European Journal of Control 51 (2020) 122–134

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.elsevier.com/locate/ejcon

Fractional order controllers for throughput and product quality control in a grinding mill circuit

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

[a] Department of Electricity, Faculty of Engineering, Universidad Tecnológica Metropolitana, Av. José Pedro Alessandri 1242, Santiago de Chile, Chile
[b] Department of Electrical Engineering, Faculty of Mathematical and Physical Sciences, University of Chile, Av. Tupper 2007, Santiago de Chile, Chile
[c] Advanced Mining Technology Center, Av. Tupper 2007, Santiago de Chile, Chile

ARTICLE INFO

Article history:
Received 24 May 2018
Revised 10 September 2018
Accepted 8 August 2019
Available online 13 August 2019

Recommended by T. Parisini

Keywords:
Fractional order multivariate controllers
Grinding mill circuit
Optimization
Process control
Comminution

ABSTRACT

This paper presents the design and application of a multiple-input–multiple-output fractional order proportional-integral (MIMO FOPI) controller to a grinding mill circuit. The MIMO FOPI controller parameters are tuned using an off-line optimization process based on Particle Swarm Optimization (PSO). Its performance is compared to a single-input–single-output fractional order proportional-integral (SISO FOPI) controller designed and tuned using the same procedure based on PSO. The results show that the MIMO FOPI achieves better results compared to the SISO FOPI controller in most of the cases simulated, even in the presence of hardness and composition variations in the processed ore, and also in the presence of process noise.

1. Introduction

The coupling and interaction between process variables, together with nonlinearities, disturbances, time delays and unmodeled dynamics, make grinding mill circuits difficult processes to control. For that reason, multivariate control techniques have been proposed and successfully implemented for grinding processes, improving their performance compared to the cases when decentralized SISO controllers are used. We can mention for instance the application of Model Predictive Control (MPC) to grinding circuits, with a large number of works reported in literature. See for instance [3,4,6,25,29,36]. Besides MPC, other multivariate control techniques have been proposed for grinding circuits as well, such as direct Nyquist array [9,10,12], extended horizon [9,10], pole placement [9,10], multivariate model reference adaptive control [9,10], sequential loop closing [9,10], and predictive multivariate neural control [11].

Although these multivariate techniques can significantly improve process performance compared to decentralized SISO controllers [24,36], still a great percentage of mining industries use proportional integral and derivative (PID) controllers (or only PIs) in their milling circuits. This is mainly due to the difficulty of implementing and maintaining advanced process control like nonlinear MPC, and the lack of sufficient dynamic and fundamental models for mineral processing circuits [8].

On the other hand, the use of fractional order operators [15] in the design of controllers have gained considerable attention in the last two decades. Several control techniques have been extended to the fractional case, studied and applied to the control of several process. We can mention, just to cite a few works, Fractional Order Proportional Integral control (FOPI) [22], Fractional Order Model Reference Adaptive Control (FOMRAC) [35], adaptive gain-order fractional order control [33], among others. Although the application of these fractional control techniques is found mainly in SISO processes [1,32], some were proposed for MIMO processes [17–19,31]. Due to the introduction of fractional operators in these control strategies, important advantages have been reported, such as improvements on system robustness in the presence of disturbances, noisy environments, time-varying parameters, and also in the management of the control energy as stated by Aguila-Camacho and Duarte-Mermoud [1]. For that reason, in Aguila-Camacho et al. [2] it is proposed the design and application of a SISO FOPI controller and a SISO Model Reference Adaptive Controller to a grinding mill circuit, obtaining similar or better results than a Lineal Model Predictive Controller (LMPC) when external disturbances and process noise were present in the circuit. The work by Aguila-Camacho et al. [2], together with the work by Olivier et al. [20], are the only two attempts (to the best of the

* Corresponding author.
E-mail addresses: norelys.aguila@utem.cl (N. Aguila-Camacho), mduartem@ing.uchile.cl (M.A. Duarte-Mermoud), morchard@ing.uchile.cl (M.E. Orchard).

https://doi.org/10.1016/j.ejcon.2019.08.002
0947-3580/

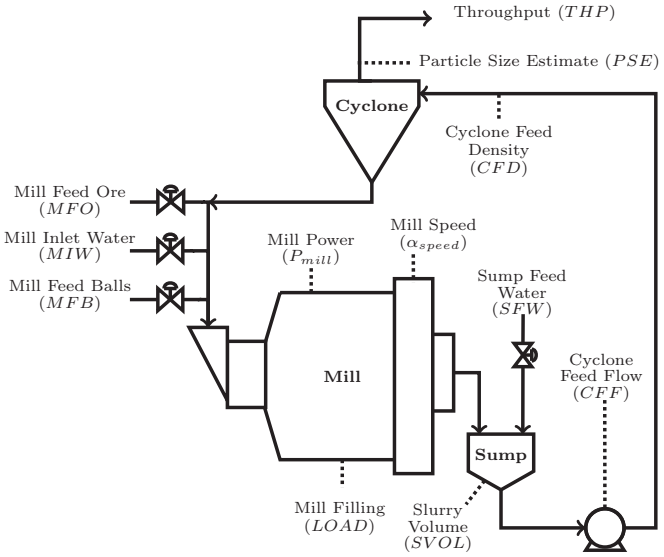

Fig. 1. Single-stage closed run-of-mine ore milling circuit.

authors knowledge) using fractional operators in the control of grinding mill circuits.

Despite integer order and fractional order controllers have been successfully applied to the control of grinding mill circuits, no attempt is made on them to control the throughput of the grinding circuit, but only the product quality. The exception would be the work by Le Roux et al. [28], where an independent control of product quality and throughput is proposed, using non-linear MPC. Since the main objective of the milling circuit is to grind mined ore into a fine product in order to extract the most minerals of it in the downstream process, and increasing the processed material with the correct quality standards leads to an increase in productivity, thus we should be looking at both, product quality and throughput. However, as stated in [28], the finer product of the comminution circuit comes at the cost of a reduced throughput, thus controlling both quality and throughput is desirable.

Precisely, this paper proposes a MIMO FOPI controller for a grinding mill circuit, where both product quality and throughput of the circuit are controlled. The controller aims to preserve the simplicity of the fractional PI and, at the same time, to manage the variables couplings and interactions using a multivariate design, also with the advantages that the fractional operators can add to the controller performance.

The paper is organized as follows: Section 2 describes the grinding mill circuit and Section 3 the corresponding nonlinear model used for simulation. Section 4 introduces the design of the proposed MIMO FOPI controller and Section 5 the tuning of the controller parameters. Section 6 presents the results obtained from simulations and in Section 7 main conclusions are drawn.

2. Process description

The single-stage closed run-of-mine (ROM) ore milling circuit shown in Fig. 1 is considered in this study, and it 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*, and it is assumed in this study that a variable speed drive is fitted on the mill motor, which can be used to manipulate the mill speed, specifically through the fraction of critical mill speed (α_{speed}). The ground ore in the mill mixes with water to form a slurry, which 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.

Table 1
Manipulated and controlled variables.

Variable	Description	Value	Unit
Manipulated variables			
CFF	Cyclone feed flow-rate	374	[m^3/h]
MFO	Mill feed-rate of ore	65.2	[t/h]
SFW	Sump feed water flow-rate	140.5	[m^3/h]
α_{speed}	Fraction of critical mill speed	0.72	[Fraction]
MIW	Mill inlet water flow-rate	4.64	[m^3/h]
MFB	Mill feed-rate of balls	5.68	[t/h]
Controlled variables			
PSE	Product particle size	67	[% < 75 μm]
LOAD	Fraction of mill filled	33	[%]
SVOL	Slurry volume in the sump	11.8	[m^3]
THP	Volumetric throughput of solids	20.375	[m^3]

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 (*THP*) 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, together with their values at the operating point.

3. Model description

The continuous time dynamic phenomenological nonlinear population balance model validated by Le Roux et al. [27] is used in this study to describe the circuit shown in Fig. 1. For the sake of space, a detailed description of the model is not presented in this paper, but the reader can found it in Le Roux et al. [27]. Also, in Aguila-Camacho et al. [2] the model is clearly specified, even in the same operating point used in this paper. Note that the process description and the model description are very similar to that founded in [2,28], since here it is used the same process and the same model.

In general terms, each process unit in the circuit of Fig. 1 is modelled separately. The model is suitable for control purposes as it uses as few parameters and states as 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 it is larger than the specification size. Fine ore also passes through the mill discharge grate but it 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 [26].

Table 2
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

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}). The parameter values of the model used in this paper can be seen in Table 2.

The operating condition in Table 1 and the model parameter values in Table 2 were taken from the controller comparison study of Le Roux et al. [29] and were also used in Aguila-Camacho et al. [2].

The manipulated variables specified in Table 1 have the following limits:

$$u_{lower} = [0\ 0\ 0\ 0\ 100\ 0.7]^T$$
$$u_{upper} = [80\ 100\ 10\ 400\ 450\ 0.85]^T \quad (1)$$

where the vector u is defined as $u = [MIW\ MFO\ MFB\ SFW\ CFF\ \alpha_{speed}]^T$, the u_{lower} represents the lower bound, and u_{upper} represents the upper bound. Furthermore, the sump has a maximum capacity of 16 m^3.

4. Controller design

This section presents the design of a MIMO FOPI controller for the grinding mill circuit. Some basic definitions of fractional calculus are introduced first in Section 4.1, since they are used in the controllers design. Also, the technical specifications of the approximation used to implement the fractional operators in simulations are presented in Section 4.2. Finally, Section 4.3 presents the controller design itself.

4.1. Basic concepts of fractional calculus

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. [15], the Riemann–Liouville fractional integral of order $\alpha \in \mathbb{R}$, with $\alpha \geq 0$, is defined as:

$$I^{\alpha}_{t_0} f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^{t} \frac{f(\tau)}{(t-\tau)^{1-\alpha}} d\tau, \quad t > t_0, \quad (2)$$

where $\Gamma(\alpha)$ is the Gamma function [15].

Regarding the fractional derivative of order $\beta \geq 0$, there exist several definitions. In this work, the Caputo definition [15] presented in (3) will be used, since it is one commonly used in engineering applications. The Caputo fractional derivative is defined as

$${}^{C}D^{\beta}_{t_0} f(t) = \frac{1}{\Gamma(n-\beta)} \int_{t_0}^{t} \frac{f^{(n)}(\tau)}{(t-\tau)^{\beta-n+1}} d\tau, \quad t > t_0, \quad (3)$$

where $f \in AC^n[t_0, t]$, $n-1 < \beta < n$, $n \in \mathbb{Z}^+$. If $\beta \in \mathbb{Z}^+$, then $n = \beta$.

The Laplace transform of the Riemann–Liouville fractional integral (2) corresponds to

$$\mathcal{L}\left\{ I^{\alpha}_{t_0} f(t) \right\} = s^{-\alpha} F(s), \quad (4)$$

while the Laplace transform of the Caputo fractional derivative (3), according to Podlubny [23] is

$$\mathcal{L}\left\{ {}^{C}D^{\beta}_{t_0} f(t) \right\} = s^{\beta} F(s) - \sum_{k=0}^{n-1} s^{n-k-1}\ f^{(k)}(t_0), \quad (5)$$

where the initial conditions of the function and its integer order derivatives appear.

4.2. Numerical approximation for implementing fractional integrals and derivatives

Fractional integrals and derivatives are commonly implemented in simulations and practical applications by means of numerical

approximations of these operators. Using these approximations, equivalent integer-order transfer functions are obtained, whose behavior approximate the fractional order Laplace operator and are easily implemented. In this paper, the Oustaloup's, method [21] is used to approximate the Laplace transform of the fractional operator, that is

$$C(s) = ks^{\alpha} \tag{6}$$

which is approximated using a recursive distribution of N poles and N zeros of the form:

$$C(s) = k' \prod_{n=1}^{N} \frac{1 + s/\omega_{zn}}{1 + s/\omega_{pn}}. \tag{7}$$

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 [34] 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.

4.3. Design of MIMO FOPI controller

The transfer function of a SISO FOPID controller is represented as:

$$G_c(s) = K_P + \frac{K_I}{s^{\alpha}} + K_D\, s^{\beta}, \tag{8}$$

where K_P is the proportional gain, K_I is the integral gain, K_D is the derivative gain, and α, $\beta \in (0, 2)$ are the order of the integral and the order of the derivative, respectively, considered in this interval for stability purposes [16]. When $\alpha = \beta = 1$, then (8) represents the classic PID controller.

Why is a good idea considering FOPID in control systems is a question that has its answer in many reported advantages of this controllers above classic PID. For instance, in Shah and Agashe [30] it is reported that using FOPIDs, five degrees of freedom are available for the controller design, while only three degrees of freedom are available for the integer order PID controller design, which implies that more design criteria can be satisfied using a FOPID controller than using a classic PID controller.

Also, in Shah and Agashe [30] it has been reported that for higher order systems, the performance of PID controller deteriorates, whereas FOPID controller can provide better results; and for a system with long time delays, FOPID controllers can provide better results than PID controllers. We can also find evidence of a better robust stability for FOPID compared to PID controllers, as well as a better performance for systems with nonlinearities and for non-minimum phase systems.

Based on these facts, SISO FOPI controllers were proposed and studied in the work by Aguila-Camacho et al. [2] for the grinding mill circuit studied in this paper, obtaining a good behavior compared to SISO Fractional Order Model Reference Adaptive Controllers (SISO FOMRAC) as well as to a LMPC controller, which is a MIMO control technique. However, in the work by Aguila-Camacho et al. [2], no attempt is made to control *THP*, which implies that the productivity of the circuit was determined by the amount of material that the mill can handle, using a constant speed, and influenced by the desired size of the particles (*PSE*) at the circuit output.

On the other hand, in this paper *THP* is also included as a controlled variable, and the mill speed is introduced as an additional manipulated variable. Also, the proposed controller is MIMO, expecting a lower effect of the coupling between variables in the control scheme, compared to SISO FOPI controllers.

Thus, a MIMO FOPI controller is proposed in this work, with a transfer matrix given by

$$\begin{bmatrix} \Delta CFF \\ \Delta MFO \\ \Delta SFW \\ \Delta\alpha_{speed} \end{bmatrix} = \begin{bmatrix} G_{c_{11}}(s) & G_{c_{12}}(s) & G_{c_{13}}(s) & G_{c_{14}}(s) \\ G_{c_{21}}(s) & G_{c_{22}}(s) & G_{c_{23}}(s) & G_{c_{24}}(s) \\ G_{c_{31}}(s) & G_{c_{32}}(s) & G_{c_{33}}(s) & G_{c_{34}}(s) \\ G_{c_{41}}(s) & G_{c_{42}}(s) & G_{c_{43}}(s) & G_{c_{44}}(s) \end{bmatrix} \begin{bmatrix} e_{PSE} \\ e_{LOAD} \\ e_{SVOL} \\ e_{THP} \end{bmatrix} \tag{9}$$

where $G_{c_{ij}}(s) = k_{P_{ij}} + \frac{k_{I_{ij}}}{s^{\alpha_{ij}}}$, $\forall i, j : 1 \ldots 4$, with $k_{P_{ij}}$, $k_{I_{ij}}$ and α_{ij} accordingly with the description made in (8). These controller parameters will be tuned and explained in Section 5.

The variable e_{PSE} corresponds to the error between the desired set point and the actual value of *PSE*, e_{LOAD} is the error between the desired set point and the actual value of *LOAD*, e_{SVOL} is the error between the desired set point of the sump volume *SVOL* and its actual value, and e_{THP} is the error between the desired set point and the actual value of *THP*. On the other hand, ΔCFF, ΔMFO, ΔSFW and $\Delta\alpha_{speed}$ correspond to the variation of *CFF*, *MFO*, *SFW* and α_{speed} around the operating point, respectively.

In the case of the two additional manipulated variables of this circuit, *MFB* and *MIW*, they are considered as in [2,29]:

- *MFB* is kept as a constant ratio of 16.7 of the mill filling *LOAD*.
- *MIW* is kept as a constant ratio of 7% of *MFO*. In that way, if *MFO* increases, *MIW* also increases to help push the extra material through the circuit and to ensure the density of the charge inside the mill does not go too high.

Fig. 2 shows the general diagram of the controlled system. Note that in Fig. 2, variables *PSE_SP*, *LOAD_SP*, *SVOL_SP* and *THP_SP* correspond to the desired set point values for *PSE*, *LOAD*, *SVOL* and *THP*, respectively.

5. Controller tuning

This section presents the tuning procedure used for the MIMO FOPI controller parameters. Off-line Particle Swarm Optimization (PSO) is used to select the controller parameters. Besides, a SISO FOPI controller is also tuned, which will be used for comparison purposes.

5.1. Particle swarm optimization basics

For tuning the controller parameters, an optimization procedure is carried out. PSO is the optimization technique used in this paper, which is an heuristic global optimization technique that belongs to the category of swarm intelligence.

As an heuristic technique, PSO has the advantage of being able to drive the most versatile 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, which is reflected in the increase of successful applications based on PSO [14].

The algorithm defines a population where every possible solution of the optimization problem 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

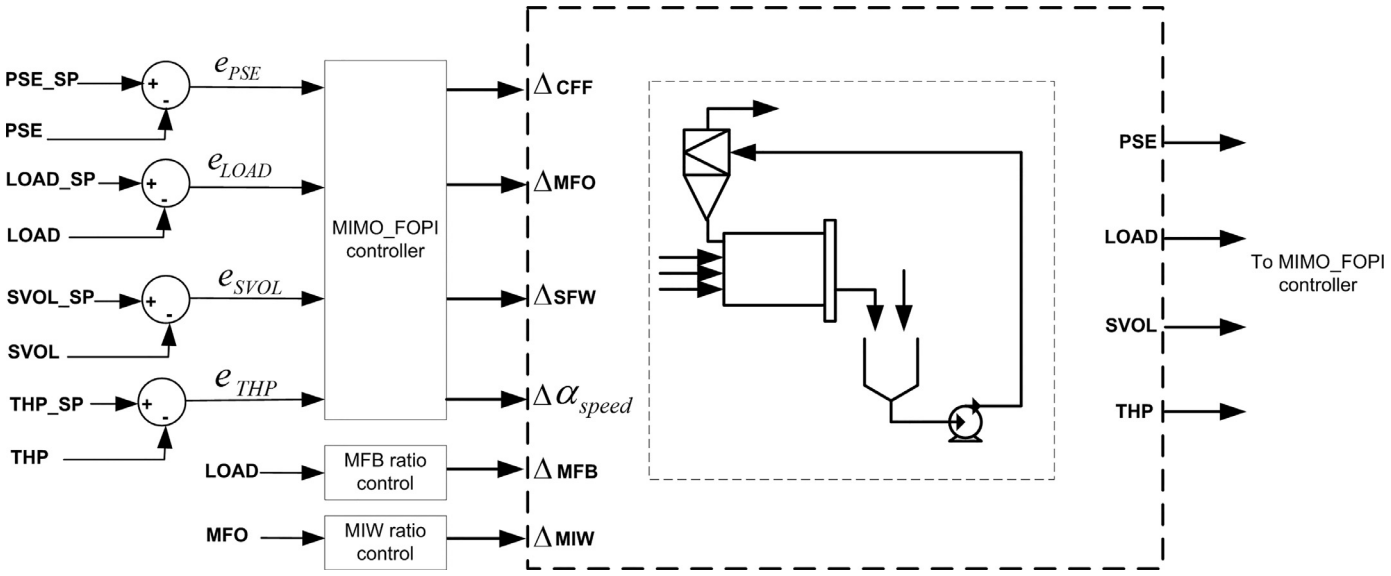

Fig. 2. General diagram of the MIMO FOPI controller configuration.

function). The evolution of the particles is given by:

$$\begin{aligned} v_i(k+1) &= \omega\, v_i(k) + r_1(k)c_1[p_i(k) - x_i(k)] \\ &\quad + r_2(k)c_2\big(g(k) - x_{i,d}(k)\big) \\ x_i(k+1) &= x_i(k) + v_i(k+1), \\ p_i(k) &= \begin{cases} p_i(k-1) & if \quad f(p_i(k-1)) \le f(x_i(k)) \\ x_i(k) & if \quad f(p_i(k-1)) > f(x_i(k)) \end{cases} \\ g(k) &= argmin\{f(p_1(k)), \ldots, f(p_s(k))\} \end{aligned} \tag{10}$$

where $v_i(k)$ and $x_i(k)$ represent the speed and the position of the ith particle at the kth 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, $x_{i,d}(k)$ is the best global position founded at the kth iteration, f is the fitness function to be optimized, and s is the number of particles in the swarm [13].

5.2. Tuning of MIMO FOPI controller parameters using PSO

As the reader may check from Section 4.3, according to the proposed structure of the MIMO FOPI controller, 48 parameters need to be tuned in this optimization stage. Nevertheless, it must be noted that, although the optimization process involves 48 tuning parameters, it can result that optimal values for some of these parameters are zero, leading to a MIMO controller not containing FOPI controllers for every position of the transfer matrix (9), but proportional controllers, pure fractional integral controllers, or even no controller at all.

The cost function used in the optimization process was selected including the control errors and the control energy, as:

$$J = \int_{t_0}^{T} \left[\begin{array}{l} w_1 e_{PSE}^2(t) + w_2 e_{LOAD}^2(t) + w_3 e_{SVOL}^2(t) + w_4 e_{THP}^2(t) + \\ + w_5 \Delta CFF^2(t) + w_6 \Delta MFO^2(t) + w_7 \Delta SFW^2(t) + w_8 \Delta \alpha_{speed}^2(t) \end{array} \right] dt \tag{11}$$

where T is the simulation time and w_i, $i = 1, \ldots, 8$ are weighting factors to give more or less importance to every term in the cost function.

The weighting factors were selected based on some of the criteria used in Le Roux et al. [28]. The weighting factors w_1, w_2, w_3 and w_4 were determined such that a 0.5% deviation from the *THP* set-point will produce an error in the cost function equal to a 1% deviation from *PSE* set-point, equal to a 5% change from *LOAD* set-point and equal to a 20% change from *SVOL* set point. Thus

$$\begin{aligned} w_1(1\%PSE\_SP)^2 &= w_2(5\%LOAD\_SP)^2 = w_3(20\%SVOL\_SP)^2 \\ &= w_4(0.5\%THP\_SP)^2. \end{aligned} \tag{12}$$

If we choose $w_2 = 1$, based on the operating condition shown in Table 1 and using (12), the controlled variable weighting factors result i.e.

$$w_1 = 6.0648 \quad w_2 = 1 \quad w_3 = 0.488 \quad w_4 = 261.77 \tag{13}$$

On the other hand, the weighting factors for the manipulated variables were determined such that a 1% change of the half ranges of *CFF* and *SFW* will produce the same error as a 2% change of the half range of *MFO* and as a 15% change of the half range of α_{spped}, in the cost function. Also, the corresponding weighting factors were scaled to produce 1% of the error compared to the weighting factors of the corresponding controlled variables i.e.

$$100\, w_6 \left(\frac{2\%MFO_{range}}{2} \right)^2 = w_2 (5\%LOAD_{SP})^2$$

and

$$\begin{aligned} w_5 \left(\frac{1\%CFF_{range}}{2} \right)^2 &= w_6 \left(\frac{2\%MFO_{range}}{2} \right)^2 = w_7 \left(\frac{1\%SFW_{range}}{2} \right)^2 \\ &= w_8 \left(\frac{15\%\alpha_{speed_{range}}}{2} \right)^2. \end{aligned}$$

Therefore, the manipulated variable weighting factors are:

$$\begin{aligned} w_5 &= 8.8898 \times 10^{-7} \quad w_6 = 2.7225 \times 10^{-6} \\ w_7 &= 6.8063 \times 10^{-7} \quad w_8 = 0.0215 \end{aligned} \tag{14}$$

For the optimization process, the range of search for those parameters corresponding to controller gains was set as $[0, 10^8]$, which guarantees a wide searching space for the optimization process.

According to Matignon [16], stability cannot be guaranteed for any linear time invariant system if the fractional order is equal or higher than 2. Although the grinding mill circuit is a nonlinear time varying system, the range of search for those parameters

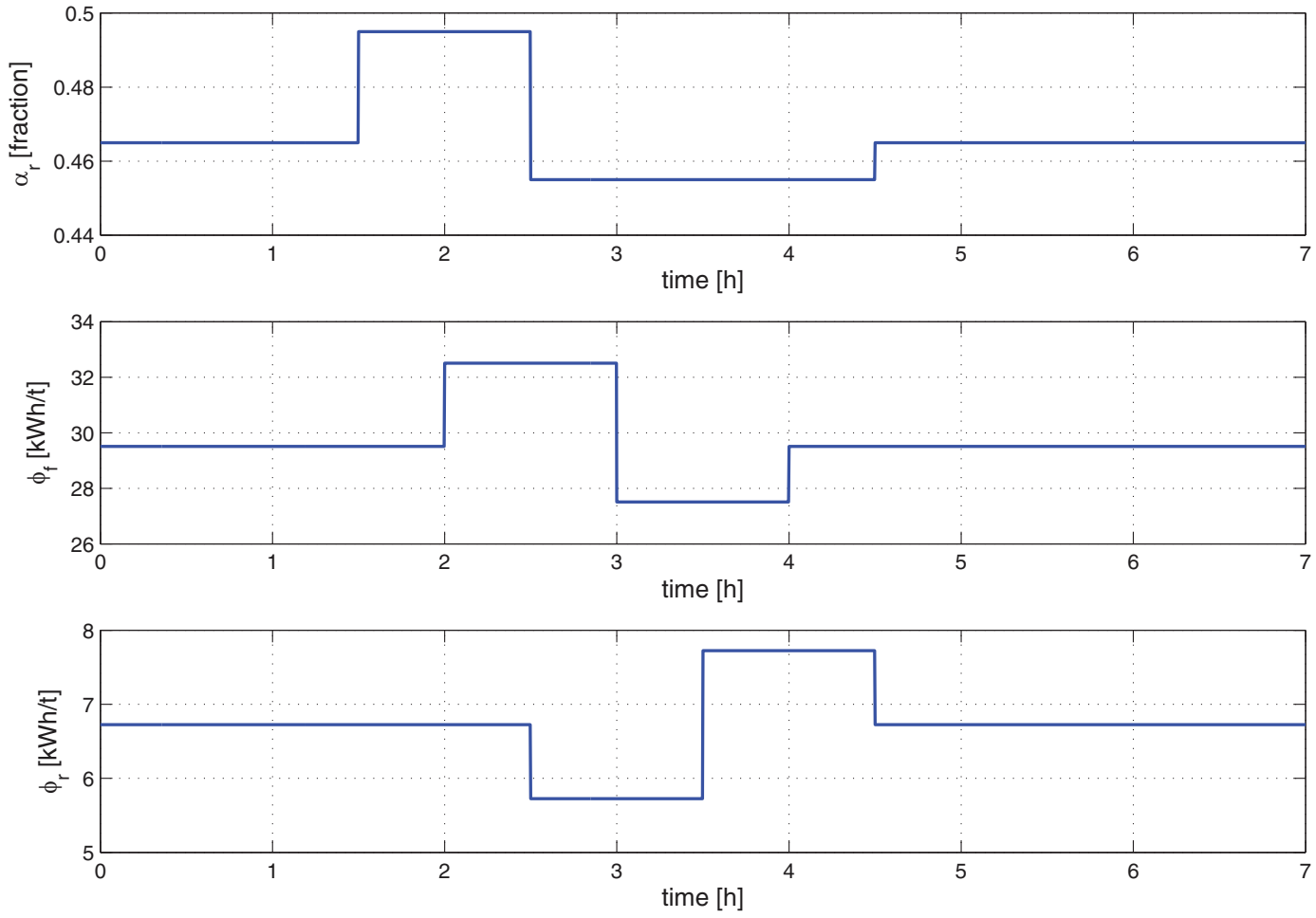

Fig. 3. Parameter variations introduced in the milling circuit in the second and third simulation scenarios.

corresponding to fractional orders was limited to $(0, 2)$, as a conservative bound. Nevertheless, stability issues due to combinations of gains and orders are taken into consideration in the optimization process 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.

As the reader may note, given the lower and upper limits used for the fractional orders, the resulting controllers could be classic PI controllers if the optimal fractional orders found are 1.

With all the parameters previously defined, the optimization process was carried out using the Constrained Particle Swarm Optimization toolbox for Matlab [7]. Most of the PSO parameters were used at their default values, except:

- Population size: 500.
- Number of generations: 500.

For every generation, the milling circuit is simulated in a time window of 6 h under ideal conditions for every particle, that is, no disturbances or noise are present in the scheme. Step references are applied to the four inputs at $t = 1$ h, specifically $PSE\_SP = 72\ \%$, $LOAD\_SP = 34\ \%$, $SVOL\_SP = 13.8\ \text{m}^3$ and $THP\_SP = 22.375\ \text{m}^3$. From the simulation results, cost function (11) 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 500 generations are reached and the optimal controller parameters are found.

As a result of this optimization process, the resulting MIMO FOPI controller detailed in (9) is:

$$\begin{bmatrix} \Delta CFF \\ \Delta MFO \\ \Delta SFW \\ \Delta \alpha_{speed} \end{bmatrix} = \begin{bmatrix} 738.18 + \dfrac{35452}{s^{0.858}} & 0 & 0.011 & 0.010 \\ 0 & 1.31 \times 10^5 + \dfrac{3.42 \times 10^5}{s^{1.9}} & 0.18 & 0.39 \\ 0.009 & 0 & 345.3 + \dfrac{141.19}{s^{0.178}} & 3.96 \\ 0.015 & 0 & 1.76 & 4.99 \times 10^3 + \dfrac{82076}{s^{0.483}} \end{bmatrix} \begin{bmatrix} e_{PSE} \\ e_{LOAD} \\ e_{SVOL} \\ e_{THP} \end{bmatrix} \tag{15}$$

It can be seen from (15) that the resulting MIMO FOPI controller has FOPI controllers on its main diagonal, and the rest of the controller parameters are zero, except for the resulting proportional controllers in the positions $G_{c_{13}}$, $G_{c_{14}}$, $G_{c_{23}}$, $G_{c_{24}}$, $G_{c_{31}}$, $G_{c_{34}}$, $G_{c_{41}}$ and $G_{c_{43}}$. This is already a big difference with the FOPI controllers tunned in [2], where the controllers were SISO and consequently the coupling between variables were not taken into account. Re-

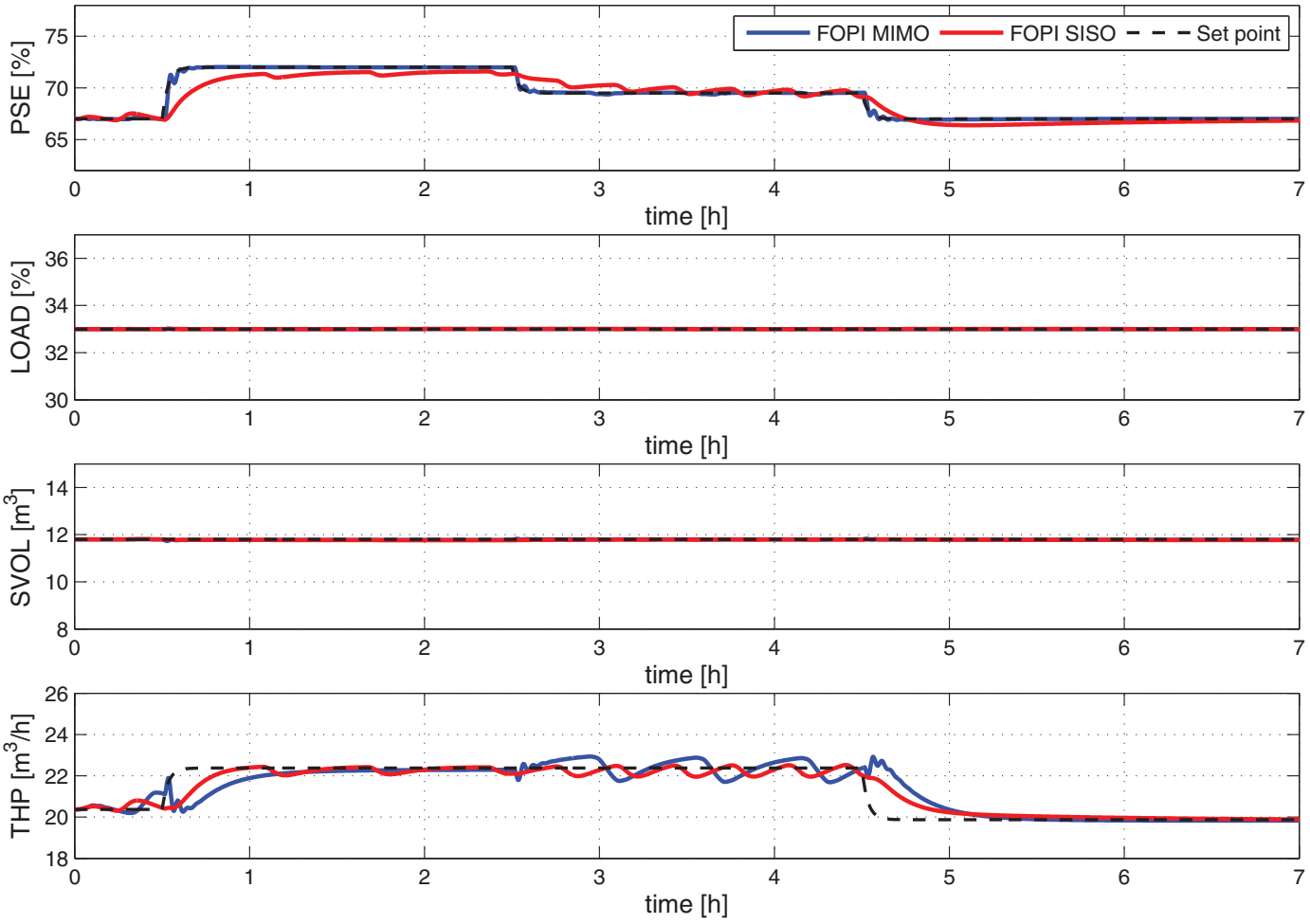

Fig. 4. Controlled variables for first simulation scenario.

garding the order of the FOPI controllers, it can be seen that it is fractional for all cases, and far from the integer order case where $\alpha = 1$.

$$\begin{bmatrix} \Delta CFF \\ \Delta MFO \\ \Delta SFW \\ \Delta\alpha_{speed} \end{bmatrix} = \begin{bmatrix} 100.86 + \dfrac{940.24}{s^{0.865}} & 0 & 0 & 0 \\ 0 & 1.157 \times 10^5 + \dfrac{3.63 \times 10^5}{s^{1.98}} & 0 & 0 \\ 0 & 0 & 206.7 + \dfrac{24.49}{s^{0.182}} & 0 \\ 0 & 0 & 0 & 3.9 \times 10^5 + \dfrac{86090}{s^{0.47}} \end{bmatrix} \begin{bmatrix} e_{PSE} \\ e_{LOAD} \\ e_{SVOL} \\ e_{THP} \end{bmatrix} \tag{16}$$

5.3. Tunning of a SISO FOPI controller for comparison purposes

For comparison purposes, an equivalent SISO FOPI controller was also tuned in this work. In this case, the selection of the controlled-manipulated variable pairing to be used is of great importance, since the process presents important interactions between variables. The pairings $CFF - PSE$, $MFO - LOAD$ and $SFW - SVOL$ are the most appropriate for this circuit, according to [5], as they can improve the robustness to feed disturbances compared to other pairings. Also, the pairing $\alpha_{speed} - THP$ is selected, since according to Le Roux et al. [28], manipulation of α_{speed} allows further control over THP.

Thus, the structure of the SISO FOPI controller can be seen as an special case of the transfer matrix (9), considering all the elements outside the main diagonal being zero. This implies that only 12 parameters need to be tuned for the SISO FOPI controller. The optimization process was carried out under the same conditions used for the MIMO FOPI controller, resulting in the following SISO FOPI controller:

6. Simulations and results

This section presents the simulation results obtained using the proposed controllers in Sections 4 and 5 for the milling circuit, under disturbances and noise conditions.

6.1. Simulation environment

The parameter values and the operating point can be seen in Tables 1 and 2, respectively. The grinding mill circuit was simulated considering a simulation time of 7 h and a sampling rate of 10 s. The following scenarios were simulated:

1. The first simulation scenario corresponds to step changes in the set points of the two most important controlled

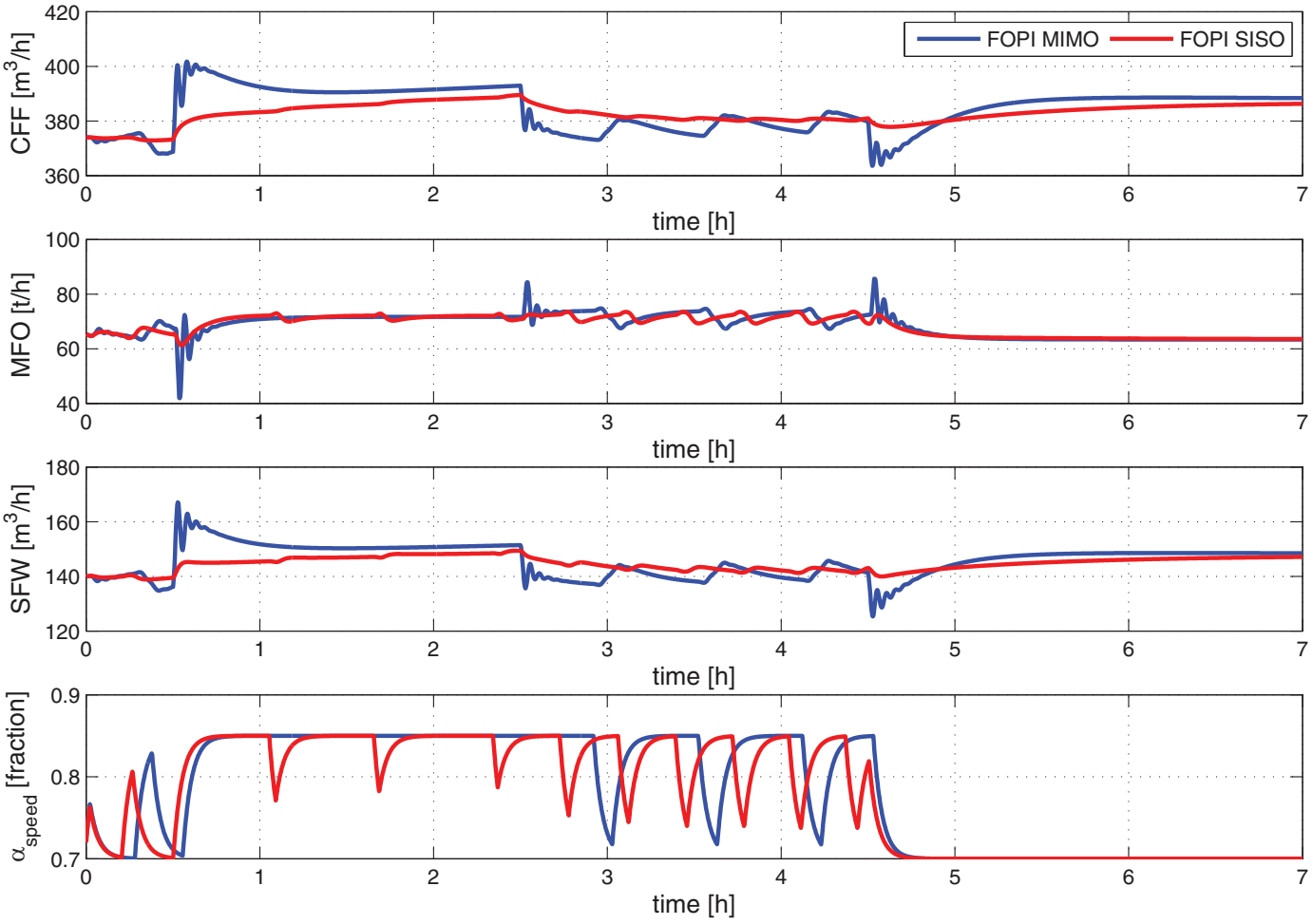

Fig. 5. Manipulated variables for first simulation scenario.

variables in the circuit, that is *PSE* and *THP*. The set point for *PSE* is increased in 0.05% from 67%, at $t = 0.5$ h. Later, it is decreased in 0.025 % at $t = 2.5$ h and finally it is returned to 67% at $t = 4.5$ h. In the case of *THP*, its set point is increased in 2 m^3/h from 20.375 m^3, at $t = 0.5$ h. Later, it is decreased in 2.5 m^3/h at $t = 4.5$ h.

2. In the second simulation scenario, a disturbance in the mill feed size distribution is simulated by increasing α_r from 0.465 to 0.495 at $t = 1.5$ h, decreased to 0.455 at $t = 2.5$ h, and increased back to 0.465 at $t = 4.5$ h. Also, 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 from 29.5 to 32.5 kWh/t at $t = 2$ h, decreased to 27.5 at $t = 3$ h, and back to 29.5 kWh/t at $t = 4$ h. Besides, a disturbance in the energy required to break rocks into solids is simulated by decreasing ϕ_r from 6.72 to 5.72 kWh/t at $t = 2.5$ h, increased to 7.55 at $t = 3.5$ h, and back to 6.72 kWh/t at $t = 4.5$ h. The parameter variations can be seen in Fig. 3.
3. In order to evaluate the noise rejection capabilities of the controllers, in the third scenario, disturbances as specified above in scenario 2 are applied, and process noise is added to the model states [27], which correspond to: mill water hold up (X_{mw}), mill solids hold up (X_{ms}), mill fines hold up (X_{mf}), mill rocks hold up (X_{mr}), mill balls hold up (X_{mb}), sump water hold up (X_{sw}), sump solids hold up (X_{ss}) and sump fines hold up (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. Results and discussion

This section presents the results of the simulations for the three scenarios detailed above, and the corresponding discussion of these results.

6.2.1. Performance functions to evaluate the grinding mill circuit behavior

The behavior of the grinding mill circuit is evaluated through the following three performance functions.

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

$$\text{NRMSE} = \frac{\sqrt{\frac{\sum^N (y - y_{SP})^2}{N}}}{y_{max} - y_{min}}, \tag{17}$$

where y is the controlled variable at instant of time t, 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 over the observation interval, respectively. This performance index checks how far the controlled variables are from their set points over the observation interval, thus checking how well the controller is working.

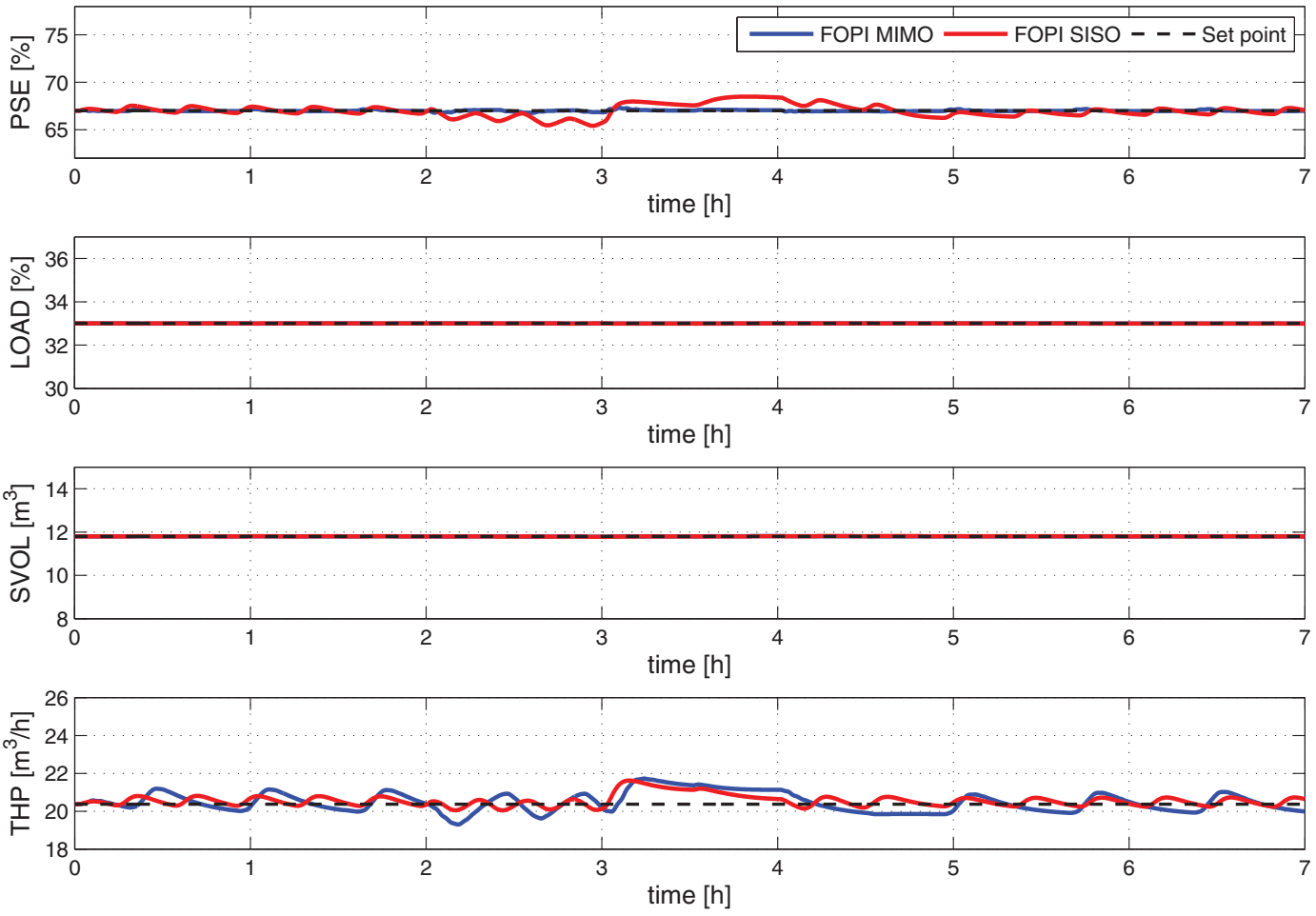

Fig. 6. Controlled variables for second simulation scenario.

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

$$\text{NRMSI} = \frac{\sqrt{\frac{\sum^{N}(u)^2}{N}}}{u_{max} - u_{min}}, \tag{18}$$

where u is the manipulated variable at instant of time t, N is the number of sample data in the observation interval, and $u_{\max}$ and $u_{\min}$ are the maximum and minimum values of u over the observation interval, respectively. This performance index is somehow aimed to check how much energy is being spent in the control.

3. Finally, the third performance index corresponds to the variance of *PSE* given by

$$\sigma_{PSE} = \frac{\sum^{N}|PSE - \mu|^2}{N}, \tag{19}$$

where N is the number of sample data in the observation interval, *PSE* is the value of this variable at the time instant t and μ is the mean value of *PSE*. This performance index is of great relevance, given the importance of reducing small variations in *PSE* in the circuit. Thus, this performance index can be seen as a measure of the quality of the final product in the grinding stage.

Table 3
Values of performance functions for the first simulation scenario.

		MIMO FOPI	SISO FOPI
NRMSE	*PSE*	0.0247	0.1312
	LOAD	0.0882	0.1047
	SVOL	0.1496	0.2281
	THP	0.1975	0.1310
σ_{PSE}		0.0194	0.4929
NRMSI	*CFF*	10.0720	10.0319
	MFO	1.5696	1.5672
	SFW	3.4830	3.4567
	α_{speed}	5.2268	5.2013

6.2.2. Simulation scenario 1

Figs. 4 and 5 show the controlled and manipulated variables, respectively, for the first simulation scenario. It can be seen from Fig. 4 that the *PSE* behaves better for the case of the MIMO FOPI controller, since it remains closer to the set point value than with the SISO FOPI controller. Also, it has less oscillations, which is a desirable feature in the grinding mill circuit. This behavior is in agreement with the NRMSE value for *PSE* and the σ_{PSE} shown in Table 3.

Regarding the *THP*, it can be seen that both controllers achieve the control goal after the first step reference in $t = 0.5$ h. At $t = 2.5$ h, when the set point for *PSE* is decreased, it can be seen that *THP* presents some oscillatory performance, with less oscillations for the MIMO FOPI controller but with less amplitude in the case

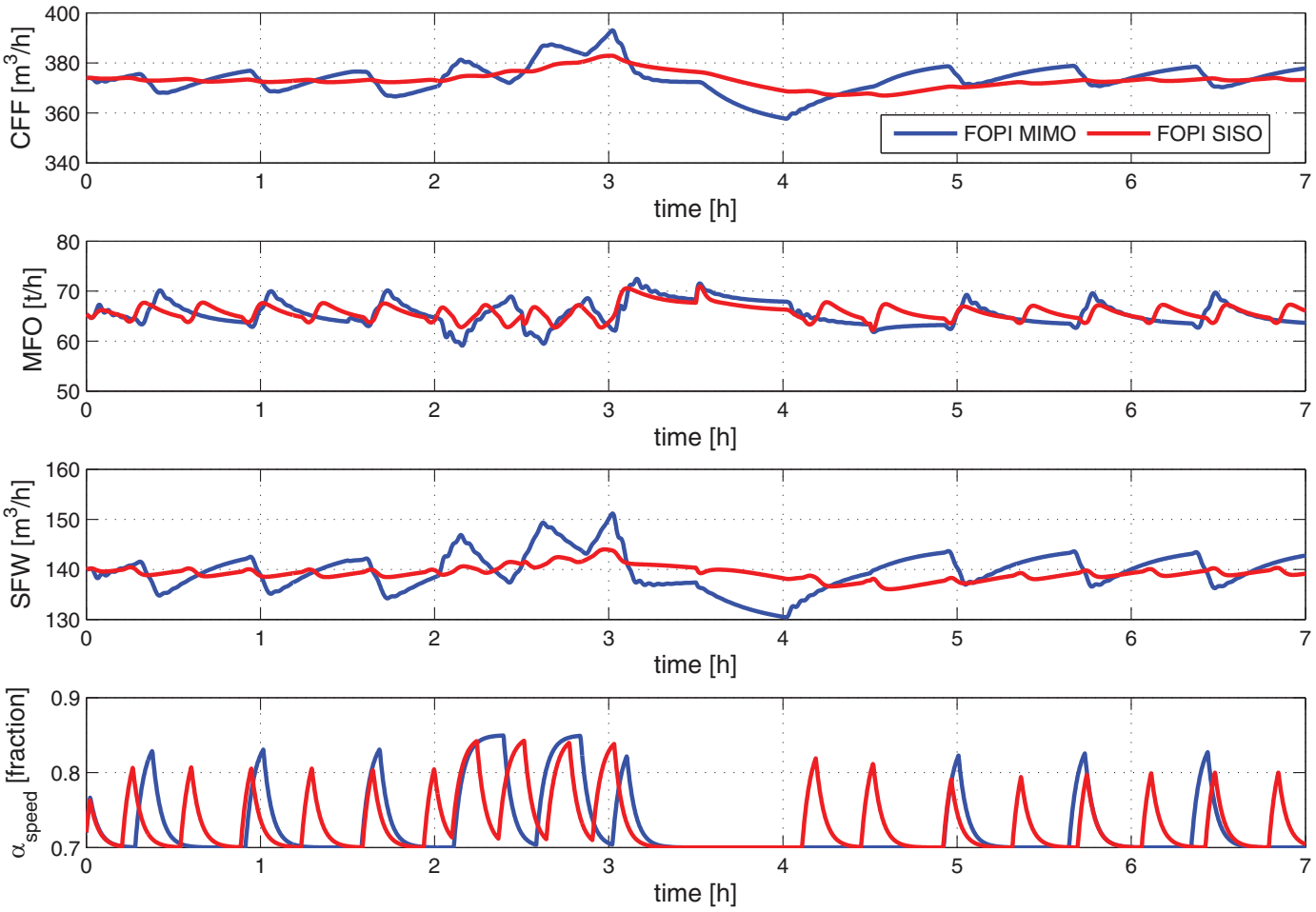

Fig. 7. Manipulated variables for second simulation scenario.

of the SISO FOPI controller. It is natural that the performance of the circuit deteriorates a little in this time frame, since the circuit needs to produce the same amount of grind material but with a smaller size, so it is a stressful situation for the control. After $t =$ 4.5 h, where the set point for *THP* is decreased, both controllers are capable to achieve the control goal, with similar stabilization times, although the control error is a little smaller for the MIMO FOPI controller. *LOAD* and *SVOL*, on the other hand, remain in their operating values as it is expected. The performance indexes for all the controlled variables are presented in Table 3.

As far as the manipulated variables is concerned, it can be seen from Fig. 5 that they remain within their bounds, except for the case of α_{speed}, which reaches its upper bound during a part of the simulation. Nevertheless, it can be noticed that it remains at its upper bound longer for the case of the MIMO FOPI controller, generating less speed fluctuations to the mill. The rest of the manipulated variables have similar values for NRMSI, as in the case of the SISO FOPI, as seen from Table 3.

6.2.3. Simulation scenario 2

Figs. 6 and 7 show the controlled and manipulated variables, respectively, for the second simulation scenario. During this time window, the external disturbances described in Section 6.1 and showed in Fig. 3 are present in the grinding mill circuit. As seen from Fig. 6, *PSE* behaves better for the case with MIMO FOPI controller, keeping the variable around its set point with less oscillations. It can also be seen that MIMO FOPI controller tends more quickly to return to the set point of *PSE* after external disturbances appears, while SISO FOPI has a slower response.

Table 4
Values of performance functions for the second simulation scenario.

		MIMO FOPI	SISO FOPI
NRMSE	*PSE*	0.0243	0.2042
	LOAD	0.2158	0.1498
	SVOL	0.1498	0.1744
	THP	0.2140	0.1439
σ_{PSE}		0.0056	0.3954
NRMSI	*CFF*	10.5679	10.5565
	MFO	4.8941	4.9133
	SFW	6.7312	6.7129
	α_{speed}	4.8678	4.8901

In the case of *LOAD* and *SVOL*, no significant differences are observed between the two control strategies. However, it can be corroborated from their corresponding NRMSE values showed in Table 4 that the MIMO FOPI controller behaves better than the SISO FOPI controller for these two controlled variables. Nevertheless, the two controlled variables remain very close to their set points values, even in the presence of the disturbances, which is a very good result for the control.

In the case of *THP*, no big differences are appreciated between the two control strategies in Fig. 6. However, looking at the NRMSE value for *THP* in Table 4, it can be seen that it is smaller for the

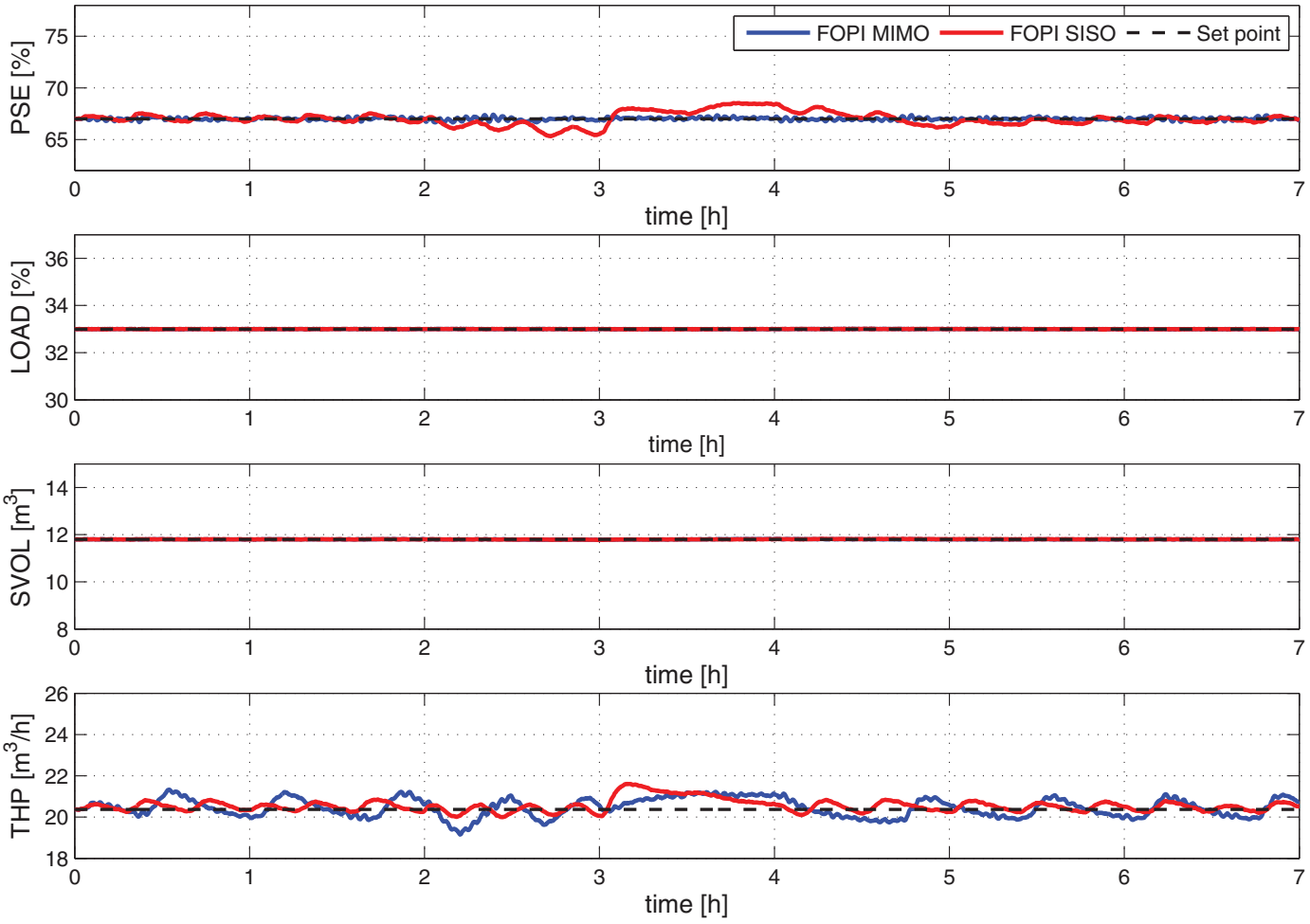

Fig. 8. Controlled variables for third simulation scenario.

SISO FOPI controller. This is because *THP* has less oscillations for the MIMO FOPI controller, but these oscillations are longer in time and amplitude than for the SISO FOPI controller.

Regarding the manipulated variables, Fig. 7 shows their evolution for this simulation scenario. As seen from Fig. 7, the two control strategies keep the manipulated variables within their limits, except for the α_{speed} in the MIMO FOPI controller, which reaches its upper bound in a few instants of time. The α_{speed} shows, however, a more oscillatory behavior for the SISO FOPI controller, contrary to the MIMO FOPI controller, where the speed control is smoother. The NRMSI values for the four manipulated variables are similar for the two control strategies, as can be seen from Table 4.

In the case of the variance of *PSE*, it is higher for the SISO FOPI controller. Since *PSE* is the most important variable in the circuit, the difference between the two controllers can be seen as an improvement of the MIMO FOPI in the grinding mill circuit operation in the presence of disturbances.

6.2.4. Simulation scenario 3

The results obtained for the third simulation scenario can be seen in Figs. 8 and 9. Table 5 shows the corresponding values of the performance indexes.

From Fig. 8 it is seen that all the controlled variables are affected by the process noise described in Section 6.1, for both control strategies, although they remain close to their set points, which is an indication of satisfactory operation. The overall behavior is similar to that of the case with external disturbances in the second simulation scenario, but with noise affecting the variables, and it is more visible in the cases of *PSE* and *THP*.

Table 5
Values of performance functions for the third simulation scenario.

		MIMO FOPI	SISO FOPI
NRMSE	*PSE*	0.0391	0.1964
	LOAD	0.1672	0.1473
	SVOL	0.1463	0.1547
	THP	0.1840	0.1406
σ_{PSE}		0.0159	0.4002
NRMSI	*CFF*	10.9435	10.9294
	MFO	1.9669	1.9721
	SFW	5.6390	5.6222
	α_{speed}	4.8726	4.8891

The results are 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.

Regarding the manipulated variables, it is observed from Fig. 9 that they are also affected by the process noise, more visible in the case of *MFO* and *SFW*. The NRMSI values are similar for the two controllers.

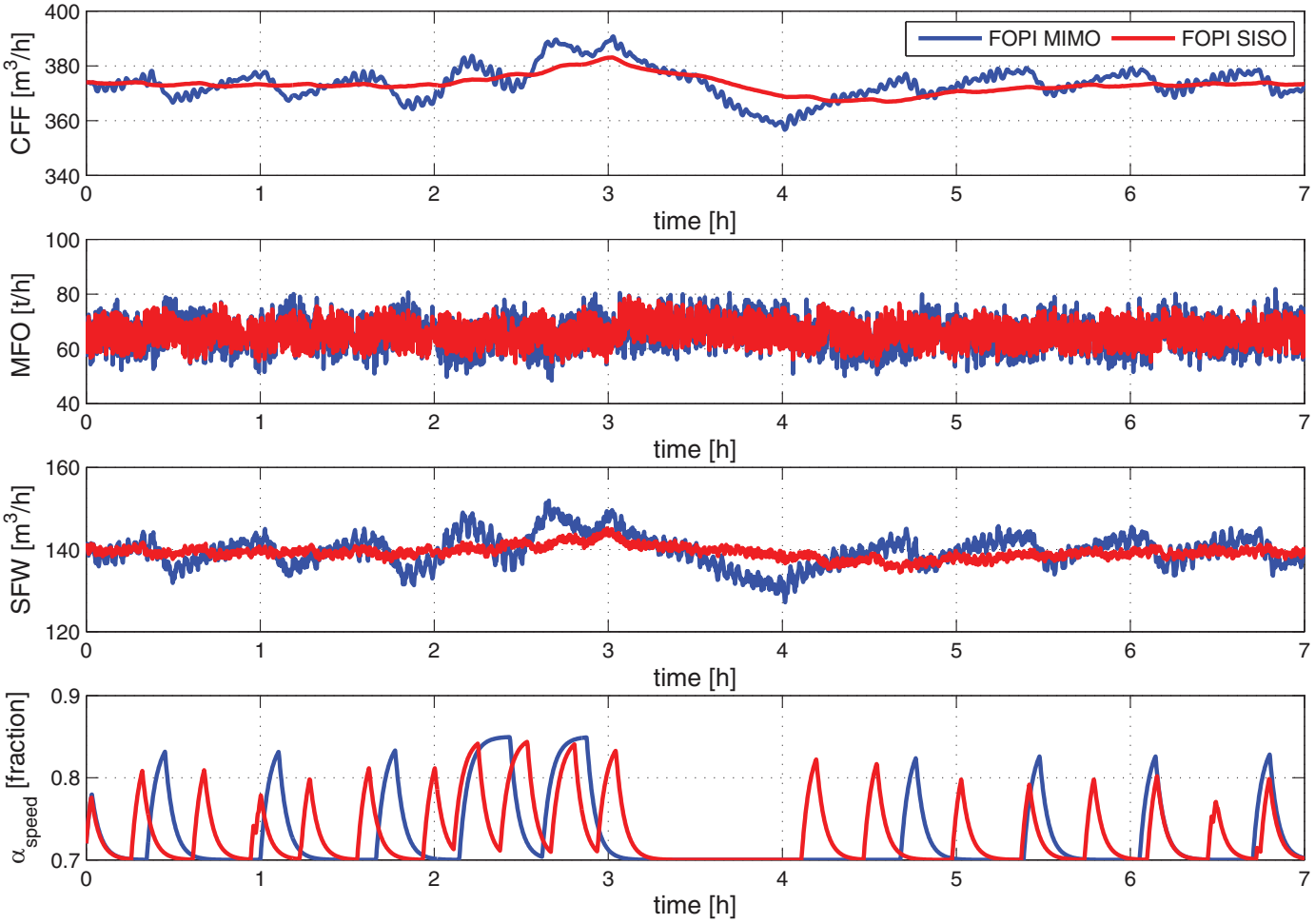

Fig. 9. Manipulated variables for third simulation scenario.

7. Conclusions

The design and application of a MIMO FOPI controller to a grinding mill circuit has been presented in this paper. Its performance was evaluated under different scenarios and compared with the performance of a SISO FOPI controller, both designed and tunned using the same optimal PSO procedure. Simulations were conducted when external disturbances and process noise were present in the circuit and different performance indexes were calculated to quantify the effects of disturbances on the milling circuit for both controllers.

Results indicate that the MIMO FOPI controller achieves independent control of product quality and throughput. Also, results indicates that MIMO FOPI controller achieves better results compared to the SISO FOPI controller in the presence of parametric disturbances and process noise, according to the performance functions computed.

Acknowledgments

The results reported in this paper have been financed by CONICYT Project AFB180004, FONDECYT Project 1150488, “Fractional Error Models in Adaptive Control and Applications”, FONDECYT Project 3150007, “Postdoctoral Program2015”, FONDECYT Project 1190959, “Development of fractional order tools for stability, estimation and control of systems and applications” and FONDECYT Project 11170154, “Improving the energy efficiency using fractional controllers: Design, analysis and applications”.

References

[1] N. Aguila-Camacho, M.A. Duarte-Mermoud, Fractional adaptive control for an automatic voltage regulator, ISA Trans. 52 (2013) 807–815.
[2] N. Aguila-Camacho, J.D.L. Roux, M.A. Duarte-Mermoud, M.E. Orchard, Control of a grinding mill circuit using fractional order controllers, J. Process Control 53 (2017) 80–94.
[3] X. Chen, Q. Li, S. Fei, Constrained model predictive control in ball mill grinding process, Powder Tech. 186 (2008) 31–39.
[4] X. Chen, J. Zhai, S. Li, Q. Li, Application of model predictive control in ball mill grinding circuit, Minerals Eng. 20 (2007) 1099–1108.
[5] L.C. Coetzee, Robust nonlinear model predictive control of a closed run-of-mine ore milling circuit, University of Pretoria, 2009 PhD. thesis.
[6] L.C. Coetzee, I.K. Craig, E. Kerrigan, Robust nonlinear model predictive control of a run-of-mine ore milling circuit, IEEE Trans. Control Syst. Tech. 18 (2010) 222–229.
[7] Constrained particle swarm optimization, http://www.mathworks.com/matlabcentral/fileexchange/25986-constrained-particle-swarm-optimization (2017).
[8] 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).
[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, 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.
[11] M.A. Duarte-Mermoud, A. Suárez, D. Bassi, Control of grinding plants using predictive multivariable neural control, Powder Tech. 115 (2001) 193–206.
[12] D.G. Hulbert, I.K. Craig, M.L. Coetzee, D. Tudor, Multivariable control of a run-of-mine milling circuit, J. South African Inst. Min. Metal. 90 (1990) 173–181.

[13] 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, University of Chile, 2012 PhD. thesis.
[14] R.H. Ordóñez Hurtado, M.A. Duarte-Mermoud, Finding common quadratic Lyapunov functions for switched linear systems using particle swarm optimization, Int. J. Control 85 (2012) 12–25.
[15] A. Kilbas, H. Srivastava, J. Trujillo, Theory and Applications of Fractional Differential Equations, Elsevier, 2006.
[16] D. Matignon, Stability results on fractional differential equations with applications to control processing, in: Proceedings of the Computational Engineering in Systems Applications. IMACS, 2, 1996, pp. 963–968. IEEE-SMC, Lille, France
[17] M. Moradi, A genetic-multivariable fractional order PID control to multi-input multi-output processes, J. Process Control 24 (2014) 336–343.
[18] 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. Vib. Control 22 (2016) 2187–2195.
[19] C.I. Muresan, E.H. Dulf, C. Ionescu, Robustness evaluation of a multivariable fractional order PI controller for time delay processes, Control Intell. Syst. 42 (2014) 112–118.
[20] 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.
[21] A. Oustaloup, La Commande CRONE: Commande Robuste d'ordre Non Entier, Hermes, Paris, 1991.
[22] I. Petrás, Tunning and implementation methods for fractional-order controllers, Fract. Calc. Appl. Anal. 15 (2012) 282–303.
[23] I.S. Podlubny, Fractional Differential Equations, Academic Press, San Diego, CA, 1999.
[24] 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.
[25] M. Ramasamy, S. Narayanan, C. Rao, Control of ball mill grinding circuit using model predictive control scheme, J. Process Control 15 (2005) 273–283.
[26] 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.
[27] 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, Miner. Eng. 43–44 (2013) 121–134.
[28] J.D. Le Roux, L.E. Olivier, M.A. Naidoo, R. Padhi, I.K. Craig, Throughput and product quality control for a grinding mil circuit using non-linear MPC, J. Process Control 42 (2016) 35–50.
[29] 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.
[30] P. Shah, S. Agashe, Review of fractional PID controller, Mechatronics 38 (2016) 29–41.
[31] X. Song, Y.Q. Chen, I. Tejado, B.M. Vinagre, Multivariable fractional order PID controller design via LMI approach, in: Proceedings of the Eighteenth IFAC World Congress, Milano, Italy, 2011.
[32] J. Suárez, B.M. Vinagre, Y.Q. Chen, A fractional adaptation scheme for lateral control of an AGV, J. Vib. Control 14 (9–10) (2008) 1499–1511.
[33] I. Tejado, S.H. HosseinNia, B.M. Vinagre, Adaptive gain-order fractional control for network-based applications, Fract. Calc. Appl. Anal. 17 (2014) 462–482.
[34] D. Valerio, J.S.D. Costa, Ninteger: a non-integer control toolbox for matlab, in: Proceedings of the Fractional Derivatives and Applications. IFAC, Bordeaux, Italy, 2004.
[35] 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 Dyn. 29 (2002) 269–279.
[36] D. Wei, I.K. Craig, Economic performance assessment of two ROM ore milling circuit controllers, Miner. Eng. 22 (2009) 826–839.