Taylor & Francis
Taylor & Francis Group

MPSO-based PID control design for power factor correction in an AC-DC boost converter

J. G. Guarnizo, C. A. Torres-Pinzón, J. Bayona, D. Páez, J. P. Romero, B. Noriegaa & L. Paredes-Madrid

Published online: 04 Jul 2023.

Submit your article to this journal ⎈

View related articles ⎈

View Crossmark data ⎈

Taylor & Francis
Taylor & Francis Group

MPSO-based PID control design for power factor correction in an AC-DC boost converter

J. G. Guarnizo [a], C. A. Torres-Pinzón [a,b], J. Bayona [b], D. Páez[a], J. P. Romero[a], B. Noriega[a] and L. Paredes-Madrid [c]

[a]Faculty of Electronic Engineering, Universidad Santo Tomás, Bogotá, Colombia; [b]Faculty of Engineering, Universidad ECCI, Bogotá, Colombia; [c]Faculty of Mechanic, Electronic and Biomedical Engineering, Universidad Antonio Nari no, Tunja, Colombia

ABSTRACT

This paper presents the implementation for the first time of a Multi-Particle Swarm Optimization (MPSO) algorithm in the tuning of a PID controller for Power Factor Correction (PFC), applied to a 100W AC-DC boost converter. MPSO algorithm navigates in a search space where each dimension of the space corresponds to the controller constants (Proportional, Integral, Derivative and the Derivative Filter), prioritizing communication over exploration in the algorithm. The controller parameters are randomly initialized in a reduced sector of the space $[K_p, K_i, K_d, K_n]$, to optimize the search for a PID solution. In the first step, the algorithm is validated using a simulation model in Simulink and Matlab. Subsequently, a final implementation using a real converter is implemented with the PID tuned by MPSO, improving the PFC obtained in previous work. Although previous works have used evolutionary algorithms applied to heuristic optimization to tunning PID controllers, the MPSO algorithm is not usually used for this purpose, particularly to tunning a PID controller in a power electronics system. One advantage of MPSO over the PSO classical algorithm is the search at different points if the vectorial field looks for an optimal solution. PSO presents problems such as getting stuck in a locally optimal solution. The PID controller is trained offline, with the advantage of allowing the risk of damage in the Boost converter for transitory response, increasing the performance of the Power Factor Correction in the converter. This research opens the possibility to use the extended version of the PSO bioinspired algorithm to tune offline controllers to improve the power converter's performance, minimizing the risk presented in the real-time tuning process.

Nomenclature

Acronyms

PFC	Power factor correction
c_1, c_2	Acceleration coefficients
$[p\vec{B}_n - \vec{p}_n]$	Cognitive component
C_{MAT}	PID-Controller tuned by conventional methods
C_{MPSO}	MPSO tuned controller
$[p\vec{B}_n - \vec{p}_n]$	Social component
MPSO	Multi-Particle swarm optimization
K_p, K_i, K_d, K_n	PIDN controller gains
r_1, r_2	Random numbers
THD	Total harmonic distortion
v_{n+1}	Velocity of each particle
w	Inertia weight

1. Introduction

Power factor correction (PFC) is an element of constant study and research, due to the need to guarantee and improve efficiency in power systems to comply with international energy quality standards [1]. Several

topologies of converters with PFC have been reported, such as the half-bridge boost converter and the bridgeless SEPIC converter [2,3], however, the boost converter is one of the best options when correcting power factor due to different intrinsic design features such as semiconductor efficiency, smooth current waveform, reduced input filter, and simple control circuitry [4,5].

One of the types of controllers used in PFC is the average current control, which plays a very important role in this type of application [6–9], where the current will follow the rectified line voltage signal. Conventional controller tuning methods have a great variety of studies which allow them to be an excellent tuning option when implemented and represented by dynamic system models [10], however, there is the possibility of facing drawbacks related to their tuning. Some of the most common problems can be a long settling time or a large over peak, in addition, there is the possibility of facing too complex systems where a conventional method cannot be used with certainty and when using some tuning methods in the controller, it is always necessary to have a descriptive mathematical model of the

converter, in that case, is where intelligent methods can be used to improve the result that a conventional analysis may not be able to detect [11].

There are different works in which exceptional results have been seen by using metaheuristic algorithms or different models. For example, in Ref. [12], the authors present a cascade control using particle swarm optimization (PSO) for the tuning of both loops, in which they achieved unity power factor and low percentage THD. On the other hand, in Ref. [13], power factor correction results of 0.93956 and 0.9999 are presented using hysteresis and fuzzy logic-based controllers, respectively, in the boost topology. It is worth highlighting the use of PSO or genetic algorithms for the tuning of PID controllers in different topologies of switched converters [14,15]. In Ref. [14], the authors present the design of a PID control tuned with a PSO algorithm for the output regulation of an interleaved buck-boost converter. However, they do not describe any experimental results and only verify the effectiveness of the proposed method by means of simulations. On the other hand, in Ref. [15], they present the design of a control law for power factor correction in a boost converter using the evolutionary principles of genetic algorithms. The accuracy of this approach is verified by an experimental prototype, which shows the effectiveness of the design.

In the literature, there is a wide variety of techniques for tuning controllers with computational intelligence, from those based on genetic algorithms [16,17], through ant colonies [18] to particle swarm optimization [19], to neural networks [20]. However, it is the versatility and speed of convergence of PSO that makes it a prime candidate for this work. Despite its qualities, the PSO algorithm also has some issues, among which the randomness in its exploration and the tendency to fall into local optimal, makes its speed of convergence to a solution be seriously affected, so it must be taken into account that the randomness of the algorithm can be modified in different ways [21], and thus adapt to the problem. In this case, the learning constants that make up the algorithm (inertia, best individual value, and best group value), are determinants. So for this work, it is proposed at first a system of multiple swarms where the communication is reduced, to avoid biasing the search of each swarm, and also a small value to the inertial constant, where the movement is given by the communication of best values, and not by the speed of the system. In works where the MPSO is used, the results are improved in the speed of convergence and in a wider exploration of candidates [22] by increasing the search clusters in the space. This increases the chances to search in more areas and different directions, as the learning constant related to the best value of the group will be now from each swarm, sharing information only towards the end of the algorithm where the one with the best results will be highlighted.

Based on the above statements, the motivation of this work consists of the use of the Multiple Particle Swarm Optimization-MPSO algorithm for the tuning of a PID control for power factor correction in an ac-dc boost converter. The main contribution is the use of MPSO techniques to offline tuning controllers, in particular to power converters. When MPSO is applied to tune the PID controller the PFC increases compared to previous results. Another advantage is the offline tuning, reducing the risk of damage in the converter, but obtaining a better performance.

The rest of the paper is organized as follows. In Section 1, preliminaries are discussed, which will explain the PID controller, the PSO, and the MPSO algorithm for the tuning of the controller, indicating the relationship between the search space of the algorithm and the constants of the PID controller and the basic operation of the boost converter, the power factor correction using a conventional controller and with a controller tuned by MPSO. Then, in Section 2 the synthesis of the simulation and its implementation are presented. Simulations and experimental results verifying the theoretical predictions are given in Section 3. Finally, in Sections 4 and 5 a brief discussion and conclusions are presented, respectively.

1.1. Main contributions of the paper

- The tuning of the PID controller by means of a machine learning algorithm allows to improve the behaviour, starting from a previous controller without the need for analytical knowledge of the system to be controlled.
- Offline tuning allows obtaining a better performance of a previous controller designed with no complex mathematical algorithms and avoids the risk of damage in the systems that could be presented in an online tuning.
- The system perturbations are monitored so that when there is an abrupt change, the system can train and adapt to this change, generating an appropriate controller.
- The problem of the local optimal present in the intrinsic behaviour of the algorithm is addressed with the proposal of multiple swarms without communication between them, in addition to a dispersion system in the case of showing no improvement after a certain number of epochs.

2. Preliminaries

2.1. PIDN constants

PID controllers constitute approximately half of the controllers implemented in industrial processes. Their high use is rooted in their applicability to most control

systems, simplicity, and outstanding results [23].

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, \mathrm{d}\tau + K_d \frac{\mathrm{d}}{\mathrm{d}t} e(t) \quad (1)$$

According to three simple parameters as seen in Equation (1), the PID controller can tune a system in an outstanding way, however, the PID controller has some problems at the time of its implementation in a real system, because the noise and non-linearity present in some systems such as the boost converter, can affect its performance, which is why a constant called k_N appears, where its function is to generate a filter to the derivative constant, preventing the growth of the error over time.

There is another problem by which the PID controller can have problems in a real system, in this case, it has to do with the integral action of the controller since when the controller is saturated, the error is maintained, resulting in large oscillations, in this work the limitation of the integral action or "clamping" will be used to avoid this problem.

2.2. PSO and MPSO algorithm

The PSO was developed by James Kennedy and Russell Eberhart in 1995 [24], and since then it has been observed how over time the PSO can be implemented to address problems related to optimization, it also allowed to relate the algorithm in fields of science and engineering [25,26].

$$\vec{v_{n+1}} = \omega \vec{v_n} + c_1 r_1 [\vec{pB_n} - \vec{p_n}] + c_2 r_2 [\vec{gB_n} - \vec{p_n}] \quad (2)$$

In Equation (2), the update of the velocity is described, where the inertial term is represented by this expression $\omega v_n(t)$ and ω is a constant that represents the inertial value. This expression $c_1 r_1 [\vec{pB_n} - \vec{p_n}]$ represents the cognitive component, and the latter expression $c_2 r_2 [\vec{gB_n} - \vec{p_n}]$ represents the social component. It should be noted that c_1 is the constant that rewards the personal experience of each particle and c_2 is the constant that rewards the collective experience of the swarm. Both represent acceleration constants that vary their value between 0 and 1. The PID controller and the MPSO algorithm are related according to the search space, where each particle must navigate in a space where the position is evaluated, the position of the particles is a vector composed of the number of dimensions of the space as well as the velocity. An implementation of MPSO is provided in this section as Algorithm 1. Due to the fact that the controller has 4 constants to tune, the search space of the particles corresponds to a space of 4 dimensions where each dimension corresponds to the different values of $[K_p, K_i, K_d, K_n]$, so the search space would be described as is shown in Equation (3).

$$S_s = \{\mathbb{R}^4\}, \quad \mathbb{R}^4 = \{(x_1, x_2, x_3, x_4) \in \mathbb{R}, \ x_1 = P,$$

Algorithm 1 Multi - PSO.

Inicialize Parameters: $\vec{c}$, dt, epochs, s, p, Lim
for $swarm = 1, 2, \ldots, s$ **do**
    for $particle = 1, 2, \ldots, n$ **do**
        Inicialize: $\vec{x}, \vec{v}$
    end for
end for

for $epochs = 1, 2, \ldots, epochs$ **do**
    for $swarm = 1, 2, \ldots, s$ **do**
        for $particle = 1, 2, \ldots, p$ **do**
            Evaluate particle position $\vec{x}$
        end for
    end for

    for $swarm = 1, 2, \ldots, s$ **do**
        Find Best Value
        Find Best Position $g\vec{B_n}$
    end for

    for $swarm = 1, 2, \ldots, s$ **do**
        for $particle = 1, 2, \ldots, p$ **do**
            Update Velocity with ??
        end for
    end for
end for
Select best Swarm behaviour: x_n

$$x_2 = I, \ x_3 = D, \ x_4 = N\} \quad (3)$$

2.3. Specifications of the converter

The values of the converter parameters as well as other specifications are shown in the Table 1. These were designed and carried out at the work [27], where an analytical procedure of the converter is performed to obtain the first controller tuned by conventional methods, the most important variable to take into account due to the nature of the converter is the useful cycle, which varies depending on the input voltage and the current flowing through the circuit.

Table 1. Specifications of the converter.

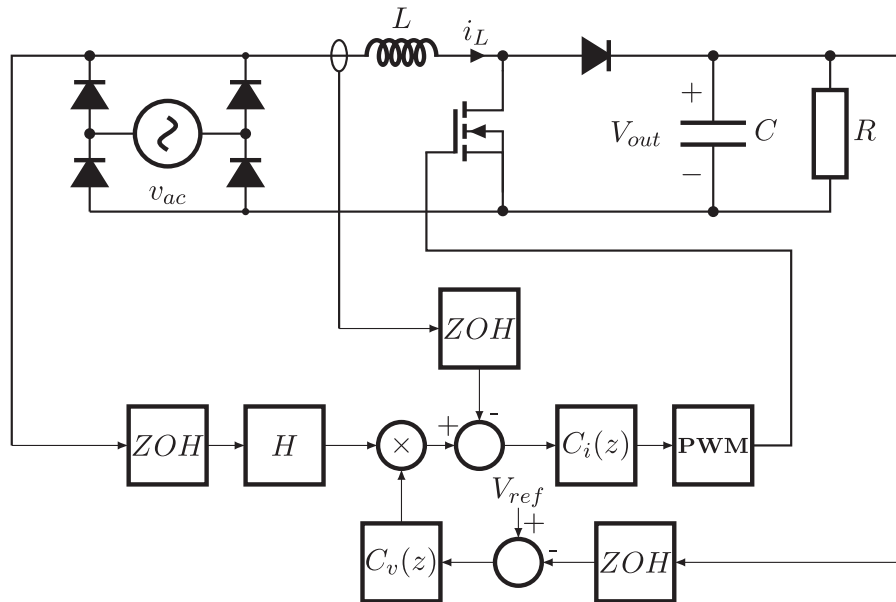| SL.NO	Specifications	Value
1	Input Voltage, V_{in}	120 V
2	Output Voltage, V_{out}	220 V
3	Output Resistance, R	500 Ω
4	Inductance, L	5.5 mH
5	Inductor ESR, R_L	0.4550 Ω
6	Capacitance, C	256 μF
7	Switch Resistance R_{sw}	0.36 Ω
8	Diode Resistance R_d	0.001 Ω
9	Diode Forward Voltage V_d	1.05 V
10	Duty Cycle, D	D
11	Switching Frequency, F_s	50 kHz

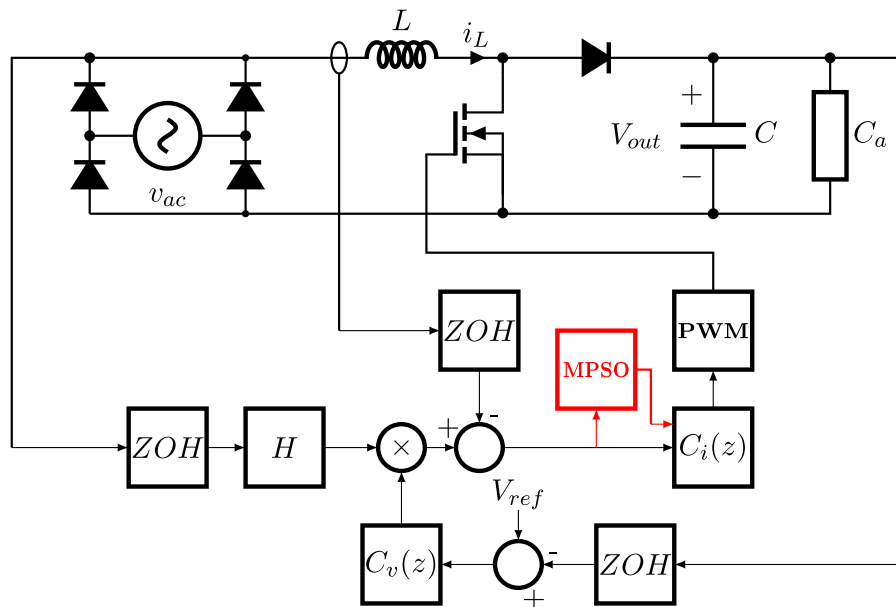**Figure 1.** Boost Converter C_{MAT} -(Reference controller) (AC/DC).

Figure 2. Boost Converter C_{MPSO} (AC/DC).

Figure 1 shows the boost converter that has a circuit for power factor correction (PFC), based on the average current control, where a double loop control is evident. Maintaining the output voltage V_{out} at the reference level V_{ref}) and for the current following the input voltage signal V_{in} as reference i_{ref}.

In the case of Figure 2, the boost converter is shown again with a controller for power factor correction, in this case, there is the MPSO algorithm located between the current error e_C and the current controller C_i, so that the algorithm delivers the constants K_p, K_i, K_d, K_n to tune the controller.

The design of the boost converter is presented in Ref. [27], which was complemented with the PSO method in the work [28].

3. Simulation model, control strategy and implementation

The converter is recreated in MATLAB/Simulink, with parameters as close as possible to the real model. The simulation consists of 3 main elements, where the boost converter, the voltage, and current controllers, and the tuning algorithm are located. There is also a visualization region for analysis and processing.

3.1. Simulation model

Figure 3 shows the boost converter located in the middle with 5 load variations automatically activated throughout the simulation time, with the objective of
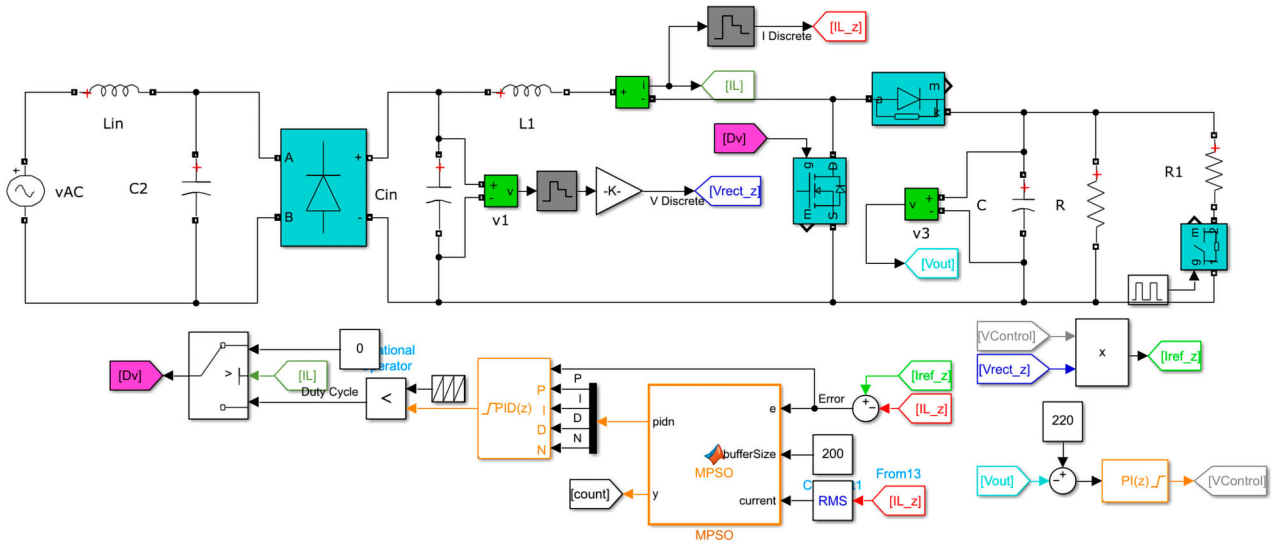
Figure 3. MATLAB/Simulink simulation of the AC/DC boost converter.

showing the disturbance rejection of the system once a controller is found by the MPSO.

The simulation has a total of 50 seconds of simulation where every 10 seconds a load is activated with a value of 500 Ω, 625 Ω, 833 Ω, 1000 Ω, and 2500 Ω respectively. The main blocks of the simulation are the boost converter, the control system distributed in adders, multipliers, the controllers themselves, and the elements of measurement of the power factor and other signals present in the circuit to be monitored.

3.2. Specifications of the algorithm MPSO

Table 2 shows the specifications of the algorithm MPSO. It is worth highlighting MPSO algorithm has the advantage that other bio-inspired technical in search of a PID-tuned control. Due to MPSO allows a search in multiple dimensions, with the purpose to obtain a response (The PID parameters in this case), by multiple searches, seeking to converge a response as close as possible to the global optimum. The number of epochs, swarm particles, and learning constants were obtained based on a previous simulated work whose results are presented in Ref. [28]. The other parameters correspond to digital control specifications, which are obtained based on an experimental way.

3.3. Control strategy

The 120 V_{RMS} signal that enters the converter (V_{in}), is rectified by a full wave bridge, this is the reference for the current controller (i_{ref}), where the angle between both signals ($\cos(\phi)$) will be as small as possible, so the first thing to do is to multiply i_{ref} by a value k which will normalize the peak voltage value ($V_{P_{in}}$) to the desired peak current value ($I_{P_{in}}$), in this case, the desired value is 1 A so k would be expressed as $k = i_{Pin}/V_{Pin}$, so that V_{in} is normalized to the value of I_{Pin}. However, the value

Table 2. Specifications of the algorithm MPSO.

SL.NO	Specifications	Value
1	Epochs n	50
2	Swarms s	5
3	Particles p	10
4	Learning Constants c	[0.05, 0.35, 0.75]
5	Search Delay d_s	0.5 s
6	Step d_t	0.5
7	Restart Epochs n_r	25
8	Error Buffer b	200
9	Searching Time t_s	10 s
10	Proportional Limit L_P	[0 − 10]
11	Integral Limit L_I	[0 − 10 × 10³]
12	Derivative Limit L_D	[0 − 1 × 10⁻⁶]
13	Derivative Filter Limit L_N	[0 − 100 × 10⁸]

of k cannot be constant, since depending on the output voltage (V_{out}), and the value of R at the output, the current will change, becoming smaller at lower load, so the value of k will depend on a voltage controller, where the error will be given by $e_v = V_{ref} - V_{out}$, where V_{ref} is the reference value of V_{out}, depending on the error in the signal, the controller at the output will have a signal of a value between 0 and 1 $C_V = [0, \ldots, 1]$ to multiply by k to keep the voltage reference when R changes. The error between i_{ref} and i_L is done by means of a Shunt resistor, having the same amplitude in the reading, the error between both will be only its phase so that the reading of the error for the controller will be $e_C = C_V i_{ref} - i_L$, the output of the current controller will be also a value between 0 and 1 $C_i = [0, \ldots, 1]$ which is the value of the useful cycle of the converter.

Considering the behaviour of the controllers, the current controller is the main one in the PFC power factor correction, the useful cycle of the converter is variable so depending on V_{in} and i_L, the value of D will change over time.

The constants of the current PID controller will be tuned with the MPSO algorithm, where the S_s is composed of the 4 constants (K_p, K_i, K_d, K_n). The algorithm detects the current variation Δi in the converter when

$i_L > i_L + \Delta i$ or $i_L < i_L - \Delta i$ a counter (d_s) is started with a value of 0.5 s, which waits for the converter to reach the value of V_{ref} (because the constant search cannot be performed during this time due to the action of the voltage controller, which could affect the e_C readings), when d_s ends, the MPSO algorithm starts. Where the different swarms (s) and particles (p) are initialized according to some limits (L_{PIDN}), the values of L_{PIDN} are extremely important to avoid local optima and instability, so in L_{PIDN} they are limited to only positive real numbers and according to the reference controller (C_{MAT}), the upper limit of L_{PIDN} corresponds to an integer close to the value of the corresponding constant, so the limits would be expressed as follows.

$$L_{PIDN} = \{([L_P], [L_I], [L_D], [L_N]) \in \mathbb{R}\},$$
$$L_P = \{0, \dots 10\}, \quad L_I = \{0, \dots 10 \times 10^3\},$$
$$L_D = \{0, \dots 1 \times 10^{-6}\}, \quad L_N = \{0, \dots 100 \times 10^8\} \tag{4}$$

It can be seen from the Equation (4) that L_{PIDN} in L_D and L_N, has values far from C_{MAT}, this is because there is a tendency to very small values in k_D and a tendency to very large values in k_N in C_{MAT}, so the limits are adjusted to the trend.

Once the different swarms (s) are initialized, C_{MAT} is placed as the position of the first particle of the first swarm, this to serve as a reference when communicating with the other members of the swarm, indicating a possible solution in the direction in which this particle is located. Once the first controller is placed, the error reading (e_C) is stored according to F_s, where e_C is recorded 200 times. The vector where the different values of e_C are stored is b and when b fills all the values, error is evaluated.

$$ISE = \int_0^\infty e^2(t) \, dt \tag{5}$$

For this work, the ISE error from Equation (5) is used, because in the work [11], it shows a better performance against other proposals in a similar problem.

Each p of each s is evaluated, where the position ($\vec{x}$) of each p is a point in S_s, where the performance (f) of $\vec{x}$, determines the best position in the history of each p ($p\vec{B}_n$) as observed in Equation (2), so it is understood that each p is a plant controller candidate, where $\vec{x}$ are the values of (K_p, K_i, K_d, K_n).

When the evaluation of all plant controller candidates is completed, the next step is to update the velocity ($\vec{v}$) and position ($\vec{x}$) vectors of each p, as seen in Equation (2), every step is repeated until the value of n epochs is reached, where the one that had the best f among all s is selected.

Thus, taking into account the specifications in Table 2 and solving the MPSO Algorithm 1, the PIDN controller gains are obtained.

$$K_p = 0.504 \quad K_i = 3000 \quad K_d = 4.1 \times 10^{-6}$$
$$K_n = 4.8 \times 10^6 \tag{6}$$

3.4. Implementation

The implementation of the system consists of 2 main elements, the boost converter and the processing device (DSP). The controller is discretized and reads the variables as shown in Figure 4, thus reading the errors and generating the PWM output.

For the reading of the signals of interest of the converter, a digital signal processor (DSP) was used together with a PWM output, the card used is from the C2000 line of Texas instrument microcontrollers line F28069M, which consists of an ADC with a resolution of 12 bits. There are 3 variables of interest in the system, the circuit current i_L, the input voltage V_{in}, and the output voltage V_{out}. Inside the card is implemented the control system which consists of the discretization of a PIDN controller, which has a trapezoidal discretized integrator and a derivative of inverse Euler form, in addition to an anti Wind up method. Finally, the controller has an input voltage V_{in} verification system, to
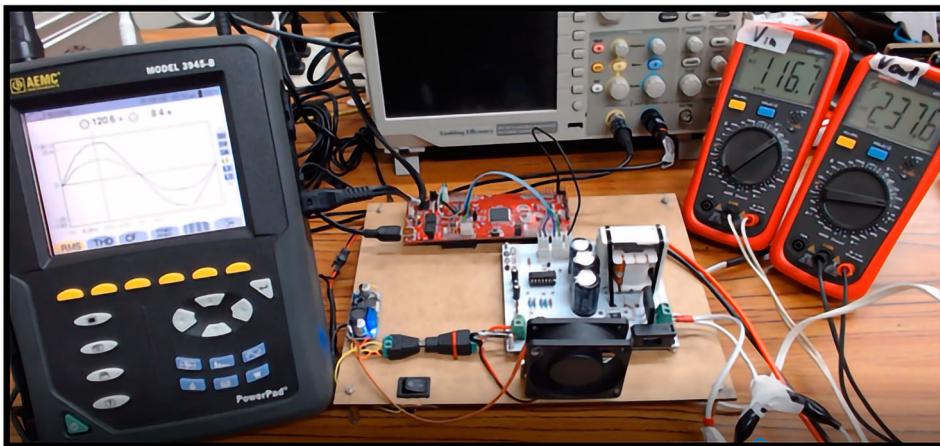

Figure 4. Experimental Setup.

have a smoothed behaviour in the variation of the useful cycle D when the system changes abruptly from an off to an on state.

4. Results

The results section is divided into two main elements, the simulation section where the algorithm is implemented and verified in an approximate model to the real one, also in this section it is compared with the conventional controller. The implementation section focuses on the results of the controller implemented in the real plant and the results obtained by using an electrical reading instrument.

At the end of each section, there is a comparative table, with the values of the power factor of each of the controllers in the variation of each of the loads, to demonstrate and analyse the controller tuned by the MPSO algorithm with respect to the conventional controller. The final result is that the controller tuned by the algorithm has better behaviour and better tracking of the reference signal, both in simulation and implementation.

4.1. Simulation

It can be observed from Figure 5(a), that the signals are out of phase in the zone close to zero, also a higher distortion level with respect to Figure 5(b), which corresponds to the C_{MPSO} controller. In this simulation, the amplitude values remain the same, test is also performed at 20% load to appreciate the tracking to the reference signal, by the current. Table 3 shows an improvement in power factor for the PSO-tuned controller, while the power factor of the C_{MAT} controller drops below 0.9, the C_{MPSO} controller remains above 0.9 at all times with minimal change in power factor as the load decreases.
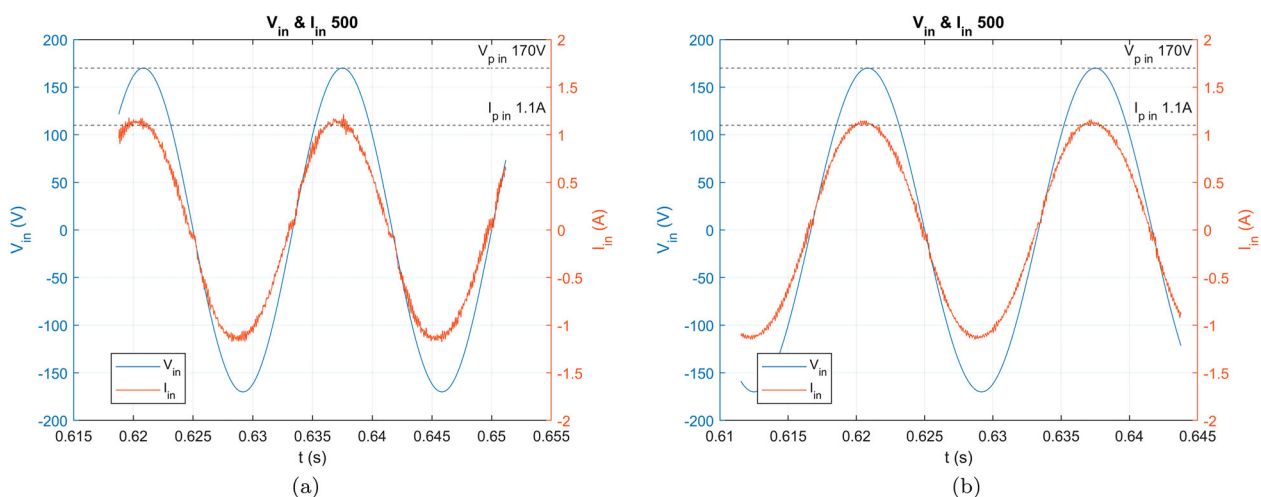
Table 3. Simulation results – comparative.

SL.NO	Resistance value	Power factor – C_{MAT}	Power factor – C_{MPSO}
1	500 Ω	0.99055	0.99680
2	625 Ω	0.98550	0.99510
3	833 Ω	0.97600	0.99135
4	1000 Ω	0.96650	0.98765
5	2500 Ω	0.86800	0.93150

4.2. Implementation

Figure 6 shows the results of the implementation of both controllers, where it can be seen that the Figure 6(a) controller has a greater phase shift with respect to the Figure 6(b) controller. In Figure 7, the load variations affect the power factor to a greater extent in Figure 7(a) with respect to Figure 7(b). The load was implemented with a total of 10 steps for different values. In Table 4, due to the precision of the measuring instrument, only 3 significant digits can be obtained in the measurement of each of the controllers, but still, it serves perfectly to demonstrate how the C_{MPSO} has a superior performance in terms of tracking the reference signal i_{ref}, also in terms of the power factor, which at only 20% load remains above 0.98. A classical PID controller was used due to this kind of controller was used in the previous work related [27], using an adequate number of parameters that allowed to control of the PF in a Boost, in the presence of uncertainties and other disturbances typical of this type of systems. On the other hand, Figure 8 depicts waveforms of the output voltage v_{out} of the ac/dc boost converter with C_{MPSO}, under nominal conditions and in the face of changes in load current. It should be noted that the voltage response has a very good settling time after each disturbance, thus guaranteeing an adequate regulation in 220 V of the converter output.

5. Discussion

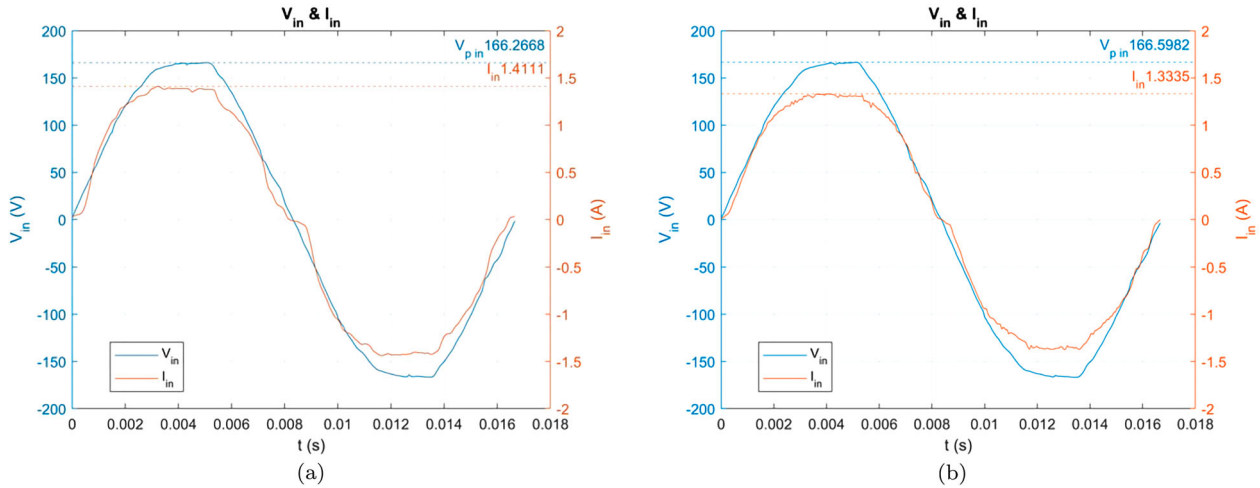This paper focuses on the study and implementation of a PID controller tuned through the MPSO online,

Figure 5. (a) C_{MAT} Controller. (b) C_{MPSO} Controller. Comparison of input voltage and current carried out in MATLAB/Simulink.

Figure 6. (a) C_{MAT} Controller. (b) C_{MPSO} Controller. Comparison of input voltage and current in a real plant.
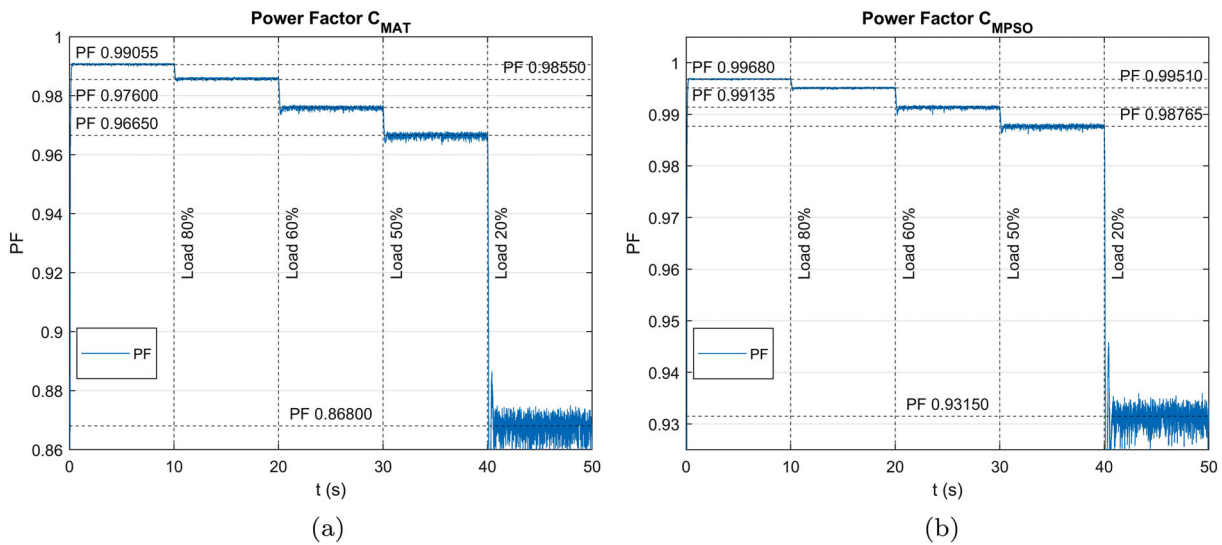

Figure 7. (a) Controller tuned by conventional methods, it can be observed that the power factor decays as the load decreases. (b) MPSO tuned controller, like the C_{MAT}, the controller also loses power factor as the load changes, however, it's tracking of the reference is much better, and therefore its power factor is also higher. Experimental response – power factor comparison with load variations at 80%, 60%, 50% and 20%.

carried out in a simulation of the system with parameters approximated to the real model, and where the constants of the controller are initially those that were tuned by a conventional analytical method (PID Tuner Matlab). Thanks to this reference controller, the limits of the navigation space of the particles (K_p, K_i, K_d, K_n) are determined, where the particles will navigate evaluating their position in this space according to a minimum number of samples of the discretized system, according to the existing sampling period in the signal. Finally, this tuned controller is tested in a real plant, comparing it with the PID controller tuned by conventional methods that serve as the reference controller of the system, the controller tuned by MPSO shows a better response against the inductor current tracking to the reference signal and a better rejection to disturbances.

Table 5 corresponds to the implementation of an experimental prototype of the boost converter with the PID controller tuned by an MPSO algorithm, to validate

Table 4. Implementation results – comparative.

SL.NO	Resistance value	Power factor – C_{MAT}	Power factor – C_{MPSO}
1	500 Ω	0.996	0.998
2	625 Ω	0.993	0.998
3	833 Ω	0.988	0.997
4	1000 Ω	0.976	0.997
5	2500 Ω	0.927	0.985

the performance of the PF. Table 5 presents a comparison of the proposed controller with work done by other authors. Within this work, in Ref. [29] an optimal utilization control of the switching cycles for a step-up converter operating in discontinuous conduction mode was proposed; a simple control scheme for single-phase power factor corrected regulators was presented in Ref. [30]; in Ref. [31], a simple method was presented to improve the dynamic response of the rectifier without affecting its steady state performance; in Ref. [32], an input-output linearization control strategy was proposed to achieve a high power factor and constant
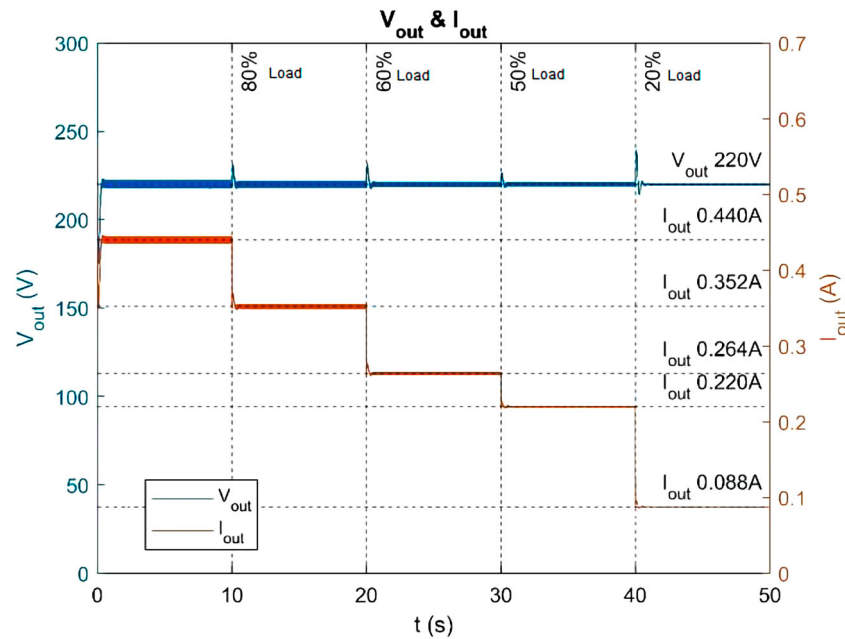
Figure 8. Voltage response v_{out} of the ac-dc boost converter under output current transients with C_{MPSO}.

Table 5. Comparison between the proposed controller and other works.

	[29]	[30]	[31]	[32]	This work
PF	0.975	0.996	0.996	0.99	0.9968
P$_o$(W)	120	150	150	200	100
V$_{AC}$(V)	175	115	110	100	120
V$_{out}$(V)	400	400	240	200	220

output voltage in a boost converter. From the results recorded in Table 5, it can be affirmed that the power factor of the proposed controller is higher than that obtained in other works so that the good performance obtained in the power factor of the PID controller tuned by an MPSO algorithm is ratified.

6. Conclusions

It is possible to tune a controller using a MPSO algorithm for power factor correction, from a simulation model, implement the resulting controller of that process, and that it improves the PF in the plant. Although the S_s search limits of this work are from a previous controller C_{MAT}, it is possible to find such limits, with an algorithm focused on exploration over communication. The PSO and MPSO algorithms show fast convergence and easy scalability, and the algorithm can be taken to the desired values of s and p without any problem. The value of T_s will depend directly on F_s and the higher the F_s, the lower the T_s; however, a digital processing element with a higher speed will be also necessary.

Starting from a PID pre-designed, MPSO offline tuning allows obtaining a controller with better performance (Table 4 shows the comparison with other studies). In this particular case is improving the Power Factor, avoiding risks associated with online tunings,

such as power peaks, shortcuts, or system destabilization induced by an incorrect controller selection. In future works, MPSO online algorithm will be used to tune PID controller in order to improve the PF in a boost converter, in light of variations in a nonlinear load.

ORCID

J. G. Guarnizo http://orcid.org/0000-0002-8401-4949
C. A. Torres-Pinzón http://orcid.org/0000-0003-0367-8143
J. Bayona http://orcid.org/0000-0001-6688-1988
L. Paredes-Madrid http://orcid.org/0000-0001-8609-0681

References

[1] Singh B, Singh BN, Chandra A, et al. A review of three-phase improved power quality AC-DC converters. IEEE Trans Ind Electron. 2004;51(3):641–660. doi: 10.1109/TIE.2004.825341
[2] Bouafassa A, Fernández-Ramírez LM, Babes B. Power quality improvements of arc welding power supplies by modified bridgeless SEPIC PFC converter. J Power Electron. 2020;20(6):1445–1455. doi: 10.1007/s43236-020-00143-2
[3] Bayona J, Gélvez N, Espitia H. Design, analysis, and implementation of an equalizer circuit for the elimination of voltage imbalance in a half-bridge boost

converter with power factor correction. Electronics. 2020;9(12):2171. doi: 10.3390/electronics9122171

[4] Redl R. Electromagnetic environmental impact of power electronics equipment. Proc IEEE. 2009;89(6): 926–938. doi: 10.1109/5.931490

[5] Van de Sype DM, De Gusseme K, Van den Bossche AP, et al. A sampling algorithm for digitally controlled boost PFC converters. IEEE Trans Power Electron. 2004;19(3):649–657. doi: 10.1109/TPEL.2004.826439

[6] Kamnarn U, Chunkag V. Analysis and design of a modular three-phase AC-to-DC converter using CUK rectifier module with nearly unity power factor and fast dynamic response. IEEE Trans Power Electron. 2009;24(8):2000–2012. doi: 10.1109/TPEL.2009. 2019575

[7] Wang L, Wu QH, Tang WH, et al. CCM-DCM average current control for both continuous and discontinuous conduction modes boost PFC converters. In: IEEE electrical power and energy conference; 2017. p. 1–6.

[8] Bouafassa A, Rahmani L, Kessal A, et al. Unity power factor converter based on a fuzzy controller and predictive input current. ISA Trans. 2014;53(6):1817–1821. doi: 10.1016/j.isatra.2014.08.001

[9] Bouafassa A, Rahmani L, Babes B, et al. Experimental design of a finite state model predictive control for improving power factor of boost rectifier. In: 2015 IEEE 15th international conference on environment and electrical engineering (EEEIC). Rome, Italy; 2015. p. 1556–1561.

[10] Guo Y. Design and analysis of controllers for boost converter using linear and nonlinear approaches. 1st ed. USA: Temple University; 2018.

[11] Kofinas P, Dounis AI. Online tuning of a PID controller with a fuzzy reinforcement learning MAS for flow rate control of a desalination unit. Electronics. 2019;8(2):231. doi: 10.3390/electronics8020231

[12] Durgadevi S, Umamaheswari MG. Analysis and design of single phase power factor correction using DC-DC SEPIC converter with bang-bang and PSO-based fixed PWM techniques. Energy Procedia. 2017;117:79–86. doi: 10.1016/j.egypro.2017.05.109

[13] Krishna SA, Abraham L. Boost converter based power factor correction for single phase rectifier using fuzzy logic control. In: 2014 first international conference on computational systems and communications (ICCSC); 2014.

[14] Sarmiento FHM, Gómez EJ, Ortiz MC. Power factor corrector with PID loop fit by genetic algorithm. Tecnura. 2013;17(36):10–17. doi: 10.14483/udistrital.jour. tecnura.2013.2.a01

[15] Mohammad A, Mohammed IA, Mohammed A-H, et al. Optimisation of PID controller employing PSO algorithm for interleaved buck-boost power electronic converter. Int J Ind Electron Drives. 2019;5(1):49–55. doi:10.1504/IJIED.2019.102385.

[16] Amaral JFM, Tanscheit R, Pacheco MAC. Tuning PID controllers through genetic algorithms. In: Proceedings of the 2001 WSES international conference on evolutionary computation. Vol. 2. No. 3; 2001.

[17] Jayachitra A, Vinodha R. Genetic algorithm based PID controller tuning approach for continuous stirred tank reactor. Adv Artif Intell. 2014;2014:Article ID 791230. doi: 10.1155/2014/791230

[18] Ibrahim HESA, Ahmed MSS, Awad KM. Speed control of switched reluctance motor using genetic algorithm and ant colony based on optimizing PID controller. In: ITM web of conferences. Vol. 16; 2018. p. 01001.

[19] Aouf A, Boussaid L, Sakly A. A PSO algorithm applied to a PID controller for motion mobile robot in a complex dynamic environment. In: International conference on engineering & MIS; 2017. p. 1–7.

[20] Guarnizo JG, Guacaneme JA, Trujillo CL. General inverse neural current control for buck converter. In: Novel algorithms and techniques in telecommunications, automation and industrial electronics; 2008. p. 117–122.

[21] Shafqat W, Malik S, Lee KT, et al. PSO based optimized ensemble learning and feature selection approach for efficient energy forecast. Electronics. 2009;10(18):2188. doi: 10.3390/electronics10182188

[22] Zheng X, Liu H. A different topology multi-swarm PSO in dynamic environment. In: AIEEE international symposium on IT in medicine & education. No. 2. Vol. 1; 2009. p. 790–795.

[23] Salem A, Hassan MM, Ammar ME. Tuning PID controllers using artificial intelligence techniques applied to DC-motor and AVR system. Asian J Eng Technol. 2014;2(02):129–138.

[24] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. Vol. 4; 1995. p. 1942–1948.

[25] Yoshida H, Kawata K, Fukuyama Y, et al. A particle swarm optimization for reactive power and voltage control considering voltage security assessment. IEEE Trans Power Syst. 2000;15(4):1232–1239. doi: 10.1109/59.898095

[26] Gaing ZL. Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE Trans Power Syst. 2003;18(3):1187–1195. doi: 10.1109/TPWRS.2003.814889

[27] Bayona J, Gélvez N, Espitia H. Design, analysis, and implementation of an equalizer circuit for the elimination of voltage imbalance in a half-bridge boost converter with power factor correction. Electronics. 2020;5(03):812–820. doi:10.3390/electronics9122171.

[28] Romero JP, Paez D, Noriega B, et al. Bio-inspired PSO technique applied to PID sintonization for power factor correction in a boost converter. In: IEEE 5th colombian conference on automatic control (CCAC); 2021.

[29] Yao K, Li Q, Lv J. DCM boost PFC converter with optimum utilization control of switching cycles. 2015 IEEE energy conversion congress and exposition (ECCE); 2015. p. 2048–2055.

[30] Jian-Min W, Hung-Chun C, Sen-Tung W, et al. Analysis and design of a boost PFC converter with sample-and-hold control techniques. 2015 Int Trans Electr Energy Syst. 2014;25(11): 3122–3138. doi:10.1002/etep.2026.

[31] Ghosh R, Narayanan G. A simple method to improve the dynamic response of single-phase PWM rectifiers. IEEE Trans Ind Electron. 2008;55(10):3627–3634. doi: 10.1109/TIE.2008.928113

[32] Gang L, Huilan H, Shaojian S, et al. A nonlinear control scheme based on input–output linearized method achieving PFC and robust constant voltage output for boost converters. Energy Rep. 2021;7(10):5386–5393. doi:10.1016/j.egyr.2021.08.169.