

# Marine Predators Algorithm Approaches on a Multivariable Fractional PID Controller Tuning

Yan Lieven Souza Lúcio  
Electrical Engineering Graduate  
Program (PPGEE)  
Federal University of Parana (UFPR)  
Curitiba, Brazil  
yanlieven@ufpr.br

Luiza Scapinello Aquino  
Electrical Engineering Graduate  
Program (PPGEE)  
Federal University of Parana (UFPR)  
Curitiba, Brazil  
luiza.scapinello@ufpr.br

Leandro dos Santos Coelho  
Electrical Engineering Graduate  
Program (PPGEE)  
Federal University of Parana (UFPR)  
and Industrial and Systems Engineering  
Graduate Program (PPGEPS),  
Pontifical Catholic University of  
Parana (PUCPR)  
Curitiba, Brazil  
leandro.coelho@pucpr.br

**Abstract**— In this paper, a performance comparison between the Marine Predators Algorithm (MPA), a metaheuristic paradigm, and two other designed variants for the tuning of a fractional proportional-integrative-derivative (PID) controller in a multiple-input multiple-output (MIMO) application is presented. The practical system plant corresponds to a ball mill pulverizing system, whose structure presents two inputs and two outputs. To encounter the optimal response on the MIMO control of this system a MPA approach applied to PID tuning is suitable, as it presents both the capability to diversify the search space (exploration) and to improve the quality of current solutions (exploitation) in search space. The MPA is a metaheuristic inspired by the extensive hunting strategy of ocean predators called Lévy and Brownian movements, it focuses on an optimal confront rate procedure in natural interaction between predator and prey in the marine ecosystem. The original MPA itself presents a satisfactory performance, in terms of statistical metrics. Nevertheless, it can be improved through the modification and addition of distinct techniques. In order to achieve those modifications, three variants are implemented exploring different procedures namely the oppositional-based learning and application of quantum mechanics. The optimal parameter values for the PID controller are analyzed by minimizing the integral time squared error (ITSE) index of the system's response. The simulations are performed using the SIMULINK/MATLAB computational environment. Statistical measures including best, mean, median and standard deviation of the system response error for the tuned controllers are evaluated and compared over fifty runs. The obtained results suggest that the use of the mentioned proposals has an advantage in enhancing the tuning efficiency of the MPA in this application.

**Keywords**—fractional proportional-integrative-derivative control, marine predators algorithm, evolutionary computation, swarm intelligence, multiple-input multiple-output application

## I. INTRODUCTION

The proportional-integral-derivative (PID) is the most widely used feedback controller industrially since its proposition in 1910, this fact is due to its clear functionality, applicability, and efficiency [1]. Robust systems are those which cope with errors during its executions, and most robust performance systems, however, are currently achieved by fractional-order PID controllers [2], which were first introduced by Podlubny in 1994 [3]. The greatest difference between the classic and the fractional PID is the number of parameters, the last one has two more parameters as compared to the classical PID controller, which means a more challenging work in the tuning process [4].

The goal of a PID controller is to decrease the error value between a required setpoint and the real measured process

variable as close to zero as possible. This operation is influenced by the input parameters, which are preset before the actual control activity [5]. During the tuning process, the optimal values for these parameters are achieved, and similar to the classical PID controller, common methods for this process in a fractional PID include Ziegler-Nichols and Cohen-Coon methods [3], artificial neural networks [6], fuzzy-based approaches [7], neuro-fuzzy [8], evolutionary and swarm intelligence-based algorithms [9].

Evolutionary algorithms (EAs) and swarm intelligence (SI) approaches are metaheuristics paradigms. They have methods capable of achieving a reasonable solution in a better computational time when compared to the other approaches mentioned [10]. They are known for using the knowledge of the problem to solve it. The Marine Predators Algorithm (MPA) [11] is based on the Lévy flight pattern, which consists of numerous small steps associated with longer relocations, extracted from a probability distribution with a power-law tail. This search strategy is used by many marine predators including sharks, tunas, and swordfishes in searching for prey [12].

The contribution of this paper is to compare the performance of two MPA variants, such as the use of oppositional-based learning and the application of quantum mechanics, in the tuning process of a fractional PID controller for a ball mill pulverizing system, which is a process defined as a multiple-input multiple-output (MIMO) system. Although other metaheuristics are used in the same application, the aim of the paper is to compare a single metaheuristic with its variants and prove that optimal values can be achieved when enhancing the MPA by its variants.

The remaining paper is organized as follows. Section II reviews the fundamentals of a fractional PID controller, its tuning process, as well as a description of the MPA. The third section explains the methodology used in this case study. Sequentially, the results are discussed in Section IV. Finally, the last section includes the final scopes and conclusion.

## II. BRIEF REVIEW OF THEORETICAL FUNDAMENTALS

A classical PID controller is composed of proportional  $K_P$ , integrative  $K_I$ , and derivative  $K_D$  factors, which are used in the general controller structure is given by

$$C(s) = K_P + \frac{K_I}{s} + K_D s. \quad (1)$$

The first factor in Eq. (1) represents a proportional action in the error between the setpoint and the output, whereas the integrative one decreases the steady-state error of the system

response. The last factor, however, enhances the transient response [13].

The understanding of fractional calculus embodies the theory of derivatives and integers of approximate numbers. Furthermore, it has the advantage of exposing both short- and long-term memory. The first one consisting of the distribution of time constants, while the second one leads to a dearth of a specific timescale [14].

There are many definitions of fractional calculus, but the most common two used in the area of control systems are the Grünwald–Letnikov (GL) definition and the Riemann–Liouville (RL) definition, both of which are expressed in Eqs. (2) and (3) respectively in which

$${}_a D_t^\alpha = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{r=0}^{\lfloor \frac{t-a}{h} \rfloor} (-1)^r \binom{\alpha}{r} f(t-rh) \quad (2)$$

$${}_a D_t^\alpha = \frac{1}{\Gamma(n-\alpha)} \left(\frac{d}{dt}\right)^n \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \quad (3)$$

where  $\lfloor \frac{t-a}{h} \rfloor$  is the integer part, being  $a$  and  $t$  the limits of the operator.  $n$  is the integer value that satisfies the condition  $n - 1 < \alpha < n$ . Also,  $\Gamma(\cdot)$  is Euler's gamma function.

A fractional PID controller, also called  $PI^\lambda D^\mu$ , firstly proposed by Podlubny [3] is an extension of the classical PID controller, whose difference lays in the addition of an integrator of order  $\lambda$  and a differentiator of order  $\mu$  [15]. The  $PI^\lambda D^\mu$  controllers achieved, according to [3] a better response when compared to the classical PID controller, since it is less susceptible to alterations of parameters of a controlled system. The generalized transfer function of a fractional PID controller is given by

$$C(s) = K_p + \frac{K_I}{s^\lambda} + K_D s^\mu, (\lambda, \mu \geq 0) \quad (4)$$

Traditional PID controller cases occur when  $\lambda$  and  $\mu$  are equal to one, as can be seen in Fig. 1 [16]. Generally, the range of fractional order is considered between zero and two.

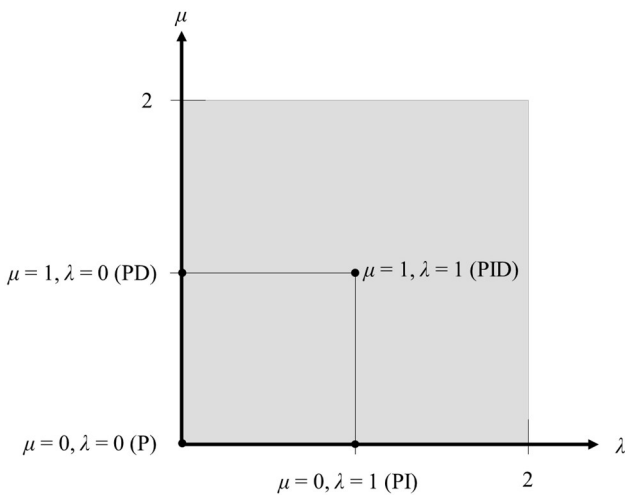


Fig. 1. Fractional PID controller operation region.

Fig. 2 shows the general structure of a closed-loop control system, in which a fractional PID controller is adopted, in which  $e(t)$  is the error,  $i(t)$  the input variable (setpoint) and  $y(t)$  the output variable.

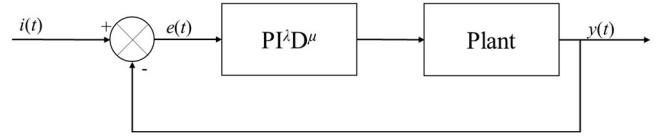


Fig. 2. General block diagram of  $PI^\lambda D^\mu$  controller.

As it can be noted, the  $PI^\lambda D^\mu$  controller has two extra parameters compared to the classical one, consequently, more conditions need to be met, improving the system's performance. Therefore, the tuning of a fractional PID controller is of great importance in the controller's design [2], which can be done via metaheuristics optimization processes.

#### A. Marine Predators Algorithm

The MPA mimics the behavior of marine predators searching for their prey, while the prey itself is also searching for food. Therefore, the MPA is a metaheuristic influenced by the survival of the fittest strategy [11]. Those predators use the Lévy and Brownian movements as the most effective hunting mechanism, which is, as most foraging tactics, a stochastic strategy, thus the consecutive location of the animal relies on the current one and the probability transition to that point. The Lévy approach is used in areas with less crowd of prey, while the Brownian movement is more related to places with an abundance of prey [17]. And to swap the two approaches the prey and predator ratio velocity  $v$  is used. A small value of  $v$  (around 0.1) means the occurrence of the Lévy or exploitation phase, regardless of the prey. A greater  $v$  (around 1) represents a predator in Brownian movements if the prey is moving according to Lévy steps. Ultimately, when  $v$  is greater than ten, that means the prey will come, representing the exploration phase [18].

Hence, the MPA has three distinct phases, and the predators' velocity is the key to the difference among the phases. The first phase represents the initialization of the population. In this phase, the predator is moving faster than the prey. During the exploration phase, the best strategy for the predator is not to move. The mathematical model of this phase, which occurs in the first third of the maximum number of iterations is given by

$$\overrightarrow{Prey}_i = \overrightarrow{Prey}_i + 0.5 \cdot \overrightarrow{rand} \otimes \overrightarrow{stepsize}_i \quad (5)$$

where  $i = 1, 2, 3, \dots, n$ .

$$\overrightarrow{stepsize}_i = \overrightarrow{R}_B \otimes (\overrightarrow{Elite}_i - \overrightarrow{R}_B \otimes \overrightarrow{Prey}_i) \quad (6)$$

In those equations,  $\overrightarrow{R}_B$  and  $\overrightarrow{rand}$  are vectors containing random numbers based on a normal distribution, the first one represents the Brownian motion. The symbol  $\otimes$  shows entry-wise multiplications. The multiplication of  $\overrightarrow{R}_B$  by the prey simulates its movement. The vector *Elite* represents the best predators.

The second phase occurs in the middle third of the algorithm, and it represents the stage when exploration is becoming exploitation. The mathematical model for the first half of the population is given as:

$$\overrightarrow{Prey}_i = \overrightarrow{Prey}_i + 0.5 \cdot \overrightarrow{rand} \otimes \overrightarrow{stepsize}_i \quad (7)$$

$$\overrightarrow{stepsize}_i = \overrightarrow{R}_L \otimes (\overrightarrow{R}_L \otimes \overrightarrow{Elite}_i - \overrightarrow{Prey}_i) \quad (8)$$

And for the second half:

$$\overrightarrow{Prey}_i = \overrightarrow{Elite}_i + 0.5 \cdot CF \otimes \overrightarrow{stepsize}_i \quad (9)$$

$$\overrightarrow{stepsize}_i = \overrightarrow{R}_B \otimes (\overrightarrow{Elite}_i - \overrightarrow{R}_B \otimes \overrightarrow{Prey}_i) \quad (10)$$

where similar to  $\overrightarrow{R}_B, \overrightarrow{R}_L$  is a vector containing random values generated according to the Lévy approach. In this phase, the prey moves both with exploitation and exploration steps. CF is a flexible parameter to control the step size, and its mathematical formula is given by Eq. (11), where  $t$  represents the current iteration, and  $T$  is the maximum number of iterations.

$$CF = \left(1 - \frac{t}{T}\right)^{2 \frac{t}{T}} \quad (11)$$

The third and last phase represents the stage of complete exploitation. The mathematical behavior can be formulated as

$$\overrightarrow{Prey}_i = \overrightarrow{Elite}_i + 0.5 \cdot CF \otimes \overrightarrow{stepsize}_i \quad (12)$$

$$\overrightarrow{stepsize}_i = \overrightarrow{R}_L \otimes (\overrightarrow{R}_L \otimes \overrightarrow{Elite}_i - \overrightarrow{Prey}_i) \quad (13)$$

In nature, the behavior of predators is affected by the environment, and the Fish Aggregating Devices (FADs) have an important matter of change in it. As seen in [11], sharks, for instance, spend 80% near FADs searching for prey, and the rest of the time they tend to explore other environments with another prey distribution. The FADs [19] process is based on

$$\overrightarrow{Prey}_i = \begin{cases} \overrightarrow{Prey}_i + CF [\overrightarrow{X}_{min} + \overrightarrow{R} \otimes (\overrightarrow{X}_{max} - \overrightarrow{X}_{min})] \otimes \overrightarrow{U} & \text{if } r \leq FADs \\ \overrightarrow{Prey}_i + [FADs(1 - r) + r](\overrightarrow{Prey}_{r_1} - \overrightarrow{Prey}_{r_2}) & \text{if } r \geq FADs \end{cases} \quad (14)$$

where  $\overrightarrow{U}$  is of zeros and ones, it is created by generating a random number between 0 and 1, if the generated number is greater than 0.2, then  $\overrightarrow{U}$  is set to 0, otherwise it is set to 1. FADs is equal to 0.2 and represents its effect on optimization. The indexes  $r_1$  and  $r_2$  are a random selection of the prey matrix.  $\overrightarrow{R}$  and  $r$  are a random vector and a constant, respectively, uniformly distributed between 0 and 1.

Marine predators remember places which they had successful foraging, meaning they have a good memory, which is simulated by memory saving in MPA. After each interaction, the fitness value is compared to its previous equivalent, and the more fitted one is preserved. Fig. 3 shows the pseudocode for the MPA.

---

### Algorithm 1: Pseudocode for marine Predators Algorithm (MPA)

---

```

1: Initialize search agents (Prey) populations  $i = 1, 2, \dots, 3, n$ 
2: while stop criteria not reached do
3:   if current iteration < maximum iteration  $\div 3$  then
4:     Update prey based on Eq. (5)
5:   else if maximum iteration  $\div 3 <$  current iteration <  $(2/3) \times$  maximum iteration then
6:     For the first half of the population update prey based on Eq. (7)
7:     For the second half of the population update prey based on Eq. (9)
8:   else if current iteration  $>$   $(2/3) \times$  maximum iteration then
9:     Update prey based on Eq. (12)
10:  end if
11:  Accomplish memory saving and Elite update
12:  Apply FADs effect and update prey passed on Eq. (14)
13: end while

```

---

Fig. 3. Pseudocode for MPA.

### B. Quantum Matrix Marine Predators Algorithm

Quantum Marine Predators Algorithm (QMPA) is based on quantum mechanics, characteristically it utilizes the same strategy present in the Quantum Particle Swarm Optimization (QPSO), therefore it is based on a quantum delta potential well model, along with the mean best solution in the population [20].

This quantum strategy is implemented in all three phases of the algorithm, by modifying the Eqs. (5), (7), (9) and (12), the ones responsible for changing the particle's position. The step size vector in each phase continues to be the same.

The strategy consists in the fact that in each phase, for every solution being modified, a random number  $q$  is generated. If this number is greater than or equal to 0.5, the quantum portion of the equation is summed, otherwise it is subtracted from the particle's current position.

The Brownian motion equation presented in phase 1 and in the first half of the population in phase 2, changes to Eq. (15) if  $q$  is greater than or equal to 0.5. In this case,

$$\overrightarrow{Prey}_i = \overrightarrow{Prey}_i + 0.5 \cdot r_1 \otimes |\overrightarrow{stepsize}_i| \log\left(\frac{1}{r_2}\right) \quad (15)$$

where  $r_1$  and  $r_2$  are random numbers uniformly distributed between (0,1). If  $q$  is lower than 0.5, the prey's position is modified by

$$\overrightarrow{Prey}_i = \overrightarrow{Prey}_i - 0.5 \cdot r_1 \otimes |\overrightarrow{stepsize}_i| \log\left(\frac{1}{r_2}\right) \quad (16)$$

The Lévy motion present in phase 3 and in the second half of the population in phase 2 changes to Eq. (17) if  $q$  is greater or equal to 0.5 where

$$\overrightarrow{Prey}_i = \overrightarrow{Elite}_i + 0.5 \cdot CF \otimes \overrightarrow{stepsize}_i \cdot \log\left(\frac{1}{r_1}\right) \quad (17)$$

If  $q$  is lower than 0.5, then the equation representing the population changes to Eq. (18)

$$\overrightarrow{Prey}_i = \overrightarrow{Elite}_i - 0.5 \cdot CF \otimes \overrightarrow{stepsize}_i \cdot \log\left(\frac{1}{r_1}\right) \quad (18)$$

### C. Oppositional Matrix Marine Predators Algorithm

The probability theory stands that 50% of the time a supposition is further from the solution than its opposite

estimate [21]. Every time a new solution is generated, oppositional-based learning generates its opposite and takes only the best one. Mathematically, for a solution vector  $X$ , the opposite solution would be calculated by the opposite solution  $OX$  as defined by

$$OX_t = LB + UB - X_t \quad (19)$$

where  $LB$  is the lower boundary and  $UB$  is the upper boundary.

The oppositional MPA (OMPA) generates the opposite of every newly generated vector  $X$  in the algorithm, as well as the randomly made initial solutions. It compares both solutions and keeps the one with the best fitness function value as the new solution. The rest of the process is maintained.

#### D. Ball Mill Pulverizing System

The system used to test the efficiency of each variant of the MPA was a ball mill, a complex object with pure delay, close coupling, multi-variable, non-linear, slow time-varying [22]. The system has three inputs (speed of coal feeder, opening of recycling air, opening of hot air) and three outputs (differential pressure, negative inlet pressure, outlet pressure). However, this system can be reduced to two inputs and two outputs, if the vertical vibration signal of the front of the bearing of the ball mill represents coal loading [23].

This allows for the system to have the transfer function

$$G(s) = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} = \begin{bmatrix} \frac{3.5}{(80 \cdot s + 1)^3} & \frac{-0.14}{(60 \cdot s + 1)^2} \\ \frac{-2}{(8 \cdot s + 1)^2} & \frac{-0.18}{(10 \cdot s + 1)} \end{bmatrix} \quad (20)$$

which is not diagonally row dominant. A steady-state decoupler can be used to reduce interactions among variables, having as transfer function [24]

$$D(s) = G^{-1}(0) = \begin{bmatrix} 0.1378 & -0.1538 \\ -2.1978 & -3.8462 \end{bmatrix} \quad (21)$$

The system's structure is fully shown in Fig. 4.

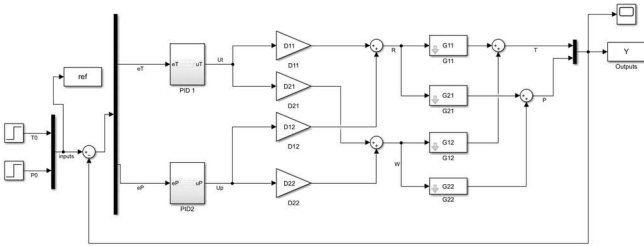


Fig. 4. System structure of the ball mill pulverizing system.

### III. METHODOLOGY

Each candidate solution  $X$  in the MPA has ten values, represented by the vector in Fig. 4. The first five values correspond to the values for the first fractional PID controller and the last five represent the values for the second one. The lower and upper limit values for the chromosomes are 0 and 50 for the proportional, integral, and derivative gains, and the integrator and differential orders stay between the limits 0 and 1 in each PID.

$K_{P1}$	$K_{I1}$	$K_{D1}$	$\lambda_1$	$\mu_1$	$K_{P2}$	$K_{I2}$	$K_{D2}$	$\lambda_2$	$\mu_2$
----------	----------	----------	-------------	---------	----------	----------	----------	-------------	---------

Fig. 5. Structure in the MPA population.

The objective function used was experimentally tested and developed. It is defined by a sum of the Integral Time Squared Error (ITSE) and the maximum output values of the system response, as presented in Eq. (20):

$$Fobj(X) = w_1 \cdot \sum_{i=1}^2 \left( \int_{t=0}^T t \cdot e_i^2(t) dt \right) + w_2 (\max(Y_1)) + w_3 \cdot \max(Y_2) \quad (20)$$

where  $x$  is the evaluated solution,  $t$  and  $T$  are the current and maximum simulation time,  $e(t)$  is the error, and  $y_1$  and  $y_2$  are the system's first and second output. The variables  $w_1$ ,  $w_2$  and  $w_3$  are weights experimentally found to give the objective function the capacity to select the solutions with the best responses. That is, with low ITSE and low overshoot, especially low overshoot in the second output, which may rise to undesirable levels. The best weight values were found as 0.3, 0.7 and 3 for  $w_1$ ,  $w_2$  and  $w_3$ , respectively.

The simulations are run with a population size of 50, and 100 as the number of generations. The tool used to run the simulations was the MATLAB-SIMULINK computational environment, and the hardware is an Intel(R) Core (TM) i7-7700HQ central processing unit (CPU) @ 2.80GHz 2.81 GHz, and 8GB random access memory (RAM). To validate the methods used and to be able to compare the results in each algorithm, the algorithms were run 40 times, and mean, median and standard deviation for the objective function values in each run were computed.

### IV. RESULTS AND DISCUSSION

The error values are present in Table I. As it can be seen, the original MPA had the smallest fitness function value. However, the statistical measures showed that both MPA variants were capable of maintaining smaller values throughout the 40 runs, evidenced by the lower mean, median and standard deviation values from OMPA and QMPA.

TABLE I. STATISTICAL RESULTS FOR THE OPTIMIZERS

Method	MPA	OMPA	QMPA
<b>Minimum</b>	414.10	419.04	414.70
<b>Maximum</b>	670.65	575.78	458.41
<b>Mean</b>	454.89	443.80	429.23
<b>Median</b>	432.68	427.21	425.56
<b>Standard deviation</b>	73.00	37.85	10.47

This shows that the generated variants are more stable and should be chosen in situations where it is fundamental for the controller to present a steady response in repetitive runs, making them more reliable, with a more concise output, as proven by the smaller standard deviation when compared to the original one.

The system responses for the solutions with the smallest fitness values in each method are shown in Fig. 6 and 8, respectively. Fig. 7 and 9 are an expended view of Figures 5 and 7.

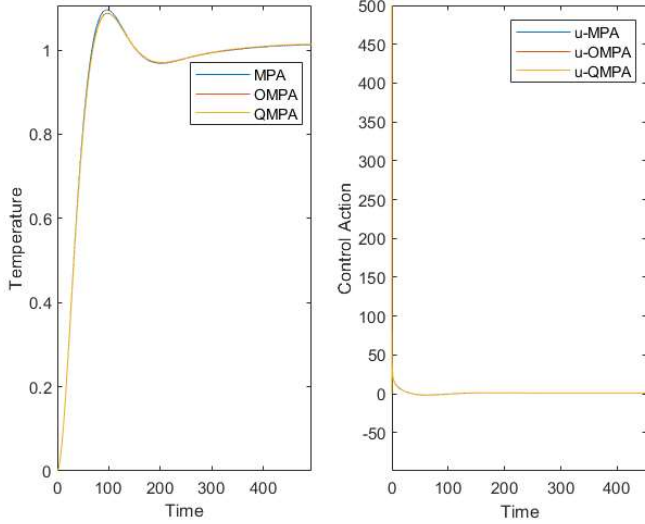


Fig. 6. System response for the first output.

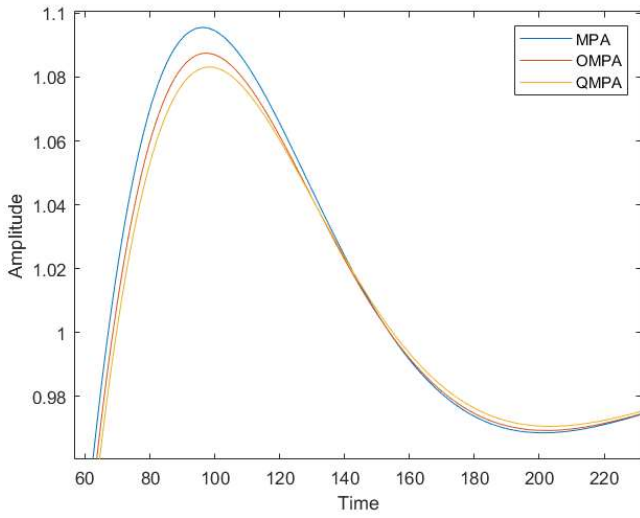


Fig. 7. Expanded view of system response for the first output.

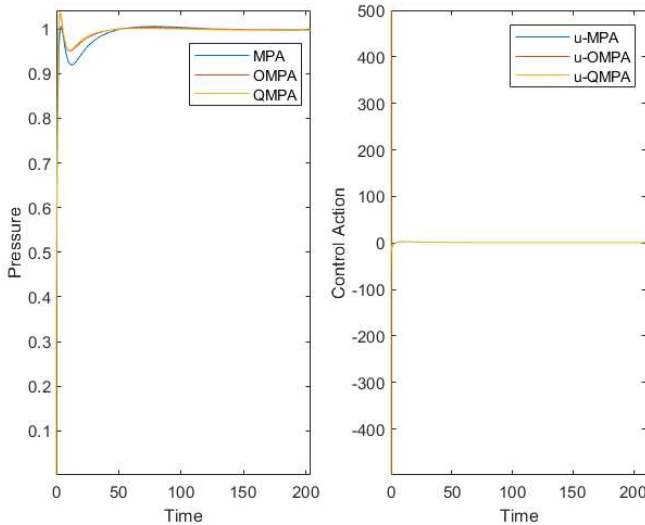


Fig. 8. System response for the second output.

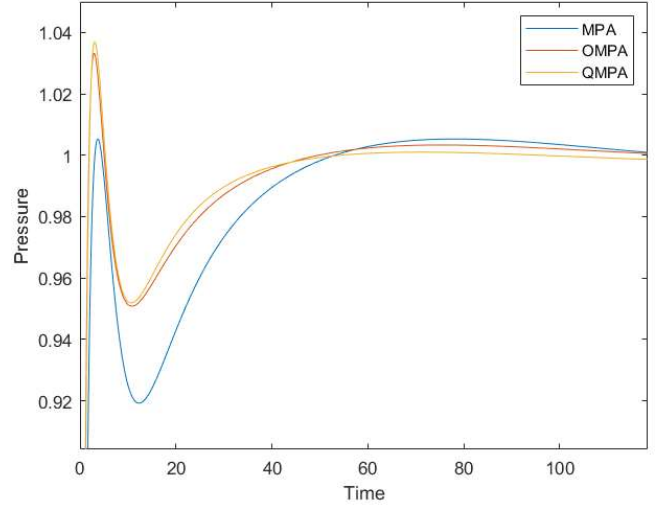


Fig. 9. Expanded view of the system response for the second output.

For the first output, the system response has a smaller overshoot in the created variants, being more stable. In the second output, the MPA has higher rising and settling times, but less overshoot compared to the variants. The best solutions in each algorithm, that is, the ones with the smallest fitness function values are presented in Table II.

TABLE II. BEST SOLUTIONS FOR EACH OPTIMIZER

Parameter	MPA	OMPA	QMPA
$K_{P1}$	0.1	0.002	0.01
$K_{I1}$	0.3	0.03	0.03
$K_{D1}$	50	50	49.99
$\lambda_1$	0.99	1	0.99
$\mu_1$	0.45	0.46	0.46
$K_{P2}$	25.92	44.63	21.94
$K_{I2}$	0.67	1.16	2.05
$K_{D2}$	49.97	50	50
$\lambda_2$	0.95	0.96	0.85
$\mu_2$	0.99	1	1

The statistical values for the time taken (in seconds) to be able to complete a run are shown in Table III. As it can be observed, the OMPA takes much longer to run, due to its oppositional nature. The original MPA takes the least time to be able to finish, with the QMPA coming close. The QMPA has the best cost-benefit if the time is taken into account, seeing that it maintains a great statistical result for the fitness values, while at the same time not having an increasing time duration.

TABLE III. PROCESSING TIMES

Method	Min	Max	Mean	Median	Std
MPA	3980	9245.2	6400.8	6592.9	1237.1
OMPA	6552	16711.7	11979.5	12284.7	2708.4

## V. FINAL CONSIDERATIONS

In this paper, a fractional order PID controller is studied in a practical system plant corresponding to a ball mill pulverizing system, whose structure presents two inputs and two outputs. As is known, each metaheuristic algorithm is predisposed by the problem in question and its available information. Even though the MPA algorithm is a relatively new method when compared to other algorithms, it has proven to be capable of tuning a fractional PID controller for a MIMO plant effectively, as well as its two created variants. The created variants all showed to be more concise when compared to their original algorithm, even though the original MPA had achieved the minimal value. They have proved to be more reliable, allowing for better results in situations where there are repetitive experiments in the plant. The QMPA with the smallest mean, median and standard deviation was the most efficient one. Further research may be motivated in improving the efficiency of the algorithms, as well as developing new variants.

### ACKNOWLEDGMENT

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 001.

### REFERENCES

- [1] M. W. Iruthayarajan, and S. Baskar. "Evolutionary algorithms based design of multivariable PID controller". *Expert Systems with Applications*, vol. 36, pp. 9159-9167, 2009.
- [2] P. Shah, and S. Agashe. "Review of fractional PID controller". *Mechatronics*, vol. 38, pp. 29-41, 2016.
- [3] I. Podlubny. "Fractional-order systems and fractional-order controllers". Institute of Experimental Physics of the Slovak Academy of Sciences, Kosice, 1994.
- [4] B. M. Vinagre, C. A. Monje, A. J. Calderón, and J. I. Suárez. "Fractional PID controllers for industry application. A brief introduction". *Journal of Vibration and Control*, vol. 13, pp.1419-1429, 2007.
- [5] D. Fister, I. Fister Jr, I. Fister, and R. Šafarič. "Parameter tuning of PID controller with reactive nature-inspired algorithms". *Robotics and Autonomous Systems*, vol. 84, pp. 64-75, 2016.
- [6] S. A. Hosseini, A. S. Shirani, M. Lotfi, and M.B. Menhaj, M. B. "Design and application of supervisory control based on neural network PID controllers for pressurizer system". *Progress in Nuclear Energy*, vol. 130, article number 103570, 2020.
- [7] D. Sain, and B. M. Mohan. "A simple approach to mathematical modelling of integer order and fractional order fuzzy PID controllers using one-dimensional input space and their experimental realization". *Journal of the Franklin Institute*, 2021, In Press.
- [8] Q. Shi, H. K. Lam, C. Xuan, and M. Chen. "Adaptive neuro-fuzzy PID controller based on twin delayed deep deterministic policy gradient algorithm". *Neurocomputing*, vol. 402, pp. 183-194, 2020.
- [9] W. D. Chang. "A multi-crossover genetic approach to multivariable PID controllers tuning". *Expert Systems with Applications*, vol. 33, pp. 620-626, 2007.
- [10] S. Salcedo-Sanz. "Modern meta-heuristics based on nonlinear physics processes: a review of models and design procedures". *Physics Reports*, vol. 655, pp. 1-70, 2016.
- [11] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi. "Marine predators algorithm: A nature-inspired metaheuristic". *Expert Systems with Applications*, vol. 152, article number 113377, 2020.
- [12] N. E. Humphries, N. Queiroz, J. R. M. Dyer, N. G. Pade, M. K. Musyl, and K. M. Schaefer, K. M. "Environmental context explains Lévy and Brownian movement patterns of marine predators". *Nature*, vol. 465, article number 1066, 2010.
- [13] K. Ogata. *Modern control engineering*. Prentice Hall. 2010.
- [14] S. Das. *Functional fractional calculus*. Springer Science & Business Media, 2011.
- [15] B. M. Vinagre, C. A. Monje, A. J. Calderón, and J. I. Suárez. "Fractional PID controllers for industry application. A brief introduction". *Journal of Vibration and Control*, vol. 13, pp. 1419-1429, 2007.
- [16] D. Xue, C. Zhao, Y. Chen. "Fractional order PID control of a dc-motor with elastic shaft: A case study". In: *Proceedings of American Control Conference*, vol. 7, pp. 3182-3187, Minneapolis, Minnesota USA 2006.
- [17] M. A. Soliman, H. M. Hasanien, and A. Alkuhayli. "Marine predators algorithm for parameters identification of triple-diode photovoltaic models." *IEEE Access*, vol. 8, pp. 155832-155842, 2020.
- [18] M. Abdel-Basset, R. Mohamed, R. K. Chakraborty, M. Ryan, and S. Mirjalili. "New binary marine predators optimization algorithms for 0-1 knapsack problems". *Computers & Industrial Engineering*, vol. 151, article number 106949, 2021
- [19] J. D. Filmlalter, L. Dagorn, P. D. Cowley, and M. Taquet. "First descriptions of the behaviour of silky sharks (*Carcharhinus falciformis*) around drifting Fish Aggregating Devices, in the Indian Ocean, using acoustic telemetry". *Indian Ocean Tuna Commission - 2010 - Working Party on Ecosystems and Bycatch - 12*, 2010..
- [20] J. Sun, C. H. Lai, and X. J. Wu. "Particle swarm optimisation: Classical and quantum perspectives". CRC Press, 2016.
- [21] Q. Xu, L. Wang, N. Wang, X. Hei, and L. Zhao. "A review of opposition-based learning from 2005 to 2012". *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 1-12, 2014.
- [22] Qi-ming, Cheng, and Wang Yong-hao. "Pulverizing control systems of intermediate storage bunker ball mills of power plants [J]." *East China Electric Power*, vol. 34 pp. 23-27, 2006.
- [23] Qiming, Cheng, et al. "The forward NN-PID controllers based on chaos PSO-BP hybrid optimization algorithms for Decoupling Control System of Ball Mill." 2009 International Conference on Artificial Intelligence and Computational Intelligence, vol. 1, IEEE, 2009.
- [24] Menhas, Muhammad Ilyas, et al. "Real/binary co-operative and co-evolving swarms based multivariable PID controller design of ball mill pulverizing system." *Energy Conversion and Management*, vol. 54, pp. 67-80, 2007.

