

Article

An Efficient Chicken Search Optimization Algorithm for the Optimal Design of Fuzzy Controllers

Leticia Amador-Angulo *, Oscar Castillo , Cinthia Peraza and Patricia Ochoa

Division of Graduate Studies and Research, Tijuana Institute of Technology, Tijuana 22414, Mexico; ocastillo@tectijuana.mx (O.C.); cinthia.peraza@tectijuana.edu.mx (C.P.); martha.ochoa18@tectijuana.edu.mx (P.O.)

* Correspondence: gloria.amador@tectijuana.edu.mx

Abstract: A proposed architecture to design the optimal parameters of Membership Functions (MFs) of Type-1 Fuzzy Logic Systems (T1FLSs) using the Chicken Search Optimization (CSO) is applied to three Fuzzy Logic Controllers (FLCs) in this paper. Two types of MFs are considered in the study: triangular and trapezoidal ones. The performance and efficiency of the CSO algorithm are particularly good when perturbations are added during the execution in each control problem. Two benchmark control problems: Water Tank Controller and Inverted Pendulum Controller are considered for testing the proposed approach. Also, the optimal design of a fuzzy controller for trajectory tracking of an Autonomous Mobile Robot (AMR) is considered to test the CSO. The main goal is to highlight the efficiency of CSO algorithm in finding optimal fuzzy controllers of non-linear plants. Two types of perturbations are considered in each control problem. Results show that the CSO algorithm presents excellent results in the field of Fuzzy Logic Controllers. Two types of Fuzzy Inference Systems: Takagi-Sugeno and Mamdani FLSs, are implemented in this paper. The most important metrics usually applied in control are used in this paper, such as: Integral Time Absolute Error (ITAE), Integral Time Squared Error (ITSE), Integral Absolute Error (IAE), Integral Square Error (ISE), Mean Square Error (MSE), and Root Mean Square Error (RMSE).

Keywords: chicken search optimization; fuzzy controller; perturbation; autonomous mobile robot; benchmark problems



Citation: Amador-Angulo, L.; Castillo, O.; Peraza, C.; Ochoa, P. An Efficient Chicken Search Optimization Algorithm for the Optimal Design of Fuzzy Controllers. *Axioms* **2021**, *10*, 30. <https://doi.org/10.3390/axioms10010030>

Academic Editor: Javier Fernandez

Received: 16 February 2021

Accepted: 5 March 2021

Published: 9 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent decades the Chicken Search Optimization (CSO) algorithm has shown good results and the efficiency of this bio-inspired algorithm has been verified in different problems. For example, in [1], Al Shayokh et al. presented a bio inspired distributed Wireless Sensor Network (WSN) localization based on chicken swarm optimization; in [2], Banerjee et al. presented an improved serially concatenated convolution turbo code (SCCTC) using chicken swarm optimization; in [3], Chen et al. presented a penalty function with a modified chicken swarm optimization for constrained optimization; in [4], Deb et al. presented a Hybrid Multi-Objective Chicken Swarm Optimization and Teaching Learning Based Algorithm for Charging Station Placement Problem; in [5], Deb et al. showed a new Teaching–Learning-based Chicken Swarm Optimization Algorithm; in [6], Deb et al. presented some recent studies on Chicken Swarm Optimization Algorithms; and in [7], Hafez et al. presented an innovative approach for feature selection based on Chicken Swarm Optimization. In addition, in [8], Kurozawa et al. showed an optimization of the enzymatic hydrolysis of chicken meat using response surface methodology; in [9], Lin et al. presented a robust recurrent wavelet neural network controller with improved particle swarm optimization for linear synchronous motor drive; in [10], Qu et al. presented a Chicken Swarm Optimization based on elite opposition-based learning; in [11], Wang et al. showed an improved Chicken Swarm Algorithm based on chaos theory and its application in wind power interval prediction; and in [12], Wu et al. presented an improved Chicken Swarm Optimization. Further, in [13], Wu et al. presented a convergence analysis and improvement of

the Chicken Swarm Optimization Algorithm; and in [14], Wu et al. showed an application of improved Chicken Swarm Optimization for Maximum Power Point Tracking (MTTP) in photovoltaic system. Finally, in [15] Yaseen et al. presented a Hybrid Invasive Weed Optimization Algorithm with Chicken Swarm Optimization Algorithm to solve Global Optimization Problems; in [16], Zarlis et al. presented a framework of training Adaptive Neuro-Fuzzy Inference System (ANFIS) using Chicken Swarm Optimization for solving classification problems, and in [17] Liang et al. presented a fast Synthetic Aperture Radar (SAR) image segmentation method based on an improved Chicken Swarm Optimization Algorithm. All these previous works allow us to observe that the CSO algorithm shows efficiency in various fields of study.

Other important contributions in which CSO algorithm has had an important presence is on the medical area. Some works include [18], where a comparative study of bio-inspired algorithms applied in the optimization of fuzzy systems is presented by Miramontes et al., and [19], where a prediction of cervical cancer using Chicken Swarm Optimization is presented by Tripathi et al.

Control is a study area of much interest for several authors; for example, in [20] the authors focused on a constrained population extremal optimization-based robust load frequency control of multi-area interconnected power system; ref [21] based their study in design of fractional order Proportional Integral Derivative (PID) controller for automatic regulator voltage system based on multi-objective extremal optimization; ref [22] presents an understanding edge computing focused on engineering evolution with artificial intelligence; ref [23] studied an improvement of Maximum Power Point Tracking (MPPT) control performance using fuzzy control and Variable Gain Proportional Integral (VGPI) in the Photovoltaic (PV) system for micro grid; ref [24] presents an optimization of fuzzy trajectory tracking in autonomous mobile robots based on bio-inspired algorithms; and in [25] they analyzed an optimization design and test bed of fuzzy control rule base for a PV system MPPT in a micro grid.

In the field of control, the CSO algorithm has also proven to be a good algorithm. For example, in [26], Li et al. showed a chicken swarm-based method for ascent trajectory optimization of hypersonic vehicles; in [27], Amador-Angulo et al. presented an optimal design of fuzzy logic systems through a Chicken Search Optimization Algorithm applied to a benchmark problem; in [28], Liang et al. presented an improved Chicken Swarm Optimization Algorithm and its application in robot path planning; in [29], Mu et al. presented an optimal trajectory planning for robotic manipulators using chicken swarm optimization; and in [30], Wu et al. presented an improved Chicken Swarm Optimization method for reentry trajectory optimization.

Bio-inspired algorithms are based on the natural behavior of species that have allowed them to be used in solving complex problems. A wide variety of algorithms have been developed in the field of control. Some related works are mentioned as follows: two applications of Bee Colony Optimization (BCO) are presented in [31,32], a Salp Swarm Algorithm (SWA) is studied in [33], a Bio-inspired Chaotic Fruit Fly Algorithm (FFA) is presented in [34], a Particle Swarm Optimization (PSO) is presented in [35], a Flower Pollination Algorithm (FPA) is presented in [36], a Cuckoo Search Algorithm (CSA) is presented in [37], a Grey Wolf Algorithm (GWA) is presented in [38], a Bat Algorithm (BA) is presented in [39], a Black Hole Algorithm (BHA) is presented in [40], a Firefly Algorithm (FA) is presented in [41], some bio-inspired algorithms are analyzed and presented in [42], a Cat Swarm Optimization (CSO) Algorithm is presented in [43], and an Ant Colony Optimization (ACO) algorithm is presented in [44]. For this reason, in this paper some benchmark control problems are studied and analyzed with Chicken Search Optimization Algorithm (CSO) for solving problems and demonstrated the efficiency in this algorithm in control problems. The benchmark problems have proven to be a good technique for testing the performance of a bio-inspired algorithm; for example, in [45], a design of a sliding mode controller for a water tank liquid level control system is presented; in [46], a controller design for rotary inverted pendulum system using evolutionary algorithms

is presented; in [47], a fuzzy controller for a benchmark problem related to computing with words approach is presented. Also, controlling the stabilization and trajectory in a robot is another interesting field in control problems; for example, in [24,31,48,49] an autonomous system to control a mobile robot is presented; in [50], a fuzzy controller in AC servo motor drive is presented; and in [51], a bio-inspired decentralized autonomous robot mobile navigation control for multi agent systems is presented. In the existing literature it is shown that the CSO method is applied as the solution of different types of problems at present time, because it is a precise and solid method for optimization problems. For this reason, the main contribution is to make use of this algorithm in the optimization of control problems.

The main motivation in this paper is to demonstrate the efficiency and performance of the Chicken Search Optimization Algorithms in control problems. Specifically, two benchmark problems are implemented, which are the water tank controller, and the inverted pendulum. Also, an autonomous mobile robot for stabilization of trajectory is implemented. Another important contribution is the statistical comparison with other meta-heuristics algorithms. Many works of the CSO algorithm have been presented in control problems, but in this paper, the more important contribution is to find the optimal design of the Membership Functions (MFs) of a Type-1 Fuzzy Logic System (T1FLS) applied to control problems. Also, all the problems were executed with perturbations in the model with the goal of analyzing the behavior that CSO algorithm when levels of noise are added in the model.

The organization of this paper is as follows. Section 2 describes the chicken search optimization algorithm. Section 3 describes each of the three control problems, the structure, equations and characteristics. Section 4 describes the proposed structure of the Type-1 Fuzzy Logic System (T1FLS) for each control problem. In Section 5 the simulation results are presented and analyzed. In Section 6 a comparative and discussion of results are presented. Finally, Section 7 presents some relevant conclusions and future works in this paper.

2. Chicken Search Optimization Algorithm

The interesting bio-inspired algorithm that this paper considers to analyze and explore is called the Chicken Swarm Optimization (CSO) based on the behaviors of cocks, hens and chicks in finding food. The main idea in this algorithm was created by Meng, Liu, Gao and Zhang in 2014 [52]. BCO algorithm uses the chicken swarm in the searching space which is mapped a specific particle individual. Also, three types of chicken are used in the swarms: Cock swarm, chick swarm and hen swarm for the whole chicken swarm. Each role has a different weight in the CSO algorithm; for example, in the methodology of this algorithm, cock swarms present a highest fitness and the chick swarms present lowest fitness. The role (hen, chick and cock) is assigned during the execution of the algorithm depends of its fitness value. In the implementation of CSO algorithm in each control problem, a chicken represents a possible solution for the control problem [52]. The CSO algorithm has as objective error minimization in each control problem. The proposed idea is represented by Figure 1.

In this algorithm each particle represented a possible solution in the problem. For example, in the case of the first control problem called water tank controller, the proposed fuzzy logic system (FLS) has triangular membership functions (MF) at the inputs and outputs. Two type of MFs are implemented in this paper. The first is a triangular MF that depends on three parameters, which are a , b and c as given by the Equation (1), where a and c represents the feet of the triangle and the b locates the peak, and the second is a trapezoidal MF that depends on four parameters which are a , b , c and d , as given by Equation (2). This triangular and trapezoidal MFs are represented graphically in Figure 2a,b, respectively.

$$f(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (1)$$

$$f(x; a, b, c, d) = \max \left(\min \left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}, 0 \right), 0 \right) \quad (2)$$

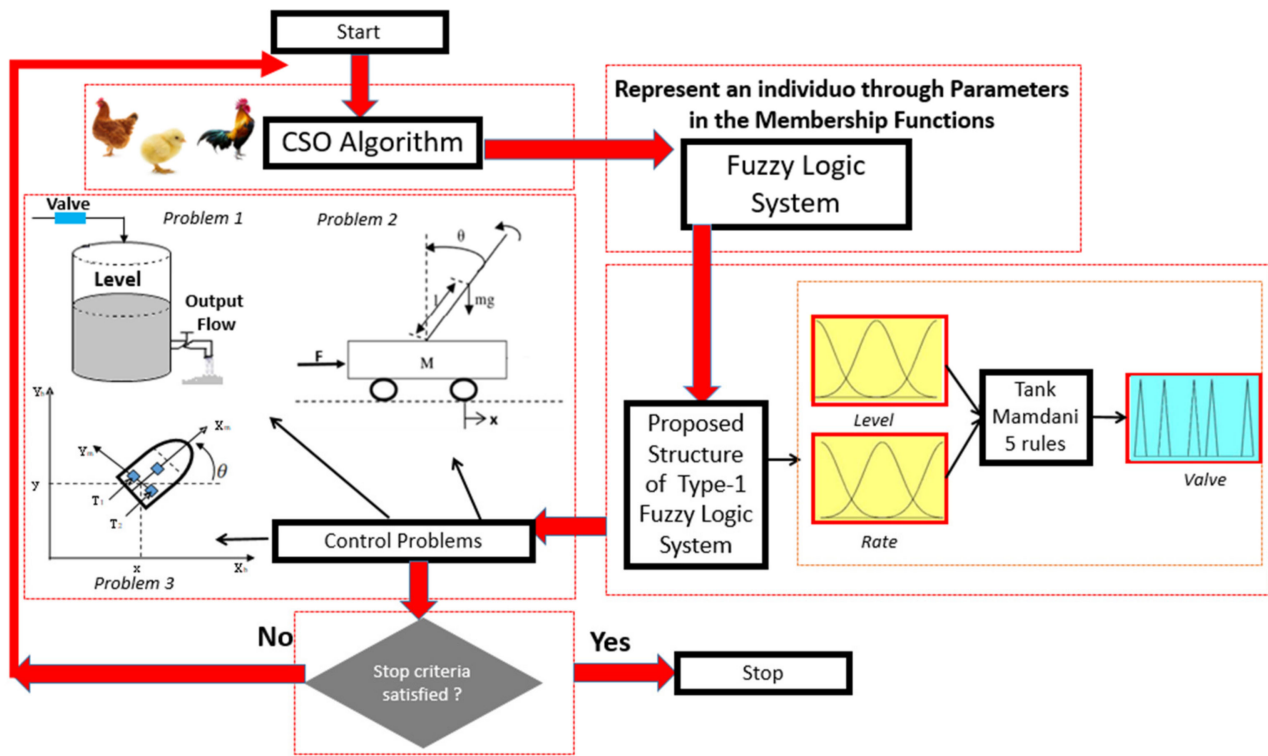


Figure 1. General idea for the Chicken Search Optimization (CSO) algorithm to solve control problems.

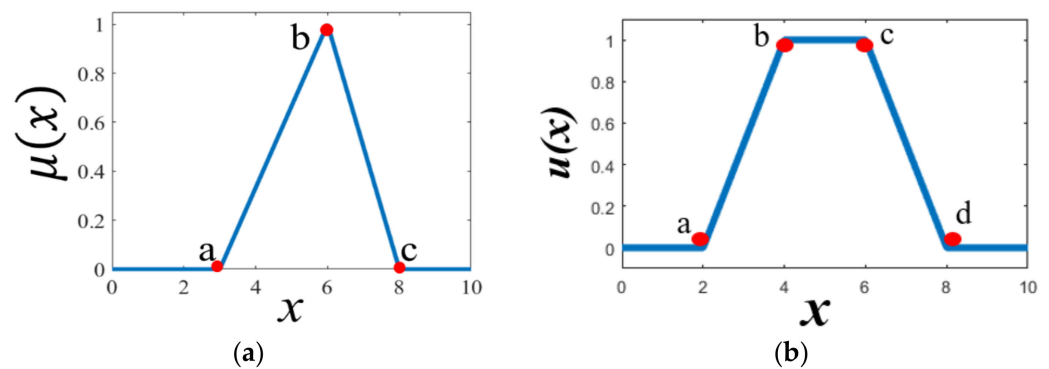


Figure 2. Points of triangular membership functions (MF) (a), and trapezoidal MF (b).

Figures 3 and 4 show the points (parameters) that make up each input and output triangular membership function.

A Type-1 FLS was optimized with the CSO algorithm, and it is necessary to have a particle (cock, hen or chick) with a size to optimize the membership functions (MFs). In this case it is called a vector representing the possible best distribution in the MFs for the first study case called water tank controller, as shown in Figure 5. The vector has 33 positions to optimize, because this control problem has two inputs each with three triangular MFs, and one output with five triangular MFs.

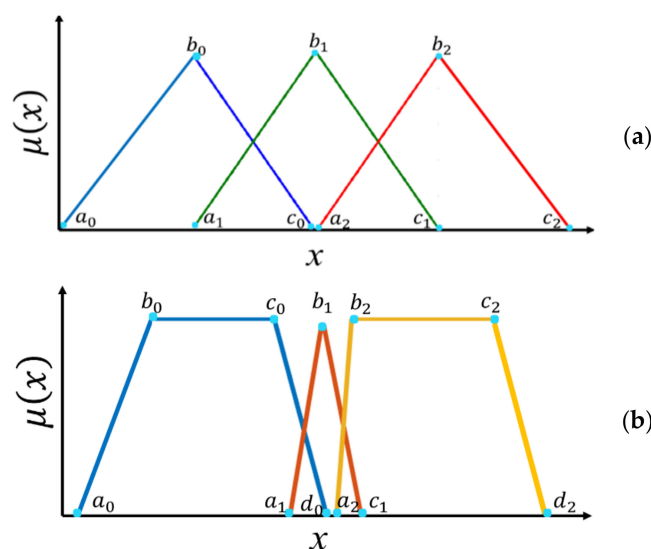


Figure 3. Points of membership functions of (a) Input 1 and Input 2 for the water tank problem controller, (b) Input 1 and Input 2 for the autonomous mobile robot problem controller.

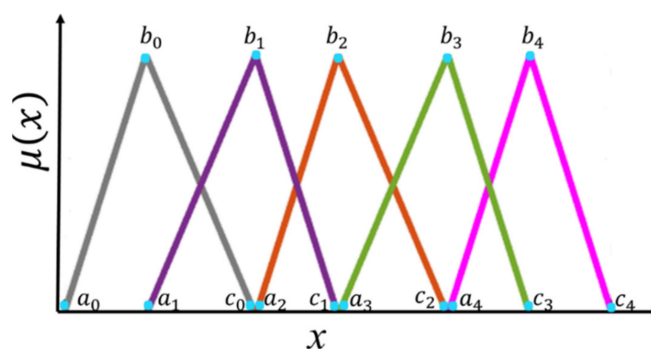


Figure 4. Points of membership functions of the Output 1 and Output 2.

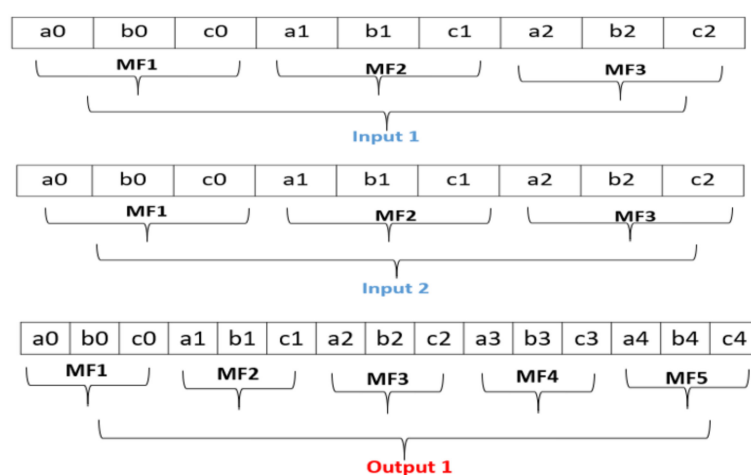


Figure 5. Vector for the optimization CSO algorithm.

Table 1 shows the total size of the vectors in each control problem and the type of membership function that CSO algorithm to find and optimize.

Table 1. Size of vector in the optimization by CSO algorithm for each control problem.

Control Problem	Input		Output		Total Size of Vector
	Total	Type of MFs	Total	Type of MFs	
Water Tank Controller	2	3—Triangular in each Input	1	5—Triangular in each output	33
Inverted Pendulum Controller	4	2—Trapezoidal in each Input		Sugeno-Takagi (Function)	32
Autonomous Mobile Robot Controller	2	Trapezoidal—Triangular—Trapezoidal	2	3—Triangular in each Output	40

The dynamism of CSO algorithm is represented in the following equations. For all the particles, the particle with best fitness is called with the role of cock particle swarm; this selection is represented by Equation (3):

$$x_{i,j}^{t+1} = x_{i,j}^t + randn(0, \sigma^2) \cdot x_{i,j}^t \quad (3)$$

where t represents the generations, and $x_{i,j}^{t+1}$ and $x_{i,j}^t$ indicate the position of j th dimension of a chicken in $t + 1$. The function $randn(0, \sigma^2)$ is implemented as a random of Gaussian distribution in Equation (4) to find the variance:

$$\sigma^2 = \begin{cases} 1, & \text{if } fit_i < fit_k \\ \exp\left(\frac{fit_k - fit_i}{(|fit_i| + \epsilon)}\right), & \text{if } fit_i \geq fit_k \end{cases} \quad (4)$$

where $i, k \in [1, rsize]$ and $i \neq k$. In each control problem, $rsize$ indicates the number of cock swarms, and the size of the swarm is 30 values. Also, a cock particle represents the number of values for the design of the MF for the Fuzzy Logic System in this paper; for each control problem the values of the MFs are different (see Table 1). fit_i and fit_k are the fitness values of cock particle i and k , respectively; ϵ is an important value that represents a number which is used to avoid zero-division-error, is the smallest constant in the computer. Otherwise, for CSO algorithm, hen swarm was selected with good fitness. Its random search is found by Equation (5) that is performed via cocks of hen population and that of others.

$$x_{i,j}^{t+1} = x_{i,j}^t + S1 \cdot rand \cdot (x_{r1,j}^t - x_{i,j}^t) + S2 \cdot rand \cdot (x_{r2,j}^t - x_{i,j}^t) \quad (5)$$

where $x_{i,j}^t$ and $x_{r2,j}^t$ is used in the position of the cock individual $r1$ in the population of hen xi and cock individual $r2$ in the other population, respectively. $rand$ is a function that represents a uniform random number with the range of $[0, 1]$. Equation (6) calculates the values of $S1$ and $S2$:

$$S1 = \exp\left(\frac{fit_i - fit_{r1}}{(|fit_{r2}| + \epsilon)}\right), S2 = \exp(fit_{r2} - fit_i) \quad (6)$$

where $r1$ (fit_{r1}) represents the fitness value of the cock individual in the population of hen xi , and $r2$ (fit_{r2}) represents the fitness value for the cock individual, respectively. In CSO algorithm all individuals, except for the cock and hen swarm, are defined as a chick swarm. Equation (7) shows this search mode follows that of the hen swarm:

$$x_{i,j}^{t+1} = x_{i,j}^t + FL \cdot (x_{m,j}^t - x_{i,j}^t), FL \in [0, 2] \quad (7)$$

In the CSO algorithm, the chick would follow its mother to forage for food, therefore $x_{m,j}^t$ is the position of j th dimension of particle m in $t + 1$, t indicate the generation, then,

this characteristic is represented by parameter FL . Hence, m represents the position of the i th chick's mother ($m \in [1, N]$), where N indicates the size of swarm (population). Algorithm 1 illustrates the pseudocode of the CSO algorithm.

Algorithm 1 General Steps of the CSO Algorithm

1. Initialize a population of N chickens and define the relation parameters;
2. Evaluate the N chickens' fitness values, $t=0$;
3. While ($t < \text{Max_Generation}$)
4. If ($t \% G == 0$)
5. Rank the chickens' fitness values and establish a hierarchal order in the swarm;
6. Divide the swarm into different groups, and determine the relationship between the chicks and mother hens in a group;
7. End if;
8. For $i = 1; N$
9. If ($i == \text{rooster}$)
10. Update its solution/location using Equation (3);
11. End if;
12. If ($i == \text{hen}$)
13. Update its solution/location using Equation (5);
14. End if;
15. If ($i == \text{chick}$)
16. Update its solution/location using Equation (7);
17. End if;
18. Evaluate the new solution;
19. If the new solution is better its previous one; update it;
20. End For;
21. End While;

With the goal to illustrate each step in the CSO algorithm, with a total of 9 steps, Figure 6 shows the Flowchart of the mechanism in this algorithm.

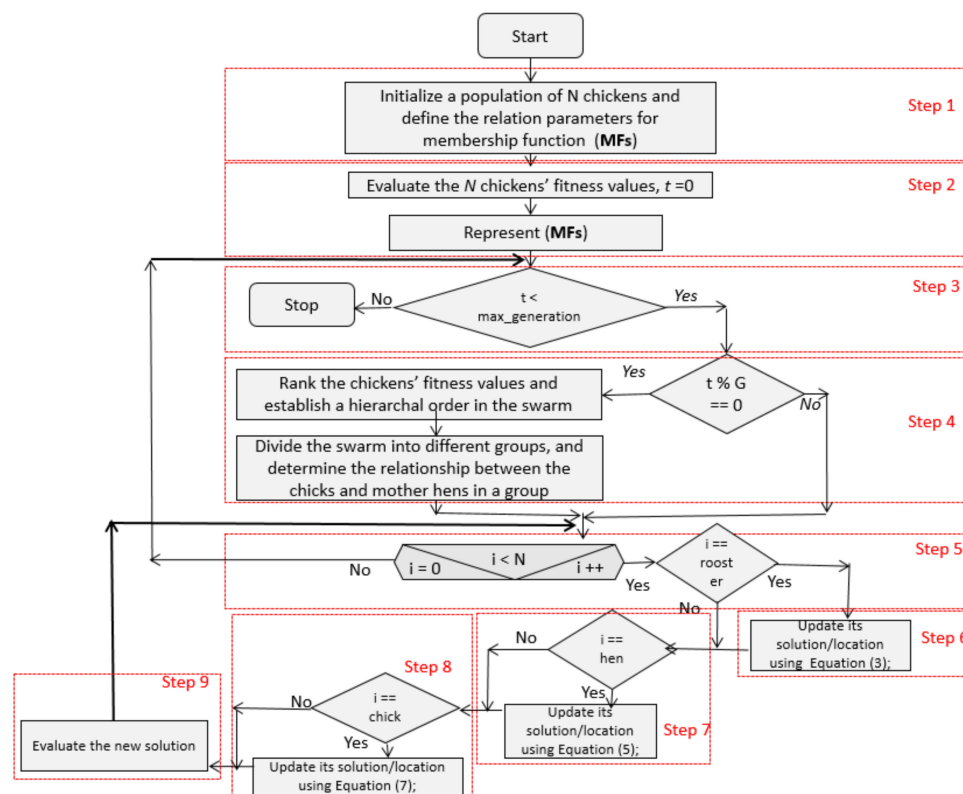


Figure 6. Flowchart of the step in the CSO algorithm.

3. Control Problems

For performance evaluation of the CSO algorithm a total of three control problems are implemented in this work. Two are benchmark problems and the other is to control the stabilization in an autonomous mobile robot (AMR). These control problems allow to analyze the behavior of proposed method to find the optimal design for the MFs through an efficient CSO algorithm. The general description of each control problem is presented in the following subsections.

3.1. Water Tank Controller

The first problem has the goal of filling the water tank, therefore, based on the actual water level in the tank the controller has to be able to provide the proper activation of the valve. Figure 7 illustrates the variables involved so that the valve opening operates, and Equation (8) shows the mathematical functions in this problem.

$$\frac{d}{dt} Vol = A \frac{dH}{dt} = bV - a\sqrt{H} \quad (8)$$

where Vol is the volume of water in the tank, A is the cross-sectional area of the tank, b is a constant related to the flow rate into the tank, and a is a constant related to the flow rate out of the tank. The equation describes the height of water H as a function of time, due to the difference between flow rates into and out of the tank.

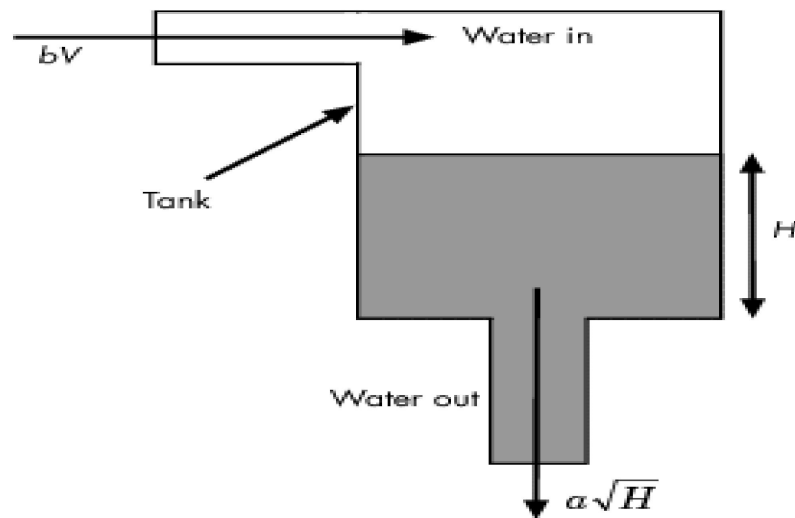


Figure 7. Graphic representation of the problem of water tank.

3.2. Inverted Pendulum Controller

The second problem consists of controlling a pendulum given a force F that is applied horizontally (in x) to a cart. The pendulum moves horizontally towards a dynamic point at the base and the goal is to get to that point quickly and with greatest balance or stability that the vertical bar (pendulum) can present. Figure 8 graphically shows the idea for this problem.

3.3. Autonomous Mobile Robot Controller

The last control problem consists of a unicycle mobile robot [24], which has two driving wheels located on the same axis and a front free wheel, and Figure 9 shows a graphical description of the robot model.

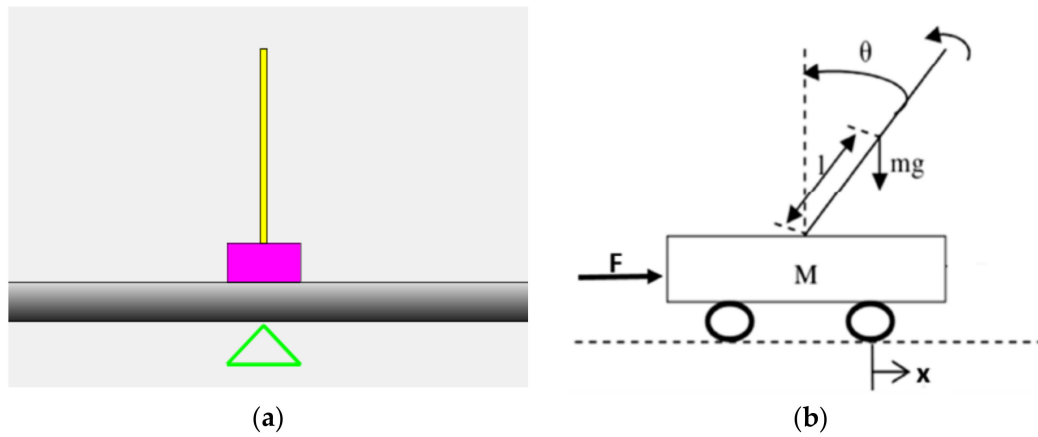


Figure 8. Second problem of control. (a) Representation in the model, and (b) graphic idea for the inverted pendulum controller.

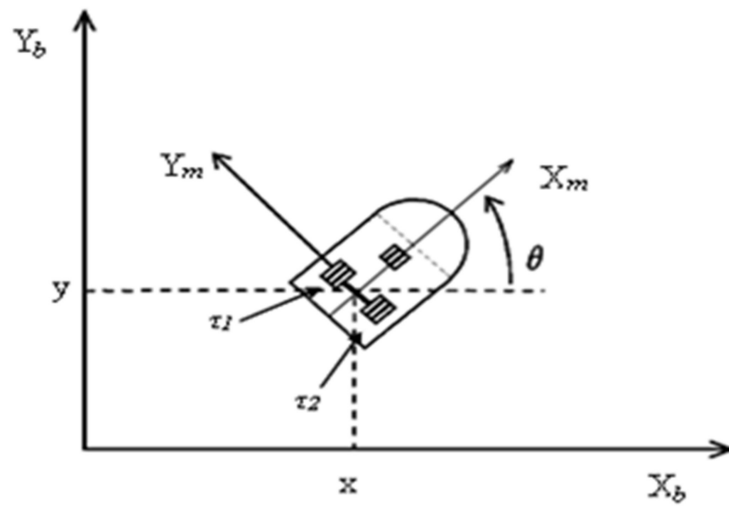


Figure 9. Graphic idea for the Autonomous Mobile Robot (AMR) problem.

The robot model assumes that the motion of the free wheel can be ignored in its dynamics, as shown by Equations (9) and (10).

$$M(q)\dot{v} + C(q, \dot{q})v + Dv = \tau + P(t) \quad (9)$$

where

$q = (x, y, \theta)^T$ is the vector of the configuration coordinates,

$v = (\dot{x}, \dot{y}, \dot{\theta})^T$ is the vector of velocities,

$\tau = (\tau_1, \tau_2)$ is the vector of torques applied to the wheels of the robot where τ_1 , and τ_2 denote the torques of the right and left wheel,

$P \in R^2$ is the uniformly bounded disturbance vector,

$M(q) \in R^{2 \times 2}$ is the positive-definite inertia matrix,

$C(q, \dot{q})\theta$ is the vector of centripetal and Coriolis forces, and

$D \in R^{2 \times 2}$ is a diagonal positive-definite damping matrix.

The kinematic system is represented by Equation (10).

$$\dot{q} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (10)$$

where (x, y) is the position in the $X - Y$ (world) reference frame, θ is the angle between the heading direction and the x -axis, v and ω are the linear and angular velocities. Furthermore, Equation (11) shows the non-holonomic constraint, which this system has, which corresponds to a no-slip wheel condition preventing the robot from moving sideways.

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0 \quad (11)$$

The system fails to meet Brockett's necessary condition for feedback stabilization, which implies that no continuous static state-feedback controller exists that can stabilize the closed-loop system around the equilibrium point.

4. Proposed Structure of Type-1 Fuzzy Logic Systems for Control

4.1. Fuzzy Logic System

A fuzzy logic system (FLS) that is defined entirely in terms of Type-1 fuzzy sets, is known as Type-1 Fuzzy Logic System (Type-1 FLS), and its elements are defined in the following Figure 10 [45,46].

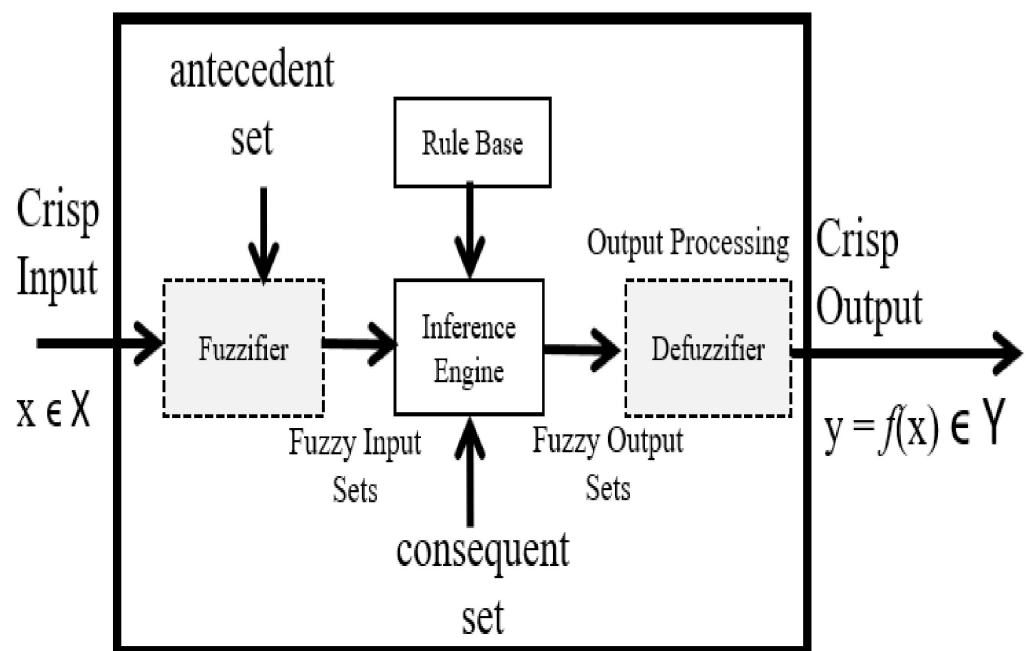


Figure 10. Architecture of a Type-1 fuzzy logic system (FLS).

A type-1 fuzzy set in the universe x is characterized by a membership function $\mu_A(x)$ taking values on the interval $[0, 1]$ and can be represented as a set of ordered pairs of an element and the membership degree of an element to the set and are defined by Equation (12) [53,54]:

$$A = \{(x, \mu_A(x)) \mid x \in U\} \quad (12)$$

4.2. Fuzzy Logic Controller

The general idea of a Fuzzy Logic Controller (FLC) was created by Mamdani in 1976. An important work of FLC is in [55] describing an application of fuzzy algorithms for control of simple dynamic plant. Another variant for the inferences in FLS is Takagi-Sugeno (in this paper the inverted pendulum control problem is of type Takagi-Sugeno), and the works in [56,57] apply this approach in the inference form of Takagi-Sugeno and Mamdani for the Type-1 FLS. Figure 11 shows the general representation of the FLCs.

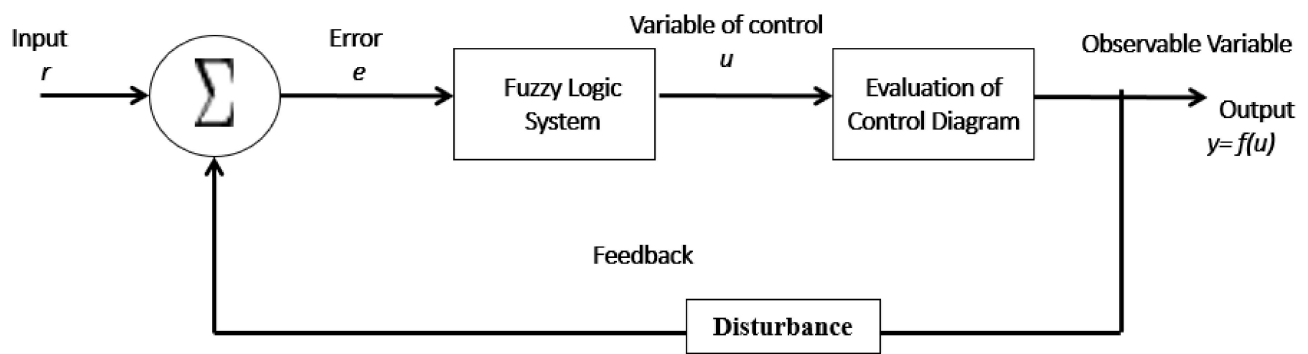


Figure 11. Architecture of a Fuzzy Logic Controller (FLC).

An important aspect in the evaluation of the FLCs is the disturbance in the feedback phase. This block allows the bio-inspired algorithm, in this case CSO, to demonstrate its efficiency in the optimization of the control problem. Some works where the behavior of the problem with level of noise in the model is analyzed are [31,58–64]. In this paper, two type of disturbances are considered in each model.

4.3. Proposed Structure of the Type-1 FLS

The proposed structures of the Type-1 FLS for each control problem are described in the following subsections.

4.3.1. Water Tank Controller

The first control problem has a proposed structure of two inputs called level and rate, each with three triangular MFs, and only one output called valve with five triangular MFs. A total of five rules are necessary to control the filling of the water tank. Figure 12 illustrates the Fuzzy Inference System (FIS) (a) and the inferences rules (b). Also, Figure 13 shows the FLC model that is implemented in the executions by CSO algorithm.

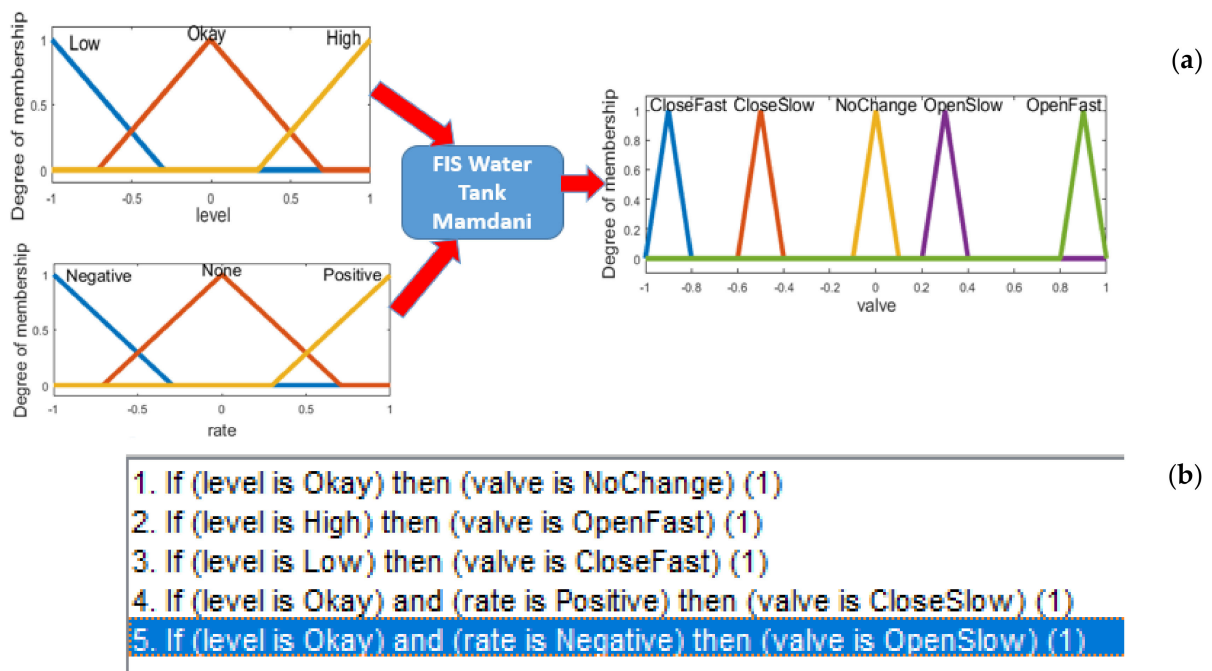


Figure 12. Proposed structure of the first Fuzzy Inference System (FIS) Control Problem. (a) Input and output of the FIS, and (b) inference rules for the Water Tank Controller.

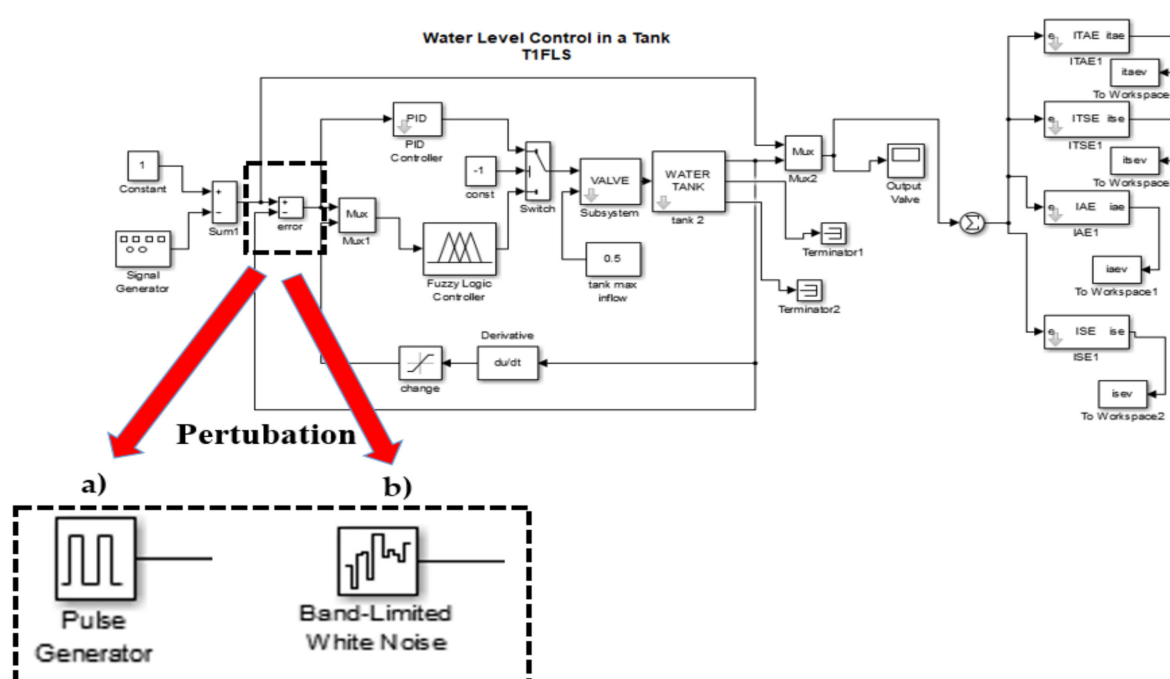


Figure 13. FLC model for execution of simulation by CSO algorithm of the Water Tank Controller. (a) Pulse generator perturbation, and (b) Band-limited perturbation.

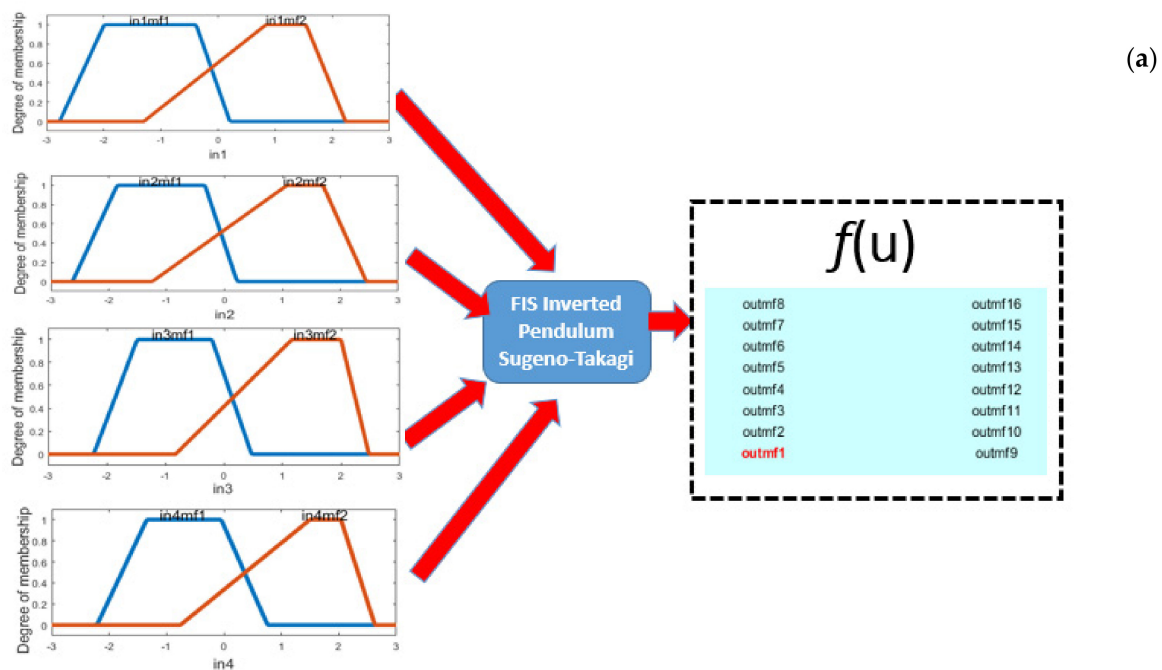
Two type of perturbations are added in the model, specifically to block the error: a pulse generator (a) and a band-limited (b) [58], for each control problem, with the goal to visualize the efficiency of CSO algorithm when disturbance is present.

4.3.2. Inverted Pendulum Controller

The second control problem has a proposed structure of four inputs called in1, in2, in3 and in4, each with two trapezoidal MFs. This FIS is of Takagi-Sugeno type. A total of 16 rules are necessary for stabilization of the pendulum. Figure 14 illustrates the Fuzzy Inference System (FIS) (a) and the inferences rules (b). Also, Figure 15 shows the FLC model for this second control problem.

4.3.3. Autonomous Mobile Robot Controller

The third control problem has the structure of two inputs called e_v (error in the linear velocity), and e_w (error in the angular velocity). Each input has three MFs of which in the extremes are trapezoidal MFs and in the middle a triangular MF, for each input. Regarding the outputs, it has two, called T1 (Torque 1) and T2 (Torque 2) with three triangular MFs, respectively. Figure 16 illustrates the Fuzzy Inference System (FIS) (a) and the inferences rules (b). Also, Figure 17 shows the FLC model for this third control problem.



- (b)
1. If (in1 is in1mf1) and (in2 is in2mf1) and (in3 is in3mf1) and (in4 is in4mf1) then (out is outmf1) (1)
 2. If (in1 is in1mf1) and (in2 is in2mf1) and (in3 is in3mf1) and (in4 is in4mf2) then (out is outmf2) (1)
 3. If (in1 is in1mf1) and (in2 is in2mf1) and (in3 is in3mf2) and (in4 is in4mf1) then (out is outmf3) (1)
 4. If (in1 is in1mf1) and (in2 is in2mf1) and (in3 is in3mf2) and (in4 is in4mf2) then (out is outmf4) (1)
 5. If (in1 is in1mf1) and (in2 is in2mf2) and (in3 is in3mf1) and (in4 is in4mf1) then (out is outmf5) (1)
 6. If (in1 is in1mf1) and (in2 is in2mf2) and (in3 is in3mf1) and (in4 is in4mf2) then (out is outmf6) (1)
 7. If (in1 is in1mf1) and (in2 is in2mf2) and (in3 is in3mf2) and (in4 is in4mf1) then (out is outmf7) (1)
 8. If (in1 is in1mf1) and (in2 is in2mf2) and (in3 is in3mf2) and (in4 is in4mf2) then (out is outmf8) (1)
 9. If (in1 is in1mf2) and (in2 is in2mf1) and (in3 is in3mf1) and (in4 is in4mf1) then (out is outmf9) (1)
 10. If (in1 is in1mf2) and (in2 is in2mf1) and (in3 is in3mf1) and (in4 is in4mf2) then (out is outmf10) (1)
 11. If (in1 is in1mf2) and (in2 is in2mf1) and (in3 is in3mf2) and (in4 is in4mf1) then (out is outmf11) (1)
 12. If (in1 is in1mf2) and (in2 is in2mf1) and (in3 is in3mf2) and (in4 is in4mf2) then (out is outmf12) (1)
 13. If (in1 is in1mf2) and (in2 is in2mf2) and (in3 is in3mf1) and (in4 is in4mf1) then (out is outmf13) (1)
 14. If (in1 is in1mf2) and (in2 is in2mf2) and (in3 is in3mf1) and (in4 is in4mf2) then (out is outmf14) (1)
 15. If (in1 is in1mf2) and (in2 is in2mf2) and (in3 is in3mf2) and (in4 is in4mf1) then (out is outmf15) (1)
 16. If (in1 is in1mf2) and (in2 is in2mf2) and (in3 is in3mf2) and (in4 is in4mf2) then (out is outmf16) (1)

Figure 14. Proposed structure of the second FIS Control Problem. (a) Input and output of the FLS, and (b) inference rules for the Inverted Pendulum Controller.

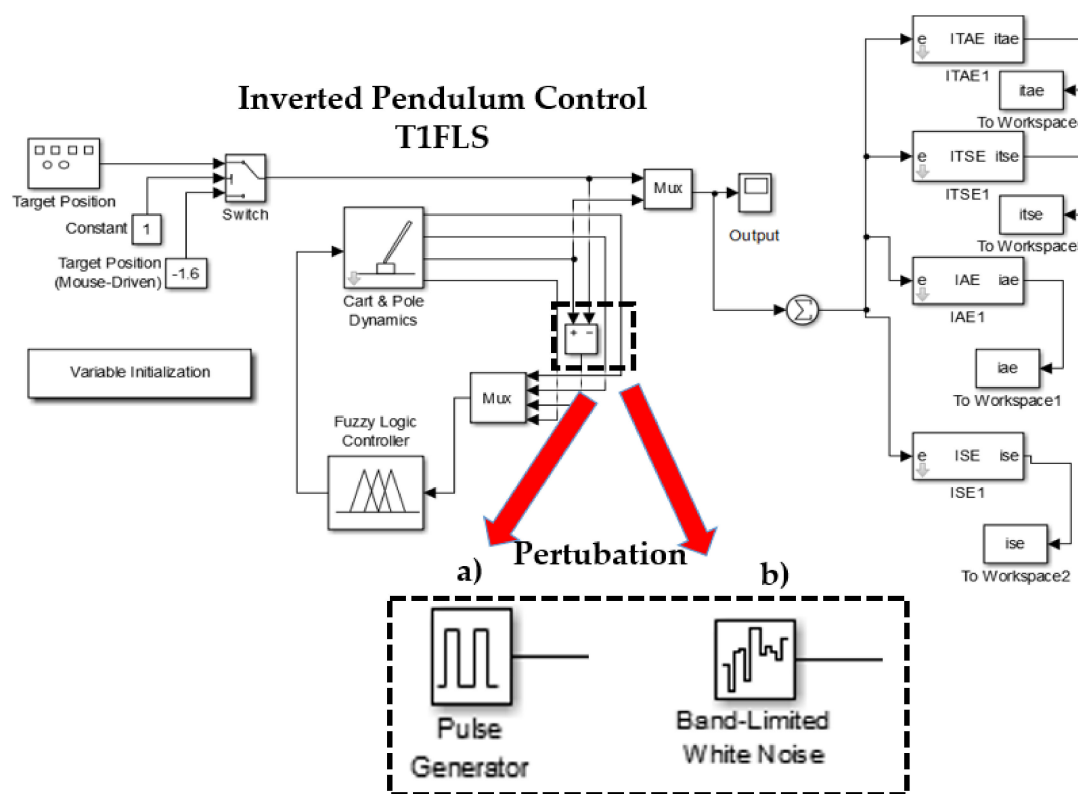


Figure 15. FLC model for the simulation by CSO algorithm of the Inverted Pendulum Controller. (a) Pulse generator perturbation, and (b) Band-Limited perturbation.

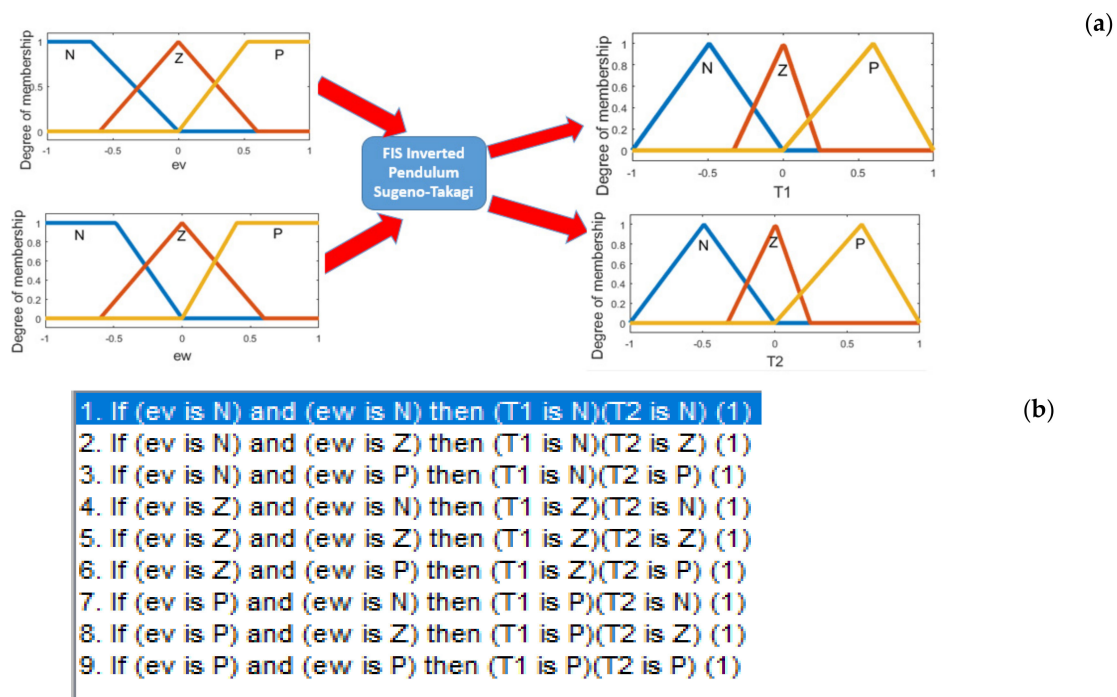


Figure 16. Proposed structure of the third FIS Control Problem. (a) Input and output of the FIS, and (b) inference rules for the Autonomous Mobile Robot Controller.

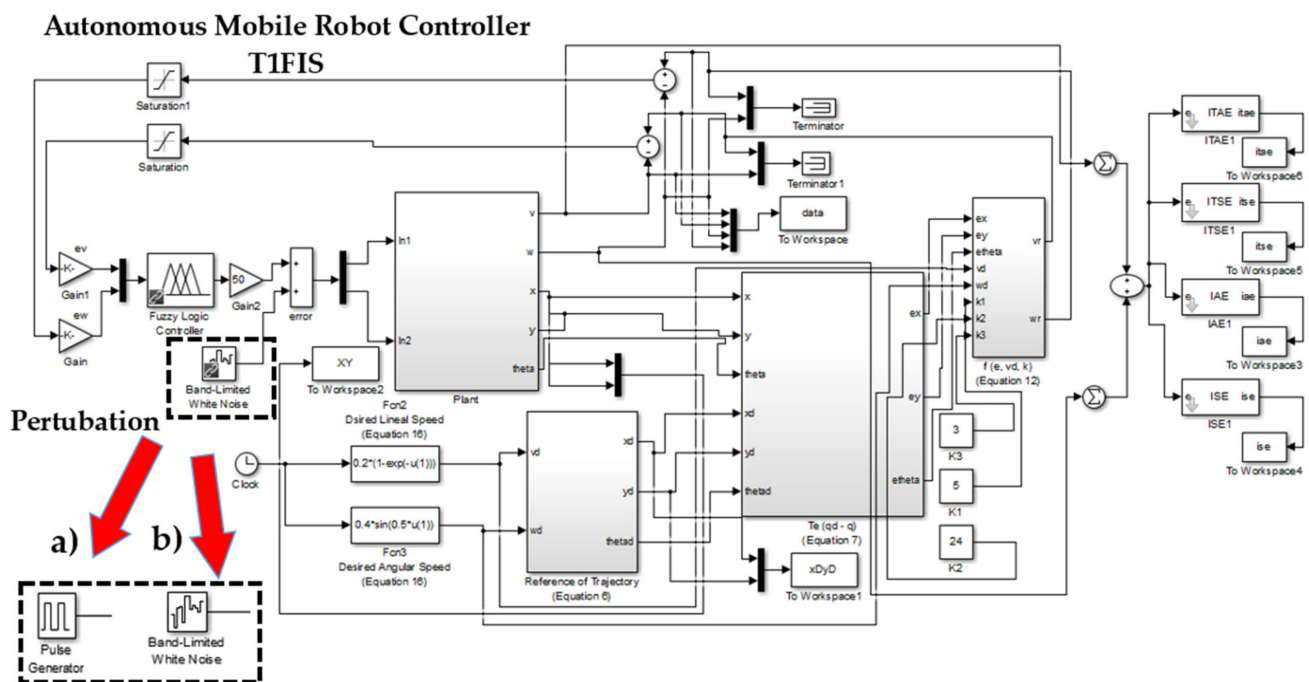


Figure 17. FLC model to execution of simulation by CSO algorithm of the AMR Controller. (a) Pulse generator perturbation, and (b) Band-Limited perturbation.

5. Simulations Results

The analysis of results is presented in this section for each control problem. The setting of parameters for CSO algorithm is presented in Tables 2 and 3.

Table 2. Settings of CSO algorithm for benchmark control problems.

Parameters	Values
N (Population)	30
RN	$0.15 * N$
HN	$0.7 * N$
MN	$0.15 * N$
G	10
FL	[0.5, 0.9]
Generations	40

Table 3. Settings of CSO algorithm for Autonomous Mobile Robot control problem.

Parameters	Values
N (Population)	20
RN	$0.15 * N$
HN	$0.7 * N$
MN	$0.15 * N$
G	10
FL	[0.5, 0.9]
Generations	15

The main reason for the decision to decrease the values in Population (N) and Generations only for third control problem is because it is not a benchmark problem, also, with previous experiments, it was analyzed that a maximum number of generations is not necessary for good errors to be found by the CSO algorithm.

To evaluate of efficiency of the CSO algorithm in control, a series of performance indices are used in non-linear plants, which are the Integral Square Error (ISE), Integral

Absolute Error (IAE), Integral Time Squared Error (ITSE), Integral Time Absolute Error (ITAE), the Root Mean Square Error (RMSE) and the Mean Square Error (MSE). The mathematical representation of each are shown in Equations (13)–(18), respectively.

$$ISE = \int_0^{\infty} e^2(t) dt \quad (13)$$

$$IAE = \int_0^{\infty} |e(t)| dt \quad (14)$$

$$ITSE = \int_0^{\infty} e^2(t) t dt \quad (15)$$

$$ITAE = \int_0^{\infty} |e(t)| t dt \quad (16)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (x_t - \hat{x}_t)^2} \quad (17)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (\bar{Y}_i - Y_i)^2 \quad (18)$$

The fitness function of CSO algorithm is the MSE (see Equation (18)), and this means that the best individual in each execution is the particle that has the lowest value of MSE of the entire population. Each FLC shown in Figures 13, 15 and 17 has two references. The first one indicates the desired reference trajectory, which is Y_i (estimated value) and the second one represents the trajectory created by the optimized fuzzy controller, in this case the observed value, which is \bar{Y}_i (signal value), and n represents the total number of observed samples.

A total of 30 experiments were realized in each non-linear plant without perturbation, with perturbation of pulse generator with values of amplitude of 1, period (seconds) of 10, pulse with (% of period) of 30 and phase delay (seconds) of 100, with a second perturbation of noise limited with value of noise power of 0.5 and sample time of 0.8. Table 4 shows the results for each model, and the average result for minimum values to find by CSO algorithm of the six-performance indexes and problem Without Perturbation (Not Applied Perturbation (Not AP), With Pulse Generator Perturbation (Type-1 Perturbation), and with Band-Limited Perturbation (Type-2 Perturbation) is presented.

Results on Table 4 show that when a certain level of noise is added in, the model CSO algorithm presents a stable behavior in the metrics of error. For example, in the first control problem, the Integral Square Error without noise the average of the error is $2.77 \times 10^{+3}$, with the noise pulse generator is $3.00 \times 10^{+3}$ and with the noise band-limited is $3.38 \times 10^{+3}$. Thus, errors remain similar even when a high level of noise is added; this is possible due to the exploitation ability that CSO algorithm presents. In the case of the third control problem the best average in RMSE is with the noise of band-limited perturbation.

Table 5 shows the results for each model, best result of the fitness function MSE to find by CSO algorithm with the metrics of best, worst, average, standard deviation, and Without Perturbation (Not Applied Perturbation (Not AP), With Pulse Generator Perturbation (Type-1 Perturbation), and with Band-Limited Perturbation (Type-2 Perturbation) is presented.

Table 4. Results of average performance index of minimums values found by the CSO algorithm.

Control Problem	Performance Index	Types of Experiment		
		Not AP	Type-1 Perturbation	Type-2 Perturbation
Water Tank Controller	ITAE	$2.77 \times 10^{+5}$	$2.94 \times 10^{+5}$	$3.05 \times 10^{+5}$
	ITSE	$7.26 \times 10^{+5}$	$7.80 \times 10^{+5}$	$8.27 \times 10^{+5}$
	IAE	$1.08 \times 10^{+3}$	$1.15 \times 10^{+3}$	$1.23 \times 10^{+3}$
	ISE	$2.77 \times 10^{+3}$	$3.00 \times 10^{+3}$	$3.38 \times 10^{+3}$
	RMSE	2.43×10^{-1}	6.86×10^{-1}	7.27×10^{-1}
Inverted Pendulum Controller	ITAE	$2.23 \times 10^{+2}$	$2.23 \times 10^{+2}$	$2.84 \times 10^{+2}$
	ITSE	$5.59 \times 10^{+2}$	$5.58 \times 10^{+2}$	$6.07 \times 10^{+2}$
	IAE	$2.78 \times 10^{+1}$	$2.77 \times 10^{+1}$	$3.12 \times 10^{+1}$
	ISE	$6.93 \times 10^{+1}$	$6.94 \times 10^{+1}$	$7.02 \times 10^{+1}$
	RMSE	1.48×10^0	1.48×10^0	8.76×10^{-1}
Autonomous Mobile Robot Controller	ITAE	$3.37 \times 10^{+2}$	$3.41 \times 10^{+2}$	$3.41 \times 10^{+2}$
	ITSE	$1.38 \times 10^{+2}$	$1.42 \times 10^{+2}$	$1.41 \times 10^{+2}$
	IAE	$1.44 \times 10^{+1}$	$1.46 \times 10^{+1}$	$1.46 \times 10^{+1}$
	ISE	6.07×10^0	6.28×10^0	6.21×10^0
	RMSE	2.06×10^{-1}	2.14×10^{-1}	2.03×10^{-1}

Table 5. Results of metrics of Mean Square Error (MSE) values to found by CSO algorithm.

Control Problem	Performance Index	Types of Experiment		
		Not AP	Type-1 Perturbation	Type-2 Perturbation
Water Tank Controller	BEST	8.22×10^{-4}	2.34×10^{-1}	2.19×10^{-1}
	WORST	8.17×10^{-2}	9.25×10^{-1}	7.69×10^{-1}
	AVERAGE	3.61×10^{-2}	4.21×10^{-1}	5.05×10^{-1}
	STANDARD DEVIATION (σ)	2.35×10^{-2}	1.86×10^{-1}	1.09×10^{-1}
Inverted Pendulum Controller	BEST	4.67×10^{-1}	4.80×10^{-1}	5.79×10^{-1}
	WORST	5.98×10^0	5.69×10^0	1.77×10^0
	AVERAGE	2.29×10^0	2.35×10^0	8.59×10^{-1}
	STANDARD DEVIATION (σ)	1.84×10^0	1.79×10^0	3.66×10^{-1}
Autonomous Mobile Robot Controller	BEST	3.87×10^{-5}	7.56×10^{-3}	3.81×10^{-4}
	WORST	1.44×10^0	2.16×10^0	3.48×10^{-1}
	AVERAGE	1.35×10^{-1}	1.79×10^{-1}	6.99×10^{-2}
	STANDARD DEVIATION (σ)	2.63×10^{-1}	4.18×10^{-1}	8.63×10^{-2}

Results on Table 5 shows an stabilization when level of noise are added in two control problems; inverted pendulum and autonomous mobile robot. For example, in the second control problem the average with band-limited perturbation is 8.59×10^{-1} compared to the average without perturbation is 2.29×10^0 ; similarly, for the third control problem the average value is 6.99×10^{-2} with band-limited perturbation more stable that the value of 1.79×10^{-1} with pulse generator perturbation and the value of 1.35×10^{-1} without perturbation in the model.

With the goal to analyze the convergence in each control problem, Figure 18 shows the best results in fitness function that the CSO algorithm finds in the results.

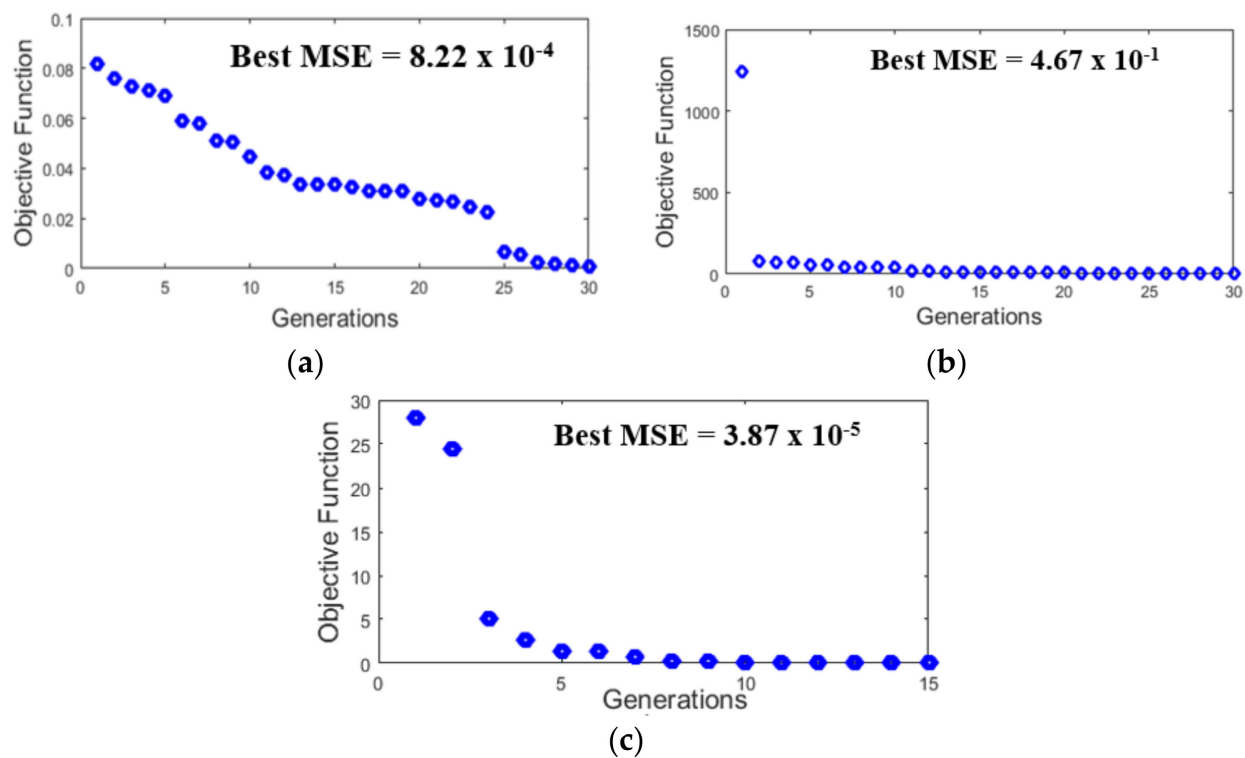


Figure 18. Convergence of Best Results. (a) Water Tank Controller, (b) Inverted Pendulum Controller and (c) Autonomous Mobile Robot Controller.

The best distribution in the MFs for the FLS that the CSO algorithm finds for each control problem is presented in Figures 19–21, respectively.

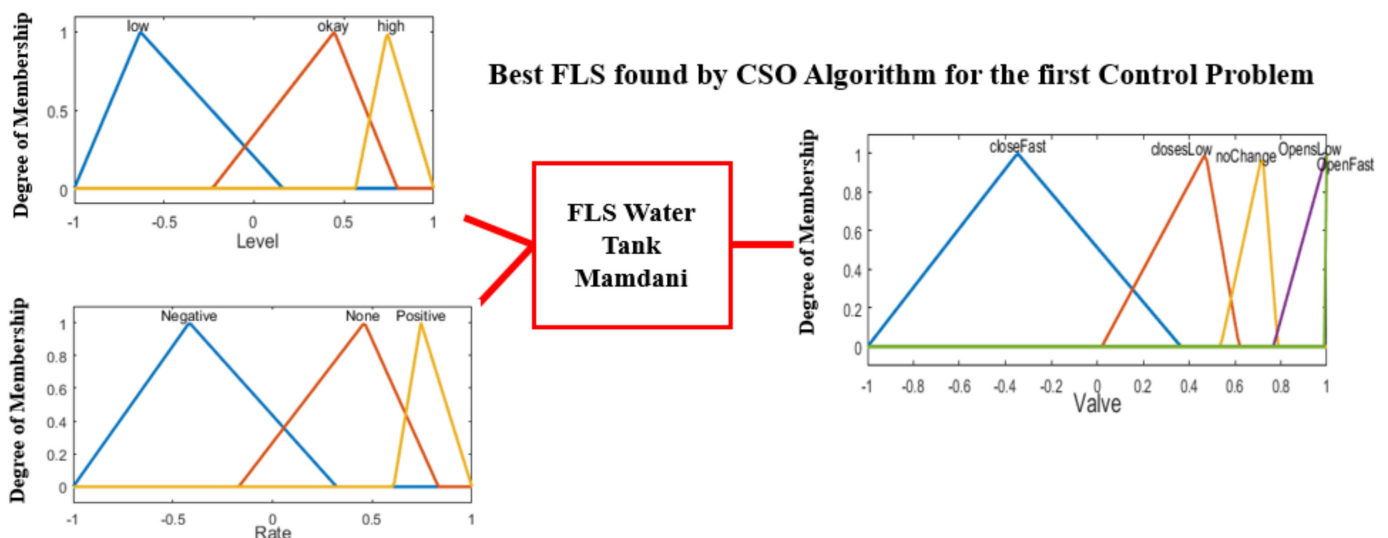


Figure 19. Best FLS that CSO algorithm finds for the Water Tank Controller.

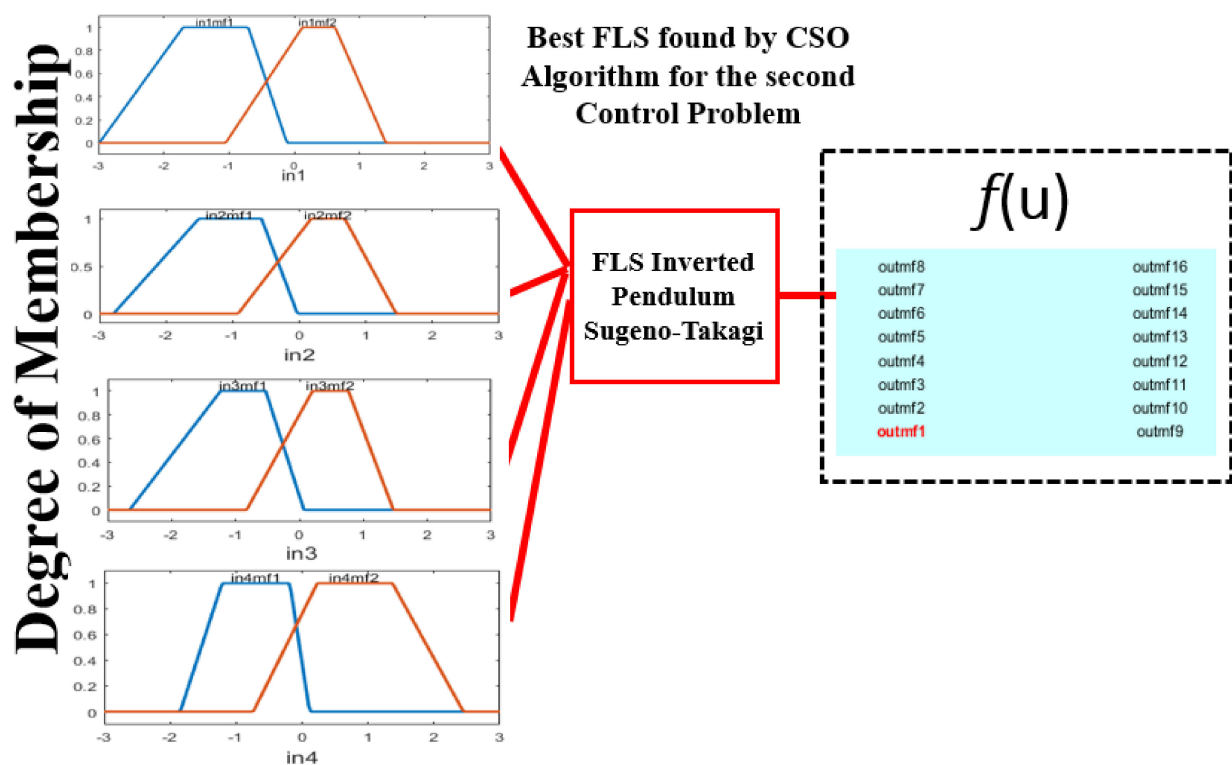


Figure 20. Best FLS that the CSO algorithm finds for the Inverted Pendulum Controller.

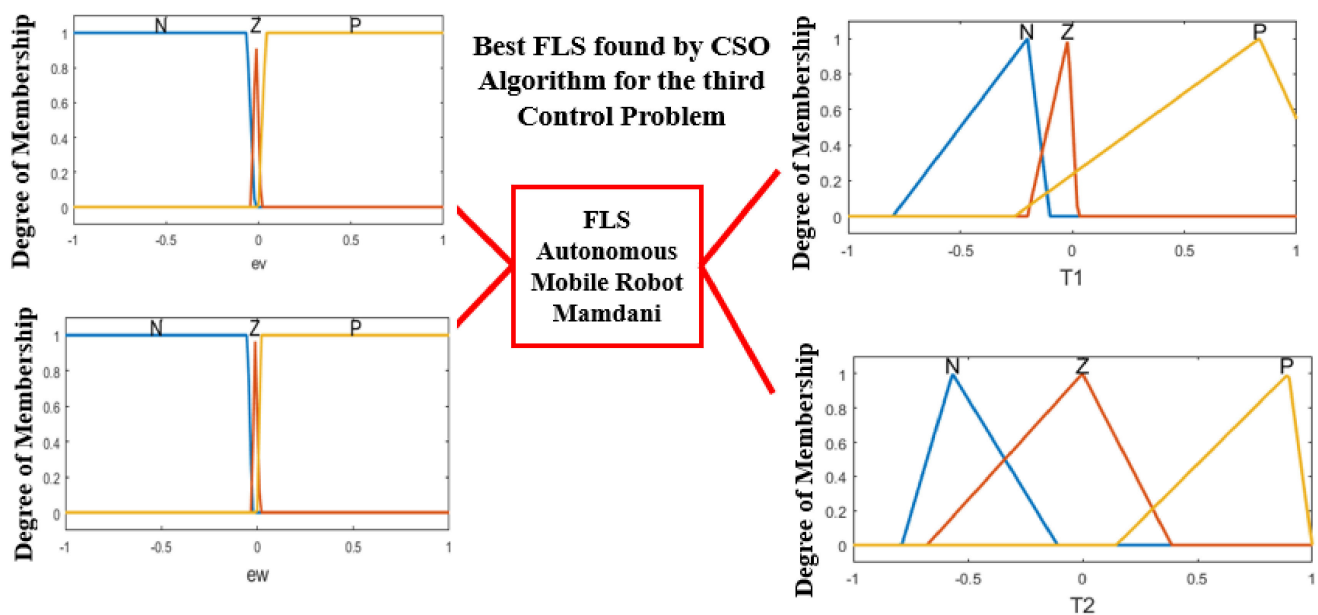


Figure 21. Best FLS that CSO algorithm finds for the Autonomous Mobile Robot Controller.

6. Comparative Analysis and Discussion of Results

This section presents a comparative analysis of the results when perturbations are added in the model. Also, a visualization of the different metrics that were used in this paper to evaluate the performance of the CSO algorithm is analyzed.

Figures 22–24 illustrate the behavior of proposed implementation of the CSO algorithm in the design of the MFs to create the FLS for the fitness function (MSE) and RMSE of the 30 experiments for each control problem, respectively.

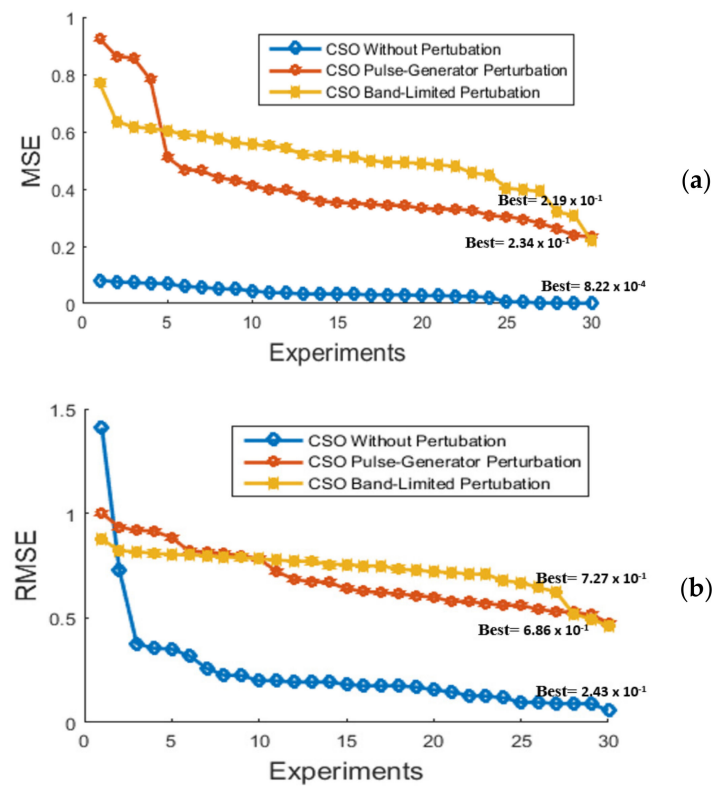


Figure 22. Comparative results of best (a) MSE and (b) RMSE for Water Tank Controller.

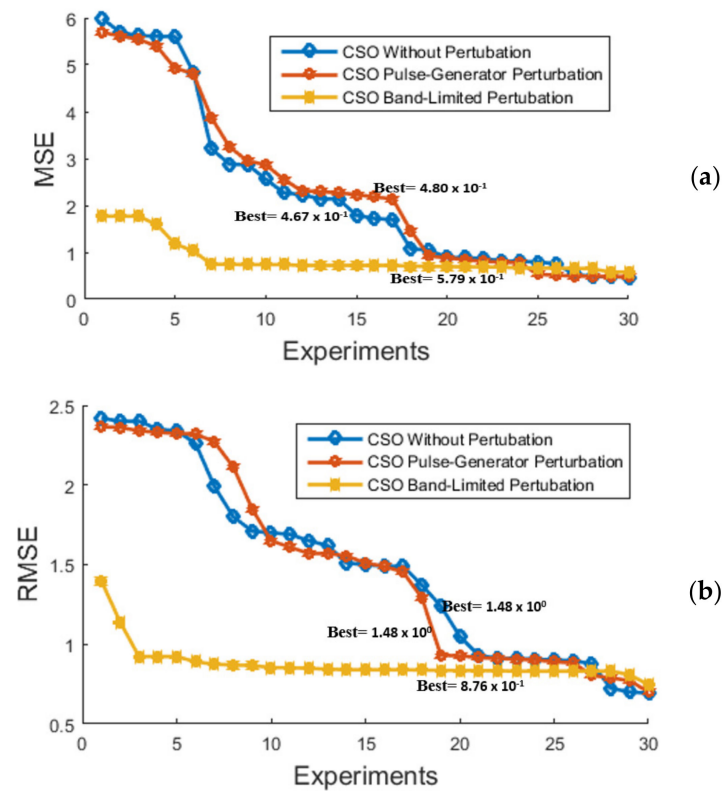


Figure 23. Comparative results of best (a) MSE and (b) RMSE for the Inverted Pendulum Controller.

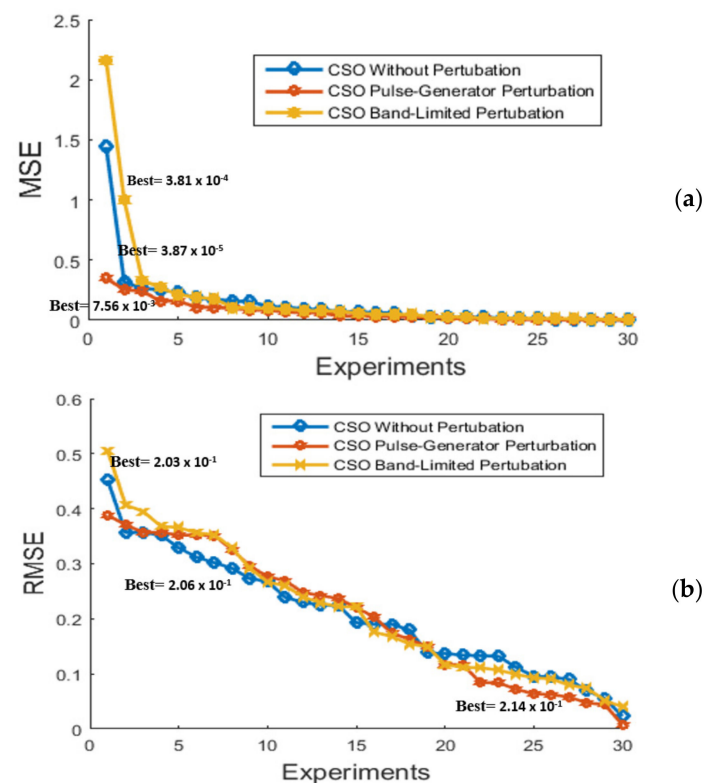


Figure 24. Comparative results of best (a) MSE and (b) RMSE for the Autonomous Mobile Robot Controller.

The results of Figure 22 show a better behavior without perturbation in the model, in this case this benchmark problem is considering a simple control problem. For the second case in the inverted pendulum (Figure 23), for both metrics when the band-limited perturbation is added the better results are found by CSO algorithm. This benchmark problem presents an efficient stabilization with the proposed implementation of CSO algorithm. Finally, the control problem of controlling the trajectory in an autonomous mobile robot shows a similar behavior even when perturbation is added in the model (Figure 24), whereby, this result allows evaluation of the efficiency and excellent performance of the CSO algorithms in the stabilization of fuzzy controllers.

Figure 25 shows the best trajectory of the autonomous mobile robot found by CSO algorithm in the design of the MFs of the FLS simulated for each type of experiment; without perturbation, with pulse generator and with band-limited perturbation, respectively.

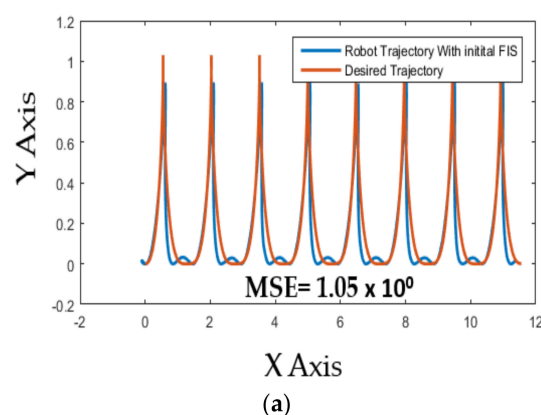


Figure 25. Cont.

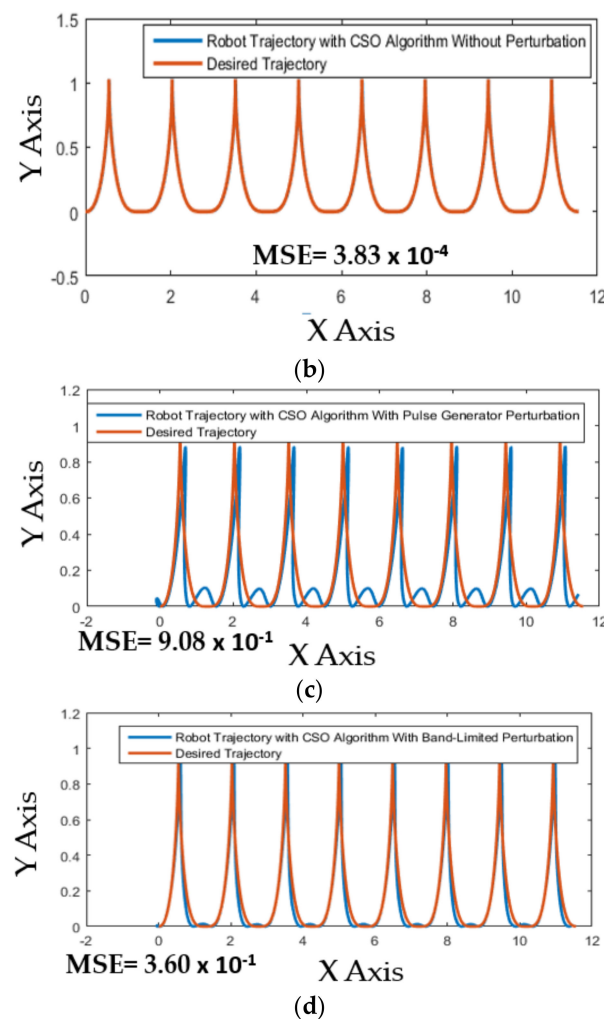


Figure 25. Comparative of trajectories for the Autonomous Mobile Robot Controller. (a) Initial FIS, (b) CSO algorithm without perturbation, (c) CSO algorithm with Pulse Generator Perturbation, and (d) CSO algorithm with Band-Limited Perturbation.

Figure 25 shows the efficiency of the CSO Algorithm when perturbation is added in the model, a stabilization of the fuzzy controller is illustrated when band-limited perturbation is added.

Table 6 shows a comparative with the BCO algorithm for the Water Tank Controller and Autonomous Mobile Robot Controller, the minimum and maximum values for the Fitness Function of MSE are presented, and the average RMSE is shown. This comparison was realized for 30 experiments in both works.

Table 6. Comparison of results with Bee Colony Optimization (BCO) algorithm.

Method	Control Problem	Minimum	Maximum	Average RMSE	Population	Iterations (BCO)— Generations (CSO)
Proposed CSO Algorithm	Water Tank Controller	8.22×10^{-4}	8.17×10^{-2}	2.43×10^{-1}	30	40
BCO Algorithm [65]	Water Tank Controller	5.50×10^{-2}	1.47×10^{-1}	5.60×10^{-1}	50	30
Proposed CSO Algorithm	Autonomous Mobile Robot Controller	3.87×10^{-5}	1.44×10^0	2.06×10^{-1}	20	15
BCO Algorithm [65]	Autonomous Mobile Robot Controller	2.00×10^{-3}	$1.40 \times 10^{+1}$	$2.26 \times 10^{+1}$	50	30

Results in Table 6 show minimal errors in the simulations with CSO algorithm. For example, in the case of water tank controller the minimal error is 8.22×10^{-4} and for BCO algorithm is 5.50×10^{-2} .

7. Conclusions

An efficient implementation of the CSO algorithm for the design of the Membership Functions for three control problems is presented in this paper. The proposed structure to find the optimal distribution of the MFs in each control problem demonstrates that when perturbation is added in the model, the CSO algorithm allows controlling a stabilization in three control problems; water tank, inverted pendulum and autonomous mobile robot controllers. The exploration and exploitation abilities in the CSO algorithm are an important aspect that is analyzed and reflected in the good results obtained. Several metrics were implemented, including ITAE, ITSE, ISE, IAE, MSE and RMSE, to evaluate of performance of the CSO algorithm.

The proposed implementation of CSO algorithm shows that it can be an excellent tool to find an optimal design in the MFs for the FLS to control and stabilize fuzzy controllers. Results in Tables 4 and 5 show minimal errors in the simulations. Figures 22–24 show that the convergence of CSO algorithm is fast and the minimal errors are obtained with few generations, this is due to its capacity of exploitation that presents this algorithm.

An interesting future work in this paper could be the implementation in the design of the MFs for an interval type-2 FLS. Also, it can be used for other fuzzy controllers; for example, flight simulations, and dynamic trajectories in autonomous mobile robots.

Author Contributions: Individual contributions by the authors are the following: Conceptualization, creation on main idea, writing-review and editing, L.A.-A.; formal analysis, O.C.; methodology, C.P., and validation, P.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank the program of the Division of Graduate Studies and Research of Tijuana Institute of Technology, specifically to Patricia Melin, program coordinator, for her interested in our research and for creating an excellent team working in collaboration with all the co-authors on this paper.

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

References

1. Shayokh, M.A.; Shin, S.Y. Bio inspired distributed WSN localization based on chicken swarm optimization. *Wirel. Personal Commun.* **2017**, *97*, 5691–5706. [\[CrossRef\]](#)
2. Banerjee, S.; Chattopadhyay, S. Improved serially concatenated convolution turbo code (SCCTC) using chicken swarm optimization. In Proceedings of the 2015 IEEE Power, Communication and Information Technology Conference (PCITC), Bhubaneswar, India, 15–17 October 2015; pp. 268–273.
3. Chen, Y.L.; He, P.L.; Zhang, Y.H. Combining penalty function with modified chicken swarm optimization for constrained optimization. In Proceedings of the First International Conference on Information Sciences, Machinery, Materials and Energy, Chongqing, China, 11–13 April 2015.
4. Deb, S.; Tammi, K.; Gao, X.Z.; Kalita, K.; Mahanta, P. A Hybrid Multi-Objective Chicken Swarm Optimization and Teaching Learning Based Algorithm for Charging Station Placement Problem. *IEEE Access* **2020**, *8*, 92573–92590. [\[CrossRef\]](#)
5. Deb, S.; Gao, X.Z.; Tammi, K.; Kalita, K.; Mahanta, P. A New Teaching–Learning-based Chicken Swarm Optimization Algorithm. *Soft Comput.* **2020**, *24*, 5313–5331. [\[CrossRef\]](#)
6. Deb, S.; Gao, X.Z.; Tammi, K.; Kalita, K.; Mahanta, P. Recent studies on chicken swarm optimization algorithm: A review (2014–2018). *Artif. Intell. Rev.* **2019**, *53*, 1737–1765. [\[CrossRef\]](#)
7. Hafez, A.I.; Zawbaa, H.M.; Emary, E.; Mahmoud, H.A.; Hassanien, A.E. An innovative approach for feature selection based on chicken swarm optimization. In Proceedings of the 2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR), Fukuoka, Japan, 13–15 November 2015; pp. 19–24.

8. Kurozawa, L.E.; Park, K.J.; Hubinger, M.D. Optimization of the enzymatic hydrolysis of chicken meat using response surface methodology. *J. Food Sci.* **2008**, *73*, C405–C412. [\[CrossRef\]](#)
9. Lin, F.J.; Teng, L.T.; Chu, H. A robust recurrent wavelet neural network controller with improved particle swarm optimization for linear synchronous motor drive. *IEEE Trans. Power Electron.* **2008**, *23*, 3067–3078. [\[CrossRef\]](#)
10. Qu, C.; Zhao, S.A.; Fu, Y.; He, W. Chicken swarm optimization based on elite opposition-based learning. *Math. Probl. Eng.* **2017**. [\[CrossRef\]](#)
11. Wang, B.; Li, W.; Chen, X.; Chen, H. Improved chicken swarm algorithms based on chaos theory and its application in wind power interval prediction. *Math. Probl. Eng.* **2019**. [\[CrossRef\]](#)
12. Wu, D.; Kong, F.; Gao, W.; Shen, Y.; Ji, Z. Improved chicken swarm optimization. In Proceedings of the 2015 IEEE international conference on cyber technology in automation, control, and intelligent systems (CYBER), Shenyang, China, 8–12 June 2015; pp. 681–686.
13. Wu, D.; Xu, S.; Kong, F. Convergence analysis and improvement of the chicken swarm optimization algorithm. *IEEE Access* **2016**, *4*, 9400–9412. [\[CrossRef\]](#)
14. Wu, Z.; Yu, D.; Kang, X. Application of improved chicken swarm optimization for MPPT in photovoltaic system. *Optim. Control Appl. Methods* **2018**, *39*, 1029–1042. [\[CrossRef\]](#)
15. Yaseen, H.T.; Mitras, B.A.; Khidhir, A.S.M. Hybrid Invasive Weed Optimization Algorithm with Chicken Swarm Optimization Algorithm to solve Global Optimization Problems. *Int. J. Comput. Netw. Commun. Secur.* **2018**, *6*, 173–181.
16. Zarlis, M.; Yanto, I.T.R.; Hartama, D. A framework of training ANFIS using chicken swarm optimization for solving classification problems. In Proceedings of the 2016 International conference on informatics and computing (ICIC), Mataram, Indonesia, 28–30 October 2016; pp. 437–441.
17. Liang, J.; Wang, L.; Ma, M.; Zhang, J. A fast SAR image segmentation method based on improved chicken swarm optimization algorithm. *Multimed. Tools Appl.* **2018**, *77*, 31787–31805. [\[CrossRef\]](#)
18. Miramontes, I.; Melin, P.; Prado-Arechiga, G. Comparative Study of Bio-inspired Algorithms Applied in the Optimization of Fuzzy Systems. In *Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine*; Springer: Cham, Switzerland, 2020; pp. 219–231.
19. Tripathi, A.K.; Garg, P.; Tripathy, A.; Vats, N.; Gupta, D.; Khanna, A. Prediction of Cervical Cancer Using Chicken Swarm Optimization. In *International Conference on Innovative Computing and Communications*; Springer: Singapore, 2020; pp. 591–604.
20. Lu, K.; Zhou, W.; Zeng, G.; Zheng, Y. Constrained population extremal optimization-based robust load frequency control of multi-area interconnected power system. *Int. J. Electr. Power Energy Syst.* **2019**, *105*, 249–271. [\[CrossRef\]](#)
21. Zeng, G.Q.; Chen, J.; Dai, Y.X.; Li, L.M.; Zheng, C.W.; Chen, M.R. Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization. *Neurocomputing* **2015**, *160*, 173–184. [\[CrossRef\]](#)
22. Huh, J.H.; Seo, Y.S. Understanding edge computing: Engineering evolution with artificial intelligence. *IEEE Access* **2019**, *7*, 164229–164245. [\[CrossRef\]](#)
23. Kim, J.C.; Huh, J.H.; Ko, J.S. Improvement of MPPT control performance using fuzzy control and VGPI in the PV system for micro grid. *Sustainability* **2019**, *11*, 5891. [\[CrossRef\]](#)
24. Amador-Angulo, L.; Castillo, O. Optimization of Fuzzy Trajectory Tracking in Autonomous Mobile Robots Based on Bio-inspired Algorithms. In *Recent Advances of Hybrid Intelligent Systems Based on Soft Computing*; Springer: Cham, Switzerland, 2021; pp. 249–271.
25. Kim, J.C.; Huh, J.H.; Ko, J.S. Optimization design and test bed of fuzzy control rule base for PV system MPPT in micro grid. *Sustainability* **2020**, *12*, 3763. [\[CrossRef\]](#)
26. Li, Y.; Wu, Y.; Qu, X. Chicken swarm-based method for ascent trajectory optimization of hypersonic vehicles. *J. Aerosp. Eng.* **2017**, *30*, 04017043. [\[CrossRef\]](#)
27. Amador-Angulo, L.; Castillo, O. Optimal Design of Fuzzy Logic Systems Through a Chicken Search Optimization Algorithm Applied to a Benchmark Problem. In *Recent Advances of Hybrid Intelligent Systems Based on Soft Computing*; Springer: Cham, Switzerland, 2021; pp. 229–247.
28. Liang, X.; Kou, D.; Wen, L. An Improved Chicken Swarm Optimization Algorithm and its Application in Robot Path Planning. *IEEE Access* **2020**, *8*, 49543–49550. [\[CrossRef\]](#)
29. Mu, Y.; Zhang, L.; Chen, X.; Gao, X. Optimal trajectory planning for robotic manipulators using chicken swarm optimization. In Proceedings of the 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 27–28 August 2016; Volume 2, pp. 369–373.
30. Wu, Y.; Yan, B.; Qu, X. Improved chicken swarm optimization method for reentry trajectory optimization. *Math. Probl. Eng.* **2018**. [\[CrossRef\]](#)
31. Amador-Angulo, L.; Mendoza, O.; Castro, J.R.; Rodríguez-Díaz, A.; Melin, P.; Castillo, O. Fuzzy sets in dynamic adaptation of parameters of a bee colony optimization for controlling the trajectory of an autonomous mobile robot. *Sensors* **2016**, *16*, 1458. [\[CrossRef\]](#) [\[PubMed\]](#)
32. Caraveo, C.; Valdez, F.; Castillo, O. Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation. *Appl. Soft Comput.* **2016**, *43*, 131–142. [\[CrossRef\]](#)

33. Baygi, S.M.H.; Karsaz, A.; Elahi, A. A hybrid optimal PID-fuzzy control design for seismic excited structural system against earthquake: A salp swarm algorithm. In Proceedings of the 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), Kerman, Iran, 28 February–2 March 2018; pp. 220–225.
34. Bhattacharjee, V.; Chatterjee, D.; Karabasoglu, O. Hybrid control strategy for a semi active suspension system using fuzzy logic and bio-inspired chaotic fruit fly algorithm. *arXiv* **2017**, arXiv:1703.08878.
35. Banjanovic-Mehmedovic, L.; Baluković, A. PSO Optimized Fuzzy Controller for Mobile Robot Path Tracking. In *International Conference “New Technologies, Development and Applications”*; Springer: Cham, Switzerland, 2020; pp. 413–421.
36. Carvajal, O.R.; Castillo, O.; Soria, J. Optimization of membership function parameters for fuzzy controllers of an autonomous mobile robot using the flower pollination algorithm. *J. Autom. Mob. Robot. Intell. Syst.* **2018**, *12*, 44–49. [[CrossRef](#)]
37. García-Gutiérrez, G.; Arcos-Aviles, D.; Carrera, E.V.; Guinjoan, F.; Motoasca, E.; Ayala, P.; Ibarra, A. Fuzzy logic controller parameter optimization using metaheuristic cuckoo search algorithm for a magnetic levitation system. *Appl. Sci.* **2019**, *9*, 2458. [[CrossRef](#)]
38. Hernández, E.; Castillo, O.; Soria, J. Design of Optimal Fuzzy Controllers for Autonomous Mobile Robots Using the Grey Wolf Algorithm. In *International Fuzzy Systems Association World Congress*; Springer: Cham, Switzerland, 2019; pp. 285–295.
39. Huang, H.C. Fusion of modified bat algorithm soft computing and dynamic model hard computing to online self-adaptive fuzzy control of autonomous mobile robots. *IEEE Trans. Ind. Inform.* **2016**, *12*, 972–979. [[CrossRef](#)]
40. Khooban, M.H.; Liaghat, A. A time-varying strategy for urban traffic network control: A fuzzy logic control based on an improved black hole algorithm. *Int. J. Bio-Inspired Comput.* **2017**, *10*, 33–42. [[CrossRef](#)]
41. Lagunes, M.L.; Castillo, O.; Soria, J. Optimization of membership function parameters for fuzzy controllers of an autonomous mobile robot using the firefly algorithm. In *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*; Springer: Cham, Switzerland, 2018; pp. 199–206.
42. Ni, J.; Wu, L.; Fan, X.; Yang, S.X. Bioinspired intelligent algorithm and its applications for mobile robot control: A survey. *Comput. Intell. Neurosci.* **2016**. [[CrossRef](#)]
43. Villa, T.C.; Castillo, O. Adaptation of Parameters with Binary Cat Swarm Optimization Algorithm of Controller for a Mobile Autonomous Robot. In *Hybrid Intelligent Systems in Control, Pattern Recognition and Medicine*; Springer: Cham, Switzerland, 2020; pp. 35–46.
44. Yen, C.T.; Cheng, M.F. A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance. *Microsyst. Technol.* **2018**, *24*, 125–135. [[CrossRef](#)]
45. Chen, H.M.; Chen, Z.Y.; Su, J.P. Design of a sliding mode controller for a water tank liquid level control system. In Proceedings of the Second International Conference on Innovative Computing, Information and Control (ICICIC 2007), Kumamoto, Japan, 5–7 September 2007; p. 335.
46. Hassanzadeh, I.; Mobayen, S. Controller design for rotary inverted pendulum system using evolutionary algorithms. *Math. Probl. Eng.* **2011**. [[CrossRef](#)]
47. Margaliot, M.; Langholz, G. Fuzzy control of a benchmark problem: A computing with words approach. *IEEE Trans. Fuzzy Syst.* **2004**, *12*, 230–235. [[CrossRef](#)]
48. Akhmedova, S.; Semenkin, E.; Stanovov, V.; Vishnevskaya, S. Fuzzy logic controller design for tuning the cooperation of biology-inspired algorithms. In Proceedings of the International Conference on Swarm Intelligence, Fukuoka, Japan, 27 July–1 August 2017; pp. 269–276.
49. Baker, A.A.; Ghadi, Y.Y. Autonomous system to control a mobile robot. *Bull. Electr. Eng. Inform.* **2020**, *9*, 1711–1717. [[CrossRef](#)]
50. Lubin, T.; Mendes, E.; Marchand, C. Fuzzy controller in AC servo motor drive. In Proceedings of the 1995 Seventh International Conference on Electrical Machines and Drives, Durham, UK, 11–13 September 1995. [[CrossRef](#)]
51. Rodriguez-Angeles, A.; Vazquez Chavez, L.F. Bio-inspired decentralized autonomous robot mobile navigation control for multi agent systems. *Kybernetika* **2018**, *54*, 135–154. [[CrossRef](#)]
52. Meng, X.; Liu, Y.; Gao, X.; Zhang, H. A new bio-inspired algorithm: Chicken swarm optimization. In Proceedings of the International Conference in Swarm Intelligence, Hefei, China, 17–20 October 2014; pp. 86–94.
53. Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning. *Part II Inf. Sci.* **1975**, *8*, 301–357. [[CrossRef](#)]
54. Zadeh, L.A. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* **1997**, *90*, 111–127. [[CrossRef](#)]
55. Mamdani, E.H. Application of fuzzy algorithms for control of simple dynamic plant. *Proc. Inst. Electr. Eng. IET* **1974**, *121*, 1585–1588. [[CrossRef](#)]
56. Kumar, A.; Saini, B. A Sugeno-Mamdani Fuzzy System Based Soft Computing Approach Towards Sensor Node Localization with Optimization. In Proceedings of the International Conference on Next Generation Computing Technologies, Dehradun, India, 30–31 October 2017; pp. 40–55.
57. Devi, M.S.; Soranamageswari, M. A hybrid technique of Mamdani and Sugeno based fuzzy interference system approach. In Proceedings of the 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), Ernakulam, India, 16–18 March 2016; pp. 340–342.
58. Siettos, C.I.; Bafas, G.V. Semiglobal stabilization of nonlinear systems using fuzzy control and singular perturbation methods. *Fuzzy Sets Syst.* **2002**, *129*, 275–294. [[CrossRef](#)]

-
59. Ontiveros-Robles, E.; Melin, P.; Castillo, O. Comparative analysis of noise robustness of type 2 fuzzy logic controllers. *Kybernetika* **2018**, *54*, 175–201. [[CrossRef](#)]
 60. Liu, H.; Hao, K.; Lai, X. Fuzzy saturated output feedback tracking control for robot manipulators: A singular perturbation theory based approach. *Int. J. Adv. Robot. Syst.* **2011**, *8*, 35. [[CrossRef](#)]
 61. Sáez, J.A.; Galar, M.; Luengo, J.; Herrera, F. INFFC: An iterative class noise filter based on the fusion of classifiers with noise sensitivity control. *Inf. Fusion* **2016**, *27*, 19–32. [[CrossRef](#)]
 62. Wu, L.; Su, X.; Shi, P. *Fuzzy Control Systems with Time-Delay and Stochastic Perturbation*; Springer: Berlin/Heidelberg, Germany, 2015.
 63. Li, T.H.S.; Lin, K.J. Composite fuzzy control of nonlinear singularly perturbed systems. *IEEE Trans. Fuzzy Syst.* **2007**, *15*, 176–187. [[CrossRef](#)]
 64. Zhang, J.; Chen, W.H.; Lu, X. Robust fuzzy stabilization of nonlinear time-delay systems subject to impulsive perturbations. *Commun. Nonlinear Sci. Numer. Simul.* **2020**, *80*, 104953. [[CrossRef](#)]
 65. Amador-Angulo, L.; Castillo, O. A new fuzzy bee colony optimization with dynamic adaptation of parameters using interval type-2 fuzzy logic for tuning fuzzy controllers. *Soft Comput.* **2018**, *22*, 571–594. [[CrossRef](#)]