




Article

Fuzzy Logic Controller Parameter Optimization Using Metaheuristic Cuckoo Search Algorithm for a Magnetic Levitation System

Gabriel García-Gutiérrez ^{1,2,*} , Diego Arcos-Aviles ^{2,3,*} , Enrique V. Carrera ^{3,4} ,
Francesc Guinjoan ⁵, Emilia Motoasca ⁶, Paúl Ayala ^{2,3} and Alexander Ibarra ^{2,3}

¹ Carrera en Electrónica, Automatización y Control, Universidad de las Fuerzas Armadas ESPE, Av. Gral. Rumiñahui s/n, Sangolquí 171-5-231B, Ecuador

² Research group of Propagation, Electronic Control, and Networking (PROCONET), Universidad de las Fuerzas Armadas ESPE, Av. Gral. Rumiñahui s/n, Sangolquí 171-5-231B, Ecuador; jpayala@espe.edu.ec (P.A.); oaibarra@espe.edu.ec (A.I.)

³ Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas ESPE, Av. Gral. Rumiñahui s/n, Sangolquí 171-5-231B, Ecuador; evcarrera@espe.edu.ec

⁴ WICOM-Energy Research Group, Universidad de las Fuerzas Armadas ESPE, Av. Gral. Rumiñahui s/n, Sangolquí 171-5-231B, Ecuador

⁵ Department of Electronics Engineering, Escuela Técnica Superior de Ingenieros de Telecomunicación de Barcelona, Universitat Politècnica de Catalunya. C. Jordi Girona 31, 08034 Barcelona, Spain; francesc.guinjoan@upc.edu

⁶ Faculty of Engineering Technology, Department of Electrical Engineering, KU Leuven, TC Ghent 9000 Ghent, Belgium; emilia.motoasca@kuleuven.be

* Correspondence: gmgarcia3@espe.edu.ec (G.G.-G.); dgarcos@espe.edu.ec (D.A.-A.); Tel.: +593-2-398-9400 (ext. 1867)

Received: 18 April 2019; Accepted: 13 June 2019; Published: 16 June 2019



Abstract: The main benefits of fuzzy logic control (FLC) allow a qualitative knowledge of the desired system's behavior to be included as IF-THEN linguistic rules for the control of dynamical systems where either an analytic model is not available or is too complex due, for instance, to the presence of nonlinear terms. The computational structure requires the definition of the FLC parameters namely, membership functions (MF) and a rule base (RB) defining the desired control policy. However, the optimization of the FLC parameters is generally carried out by means of a trial and error procedure or, more recently by using metaheuristic nature-inspired algorithms, for instance, particle swarm optimization, genetic algorithms, ant colony optimization, cuckoo search, etc. In this regard, the cuckoo search (CS) algorithm as one of the most promising and relatively recent developed nature-inspired algorithms, has been used to optimize FLC parameters in a limited variety of applications to determine the optimum FLC parameters of only the MF but not to the RB, as an extensive search in the literature has shown. In this paper, an optimization procedure based on the CS algorithm is presented to optimize all the parameters of the FLC, including the RB, and it is applied to a nonlinear magnetic levitation system. Comparative simulation results are provided to validate the features improvement of such an approach which can be extended to other FLC based control systems.

Keywords: fuzzy logic controller; meta-heuristics; cuckoo search algorithm; magnetic levitation system

1. Introduction

Most problems in the real world are often very challenging to solve, and many applications have to deal with non-deterministic polynomial-time hard (i.e., NP-hard) problems. To solve these problems,

an optimization tool can be used even if there is no guarantee that the optimal solution can be obtained, techniques such as trial and error methods may help to find “acceptable solutions” [1,2]. In this regard, researchers have focused their efforts on developing algorithms capable of solving these problems in a more efficient manner.

Real-life applications found in engineering, business activities, industrial designs, statistics, decision making, optimal control, machine learning, etc., require optimization of (a set of) objective functions that can be anything, for example, minimization of energy consumption and costs, maximization of profit, output, performance, and efficiency, of a global optimization problem [3,4]. Moreover, optimization can be defined as a field of science in which the values of parameters of the problem are explored under specified conditions (constraints) in order to find those particular values which enable an objective or set of objective functions to generate the minimum or maximum value [5,6].

In general, an optimization problem can be formulated in terms of a correctly defined cost/objective function in which all the objectives of optimization related to the problem are involved. Moreover, if any predefined constraints are to be used with respect of the parameters of the related problem, these should be considered during the design of the optimization problem and its corresponding cost/objective function [7].

During the last decades, meta-heuristic nature-inspired algorithms have been given special attention due to their great capability in solving optimization problems in a wide range of applications such as operation and control of electric power systems [8–12], chemical processes [13–15], job scheduling [16–18], vehicle routing [19,20], autonomous vehicles control [21], mobile networking [22,23], multi-objective optimization [24–26], image processing [27], etc. Some of these algorithms found in the literature are cuckoo search, bat algorithm, particle swarm optimization, firefly algorithm, ant algorithms, genetic algorithm, differential evolution, etc. However, recent studies have shown that cuckoo search (CS) is potentially more efficient than other algorithms in solving optimization problems in terms of better-obtained solutions [8], its well-balanced intensification/diversification search strategies [3,28–30], robustness and precision of obtained results [7], and faster convergence speed [31].

Like almost all nature-inspired algorithms, the CS algorithm can be referred to as a meta-heuristic algorithm with two important features that make it so efficient at solving optimization problems, one is intensification or exploitation and the other is diversification or exploration. Intensification refers to focus on the search in a local region by exploiting information when a good solution has been found while diversification means the generation of diverse solutions far enough from good solutions to explore the search space on the global scale more efficiently [2,3,7,29,32,33].

On the other hand, in contrast to classical control theory that has been successful in controlling well defined deterministic systems, many engineering problems require more sophisticated control techniques due to the presence of strong non-linearities, embedded in a changing environment with uncertainty, or are difficult to model [34]. Fuzzy logic control (FLC) technique provides a formal methodology to represent, manipulate, compute and incorporate human intelligence directly into automatic control systems [34,35] offering potential advantages over conventional control schemes in less dependency on quantitative models, natural decision making, learning capability, a greater degree of autonomy, ease of implementation, and a friendly user interface [34,36–39]. By using the linguistic approach provided by the fuzzy theory [40], human knowledge about the system to be controlled can be integrated into control theory [41].

An analysis of the literature about FLC parameter optimization using nature-inspired CS algorithm reveals some works which will be briefly described next. In [8] CS is used to adjust the shape of the controller’s triangular and trapezoidal MF for the energy management of a hybrid power system. In [42] an image edge detection system based on a type-2 fuzzy is optimized by adjusting the shape of Gaussian MF. In [43], CS is used to adjust the shape of the fuzzy controller for liquid level control. In [11], a modified CS is used to tune MF’s positions of an FLC for a stationary battery storage system.

In this regard, the objective of this paper is to propose a methodology to optimize FLC parameters which involve not only the MF's positions but also the RB, which is the novelty of the presented approach, using for this purpose the CS algorithm. This procedure can be applied to any problem which involves a system where this control technique is required. The CS is applied to a FLC to achieve a better performance in comparison to conventional design and according to some predefined objectives. Referring to the magnetic levitation system (MLS), these objectives are steady-state vertical position, maximum overshoot, and settling time.

Based on the aim discussed above, the major contribution of this research is the new methodology for FLC parameter optimization using CS, including the RB as another degree of freedom, in such a way that a better solution for the problem to optimize is more feasible to achieve.

This paper is organized as follows: The mathematical model of the MLS system is detailed in Section 2. In Section 3, The FLC design to control the MLS is presented. This FLC optimization procedure using the CS algorithm is given in Section 4, with a detailed description of the algorithm used. Finally, the simulation results and discussions are presented in Section 5.

2. Magnetic Levitation System Modeling

A magnetic levitator is a system formed by electromagnets that allow ferromagnetic objects to remain in the air by means of a strong magnetic field opposite and equal to gravity. According to [44], the model of the MLS systems using differential equations is described as follows:

$$r' = v \quad (1)$$

$$v' = -\frac{F_{em}}{m} + g \quad (2)$$

$$i' = \frac{1}{f_i(r)} \cdot (k_i \cdot u + c_i - i) \quad (3)$$

$$F_{em} = i^2 \cdot \frac{F_{emP1}}{F_{emP2}} \cdot e^{\left(-\frac{r}{F_{emP2}}\right)} \quad (4)$$

$$f_i(r) = \frac{f_{iP1}}{f_{iP2}} \cdot e^{\left(-\frac{r}{f_{iP2}}\right)} \quad (5)$$

where $r \in [0, 0.016]$ (m) is the vertical position range of the object according to [44], v the velocity, i the current, and $f_i(r)$ is the electromagnet's inductance as function of the distance r . The other parameters that define the mathematical model of the MLS are: k_i , c_i are constants that determine the static characteristic of the current i versus the control signal u , F_{emP1} and F_{emP2} are constants for the electromagnetic force F_{em} calculation, f_{iP1} and f_{iP2} are constants to determine the inductance value $f_i(x_i)$, and, u represents the control signal. These equations include strong non-linearities and the control can be analytically designed either by a linearization approach, thus not dealing with large signal perturbations, or eventually by using complex nonlinear techniques.

For simulation purposes, a Simulink model representing—and the parameters defined in [44], has been implemented. Both, the control system structure and the Simulink representation of the MLS real model are illustrated in Figures 1 and 2, respectively.

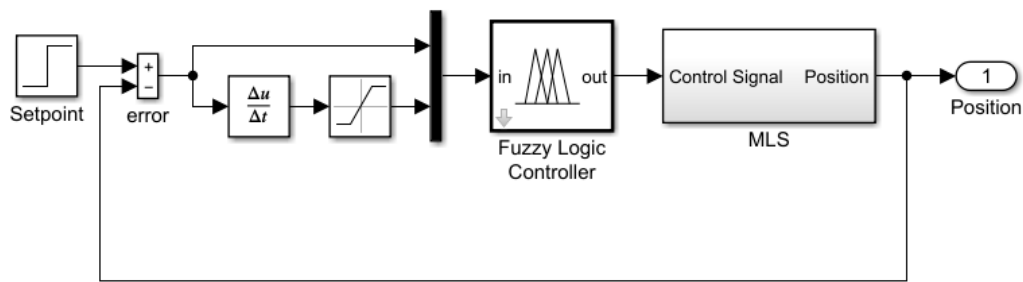


Figure 1. Control system structure for magnetic levitation system (MLS) using fuzzy logic control (FLC).

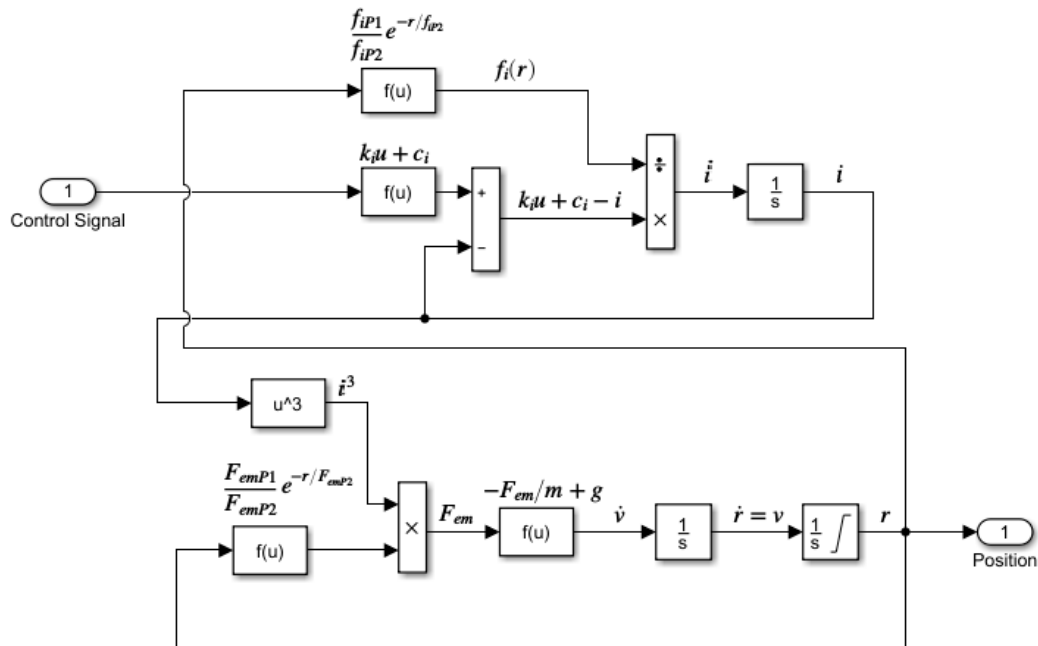


Figure 2. MLS model representation in Simulink.

3. Fuzzy Logic Controller Design

Due to the non-linear dynamics presented by the MLS [44], FLC is applied for maintaining a ferromagnetic sphere at a specific vertical position r , providing two advantages: A simple and robust non-linear control system and, an easy to implement controller since it is based on the linguistic description of the global system behavior [8]. The FLC consists of three main modules: The fuzzification process module which takes the inputs to the controller and produces a degree of membership for each fuzzy set in the membership function, the inference engine which takes these fuzzy set memberships and determines which rules should be evaluated, and the defuzzification process which takes the output values from the inference mechanism and translates them into an output dispatch signal [8,35,38,39]. Mamdani-based inference and defuzzification of the center of gravity are used in this paper.

A simple proportional-derivative (P-D) FLC type is chosen. The first input of the FLC is the position error, which represents the error between the desired sphere height and the actual sphere height. The second input of the FLC is the derivative of error, which indicates whether a positive/negative variation of the position is progressing or not. The controller will use this information to decide whether a high/low control action is needed to accomplish a predefined group of performance parameters or not. The output of the FLC is the control signal, which represents the control signal injected to the electromagnet of the MLS. The controller will use this information to decide a higher/lower control signal according to the value of its input.

The MF used for input and output data are shown in Figure 3 and the rules to be evaluated are given in Table 1, leading to a simple 25 rules FLC to reduce the controller computational complexity

and facilitate a subsequent implementation. At the beginning, following a current preset practice of the FLC parameters, five triangular MF, uniformly distributed, are assigned to the FLC inputs and output, whereas the initial RB is defined according to the heuristic knowledge of the MLS performance and corresponds to a standard P-D based FLC.

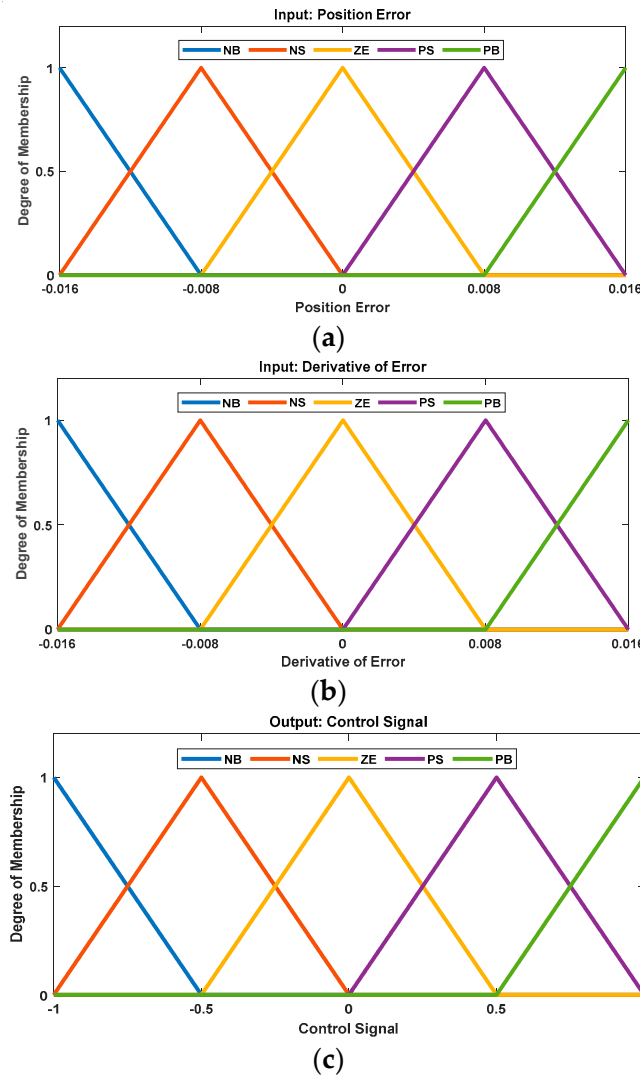


Figure 3. Membership functions (MF) used for inputs and output data: (a) Position error (input), (b) derivative of error (input), (c) control signal (output).

Table 1. Fuzzy logic initial rules for magnetic levitation system (MLS) control.

Control Signal	Position Error					
	NB	NS	ZE	PS	PB	
Derivative of error	NB	PB	PB	PS	PS	ZE
	NS	PB	PS	PS	ZE	NS
	ZE	PS	PS	ZE	NS	NS
	PS	PS	ZE	NS	NS	NB
	PB	ZE	NS	NS	NB	NB

For MF, “NB” represents the “Negative Big” fuzzy set, “NS” represents the “Negative Small” fuzzy set, “ZE” represents the “Zero” fuzzy set, “PS” represents the “Positive Small” fuzzy set and “PB” represents the “Positive Big” fuzzy set.

The CS algorithm used to optimize the MF for the input and output and the RB of the FLC are discussed next.

4. Fuzzy Logic Controller Parameter Optimization Using Cuckoo Search Algorithm

The CS algorithm is one of the newest developed nature-inspired metaheuristic algorithms and it is inspired in the obligate brood parasitism behavior of some cuckoo bird species which lay their own eggs in nests of other birds [3,29,45]. If host birds realize that a cuckoo laid an egg in their nests, they will either remove the egg from the nest or just abandon it and build a new one. In this regard, some cuckoo species have developed skills to imitate patterns and colors of specific host bird’s species increasing the probability of a cuckoo egg to survive [3,29,46].

The simplest approach of using new metaheuristic CS algorithm can be done through three ideal rules [3,29,47], which are: Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest. Each egg represents a solution. The best nests with high quality of eggs (i.e., better solutions) will carry over to the next generations. Additionally, the number of available host nests (i.e., number of solutions per generation) is fixed and the egg laid by a cuckoo is discovered by the host bird with a probability, $p_a \in [0, 1]$. In general terms, CS is used to minimize/maximize an objective function, $O(X)$, where X is a candidate solution of dimension, d , selected from a population of length, n . It is worth to remark that terms such as egg, nest, and solution represent the same in the context of the present study.

In short, CS algorithm involves two phases: At first, the initial random population is created and then, CS goes into an iterative process where two random walks, Lévy flights, and biased/selective random walk, are used to search for new solutions within the boundaries of the search space according to the problem restrictions. CS selects the best solution obtained in every iterative process according to the objective function value [3,29,48]. A detailed description of the different phases of the algorithm and its application for the optimization of the FLC described in Section 3 is exposed next and is illustrated in Figure 4.

4.1. Initialization

In the initialization phase, the CS algorithm initializes a set of solutions which are randomly sampled from the search space. For this purpose, the dimension of the solution and the length of the population must be defined.

The dimension of the solution, d , is defined according to the total number of parameters to optimize. In accordance with the FLC designed in Section 3, these parameters are related to the MF’s mapping and the RB. In this regard, both inputs (i.e., position error and derivative of error) and output (i.e., control signal) comprise five fuzzy sets and eleven parameters (since every fuzzy set is represented by a membership function of three variable positions except for both located at the maximum and minimum universe of discourse limits that have two fixed positions and one variable) whereas the RB includes twenty-five parameters. Considering all these values, each solution is represented with $d = 58$ parameters. The length of the population is fixed and a value of $n = 25$ is sufficient according to the literature [3]. In this regard, the initial random population is formulated as a $n \times d$ matrix where every row is a candidate solution of dimension d .

In order to preserve some order and symmetry in comparison with the MF’s mapping and RB distribution, every initial random solution conforming the initial random population, is generated according to the initial parameter distribution presented in Figure 3 and Table 1, as follows:

$$X_{Gi} = [X_{MF_i}, X_{RB_i}] \tag{6}$$

$$X_{MF_i} = X_{MF(I)} + R \tag{7}$$

$$R = [U(-5\% \cdot R_{Imp(1)}, 5\% \cdot R_{Imp(1)}), U(-5\% \cdot R_{Imp(2)}, 5\% \cdot R_{Imp(2)}), U(-5\% \cdot R_{Out}, 5\% \cdot R_{Out})] \tag{8}$$

$$X_{RB_i} = X_{RB(I)} + U(-1, 1) \tag{9}$$

where X_{MF_i} represents a candidate initial solution for the MF’s mapping sampled randomly from the search space which is limited according to the problem restrictions (i.e., inputs and output range

values of FLC designed in Section 3) and of dimension equal to the total number of MF parameters to optimize, $X_{MF(i)}$ represents initial membership function's parameters according to Figure 3, R is a set of displacements constituted of three different subsets that correspond to vectors of random numbers drawn from a uniform distribution where R_{Inp} , R_{Inp} , and R_{Out} are the defined range for every input/output variable, respectively, and a design value of 5% has been defined with the aim that randomly generated solutions result rather similar to initial distributed ones (Figure 3), X_{RBi} represents a candidate initial solution for the RB sampled randomly from the search space which is limited according to the problem restrictions (i.e., linguistic labels corresponding to fuzzy sets whose MF are depicted in Figure 3) and of dimension equal to the total number of RB parameters to optimize, $X_{RB(i)}$ represents initial RB parameters according to Table 1, $U(-1, 1)$ is a set of integer rule-base displacements drawn from a uniform distribution, and X_{Gi} is a candidate initial solution (nest) comprising of X_{MFi} and X_{RBi} at an initial generation G_i .

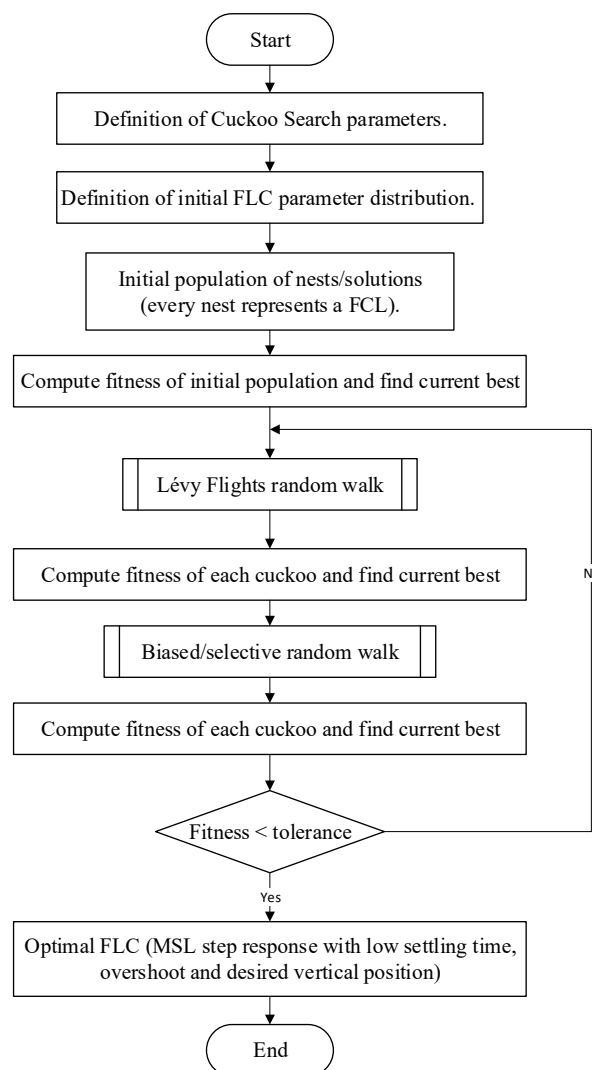


Figure 4. Flowchart of the optimization process.

Once the initial random population is created, the CS algorithm evaluates every candidate solution, so the best ones obtained a pass to the next generations. It is worth pointing out that, during a minimization/maximization process, the 'selection of best' step results crucial so the best solution according to problem's objectives of optimization can be obtained at the end of the algorithm.

4.2. Lévy Flights Random Walk

After the initialization and with the information that possible good solutions were obtained in a previous generation, CS starts an intensification (local search) process so that better solutions can be obtained. CS intensification is carried out using a Lévy flights random walk with a step-size drawn from a Lévy distribution that is characterized by its infinite mean and variance, enabling CS to explore the search space more efficiently than other algorithms that use a standard Gaussian process. At a generation $G > 0$, a new solution is generated using a random walk according to the following expression [3,29]:

$$X_{MF(G+1)} = X_{MF(G)} + \alpha \cdot Levy(\beta) \tag{10}$$

$$\alpha = \alpha_0 \cdot (X_{MF(G)} - X_{MF(best)}) \tag{11}$$

where $X_{MF(G+1)}$ is a candidate new solution for MF’s parameters obtained after the intensification process, $X_{MF(G)}$ is a candidate solution at a previous generation G selected from the cuckoo population, $\alpha > 0$ is a set of step-sizes that are related to the scale of the problem [3,29], $Levy(\beta)$ is a set of random numbers drawn from a Lévy distribution, $X_{MF(best)}$ is the best solution for MF’s parameters obtained so far in the iterative process and together with RB parameters, better optimizes the objective function, and α_0 is a scaling factor that generally is fixed and set to 0.001 or 0.1 although a varying scaling factor has shown better performance for most optimization problems [48]. In the present work, a varying scaling factor drawn from a uniform distribution is employed and formulated as follows:

$$\alpha_0 \sim U(0.01, 0.1) \tag{12}$$

In the implementation, Lévy flights are calculated via the so-called Mantegna’s algorithm for a symmetric Lévy stable distribution, as follows [3,29].

$$Levy(\beta) = \frac{\phi \cdot w}{|v|^{1/\beta}} \tag{13}$$

$$\phi = \left[\frac{\Gamma(1 + \beta) \cdot \sin\left(\frac{\pi \cdot \beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot 2^{\frac{\beta-1}{2}}}\right]^{1/\beta} \tag{14}$$

where $\beta = 1.5$ is the constant of the Lévy distribution [3], w and v are random numbers drawn from a normal distribution with zero mean and unit standard deviation, and Γ is the gamma function [3,29,49].

It is worth pointing out that the aforementioned procedure is applied only to generate new solutions for the MF’s mapping. Therefore, new solutions for the RB are calculated as follows.

$$X_{RB(G+1)} = X_{RB(G)} + S \cdot (X_{RB(G)} - X_{RB(best)}) \tag{15}$$

where $X_{RB(G+1)}$ is a candidate new solution for RB parameters obtained after the intensification process, $X_{RB(G)}$ is a candidate solution at a previous generation G selected from the cuckoo population, S is set of integer numbers drawn from a normal standard distribution $N(0, 1)$ to displace the RB values, and $X_{RB(best)}$ is the best solution for RB parameters obtained so far in the iterative process and together with membership function’s parameters, better optimizes the objective function.

It is worth pointing out that intensification is applied to every solution of the population and a solution is replaced by another one or carried through the next generations, according to the following expression:

$$X_{G+1} = \begin{cases} X_{G+1}; & O(X_{G+1}) < O(X_G) \\ X_G; & otherwise \end{cases} \tag{16}$$

where $O(X)$ is the objective of optimization function, and X_G/X_{G+1} are candidate solutions comprising of $X_{MF(G)}/X_{MF(G+1)}$ and $X_{RB(G)}/X_{RB(G+1)}$, respectively.

4.3. Biased/selective Random Walk

In this step, new solutions far enough from the current best solutions are discovered by using far-field randomization [29]. First, a random probability value of $p_a \in [0, 1]$ is assigned to every solution in the cuckoo population after the intensification process and then biased/selective random walk is implemented if the individual solution's assigned probability fits the condition $p_a \leq 0.25$. In this regard, a trial solution is built using a mutation of a candidate solution selected from the population and a differential step size from two stochastically selected solutions and then, a new solution is generated by a crossover operator from the candidate current solution and the trial solutions [48]. This procedure is formulated as follows:

$$X_{G+1} = \begin{cases} X_G + r \cdot (X_{p(G)} - X_{q(G)}); & p_a \leq 0.25 \\ X_G; & p_a > 0.25 \end{cases} \quad (17)$$

where X_{G+1} is a candidate new solution for MF and RB parameters obtained after the diversification process, X_G is a candidate solution at a previous generation G selected from the cuckoo population after the intensification process, p and q are random indexes so $X_{p(G)}$ and $X_{q(G)}$ are randomly selected solutions from the cuckoo population, $r \sim U(0, 1)$ is a random number drawn from a uniform distribution, and $p_a = 0.25$ is the solution discovery rate used for most optimization problems according to the literature [3,48,50]. It is worth pointing out that with a value of $p_a = 0.25$, the global search is slightly intensive with about 1/4 of the search time, while local search takes about 3/4 of the total search time. However, this parameter can be modified depending on the application to increase the probability to find an optimum solution globally or locally [8].

5. Results and Discussion

Tuning procedure of fuzzy design variables using the CS algorithm presented in this paper permitted not only to eliminate the requirement of expertise needed for setting these variables but also to obtain an accurate FLC that enables MLS to perform according to a sort of pre-defined parameters. The CS algorithm used MLS modeling to simulate the real performance of the system with an 8 mm height set point and tuned the FLC according to MLS desired performance parameters. In this regard, the typical parameters of a second-order linear system are overshoot, settling time and steady-state output value. Overshoot represents the maximum response value (extremum), whereas settling time is the time elapsed until the system response remains bounded around the set point tolerance usually between $\pm 5\%$. The objective function and a minimum tolerance value as a finishing condition of the optimization process, were defined as:

$$O = \alpha_1 \frac{|\%M_{p,D} - \%M_p|}{100} + \alpha_2 \frac{|t_{s,D} - t_s|}{10} + \alpha_3 |r_{ss,D} - r_{ss}| \cdot 100 \quad (18)$$

$$tolerance = 1\% \cdot (\%M_{p,D} + t_{s,D} + r_{ss,D}) \quad (19)$$

where $\%M_{p,D}$ is the desired percentage overshoot, $\%M_p$ is the actual percentage overshoot, $t_{s,D}$ is the desired settling time, t_s is the actual settling time, $r_{ss,D}$ is the desired final position, r_{ss} is the actual final position, and α_x is the weight of each component. Note that $\alpha_1 = \alpha_2 = \alpha_3 = 1$ are used in this paper to prioritize the minimization of every objective equally. In this regard, the aim is to minimize the deviation of every performance parameter from its respective goal.

For purposes of comparison, both optimization procedures: For MF and RB, and with only MF have been implemented. On one hand, the optimization process of MF and RB took approximately 8 h on an Intel(R) Core(TM) i7-4510U CPU (2.00GHz) 12 GB of RAM computer using Matlab® program (Matlab 2016b, MathWorks®, Natick, MA, USA, 2016), whereas most of the computation time is spent on solving the 40 iterations required to reach the optimum solution that minimizes the objective function according to the problem restrictions. On the other hand, the optimization process of only MF

took approximately 12 h on in the same simulation platform, whereas most of the computation time is spent for solving the 60 iterations required to reach the optimum solution that minimizes the objective function according to the problem restrictions. In this regard, Figure 5 illustrates the average objective function convergence for ten runs of each algorithm showing that optimizing MF and RB may lead to a faster convergence speed of the optimization process.

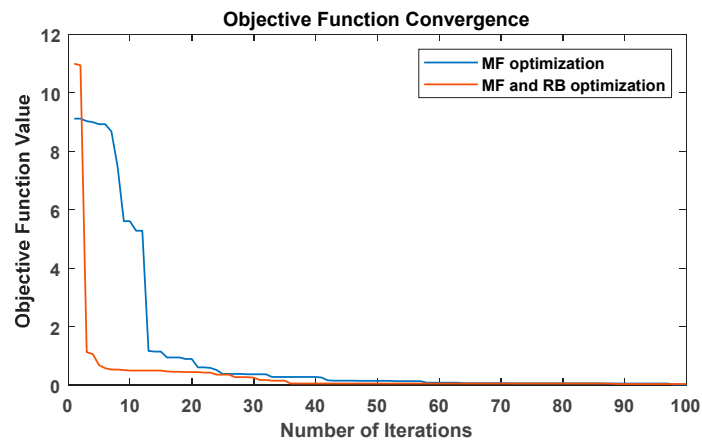
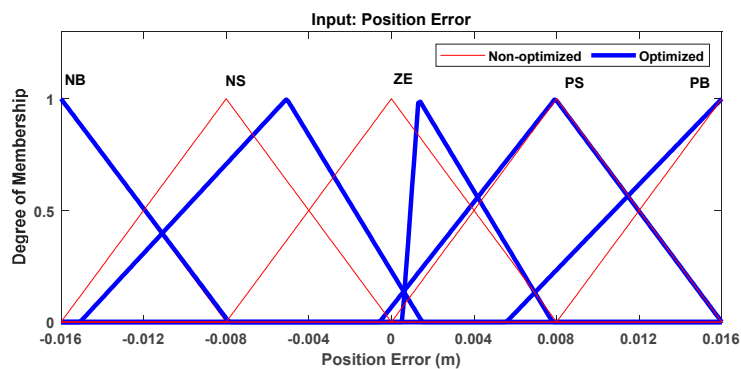
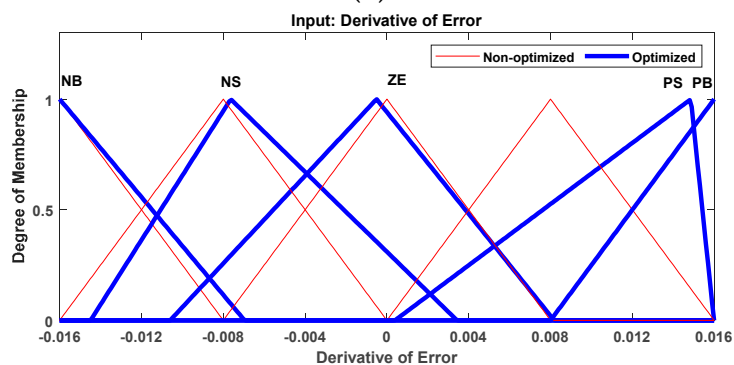


Figure 5. Objective function convergence.

In Figure 6, the optimal fuzzy MF encountered by the optimization algorithm are superimposed over non-optimized MF. The optimized MF maintain order and coherency with regards to the initial distribution illustrated in Figure 3, which was one of the primary objectives. Furthermore, as is shown in Table 2, the optimized RB distribution has been displaced in comparison to the initial RB shown in Table 1.



(a)



(b)

Figure 6. Cont.

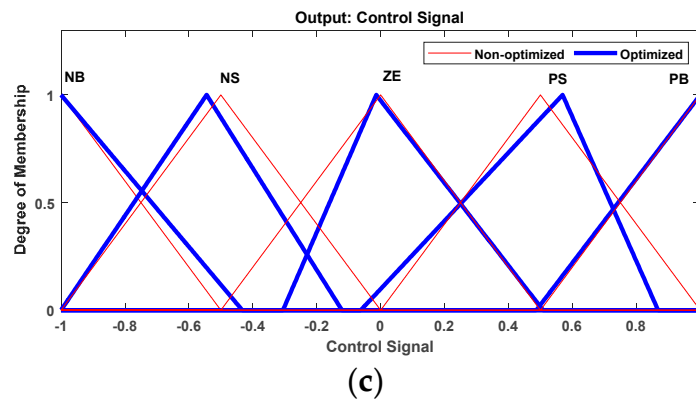


Figure 6. Optimized and non-optimized MF used for inputs and output data: (a) Position error (input), (b) derivative of error (input), (c) control signal (output).

Table 2. Fuzzy Logic optimized rule base (RB) for MLS control.

Control Signal	Position Error					
	NB	NS	ZE	PS	PB	
Derivative of error	NB	PB	PB	PB	PS	PS
	NS	PB	ZE	PS	NS	NB
	ZE	ZE	NS	NS	NS	NB
	PS	NB	NS	NB	NB	NB
	PB	NB	NS	NB	NB	NB

As aforementioned, FLC is tuned using CS to enable MLS to accomplish a specific settling time, overshoot, and steady-state vertical position. Moreover, MLS performance with optimized FLC can be compared with linear controllers designed from the system’s linearized model around the desired steady-state. For this purpose, and in order to fulfill as close as possible the required performance parameters, full-state feedback controller referred to as “state-space (SS) controller” has also been designed. In this concern, it should be pointed out that, instead of the FLC which only needs to sense the position of the sphere, the SS controller should also require sensing both the velocity and the electrical current.

It is worth pointing out that the main reason to optimize both the RB and the MF’s mapping is to reach the optimal minimum of the objective function given by the overall FLC processing in comparison to an only MF’s parameter optimization procedure, as in [8,11,42,43]. In this regard, both optimization procedures (i.e., MFs and RB optimization and only MFs optimization) have been applied to the FLC of the MLS (MLS-FLC). The obtained solutions of the aforementioned optimization procedures are compared with the MLS using a non-optimized FLC and the MLS with a full state feedback controller in terms of MLS’ performance parameters. MLS real model representation described in Section 2 is used for simulations.

The objective of optimization was to accomplish a specific set of performance parameters for MLS (i.e., $\%M_{p,D} = 5\%$, $t_{s,D} = 0.5\text{ s}$, $r_{ss,D} = 0.008\text{ m}$) by minimizing the objective function defined by (18). MF and RB optimization optimum solution shows a smaller deviation value of every defined goal in comparison with an only MF optimization. Both optimization procedures’ results are also compared with the performance of MLS with a SS controller. In this regard, according to resulting values listed in Table 3, MF and RB optimization proposed in this paper converges to a better solution which results in a set of slightly better performance parameters in comparison with only MF optimization and the results are close to the ones obtained using an analytic design approach (i.e., full state feedback controller). Furthermore, simulation results comparison is shown in Figure 7.

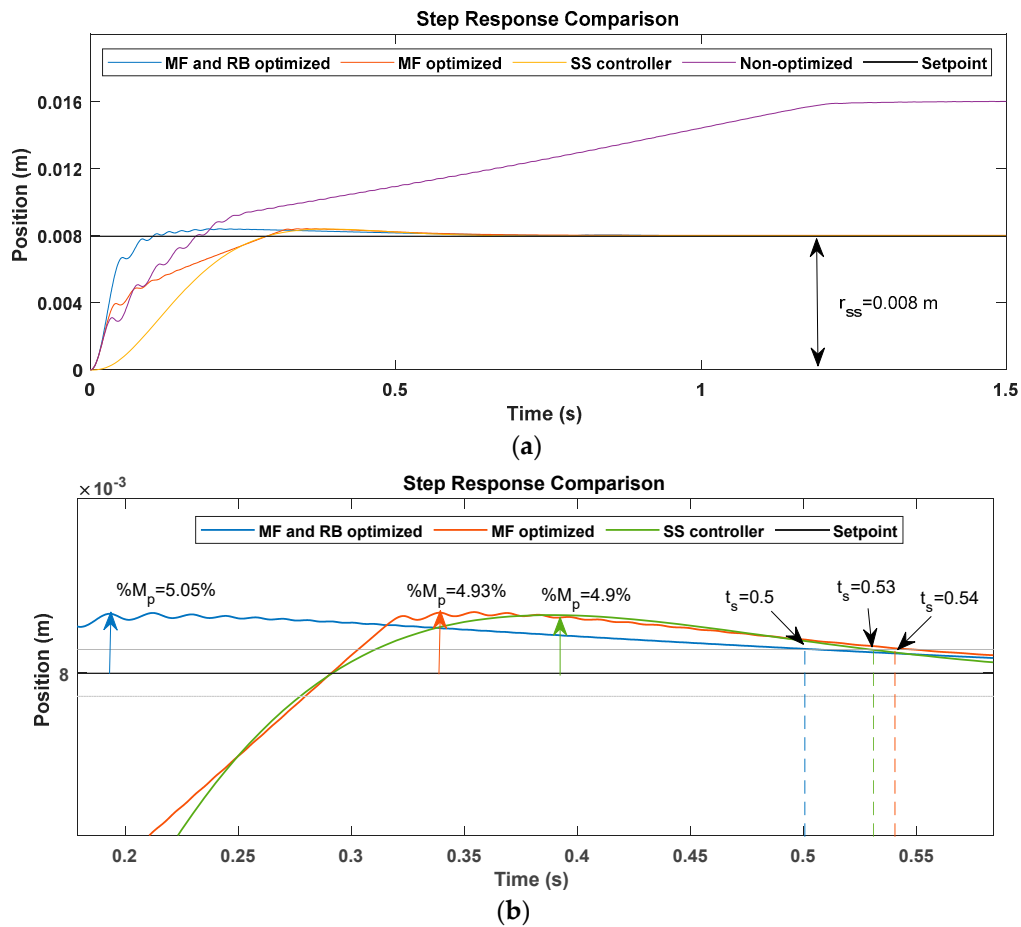


Figure 7. Fuzzy logic control of the MLS (MLS-FLC) with only MF’s positions optimized, with membership function (MF) and rule base (RB) optimized, non-optimized and MLS with state-space (SS) controller step response: (a) Steady-state response, (b) dynamic response.

Table 3. Comparison results for the non-optimized fuzzy logic control (FLC), the optimized FLC, and the state-space (SS) controller.

Desired		Non-Optimized		MF Optimized		MF and RB Optimized		SS Controller	
		Value	Deviation	Value	Deviation	Value	Deviation	Value	Deviation
r_{ss} (m)	0.008	Inf	Inf	0.008	0	0.008	0	0.008	0
M_p (%)	5	0	0	4.93	0.09	5.05	0.05	4.9	0.1
t_s (s)	0.5	Inf	inf	0.54	0.04	0.5	0	0.53	0.03

Table 4 presents the objective function values in terms of MLS performance parameters for the MF and RB optimized FLC, the MF optimized FLC, and the SS controller. Notice that the objective global optimum value obtained with the MF and RB optimization is approximately half the obtained value with only MF optimization and closer to the SS controller approach, being a better solution the one with its objective function value closer to zero.

Table 4. Objective function values comparison.

	Objective Function Value
MF optimized	0.0047
MF and RB optimized	0.0005
SS Controller	0.0040

It is worth pointing out that the analytic design of any classic or modern controller for a non-linear system (i.e., MLS) either by a linearization approach, thus not dealing with large signal perturbations, or eventually by using complex nonlinear techniques, requires a certain level of expertise for system parameter identification and the development of complex system modeling, resulting in a long implementation time. Consequently, by using the presented approach, these requirements are not completely necessary to obtain similar results than other complex control techniques.

6. Conclusions

A procedure for optimizing the parameters of an FLC by means of cuckoo search algorithm has been presented in this paper. The main advantage of this approach is that little or no expertise for tuning FLC parameters is needed since by using CS as an optimization algorithm, a global convergence to the optimum FLC according to the objective function is assured. An initial symmetrically defined set of MF and RB has been required to obtain an optimum FLC with order and coherency of parameters. By tuning the FLC MF's mapping and RB distribution, the proposed approach has successfully reached the optimization objectives such as settling time, overshoot and steady-state vertical position for an FLC-MLS. The obtained results have highlighted that the proposed algorithm can be applied to any system where the FLC technique is used, to optimize an objective function involving several system parameters.

Current work is focused on applying the presented algorithm to improve the performance of a fuzzy-based energy management strategy for a residential grid-connected microgrid considering energy state forecasting, peak shaving, and demand management.

Author Contributions: Conceptualization, G.G.-G., D.A.-A. and E.C.; data curation G.G.-G.; formal analysis D.A.-A., E.C., F.G., E.M. and P.A., and A.I.; investigation G.G.-G. and D.A.-A.; methodology G.G.-G., D.A.-A., E.C., and F.G.; project administration D.A.-A.; software G.G.-G.; supervision D.A.-A. and E.C.; validation D.A.-A. and E.C.; visualization P.A. and A.I.; writing—original draft preparation G.G.-G. and D.A.-A.; writing—review and editing D.A.-A., E.C., F.G., E.M., and P.A.

Funding: This work is part of the project 2016-PIC-044 and 2019-PIC-003-CTE from the Research Group of Propagation, Electronic Control, and Networking (PROCONET) of Universidad de las Fuerzas Armadas ESPE. This work has been partially supported by the Spanish Ministry of Industry and Competitiveness under the grant DPI2017-85404.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fister, I., Jr.; Yang, X.S.; Fister, I.; Brest, J.; Fister, D. A Brief Review of Nature-Inspired Algorithms for Optimization. *Electrotech. Rev.* **2013**, *80*, 1–7.
2. Duarte, M. *Metaheurísticas*; S.L.—DYKINSON: Madrid, Spain, 2007.
3. Yang, X.-S. *Nature-Inspired Optimization Algorithms*, 1st ed.; Elsevier: Amsterdam, The Nederland, 2014.
4. Sergeev, Y.D.; Kvasov, D.E.; Mukhametzhanov, M.S. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. *Sci. Rep.* **2018**, *8*, 453. [[CrossRef](#)] [[PubMed](#)]
5. Arora, S.; Singh, S. A conceptual comparison of firefly algorithm, bat algorithm and cuckoo search. In Proceedings of the International Conference on Control, Computing, Communication and Materials (ICCCCM), Allahabad, India, 3–4 August 2013; pp. 1–4.
6. Deepthi, S.; Ravikumar, A. A Study from the Perspective of Nature-Inspired Metaheuristic Optimization Algorithms. *Int. J. Comput. Appl.* **2015**, *113*, 53–56.
7. Civicioglu, P.; Besdok, E. A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artif. Intell. Rev.* **2013**, *39*, 315–346. [[CrossRef](#)]
8. Berrazouane, S.; Mohammedi, K. Parameter optimization via cuckoo optimization algorithm of fuzzy controller for energy management of a hybrid power system. *Energy Convers. Manag.* **2014**, *78*, 652–660. [[CrossRef](#)]

9. Taruwona, M.; Nyirenda, C.N. Particle Swarm Optimization of a Mamdani Fuzzy Logic Based Charge Controller for Energy Storage Systems. In Proceedings of the Open Innovations Conference (OI), Johannesburg, South Africa, 3–5 October 2018; pp. 73–78.
10. Elazim, S.A.; Ali, E.S. Optimal Power System Stabilizers design via Cuckoo Search algorithm. *Int. J. Electr. Power Energy Syst.* **2016**, *75*, 99–107. [[CrossRef](#)]
11. Truong, C.N.; May, D.C.; Martins, R.; Musilek, P.; Jossen, A.; Hesse, H.C. Cuckoo-search optimized fuzzy-logic control of stationary battery storage systems. In Proceedings of the IEEE Electrical Power and Energy Conference (EPEC), Saskatoon, SK, Canada, 22–25 October 2017; pp. 1–6.
12. Abdelaziz, A.Y.; Ali, E.S. Cuckoo Search algorithm based load frequency controller design for nonlinear interconnected power system. *Int. J. Electr. Power Energy Syst.* **2015**, *73*, 632–643. [[CrossRef](#)]
13. Bhargava, V.; Fateen, S.E.; Bonilla-Petriciolet, A. Cuckoo Search: A new nature-inspired optimization method for phase equilibrium calculations. *Fluid Phase Equilibria* **2013**, *337*, 191–200. [[CrossRef](#)]
14. Teimouri, R.; Sohrabpoor, H. Application of adaptive neuro-fuzzy inference system and cuckoo optimization algorithm for analyzing electro chemical machining process. *Front. Mech. Eng.* **2013**, *8*, 429–442. [[CrossRef](#)]
15. Fateen, S.E.; Bonilla-Petriciolet, A. Unconstrained Gibbs Free Energy Minimization for Phase Equilibrium Calculations in Nonreactive Systems, Using an Improved Cuckoo Search Algorithm. *Ind. Eng. Chem. Res.* **2014**, *53*, 10826–10834. [[CrossRef](#)]
16. Babukartik, R.G.; Dhavachelvan, P. Hybrid Algorithm using the advantage of ACO and Cuckoo Search for Job Scheduling. *Int. J. Inf. Technol. Conver. Serv.* **2012**, *2*, 25–34. [[CrossRef](#)]
17. Khadwilard, A.; Chansombat, S.; Thepphakorn, T.; Thapatsuwan, P.; Chainate, W.; Pongcharoen, P. Application of Firefly Algorithm and Its Parameter Setting for Job Shop Scheduling. *J. Ind. Technol.* **2012**, *8*, 49–58.
18. Bitam, S. Bees Life Algorithm for Job Scheduling in Cloud Computing. In Proceedings of the Second International Conference on Communications and Information Technology, Hammamet, Tunisia, 26–28 June 2012; pp. 186–191.
19. Bitam, S.; Mellouk, A.; Zeadally, S. Bio-Inspired Routing Algorithms Survey for Vehicular Ad Hoc Networks. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 843–867. [[CrossRef](#)]
20. Marinakis, Y.; Iordanidou, G.R.; Marinaki, M. Particle Swarm Optimization for the Vehicle Routing Problem with Stochastic Demands. *Appl. Soft Comput.* **2013**, *13*, 1693–1704. [[CrossRef](#)]
21. Mohanty, P.K.; Parhi, D.R. Optimal path planning for a mobile robot using cuckoo search algorithm. *J. Exp. Theor. Artif. Intell.* **2016**, *28*, 35–52. [[CrossRef](#)]
22. Mehboob, U.; Qadir, J.; Ali, S.; Vasilakos, A. Genetic algorithms in wireless networking: Techniques, applications, and issues. *Soft Comput.* **2016**, *20*, 2467–2501. [[CrossRef](#)]
23. Roy, B.; Banik, S.; Dey, P.; Sanyal, S.; Chaki, N. Ant Colony based Routing for Mobile Ad-Hoc Networks towards Improved Quality of Services. *J. Emerg. Trends Comput. Inf. Sci.* **2012**, *3*, 10–14.
24. Chandrasekaran, K.; Simon, S.P. Multi-objective scheduling problem: Hybrid approach using fuzzy assisted cuckoo search algorithm. *Swarm Evol. Comput.* **2012**, *5*, 1–16. [[CrossRef](#)]
25. Deb, K. Multi-objective Optimization. In *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*; Burke, E.K., Kendall, y.G., Eds.; Springer: Boston, MA, USA, 2014; pp. 403–449.
26. Yang, X.S.; Deb, S. Multiobjective cuckoo search for design optimization. *Comput. Oper. Res.* **2013**, *40*, 1616–1624. [[CrossRef](#)]
27. Woźniak, M.; Połap, D. Basic concept of Cuckoo Search Algorithm for 2D images processing with some research results: An idea to apply Cuckoo Search Algorithm in 2D images key-points search. In Proceedings of the International Conference on Signal Processing and Multimedia Applications (SIGMAP), Vienna, Austria, 28–30 August 2014; pp. 157–164.
28. Wang, T.; Meskin, M.; Grinberg, I. Comparison between particle swarm optimization and Cuckoo Search method for optimization in unbalanced active distribution system. In Proceedings of the IEEE International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 14–17 August 2017; pp. 14–19.
29. Yang, X.S.; Deb, S. Cuckoo Search via Lévy flights. In Proceedings of the World Congress on Nature Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214.
30. Yang, X.S.; Deb, S. Cuckoo search: Recent advances and applications. *Neural Comput. Appl.* **2014**, *24*, 169–174. [[CrossRef](#)]

31. Yang, X.-S. Bat Algorithm and Cuckoo Search: A Tutorial. In *Artificial Intelligence, Evolutionary Computing and Metaheuristics: In the Footsteps of Alan Turing*; Yang, X.-S., Ed.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 421–434.
32. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A New Heuristic Optimization Algorithm: Harmony Search. *Simulation* **2001**, *76*, 60–68.
33. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *213*, 267–289. [[CrossRef](#)]
34. Ng, K.C.; Li, Y. Design of sophisticated fuzzy logic controllers using genetic algorithms. In Proceedings of the 3rd International Fuzzy Systems Conference, Orlando, FL, USA, 26–29 June 1994; pp. 1708–1712.
35. Passino, K.M.; Yurkovich, S.; Reinfrank, M. *Fuzzy Control*; Addison-Wesley: Menlo Park, CA, USA, 1998.
36. Antsaklis, P.J. Intelligent Learning Control. *Control Syst. Mag.* **1995**, *15*, 5–7.
37. Rogers, E.; Li, Y. *Parallel Processing in a Control Systems Environment*; Prentice Hall: New York, NY, USA, 1993.
38. Chen, G.; Pham, T.T. *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*; CRC Press: Boca Raton, FL, USA, 2001.
39. Nguyen, H.T.; Prasad, N.R.; Walker, E.A.; Walker, C.L. *A First Course in Fuzzy and Neural Control*; CRC Press: Boca Raton, FL, USA, 2002.
40. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353.
41. Homaifar, A.; McCormick, E. Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. *Trans. Fuzzy Syst.* **1995**, *3*, 129–139.
42. Gonzalez, C.I.; Castro, J.R.; Melin, P.; Castillo, O. Cuckoo search algorithm for the optimization of type-2 fuzzy image edge detection systems. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 25–28 May 2015; pp. 449–455.
43. Balochian, S.; Ebrahimi, E. Parameter Optimization via Cuckoo Optimization Algorithm of Fuzzy Controller for Liquid Level Control. *J. Eng.* **2013**, *2013*, 1–7.
44. Checa, F. *Diseño e implementación de controladores clásicos y en el espacio de estados para el levitador magnético MLS*; Universidad de las Fuerzas Armadas ESPE: Sangolquí, Ecuador, November 2009.
45. Rajabioun, R. Cuckoo Optimization Algorithm. *Appl. Soft Comput.* **2011**, *11*, 5508–5518.
46. Payne, R.B.; Sorensen, M.D. *The Cuckoos*; Oxford University Press: New York, NY, USA, 2005.
47. Rani, K.A.; Hoon, W.F.; Malek, M.F.; Affendi, N.A.; Mohamed, L.; Saudin, N.; Ali, A.; Neoh, S.C. Modified cuckoo search algorithm in weighted sum optimization for linear antenna array synthesis. In Proceedings of the IEEE Symposium on Wireless Technology and Applications (ISWTA), Bandung, Indonesia, 23–26 September 2012; pp. 210–215.
48. Wang, L.; Yin, Y.; Zhong, Y. Cuckoo search with varied scaling factor. *Front. Comput. Sci.* **2015**, *9*, 623–635. [[CrossRef](#)]
49. Mantegna, R.N. Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes. *Phys. Rev. E* **1994**, *49*, 4677–4683. [[CrossRef](#)]
50. Fister, I., Jr.; Fister, D.; Fister, I. A comprehensive review of cuckoo search: Variants and hybrids. *Int. J. Math. Model. Numer. Optim.* **2013**, *4*, 387–409.

