}) can be calculated using a flow balance around the cyclone.

The product quality is defined as the fraction of fines to solids in the cyclone overflow, and is represented by the particle size estimate (PSE):

$$PSE = \frac{V_{cfo}}{V_{cso}}. \tag{16}$$

2.3. Representation of the plant model in nonlinear state space

The plant model can be represented in nonlinear state-space as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(t, \mathbf{x}, \mathbf{u}\right) \tag{17a}$$

$$\mathbf{y}(t) = \mathbf{g}\left(t, \mathbf{x}, \mathbf{u}\right), \tag{17b}$$

where $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{u}$ represents the plant's state, measured, and manipulated variables respectively. These are given by:

$$\begin{aligned} \mathbf{x} &= [X_{mw}, X_{ms}, X_{mf}, X_{mr}, X_{mb}, X_{sw}, X_{ss}, X_{sf}]^T \\ \mathbf{u} &= [CFF, MFO, SFW, MIW, MFB]^T \\ \mathbf{y} &= [PSE, LOAD, SVOL]^T. \end{aligned}$$

Function $\mathbf{f}$ is given by (1) and (10), and function $\mathbf{g}$ is given by (16), (8), and (12).

2.4. Operating condition

The operating condition in Table 1 and the model parameter values in Table 3 were taken from controller comparison study of Le Roux et al. [6]. The manipulated variables have the following limits:

$$\begin{aligned} u_{lower} &= [100\ 0\ 0\ 0\ 0] \\ u_{upper} &= [450\ 100\ 400\ 80\ 10] \end{aligned} \tag{18}$$

where u_{lower} represents the lower bound, and u_{upper} represents the upper bound. Furthermore, the sump has a maximum capacity of 16 m^3.

3. SISO controllers design

This section presents the design of two SISO fractional control strategies for the grinding mill circuit. Before starting with the controllers design, definitions of fractional integrals and fractional derivatives are introduced.

3.1. Fractional integral and fractional derivative basics

In fractional calculus, the traditional definitions of the integral and derivative of a function are generalized from integer orders to real orders. In the time domain, the fractional order derivative and fractional order integral operators are defined by a convolution operation.

According to Kilbas et al. [20], the Riemann-Liouville fractional integral of order $\alpha \in \mathbb{R}$, with $\alpha \geq 0$, is defined as:

$$I_{t_0}^{\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^{t} \frac{f(\tau)}{(t-\tau)^{1-\alpha}} d\tau, \quad t > t_0, \tag{19}$$

where $\Gamma(\alpha)$ is the Gamma function [20].

Several definitions exist regarding the fractional derivative of order $\beta \geq 0$, but the Caputo definition [20] presented in (20) is the most used in engineering applications, and is the one used in this work:

$${}^{C}D_{t_0}^{\beta} f(t) = \frac{1}{\Gamma(n-\beta)} \int_{t_0}^{t} \frac{f^{(n)}(\tau)}{(t-\tau)^{\beta-n+1}} d\tau, \tag{20}$$

where $n-1 < \beta < n, \;\; n \in \mathbb{Z}^{+}$. If $\beta \in \mathbb{Z}^{+}$, then $n = \beta$.

The Laplace transform of the Riemann-Liouville fractional integral corresponds to $\mathcal{L}\left\{I_{t_0}^{\alpha} f(t)\right\} = s^{-\alpha} F(s)$, while the Laplace transform of the Caputo fractional derivative is $\mathcal{L}\left\{{}^{C}D_{t_0}^{\beta} f(t)\right\} = s^{\beta} F(s) - \sum_{k=0}^{n-1} s^{n-k-1} f^{(k)}(0)$ [35].

3.2. Design of SISO FOPI controllers

It has been reported in literature that the use of fractional order proportional-integral-derivative (FOPID) controllers could make control systems perform better than PID controllers. The transfer function of a FOPID controller is represented as:

$$G_c(s) = K_P + \frac{K_I}{s^{\alpha}} + K_D\, s^{\beta}, \tag{21}$$

where K_P is the proportional gain, K_I is the integral gain, K_D is the derivative gain, and $\alpha, \beta \in (0, 2)$ are the order of the integral and the order of the derivative, respectively, considered in this interval for stability purposes [36]. When $\alpha = \beta = 1$, then (21) represents the PID.

Advantages reported for FOPID controllers are summarized below [37].

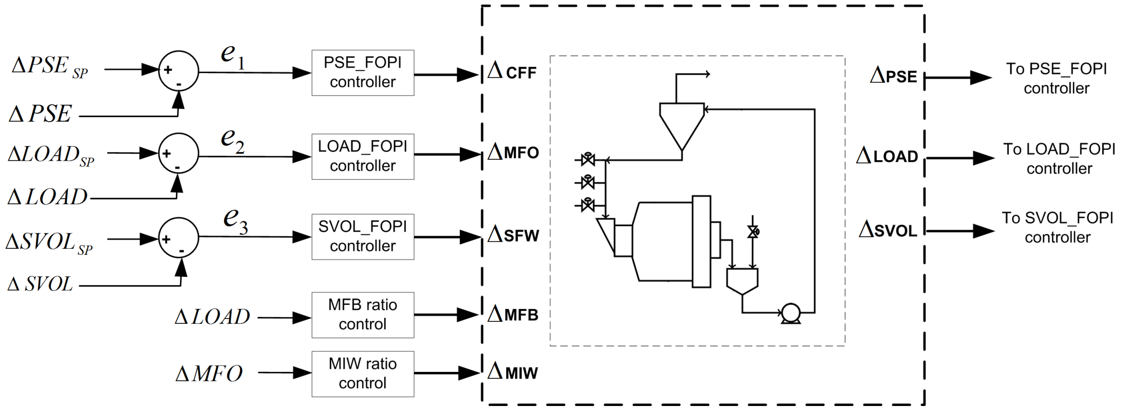

Figure 2: General diagram of SISO FOPI controllers configuration.

- Five degrees of freedom are available for the FOPID controller design, while three degrees of freedom for the PID controller design. This implies that more design criteria can be satisfied using a FOPID controller than using a PID controller.
- For higher order systems, the performance of PID controller deteriorates, whereas FOPID controller can provide better results.
- For a system with long time delays, FOPID controllers can provide better results than PID controllers.
- A FOPID controller achieves better robust stability compared to a PID controller.
- FOPID controllers can perform better than PID controllers for systems with nonlinearities.
- The FOPID controllers can achieve better responses for non-minimum phase systems.

Although all these advantages have been reported in literature, it is not always the case that FOPID controllers will perform better than PID controllers. The only way to know if it is better is to compare the FOPID controller to its integer order counterpart during the design stage.

Having all these possible advantages in mind, the first control strategy proposed in this work is the use of three SISO FOPI controllers (one per controlled variable). The method used for tuning the FOPI controller parameters, namely the proportional gain, the integral gain and the order of the integral, is explained in detail in Section 4.

Since the proposed controllers are SISO but the process is MIMO, with important interactions between variables, the selection of the controlled-manipulated variable pairing to be used is of great importance. For this particular circuit, it was found in [38] that the pairings $CFF-PSE$, $MFO-LOAD$ and $SFW-SVOL$ are the most appropriate pairings as they can improve the robustness to feed disturbances compared to other pairings.

Thus, this paper uses the mentioned pairs to implement SISO FOPI controllers. Fig. 2 shows a general diagram of the controlled system. In what follows, the proposed SISO FOPI controllers are described, using their equations in the time domain. Details for the implementation and tuning of these controllers are given in Section 4.

3.2.1. ***PSE_FOPI controller:***

This controller calculates the resulting control signal through the following equation:

$$\Delta CFF\left(t\right) = K_{P_1}\, e_1\left(t\right) + K_{I_1} I_{t_0}^{\alpha_1} e_1\left(t\right), \tag{22}$$

where $e_1(t) = \Delta PSE_{SP}(t) - \Delta PSE(t)$ is the control error, PSE_{SP} represents the set point for PSE, and the term Δ represents variations of each variable around the operating point. K_{P_1}, K_{I_1} and α_1 correspond to the proportional gain, the integral gain and the order of the integral of the PSE_FOPI controller, respectively, which are the three design parameters of this controller that need to be tuned.

3.2.2. ***LOAD_FOPI controller:***

For this FOPI controller, the control signal is:

$$\Delta MFO(t) = K_{P_2}\, e_2(t) + K_{I_2} I_{t_0}^{\alpha_2} e_2(t), \tag{23}$$

where $e_2(t) = \Delta LOAD_{SP}(t) - \Delta LOAD(t)$ represents the control error and $LOAD_{SP}$ represents the set point for $LOAD$. The controller's design parameters K_{P_2}, K_{I_2} and α_2 correspond to the proportional gain, integral gain and fractional order of the LOAD_FOPI controller, respectively.

3.2.3. ***SVOL_FOPI controller:***

Finally, for the case of this controller, the control signal is:

$$\Delta SFW(t) = K_{P_3}\, e_3(t) + K_{I_3} I_{t_0}^{\alpha_3} e_3(t), \tag{24}$$

where $e_3(t) = \Delta SVOL_{SP}(t) - \Delta SVOL(t)$ is the control error and $SVOL_{SP}$ represents the set point for $SVOL$. K_{P_3}, K_{I_3} and α_3 are the SVOL_FOPI controller design parameters that need to be tuned.

3.2.4. Additional manipulated variables:

In the case of the two additional manipulated variables of this circuit, MFB and MIW, they are used as in [6]:

- MFB is kept as a constant ratio of 16.7 of the volume of $LOAD$.
- MIW is kept as a constant ratio of 7% of MFO.

Thus, for the case of the SISO FOPI controllers, nine parameters need to be tuned: K_{P_1}, K_{I_1}, α_1, K_{P_2}, K_{I_2}, α_2, K_{P_3}, K_{I_3} and α_3.

3.3. Design of SISO FOPI-FOMRAC controllers

Adaptive controllers are used when the plant to be controlled is partially unknown. These controllers do not need the knowledge of a complete model of the plant (structure and parameters) in order to achieve control, which is a great advantage for practical applications. Therefore, adaptive controllers can achieve better performance compared to non-adaptive strategies when parameter variations and disturbances are present.

Model Reference Adaptive Control (MRAC) is one of the adaptive control techniques that can be found in literature. A great summary of this control strategy for linear plants can be found in [39]. The main objective of MRAC is that the output of a plant with unknown parameters follows asymptotically the output of a given reference model, and to that extent different MRAC approaches can be used. The simpler approach is the direct approach, where no identification of the plant parameters is attempted, but the controller parameters are directly adjusted using an integer order differential equation along with the information available in the control system.

Although the plant parameters can be unknown, some information/hypotheses about the plant are needed in order to design the controller [39], otherwise some modifications need to be made in the control scheme to deal with the additional uncertainty, which leads to more complex adaptive controllers. These four basic hypothesis are [39]:

- The sign of the high frequency gain of the plant is known.
- The relative degree (difference between the order of the denominator and the order of the numerator of the transfer function) of the plant is known.

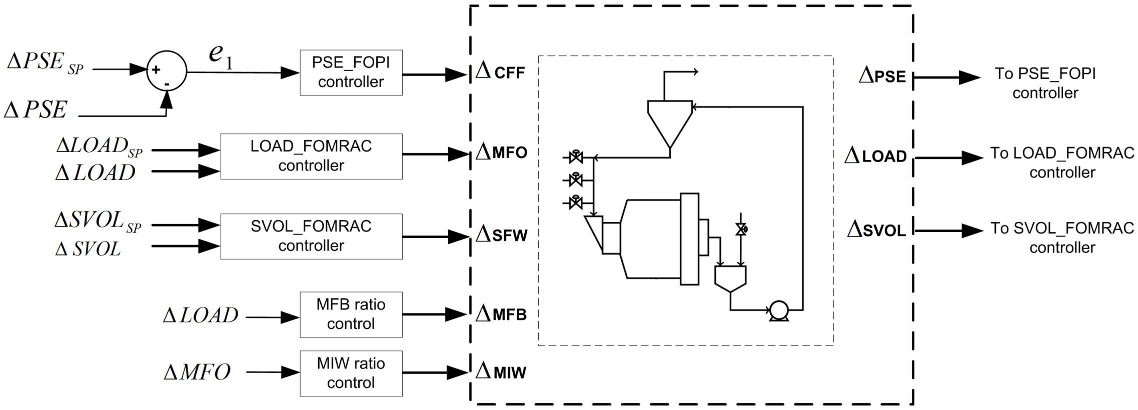

Figure 3: General diagram of SISO FOPI-FOMRAC controllers configuration.

- An upper bound on the order of the plant transfer function is known.
- The plant is minimum phase.

With the introduction of fractional operators in control systems, the idea of using MRAC together with fractional operators was investigated [23]. Afterwards, many other works were published dealing with this topic from both a theoretical [40] and a practical point of view [21, 25]. Some of the advantages of using fractional operators in MRAC schemes are better management of noise [41], better behavior under disturbances [21, 25, 42] and improvements in transient responses [21, 23].

Based on these reported advantages, the second control strategy proposed in this work correspond also to three SISO controllers, but in this case a combination of FOPI and FOMRAC is used. A general diagram of the SISO FOPI-FOMRAC controllers can be seen in Fig. 3.

Note that although the MRAC have been proposed for linear systems [39], in this work it will be applied to a nonlinear system. For checking the four basic hypotheses previously mentioned and also for selecting the amount of controller parameters to be used, the structure of a linearized model of the plant found in Le Roux and Craig [31] is used in the design stage. However, for the controller parameters tuning stage described in Section 4 and also for all the simulations presented in Section 6, the nonlinear model of the grinding circuit is used.

3.3.1. ***PSE_FOPI controller:***

For the case of controlled variable PSE, the FOPI designed in Section 3.2.1 will be maintained. This is because the relation between PSE and CFF can be approximated by a linearized model that is non-minimum phase [17, 31]. Under these circumstances, the fourth basic hypothesis for implementing MRAC is not fulfilled. Thus, the adaptive control scheme needs to be modified, leading to a more complex controller. Since the aim of this work is to use controllers with simple structure, the FOPI controller (22) is kept for PSE.

3.3.2. ***LOAD_FOMRAC controller:***

The relation between $LOAD$ and MFO can be approximated by a first order transfer function [17, 31]. This fulfils the four basic hypothesis mentioned previously, thus a direct FOMRAC is proposed in this case, using two adjustable parameters for the controller.

The control signal is generated as:

$$\Delta MFO\left(t\right) = \theta_1\left(t\right)\Delta LOAD\left(t\right) + \theta_2\left(t\right)\Delta LOAD_{SP}\left(t\right), \tag{25}$$

where $\theta_1\left(t\right), \theta_2\left(t\right) \in \mathbb{R}$ are adjustable parameters, $LOAD_{SP}\left(t\right)$ is the desired set point for the load in the mill, and Δ represents the variation of the corresponding variable around the operating point.

Adaptive laws for adjusting controller parameters θ_1, θ_2 are defined as:

$$
{}^{C}D_{t_0}^{\beta_1}\theta_1(t) = -\gamma_1 e_{LOAD}(t)\,\Delta LOAD(t), \quad \theta_1(t_0) = \theta_{1_0} \tag{26a}
$$

$$
{}^{C}D_{t_0}^{\beta_2}\theta_2(t) = -\gamma_2 e_{LOAD}(t)\,\Delta LOAD_{SP}(t), \quad \theta_2(t_0) = \theta_{2_0}. \tag{26b}
$$

where $\beta_1, \beta_2 \in (0, 2)$ are the fractional orders of the adaptive laws and $\theta_{1_0}, \theta_{2_0}$ are the initial values for the estimated parameters. $\gamma_1, \gamma_2 \in \mathbb{R}^+$ are the adaptive gains, which are used to manipulate the convergence speed of the parameters in the adaptive laws. In this work, adaptive gains are used as positive constants, but in general it is possible to use either constant or time varying adaptive gains [39].

The error $e_{LOAD}(t)$ in (26) is defined as:

$$
e_{LOAD}(t) = \Delta LOAD(t) - \Delta y_{m_{LOAD}}(t), \tag{27}
$$

where $\Delta y_{m_{LOAD}}(t)$ is the output of an asymptotically stable and known reference model, when the reference $\Delta LOAD_{SP}$ is applied to its input. Thus:

$$
\Delta y_{m_{LOAD}}(t) = \frac{a_{m_{LOAD}}}{s + a_{m_{LOAD}}}\Delta LOAD_{SP}(t), \quad a_{m_{LOAD}} > 0. \tag{28}
$$

As can be seen from (28), the reference model is defined by a first order transfer function, with a pole in $s = -a_{m_{LOAD}}$, and unit gain. For this controller, parameters $\beta_1, \beta_2, \gamma_1, \gamma_2, a_{m_{LOAD}}$ are design parameters that need to be tuned according to some criteria.

3.3.3. ***SVOL_FOMRAC controller:***

The relation between $SVOL$ and SFW can be approximated by an integrating transfer function [17, 31]. Thus, the four hypothesis mentioned early are fulfilled, and a direct FOMRAC can be proposed for this variable as well, using two adjustable parameters.

The structure of this controller is the same as for the LOAD_FOMRAC controller, thus only a summary of the equations will be shown without explanations.

$$
\Delta SFW(t) = \theta_3(t)\,\Delta SVOL(t) + \theta_4(t)\,\Delta SVOL_{SP}(t) \tag{29}
$$

$$
{}^{C}D_{t_0}^{\beta_3}\theta_3(t) = -\gamma_3 e_{SVOL}(t)\,\Delta SVOL(t), \quad \theta_3(t_0) = \theta_{3_0} \tag{30a}
$$

$$
\tag{30b}
$$

$$
{}^{C}D_{t_0}^{\beta_4}\theta_4(t) = -\gamma_4 e_{SVOL}(t)\,\Delta SVOL_{SP}(t), \quad \theta_4(t_0) = \theta_{4_0} \tag{30c}
$$

$$
e_{SVOL}(t) = \Delta SVOL(t) - \Delta y_{m_{SVOL}}(t) \tag{31}
$$

$$
\Delta y_{m_{SVOL}}(t) = \frac{a_{m_{SVOL}}}{s + a_{m_{SVOL}}}\Delta SVOL_{SP}(t), \quad a_{m_{SVOL}} > 0. \tag{32}
$$

To summarize, the FOPI_FOMRAC controller shown in Fig. 3 has 13 parameters to be tuned, which correspond to K_{P_1}, K_{I_1}, α_1, β_1, γ_1, β_2, γ_2, $a_{m_{LOAD}}$, β_3, γ_3, β_4, γ_4 and $a_{m_{SVOL}}$.

4. SISO controllers tuning

This section presents the procedure used for tuning the controller parameters. Off-line Particle Swarm Optimization (PSO) is used to select the controller parameters. Before explaining the tuning process itself, some details regarding the implementation of the controllers are given in order to facilitate the reproducibility of the results.

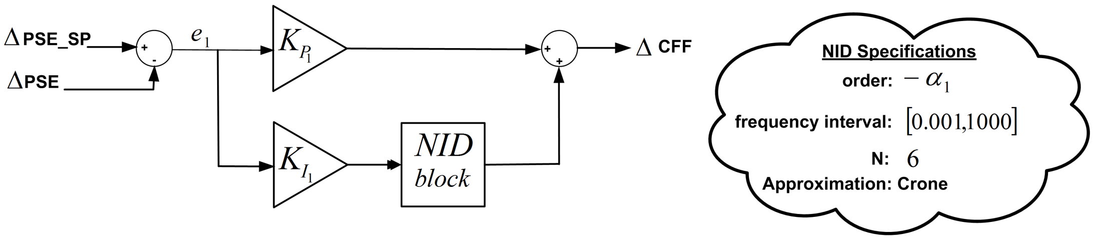

Figure 4: Block diagram for the implementation of FOPI controllers.

4.1. Numerical approximation for implementing fractional integrals and derivatives

One of the most common ways of implementing fractional integrals and derivatives in simulations and practical applications is by means of numerical approximations of these operators. The idea is to obtain integer-order transfer functions whose behavior approximates the fractional order Laplace operator:

$$C(s) = ks^{\alpha}. \tag{33}$$

Oustaloup's method [43] is one of the available frequency-domain methods for making this approximation, which uses a recursive distribution of N poles and N zeros of the form:

$$C(s) = k^{'} \prod_{n=1}^{N} \frac{1 + s/\omega_{zn}}{1 + s/\omega_{pn}}. \tag{34}$$

The gain $k^{'}$ is adjusted so that if $k = 1$ then $|C(s)| = 0$ dB at 1 rad/s and ω_{zn}, ω_{pn} represent respectively the zeros and poles of the approximation, which are placed inside a frequency interval $[\omega_l, \omega_h]$ rad/s in which the approximation is valid.

The Oustaloup's method is incorporated in the NID block of the Ninteger Toolbox for Matlab/Simulink [44] specified as the Crone approximation. In this block, if $\alpha < 0$ is set, then the NID simulates a fractional integral, otherwise if $\alpha > 0$ the NID simulates a fractional derivative.

4.2. Implementation of the controllers

The implementation of the fractional order controllers proposed in this work is made using Matlab/Simulink, along with the Ninteger Toolbox for Matlab/Simulink [44]. To ensure reproducibility of the results and future applications of fractional controllers by other researchers, block diagrams are presented here to make explicit the implementation of the two kind of controllers used in this work.

Fig. 4 shows the block diagram used to implement FOPI controllers. The diagram corresponds to the implementation of the PSE_FOPI controller (22), but it is the same structure for the LOAD_FOPI and the SVOL_FOPI controllers. It can be seen from Fig. 4 that the implementation is very simple, and the internal specifications of the NID block, which simulates a fractional integral in this case, are also shown.

Fig. 5, on the other hand, shows the block diagram used to implement FOMRAC controllers. The diagram corresponds to the implementation of the LOAD_FOMRAC controller (25)-(28), but it is the same structure for SVOL_FOMRAC controller. The NID specifications for the FOMRAC controller are the same as for the FOPI controllers, shown in Fig. 4, except for the orders which in the case of Fig. 5 are $-\beta_1$ and $-\beta_2$.

4.3. Particle Swarm Optimization basics

PSO is an heuristic global optimization technique that belongs to the category of swarm intelligence, which is also a sub-category of Evolutionary Computation. This technique allows solving optimization problems by using clusters or swarms of particles, simulating the behavior of social groups in nature (flocks, banks, crowds, etc.) in its process of searching for a common benefit.

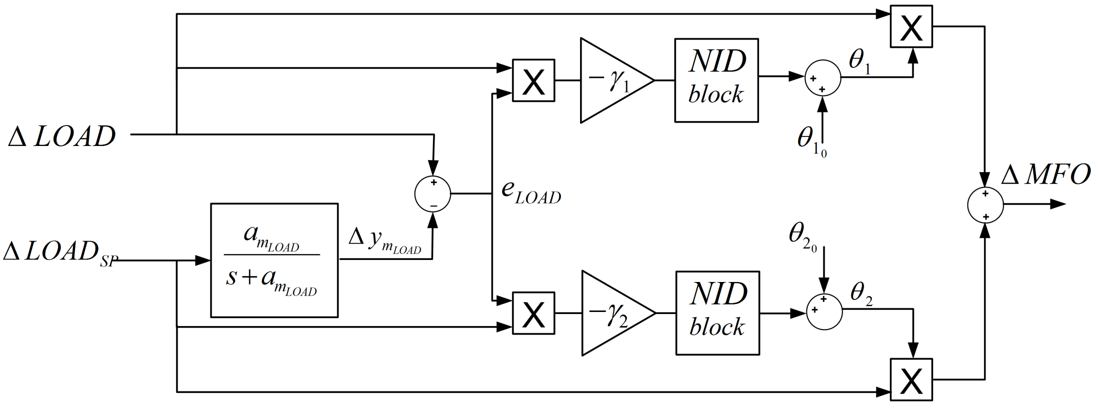

Figure 5: Block diagram for the implementation of FOMRAC controllers.

The algorithm defines a population where every possible solution is represented by a particle. These particles move in the search space iteratively according to certain rules, influenced by the particle that had found the best global position, with respect to a predefined objective function (fitness function). The evolution of the particles is given by:

$$\begin{aligned} &v_i(k+1) = \omega\ v_i(k) + r_1(k)\,c_1\,[p_i(k) - x_i(k)] + r_2(k)\,c_2\,(g(k) - x_{i,d}(k)) \\ &x_i(k+1) = x_i(k) + v_i(k+1)\,, \\ &p_i(k) = \begin{cases} p_i(k-1) & if \quad f(p_i(k-1)) \le f(x_i(k)) \\ x_i(k) & if \quad f(p_i(k-1)) > f(x_i(k)) \end{cases} \\ &g(k) = argmin\,\{f(p_1(k)),\ldots,f(p_s(k))\} \end{aligned} \tag{35}$$

where $v_i(k)$ and $x_i(k)$ represent the speed and the position of the i-th particle at the k-th iteration respectively, and $\omega \in [0,1]$ is the inertia weight, which serves to limit the particle velocity and consequently to achieve convergence to an equilibrium point. c_1 and c_2 are the social and cognitive acceleration coefficients, $r_1(k)$ and $r_2(k)$ are a pair of random numbers uniformly distributed in the interval $[0,1]$ and represents the stochastic element of any swarm, f is the fitness function to be optimized, and s is the number of particles in the swarm [45, 46].

As an heuristic technique, PSO has the advantage of being able to drive the most versatile of fitness functions, and being able to use non-differentiable, nonlinear and /or discontinuous functions. Compared to techniques such as genetic algorithms and differential evolution, PSO is a good alternative to solve global optimization problems with multiple maximum/minimum, discontinuities and deterministic solutions in non-polynomial time. This is reflected in the increase of successful applications based on PSO [47]. For these reasons it is used in this work.

4.4. Tuning of SISO FOPI controller parameters using PSO

As explained in Section 3.2, the controller proposed in this case for the whole multivariable process corresponds to three SISO FOPI controllers. This leads to nine design parameters to be tuned using the optimization process, thus the parameter vector to be optimized by PSO is:

$$x = [K_{P_1}\ \ K_{I_1}\ \ \alpha_1\ \ K_{P_2}\ \ K_{I_2}\ \ \alpha_2\ \ K_{P_3}\ \ K_{I_3}\ \ \alpha_3]\,. \tag{36}$$

4.4.1. Cost function and weighting factors:

The cost function used in the optimization process was selected as:

$$\begin{aligned} J = \; & w_1 \int\limits_{t_0}^{T} e_1^2(t)\,dt + w_2 \int\limits_{t_0}^{T} e_2^2(t)\,dt + w_3 \int\limits_{t_0}^{T} e_3^2(t)\,dt + w_4 \int\limits_{t_0}^{T} \left[\Delta CFF(t)\right]^2 dt + \\ & + w_5 \int\limits_{t_0}^{T} \left[\Delta MFO(t)\right]^2 dt + w_6 \int\limits_{t_0}^{T} \left[\Delta SFW(t)\right]^2 dt \end{aligned} \tag{37}$$

where $e_1(t) = \Delta PSE_{SP}(t) - \Delta PSE(t)$, $e_2(t) = \Delta LOAD_{SP}(t) - \Delta LOAD(t)$, $e_3(t) = \Delta SVOL_{SP}(t) - \Delta SVOL(t)$ and w_i, $i = 1, \ldots, 6$ are weighting factors to give more or less importance to every term in the cost function.

The weighting factors were determined such that a 1% deviation from set-point for PSE will produce an error in the cost function equal to a 5% deviation of $LOAD$ from set-point and equal to a 20% change in $SVOL$ from set-point:

$$w_1 \left(1\% PSE_{SP}\right)^2 = w_2 \left(5\% LOAD_{SP}\right)^2 = w_3 \left(20\% SVOL_{SP}\right)^2 .$$

For the operating condition shown in Table 1, and choosing $w_3 = 1$, the controlled variable weighting factors are:

$$w_1 = 1.2407 \times 10^5 \quad w_2 = 2.0458 \times 10^4 \quad w_3 = 1. \tag{38}$$

In the case of the manipulated variables, the weighting factors were determined such that 2% changes of half the ranges of MFO, SFW, and CFF will produce the same error in the cost function. The corresponding weighting factors were scaled to produce 1% of the error compared to the weighting factors of the corresponding controller variables:

$$100\, w_5 \left(\frac{2\% MFO_{range}}{2}\right)^2 = w_2 \left(5\% LOAD_{SP}\right)^2$$

and

$$w_4 \left(\frac{2\% CFF_{range}}{2}\right)^2 = w_5 \left(\frac{2\% MFO_{range}}{2}\right)^2 = w_6 \left(\frac{2\% SFW_{range}}{2}\right)^2 .$$

Therefore, the manipulated variable weighting factors are:

$$w_4 = 0.0045 \quad w_5 = 0.0557 \quad w_6 = 0.0035. \tag{39}$$

4.4.2. PSO specifications and results:

For the optimization process, the lower and upper bounds for the parameters were selected as:

$$\begin{aligned} x_{lower} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ x_{upper} &= \begin{bmatrix} 10^8 & 10^8 & 1.99 & 10^8 & 10^8 & 1.99 & 10^8 & 10^8 & 1.99 \end{bmatrix} . \end{aligned} \tag{40}$$

In the case of the fractional orders α_1, α_2 and α_3, their upper limit was set to 1.99 in order to keep the stability of the corresponding closed loops. According to Matignon [36], stability cannot be guaranteed for any linear time invariant system if the fractional order is equal or higher than 2. For the case of the grinding mill circuit, it is not possible to make a stability analysis at this moment, since the tools for stability analysis of nonlinear fractional systems is limited. Thus, the upper limit of 1.99 is a conservative bound.

Regarding the upper bounds of the gains of the controllers, they were set to 10^8 in order to give a wide searching space to the optimization process. An upper bound that guarantees stability cannot be provided a-priori to the algorithm, since it will depend on the fractional orders used, which are also varied during the optimization process.

Stability issues due to combinations of gains and orders are taken into consideration by the cost function. If a combination of fractional orders and gains lead to instability of the circuit during the optimization process, the cost function returns a high value to guide the optimization process away from these solutions. Given the lower and upper limits of the fractional orders used, the resulting controller could be either a classic PI controller ($\alpha_1 = \alpha_2 = \alpha_3 = 1$) or a FOPI controller, depending on the parameter values which produces the minimum of the cost function.

If the designer does not have a nonlinear model of the plant and/or does not want to use heuristic methods like PSO for the tuning process, then some different tuning techniques could be used. One of the most simple technique consists of using tuning rules like the ones presented in Bhaskaran et al. [48]. These tuning rules allow obtaining FOPI parameters based in first order plus time delay approximations of the plant around the operating point, which can be obtained using standard system identification techniques [49]. It is similar to, for instance, using the well known Ziegler-Nichols rules for the classical PI controllers. After obtaining the controller parameter values, a manual fine tuning can be done, as in many processes with classical PI controllers.

With all the mentioned parameters already defined, the optimization process was carried out using the Constrained Particle Swarm Optimization toolbox for Matlab [50]. Most of the PSO parameters were used at their default values, except:

- Population size: 150.
- Number of generations: 350.

For every generation the milling circuit is simulated in a time window of 10 hours under ideal conditions for every particle, that is, no disturbances or noise are present in the scheme. Table 1 contains the initial conditions and limits of input and output variables, as well as the initial values of the states used in simulations.

Step references are applied to the three inputs at $t = 0$, specifically $PSE_{SP} = 68$ %, $LOAD_{SP} = 34$ % and $SVOL_{SP} = 14.8$ m^3. With the simulation results, cost function (37) is calculated for every particle, and based on the corresponding results the algorithm generates the population to be used in the next generation. The process continues until 350 generations are reached and the optimal x is found.

As a result of this optimization process, the resulting $FOPI$ controller parameters are:

$$\begin{array}{rlrlrl} K_{P_1} & = 731.42 & K_{P_2} & = 12835 & K_{P_3} & = 23.97 \\ K_{I_1} & = 64698 & K_{I_2} & = 33818 & K_{I_3} & = 5.12 \\ \alpha_1 & = 0.96 & \alpha_2 & = 0.98 & \alpha_3 & = 0.95 \end{array} \tag{41}$$

The parameter values in (41) will be used in the simulation studies presented in Section 6.

4.5. Tuning of SISO FOPI-FOMRAC controller parameters using PSO

In the case of FOPI-FOMRAC controller, PSO was also used to tune the parameters. The same experiment used for the FOPI tuning was carried out for this controller, using the same specifications for PSO, the same cost function (37), and the same weighting factors (38),(39).

For the FOPI-FOMRAC controller, the vector x to be optimized corresponds to:

$$x = [K_{P_1} \quad K_{I_1} \quad \alpha_1 \quad \beta_1 \quad \gamma_1 \quad \beta_2 \quad \gamma_2 \quad a_{m_{LOAD}} \quad \beta_3 \quad \gamma_3 \quad \beta_4 \quad \gamma_4 \quad a_{m_{SVOL}}], \tag{42}$$

and the upper and lower limits were set as:

$$\begin{aligned} x_{lower} &= [0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0] \\ x_{upper} &= \left[10^8 \;\; 10^8 \;\; 1.99 \;\; 1.99 \;\; 10^3 \;\; 1.99 \;\; 10^3 \;\; 10^3 \;\; 1.99 \;\; 10^3 \;\; 1.99 \;\; 10^3 \;\; 10^3\right]. \end{aligned} \tag{43}$$

The limits used for the fractional orders were selected in order to guarantee stability of the scheme [36, 40]. The upper limit of the adaptive gains was set to 10^3, since according to the experience obtained from many previous analysis of fractional adaptive schemes, higher values usually do not improve the circuit behavior. For the poles of the reference models, the upper limit was also set to 10^3, since with this value the response speed is faster.

Manually tuning the FOMRAC controllers could be hard to achieve, since more parameters need to be adjusted than in the case of FOPI controllers. Nevertheless, if the designer does not have a nonlinear model of the plant and/or does not want to use heuristic methods like PSO for adjusting the controller parameters, a very conservative classical MRAC could be implemented. To that extent, adaptive gains equal to 1 can be used in a first tuning attempt, along with orders equals to 1. Selection of the poles of the reference models will depend on how fast the designer wants the control action to be, which depends on the control objectives. Afterwards, a manual fine tuning could be done. Nevertheless, it must be mentioned that the design of the FOMRAC, i.e. the number of controller parameters, the order of the reference model, and the structure of the controller, will depend on the process to be controlled. Therefore, it could be even harder to do a manual tuning if the process cannot be approximated by first-order-plus-time-delay transfer functions.

As a result of the optimization process, the FOPI-FOMRAC controller parameters are:

$$\begin{array}{lllll} K_{P_1} = 761.27 & K_{I_1} = 73667 & \alpha_1 = 0.98 & & \\ \beta_1 = 0.89 & \gamma_1 = 99.94 & \beta_2 = 1.82 & \gamma_2 = 13.94 & a_{m_{LOAD}} = 922.07 \\ \beta_3 = 1.1 & \gamma_3 = 0.1 & \beta_4 = 0.98 & \gamma_4 = 1.88 & a_{m_{SVOL}} = 25.347 \end{array} \tag{44}$$

The parameter values in (44) will be used in the simulation study presented in Section 6.

5. LMPC Controller

The ability of the fractional order controllers to control the grinding mill circuit is compared to the performance of an LMPC controller. LMPC is chosen as it can be considered to be the de-facto standard for advanced process control implementations in industry [51, 52], and was successfully applied to control grinding mill circuits both in simulation and practice [1, 2, 3, 53, 54]. The aim of the LMPC controller can be described as:

$$\begin{array}{ll} \min\limits_{u_k,\dots,u_{k+N_c-1}} & J\left(u_k, \dots, u_{k+N_c-1}, x_k\right) \\ \text{s.t.} & x_{k+1} = Ax_k + Bu_k \\ & y_k = Cx_k + Du_k \\ & u_l \leq u_k \leq u_u \end{array} \tag{45}$$

where u_l and u_u represent the lower and upper limits of the manipulated variables u respectively. The objective function J is defined as:

$$J(\cdot) = \frac{1}{2}\sum_{i=0}^{N_p-1} \left(y_{sp} - y_{k+i|k}\right)^T Q \left(y_{sp} - y_{k+i|k}\right) + \frac{1}{2}\sum_{i=0}^{N_c-1} \left(\Delta u_{k+i|k}\right)^T R \left(\Delta u_{k+i|k}\right), \tag{46}$$

where y_{sp} is the controlled variable set-points, N_p is the prediction horizon, N_c is the control horizon, and Q and R are the controlled and manipulated variable weighting matrices respectively.

A continuous linear state-space representation with matrices A, B, C, and D of the plant model in (17) is calculated where $A = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} |_{\mathbf{x}=x_0,\ \mathbf{u}=u_0}$, $B = \frac{\partial \mathbf{f}}{\partial \mathbf{u}} |_{\mathbf{x}=x_0,\ \mathbf{u}=u_0}$, $C = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} |_{\mathbf{x}=x_0,\ \mathbf{u}=u_0}$, and $D = \frac{\partial \mathbf{g}}{\partial \mathbf{u}} |_{\mathbf{x}=x_0,\ \mathbf{u}=u_0}$. The linear state-space model of the plant is discretized using Euler's Explicit integration rule to obtain the discrete-time model. Full-state feedback is assumed. This is a significant assumption as the measurements available in industrial grinding mill circuits are limited [55]. Although various state estimators have been investigated for grinding mills [17, 56, 57, 58], it remains a challenge to estimate mill hold-ups [19, 59].

5.1. Prediction and Control Horizon

The prediction horizon should be long enough for sufficient dynamics of the process to be visible, but short enough for the control action calculation not to be too computationally expensive. The control horizon should be short enough such that the controller is not too aggressive, but long enough for a sufficient part of the prediction horizon to contain control action [18]. To fulfill these requirements, N_p is chosen as 36 (6 min), and N_c as 9 (1.5 min).

5.2. Weighting matrices

The weighting matrix Q for the controlled variables and the weighting matrix R for the manipulated are determined following the same basis used to select the weighting factors for the cost function in the optimization process, specified in Section 4.4. Thus, the controlled variable weighting matrix is:

$$Q = \text{diag}\left(\left[1.2407 \times 10^5,\ 2.0458 \times 10^4,\ 1\right]\right), \tag{47}$$

and the manipulated variable weighting matrix is:

$$R = 10^{-3}\ \text{diag}([4.5,\ 5.57,\ 3.5]). \tag{48}$$

6. Simulation and results

This section presents the simulation results obtained using the proposed controllers for the milling circuit, under disturbances and process noise conditions[2].

6.1. Simulation environment

The parameter values and the operating point can be seen in Tables 1 and 3, respectively. The grinding mill circuit was simulated with the following general conditions:

1. Simulation time of 4 h and a sampling rate of 10 s.
2. A disturbance in the mill feed size distribution is simulated by increasing α_r to 0.765 at $t = 0.2$ h, to 0.365 at $t = 1.5$ h, and back to 0.465 at $t = 2.9$ h.
3. A disturbance in the energy required to produce fine ore, which is similar to a change in the hardness of the ore, is simulated by increasing ϕ_f to 32.5 kWh/t at $t = 0.6$ h, to 27.5 at $t = 1.9$ h, and back to 29.5 kWh/t at $t = 3.2$ h.
4. A disturbance in the energy required to break rocks into solids is simulated by decreasing ϕ_r to 5.72 kWh/t at $t = 1.0$ h, to 8.22 at $t = 2.3$ h, and back to 6.72 kWh/t at $t = 3.6$ h.

The parameter variations can be seen in Fig. 6.

In order to evaluate the noise rejection capabilities of the controllers, two different scenarios are simulated:

- In the first scenario, disturbances as specified above are applied to the circuit, but no process noise is added.
- In the second scenario, disturbances as specified above are applied, and process noise is added to the states (X_{mw}, X_{ms}, X_{mf}, X_{mr}, X_{mb}, X_{sw}, X_{ss} and X_{sf}). The process noise follows a uniform random distribution, with maximum and minimum values given by $\pm 0.01 x_0$, where x_0 is the nominal value of the corresponding state (1% of the state's nominal value).

6.2. Performance functions to evaluate the grinding mill circuit behavior

In order to make more specific conclusions from the experiments, three performance functions are proposed to evaluate the behavior of the grinding mill circuit from different perspectives.

1. The first performance function proposed is the Normalized Root Mean Square Error (NRMSE), which is calculated for each controlled variable as:

$$\text{NRMSE} = \frac{\sqrt{\dfrac{\sum\limits^{N} (y - y_{SP})^2}{N}}}{y_{max} - y_{min}}, \tag{49}$$

where y is the controlled variable, y_{SP} is the corresponding set point, N is the number of sample data, and y_{max} and y_{min} are the maximum and minimum values of y, respectively. This performance index checks how far the controlled variables are from their set points.

[2] The MATLAB simulation code is available from the authors on request.

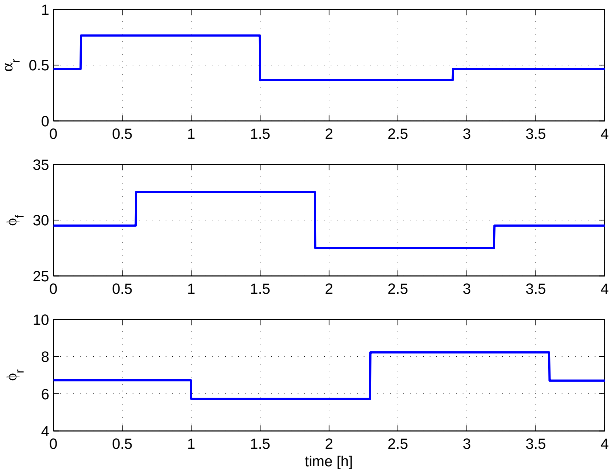

Figure 6: Variation of parameters α_r, ϕ_f and ϕ_r.

2. The second performance index is the Normalized Root Mean Square Input (NRMSI), which is calculated for each manipulated variable as:

$$\mathrm{NRMSI} = \frac{\sqrt{\frac{\sum\limits^{N} (u - u_{OP})^2}{N}}}{u_{max} - u_{min}}, \tag{50}$$

   where u is the manipulated variable, u_{OP} is the corresponding value of the manipulated variable in the operating point, N is the number of sample data, and u_{max} and u_{min} are the maximum and minimum values of u, respectively. This performance index checks how far from their operating point values the manipulated variables need to move, in order to keep the controlled variables as close as possible to their set points.

3. In grinding mill circuits, keeping PSE at its set point is of great importance, but reducing small variations in this variable is also critical. That is why the third performance function proposed in this paper is the variance of PSE, which will be referred to σ_{PSE} and calculated as:

$$\sigma_{PSE} = \frac{\sum\limits^{N} |PSE - \mu|^2}{N}, \tag{51}$$

   where N is the number of sample data, and μ is the mean value of PSE.

6.3. Results in the presence of external disturbances

Fig. 7 shows the behavior of the three controlled variables for the first simulation scenario. During this time window, the external disturbances described above and showed in Fig. 6 are present in the grinding mill circuit.

As seen from Figure 7, PSE has a similar behavior for the cases with FOPI and with FOPI+FOMRAC, keeping the variable around its set point. The variations are not higher than 4 % for FOPI and FOPI+FOMRAC controllers, while for LMPC they are not higher than 5 %. It can also be seen that fractional controllers tend more quickly to return PSE to its set point after external disturbances appears, while LMPC has a slower response.

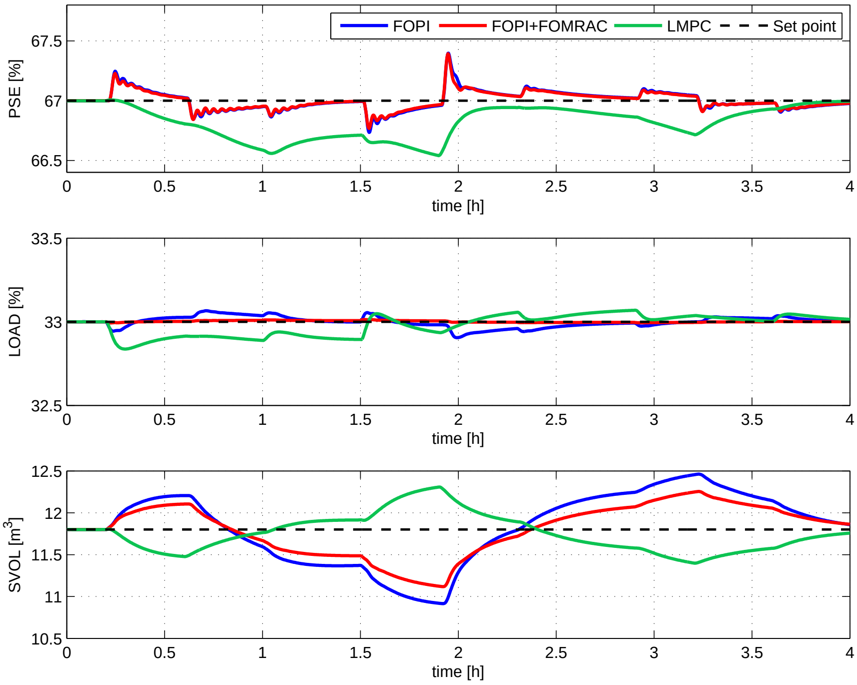

Figure 7: Behavior of the controlled variables when external disturbances are present in the single-stage closed grinding mill circuit.

In the case of $LOAD$, variations around the set point are less than 1.5%, but it can be clearly seen that the FOPI+FOMRAC controller is less affected by the external disturbances than the FOPI and LMPC controllers.

The behavior of PSE and $LOAD$ can be regarded as excellent results for the fractional controllers, since these are SISO controllers and no attempt is made to decouple the variables.

Finally, in the case of $SVOL$, no significant differences can be seen between the three control strategies. Variations of this controlled variable are not critical, as long as it remains between the corresponding limits.

Regarding the manipulated variables, Fig. 8 shows their evolution for this simulation scenario. As can be seen from Fig. 8, the three control strategies keep the manipulated variables within their limits. No significant differences can be seen between the three control strategies from Fig. 8. However, evaluation of the performance function NRMSI is more revealing.

As seen in Fig. 8, the manipulated variables follow the same trend for all three controllers. However, the response of the controlled variables in Fig. 7 for FOPI and FOPI+FOMRAC differ somewhat to the response of the controlled variables for LMPC. For example, $SVOL$ follows the same trend as CFF and SFW for LMPC, but the negative of the trend of CFF and SFW for FOPI and FOPI+FOMRAC. Although not visible in the graph, for the LMPC SFW starts to decrease before CFF starts to decrease. The opposite occurs for the FOPI and FOPI+FOMRAC controllers where CFF reacts before SFW. Since the sump acts as an integrator, the trend of SFW dominates if it reacts before CFF, but the negative of the trend of CFF will dominate if it reacts before CFF. However, even with these differences in responses, if the vertical axis scales of Fig. 7 are considered it can be seen that the three controllers are capable of maintaining the three controlled variables very close to their desired setpoints.

Table 4 shows the values of the performance functions for the simulation already presented.

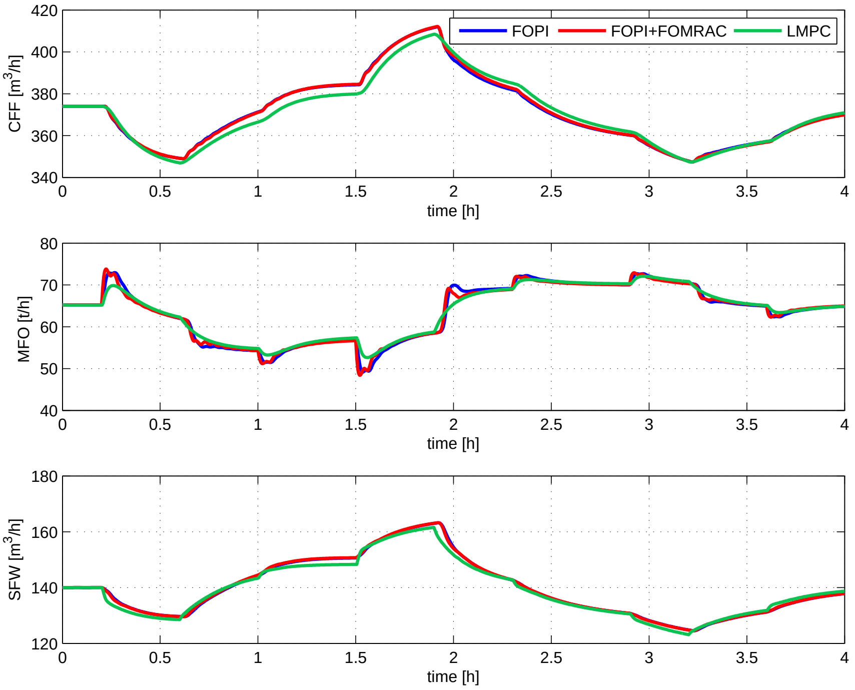

Figure 8: Behavior of the manipulated variables when external disturbances are present in the single-stage closed grinding mill circuit.

As can be seen from Table 4, the NRMSE of PSE is less for the fractional control strategies, which is in accordance with the conclusions made from the plots. The same occurs for the NRMSE of $LOAD$, where it can be seen that it is lower for the FOPI+FOMRAC controllers. Regarding the NRMSE of $SVOL$, differences are not so abrupt between the three control strategies, although LMPC has the lower value in this case.

In the case of the variance of PSE, it is higher for the LMPC controllers. Since PSE is the most important variable in the circuit, the difference between the fractional controllers and the LMPC can be seen as an improvement in the grinding mill circuit operation. Moreover, it was stated by Wei and Craig [4] that the mineral recovery downstream improves as σ_{PSE} decreases, thus the results for σ_{PSE} in the case

Table 4: Values of performance functions for the case when external disturbances are present in the grinding mill circuit

		FOPI	FOPI+FOMRAC	LMPC
NRMSE	PSE	0.0915	0.0831	0.2515
	$LOAD$	0.1552	0.0249	0.2981
	$SVOL$	0.2716	0.1951	0.1119
σ_{PSE}		0.0068	0.0056	0.0180
NRMSI	CFF	0.2544	0.2567	0.2518
	MFO	0.2646	0.2593	0.2381
	SFW	0.2548	0.2553	0.2451

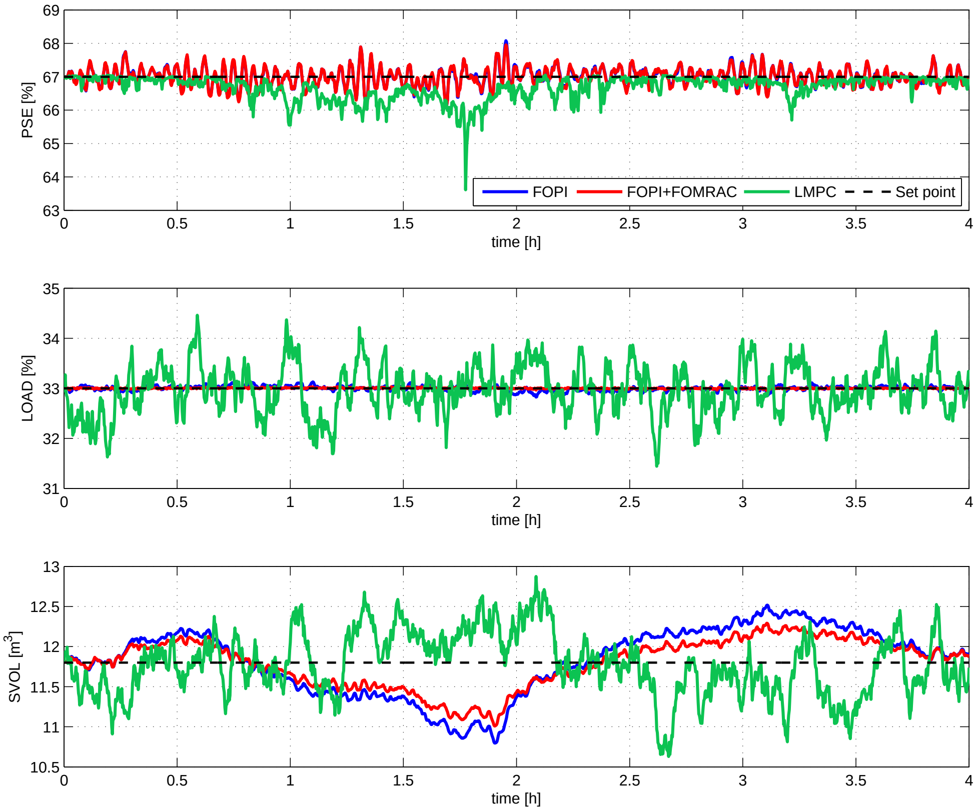

Figure 9: Behavior of the controlled variables when external disturbances and process noise are present in the single-stage closed grinding mill circuit.

using the FOPI+FOMRAC can be interpreted as an improvement in the mineral recovery.

Regarding the NRMSI, it can be seen from Table 4 that it has similar magnitudes for the three control strategies. Nevertheless, there are slight differences that could benefit LMPC in the comparison.

6.4. Results in the presence of external disturbances and process noise

The results obtained for the second scenario can be seen in Fig. 9 and Fig. 10. Table 5 shows the corresponding values of the performance functions.

It can be seen from Fig. 9 that all the controlled variables are affected by the process noise, for the three control strategies, although they remain close to their set points, which is an indication of satisfactory operation. However, the fractional controllers are considerably less affected by the process noise than LMPC, specially variables PSE and $LOAD$, which can be seen from Fig. 9 and supported by the NRMSE performance function and σ_{PSE} in Table 5. This agrees with other results that have been cited in the literature, where the capability of fractional controllers to handle the noise is one of their advantages. Although no analytical results are available to support this behavior, it seems to be related with the memory intrinsically incorporated by definition in fractional operators.

Regarding the manipulated variables, it can be seen from Fig. 10 that they are also affected by the process noise. On the contrary to the case without process noise, where the NRMSI had similar values for

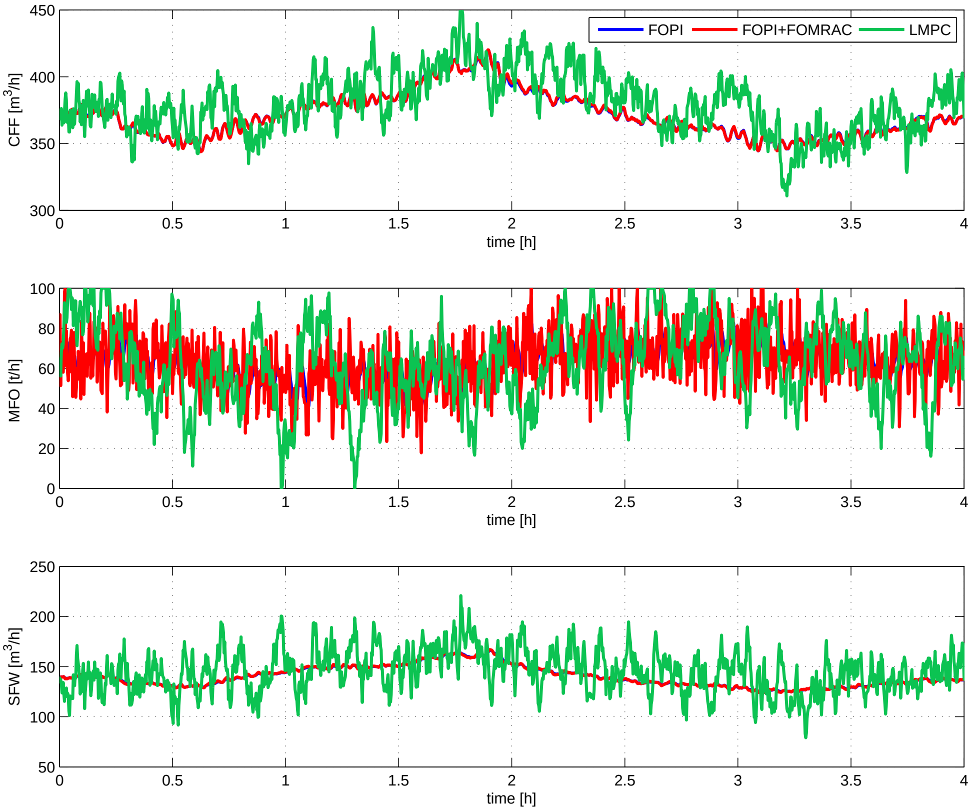

Figure 10: Behavior of the manipulated variables when external disturbances and process noise are present in the single-stage closed grinding mill circuit.

the three control strategies, in this case the difference in the NRMSI is more evident. The lowest NRMSI values are achieved by the FOPI+FOMRAC controller.

In summary, for the external disturbances incorporated in the simulation, the fractional order SISO controllers achieve similar and improved behavior to that achieved with LMPC. The fractional order SISO controllers present an improved response when process noise is present in the system. The use of these fractional order SISO controllers can be seen as a potential solution to improve milling circuit operation.

7. Conclusions

This paper presented the design and application of two kinds of fractional order SISO controllers to a grinding mill circuit. The performance of the proposed control techniques was evaluated against the performance of an LMPC controller, which is an advanced MIMO control technique. Simulations were conducted when external disturbances and process noise were present in the circuit.

Results indicate that the fractional order SISO controllers achieve similar or better results compared to LMPC in the presence of parametric disturbances and process noise.

Table 5: Values of performance functions for the case when external disturbances and process noise are present in the grinding mill circuit

		FOPI	FOPI+FOMRAC	LMPC
	PSE	0.0608	0.0625	0.1065
NRMSE	$LOAD$	0.0157	0.0036	0.1643
	$SVOL$	0.1855	0.1337	0.1754
σ_{PSE}		0.0738	0.0780	0.1220
	CFF	0.1194	0.1206	0.1674
NRMSI	MFO	0.0810	0.1362	0.2036
	SFW	0.0717	0.0721	0.1533

Acknowledgments

The results reported in this paper have been financed by CONICYT- Chile, under the Basal Financing Program FB0809 “Advanced Mining Technology Center”, FONDECYT Project 1150488, “Fractional Error Models in Adaptive Control and Applications”, and FONDECYT 3150007, “Postdoctoral Program 2015”.

[1] M. Ramasamy, S. Narayanan, Ch. Rao, Control of ball mill grinding circuit using model predictive control scheme, J. Process Control 15 (2005) 273 – 283.

[2] X. Chen, J. Zhai, S. Li, Q. Li, Application of model predictive control in ball mill grinding circuit, Minerals Eng. 20 (2007) 1099 – 1108.

[3] X. Chen, Q. Li, S. Fei, Constrained model predictive control in ball mill grinding process, Powder Tech. 186 (2008) 31 – 39.

[4] D. Wei, I.K. Craig, Economic performance assessment of two ROM ore milling circuit controllers, Minerals Eng. 22 (2009) 826 – 839.

[5] L.C. Coetzee, I.K. Craig, E. Kerrigan, Robust nonlinear model predictive control of a run-of-mine ore milling circuit, IEEE Trans. Control Systems Tech. 18 (2010) 222 – 229.

[6] J.D. Le Roux, R. Padhi, I.K. Craig, Optimal control of grinding mill circuit using model predictive static programming: A new nonlinear MPC paradigm, J. Process Control 24 (2014) 29 – 40.

[7] D.G. Hulbert, I.K. Craig, M.L. Coetzee, D. Tudor, Multivariable control of a run-of-mine milling circuit, J. South African Inst. Mining and Metallurgy 90 (1990) 173 – 181.

[8] M.A. Duarte-Mermoud, F. Sepúlveda, A. Castillo, A. Contreras, V. Lazcano, P. Giménez, L. Castelli, A comparative experimental study of five multivariable control strategies applied to a grinding plant, Powder Tech. 104 (1999) 1 – 28.

[9] M.A. Duarte-Mermoud, A. Castillo, F. Sepúlveda, A. Contreras, P. Giménez, L. Castelli, Multivariable control of grinding plants: A comparative simulation study, ISA Trans. 41 (2002) 57 – 79.

[10] M.A. Duarte-Mermoud, A. Suárez, D. Bassi, Control of grinding plants using predictive multivariable neural control, Powder Tech. 115 (2001) 193 – 206.

[11] A. Pomerleau, D. Hodouin, A. Desbiens, E. Gagnon, A survey of grinding circuit control methods: from decentralized PID controllers to multivariable predictive controllers, Powder Tech. 108 (2000) 103 – 115.

[12] D. Wei, I.K. Craig, Grinding mill circuits - A survey of control and economic concerns, Int. J. Miner. Process. 90 (2009) 56 – 66.

[13] I.K. Craig, C. Aldrich, R. Braatz, F. Cuzzola, E. Domlan, S. Engell, J. Hahn, V. Havlena, A. Horch, B. Huang, M. Khanbaghi, A. Konstantellos, W. Marquardt, T. McAvoy, T. Parisini, S. Pistikopoulos, T. Samad, S. Skogestad, N. Thornhill, J. Yu, The impact of control technology: Control in the process industries, www.ieeecss.org (Last accessed on 2016-06-28).

[14] X. Chen, S. Li, J. Zhai, Q. Li, Expert system based adaptive dynamic matrix control for ball mill grinding circuit, Expert Systems Applications 36 (2009) 716 – 723.

[15] X. Chen, J. Yang, S. Li, Q. Li, Disturbance observer based multi-variable control of ball mill grinding circuits, J. Process Control 19 (2009) 1205 – 1213.

[16] J. Yang, S. Li, X. Chen, Q. Li, Disturbance rejection of ball mill grinding circuits using DOB and MPC, Powder Tech. 198 (2010) 219 – 228.

[17] L.E. Olivier, I.K. Craig, Y.Q. Chen, Fractional order and BICO disturbance observers for a run-of-mine ore milling circuit, J. Process Control 22 (2012) 3 – 10.

[18] L.E. Olivier, I.K. Craig, Model-plant mismatch detection and model update for a run-of-mine ore milling circuit under model predictive control, J. Process Control 23 (2013) 100 – 107.

[19] J.D. Le Roux, L.E. Olivier, M.A. Naidoo, R. Padhi, I.K. Craig, Throughput and product quality control for a grinding mill circuit using non-linear MPC, J. Process Control 42 (2016) 35 – 50.

[20] A. Kilbas, H. Srivastava, J. Trujillo, Theory and applications of fractional differential equations, Elsevier, 2006.

[21] N. Aguila-Camacho, M.A. Duarte-Mermoud, Fractional adaptive control for an automatic voltage regulator, ISA Trans. 52 (2013) 807–815.

[22] I. Petrás, Tunning and implementation methods for fractional-order controllers, Fractional Calculus and Applied Analysis 15 (2012) 282–303.

[23] B.M. Vinagre, I. Petrás, I. Podlubny, Y.Q. Chen, Using fractional order adjustment rules and fractional order reference models in model-reference adaptive control, Nonlinear Dynamics 29 (2002) 269 – 279.

[24] I. Tejado, S.H. HosseinNia, B.M. Vinagre, Adaptive gain-order fractional control for network-based applications, Fractional Calculus and Applied Analysis 17 (2014) 462–482.

[25] J. Suárez, B.M. Vinagre, Y.Q. Chen, A fractional adaptation scheme for lateral control of an AGV, J. Vibration and Control 14 (9-10) (2008) 1499–1511.

[26] X. Song, Y.Q. Chen, I. Tejado, B.M. Vinagre, Multivariable fractional order PID controller design via LMI approach, in: 18th IFAC World Congress, Milano, Italy, 2011.

[27] M. Moradi, A genetic-multivariable fractional order PID control to multi-input multi-output processes, J. Process Control 24 (2014) 336–343.

[28] C.I. Muresan, E.H. Dulf, C. Ionescu, Robustness evaluation of a multivariable fractional order PI controller for time delay processes, Control and Intelligent Systems 42 (2014) 112–118.

[29] C.I. Muresan, E.H. Dulf, C. Copot, R. De Keyser, C. Ionescu, Design and analysis of a multivariable fractional order controller for a non-minimum phase system, J. Vibration and Control 22 (2016) 2187–2195.

[30] J.D. Le Roux, I.K. Craig, D.G. Hulbert, A.L. Hinde, Analysis and validation of a run-of-mine ore grinding mill circuit model for process control, Minerals Eng. 43-44 (2013) 121–134.

[31] J.D. Le Roux, I.K. Craig, Reducing the number of size classes in a cumulative rates model used for process control of a grinding mill circuit, Powder Tech. 246 (2013) 169–181.

[32] A.L. Hinde, J.T. Kalala, The application of a simplified approach to modelling tumbling mills, stirred media mills and HPGR's, Minerals Eng. 22 (2009) 633–641.

[33] R. Amestica, G.D. Gonzalez, J. Menacho, J. Barria, A mechanistic state equation model for semiautogenous mills, Int. J. Miner. Process. 44-45 (1996) 349–360.

[34] M. Powell, A. van der Westhuizen, A. Mainza, Applying grindcurves to mill operation and optimisation, Minerals Eng. 22 (2009) 625–632.

[35] I.S. Podlubny, Fractional Differential Equations, Academic Press, San Diego, CA, 1999.

[36] D. Matignon, Stability results on fractional differential equations with applications to control processing, in: Computational Engineering in Systems Applications. IMACS, IEEE-SMC, Lille, France, 2 (1996) 963–968.

[37] P. Shah, S. Agashe, Review of fractional PID controller, Mechatronics 38 (2016) 29–41.

[38] L.C. Coetzee, Robust Nonlinear Model Predictive Control of a Closed Run-of-Mine Ore Milling Circuit, PhD. thesis, University of Pretoria, 2009.

[39] K.S. Narendra, A.M. Annaswamy, Stable Adaptive Systems, Dover Publications, Mineola, N.Y., 2005.

[40] N. Aguila-Camacho, M.A. Duarte-Mermoud, Boundedness of the solutions for certain classes of fractional differential equations with application to adaptive systems, ISA Transactions 60 (2016) 82–88.

[41] S. Ladaci, S. Loiseau, A. Charef, Fractional order adaptive high-gain controllers for a class of linear systems, Communications in Nonlinear Science and Numerical Simulation 13 (2008) 707–714.

[42] S. Ladaci, S. Loiseau, A. Charef, Using fractional order filter in adaptive control of noisy plants, in: Third International Conference on Advances in Mechanical Engineering and Mechanics. Hammamet, Tunisia, 2006.

[43] A. Oustaloup, La commande CRONE: commande robuste d'ordre non entier, Hermes, Paris, 1991.

[44] D. Valerio, J.S. Da Costa, Ninteger: a non-integer control toolbox for Matlab, in: Fractional Derivatives and Applications. IFAC, Bordeaux, France, 2004.

[45] J. Kennedy, R. Eberhart, Particle Swarm Optimization, in: IEEE International Conference on Neural Networks. Perth, Australia, 1995.

[46] R.H. Ordóñez-Hurtado, Aplicación de la técnica PSO a la determinación de funciones de Lyapunov cuadráticas comunes y a sistemas adaptables basados en modelos de error, PhD. thesis, University of Chile, 2012.

[47] R.H. Ordóñez-Hurtado, M.A. Duarte-Mermoud, Finding common quadratic Lyapunov functions for switched linear systems using particle swarm optimization, International Journal of Control 85 (2012) 12–25.

[48] T. BhasKaran, YQ. Chen, D. Xue, Practical tunning of fractional order proportional and integral controller (1): Tunning rule development, in: ASME 2007 International Design Engineering Technical Conferences and Computers Information in Engineering Conference, IDETC/CIE 2007, 2007.

[49] L. Ljung, L. Ljung, System Identification Theory for the User, Prentice Hall, Englewood Cliffs, NJ., 1987.

[50] Constrained Particle Swarm Optimization, http://www.mathworks.com/matlabcentral/fileexchange/25986-constrained-particle-swarm-optimization (January of 2017).

[51] S.J. Qin, T.A. Badgwell, A survey of industrial model predictive control technology, Control Eng. Practice 11 733–764.

[52] M.L. Darby, M. Nikolaou, MPC: Current practice and challenges, Control Eng. Practice 20 2012 328–342.

[53] T.A. Apelt, N. Thornhill, Inferential measurement of SAG mill parameters V: MPC simulation, Miner. Eng. 22 2009 1045–1052.

[54] J. Salazar, H. Valdez-Gonzalres, E. Vyhmesiter, F. Cubillos Model predictive control of semi-autogenous mills, Miner. Eng. 64 2009 92–96.

[55] D. Wei, I.K. Craig, Grinding mill circuits a survey of control and economic concerns, Int. J. Miner. Process. 90 (2009) 56–66.

[56] J.A. Herbst, W.T. Pate, Object components for comminution system softsensor design, Powder Tech. 105 (1999) 424–429.

[57] T.A. Apelt, S.P. Asprey, N.F. Thornhill, Inferential measurement of SAG mill parameters. II: State estimation, Miner. Eng. 15 (2002) 1043–1053.

[58] L.E. Olivier, B. Huang, I.K. Craig, Dual particle filters for state and parameter estimation with application to a run-of-mine ore mill, J. Process Control 22 (2012) 710–717.

[59] J.D. le Roux, A. Steinboeck, A. Kugi, I.K. Craig, An EKF observer to estimate semi-autogenous grinding mill hold-ups, J. Process Control 51 (2017) 27–41.