



Article Optimization of Sliding Mode Control to Save Energy in a SCARA Robot

Luis Arturo Soriano ¹, José de Jesús Rubio ²,*^(D), Eduardo Orozco ², Daniel Andres Cordova ², Genaro Ochoa ², Ricardo Balcazar ², David Ricardo Cruz ³, Jesus Alberto Meda-Campaña ⁴, Alejandro Zacarias ², and Guadalupe Juliana Gutierrez ²

- ¹ Departamento de Ingeniería Mecánica Agrícola, Universidad Autónoma Chapingo, Km. 38.5 Carretera México-Texcoco, Texcoco 56235, Mexico; Isorianoa@chapingo.mx
- ² Sección de Estudios de Posgrado e Investigación, ESIME Azcapotzalco, Instituto Politécnico Nacional, Av. de Las Granjas No. 682, Col. Santa Catarina, Mexico City 02250, Mexico; orozco_ing@hotmail.com (E.O.); cordovadaniel93@hotmail.com (D.A.C.); genaro.ochoa@itstb.edu.mx (G.O.); alaks1331@gmail.com (R.B.); azacarias@ipn.mx (A.Z.); ggutierrezp@ipn.mx (G.J.G.)
- ³ Instituto Tecnológico de Tlalnepantla, Tecnológico Nacional de México, Av. Instituto Tecnológico s/n, Col. La Comunidad, Tlalnepantla de Baz 54070, Mexico; ingdavidrcruz@gmail.com
- ⁴ Sección de Estudios de Posgrado e Investigación, ESIME Zacatenco, Instituto Politécnico Nacional, Av. IPN s/n, Col. Lindavista, Gustavo A. Madero, Mexico City 07738, Mexico; jmedac@ipn.mx
- Correspondence: rubio.josedejesus@gmail.com or jrubioa@ipn.mx; Tel.: +52-55-57296000-64497

Abstract: Sliding mode control is a robust technique that is used to overcome difficulties such as parameter variations, unmodeled dynamics, external disturbances, and payload changes in the position-tracking problem regarding robots. However, the selection of the gains in the controller could produce bigger forces than are required to move the robots, which requires spending a large amount of energy. In the literature, several approaches were used to manage these features, but some proposals are complex and require tuning the gains. In this work, a sliding mode controller was designed and optimized in order to save energy in the position-tracking problem of a two-degree-of-freedom SCARA robot. The sliding mode controller gains were optimized usinga Bat algorithm to save energy by minimizing the forces. Finally, two controllers were designed and implemented in the simulation, and as a result, adequate controller gains were found that saved energy by minimizing the forces.

Keywords: energy; optimization; Bat algorithm; sliding mode; SCARA robot

1. Introduction

Sliding mode control (SMC) is a robust control technique that is used to command nonlinear systems, such as robots. SMC presents several advantages that can be employed in nonlinear systems due to its robustness against parametric variations, external disturbances, unmodeled dynamics, and uncertainties [1]. However, one of the major drawbacks in the control of robots with SMC is the use of big forces in the reaching law that is required to move the robots. During industrial operations, the big forces in the electric actuator control of the robots can result in spending large amounts of energy [2].

In order to save energy by minimizing the forces in the sliding mode control, several methods were proposed in the literature. In [3], the authors proposed a neuro-sliding mode controller in which discontinuous control was replaced by a neural network that was combined with an error estimator, and then the controller was applied to a two-degree-of-freedom (DOF) planar parallel manipulator. In [4], a method was presented and implemented using the current regulation of motor drives, and their results showed that the forces were minimized. In [5], cascaded control was used to manage a converter at the same time that the forces were minimized. In [6], integral sliding mode control was proposed to enforce the sliding mode robustness from the initial time instant. In [7], a robust controller



Citation: Soriano, L.A.; Rubio, J.d.J.; Orozco, E.; Cordova, D.A.; Ochoa, G.; Balcazar, R.; Cruz, D.R.; Meda-Campaña, J.A.; Zacarias, A.; Gutierrez, G.J. Optimization of Sliding Mode Control to Save Energy in a SCARA Robot. *Mathematics* **2021**, *9*, 3160. https://doi.org/10.3390/ math9243160

Academic Editor: Svetlana V. Solodusha

Received: 11 November 2021 Accepted: 6 December 2021 Published: 8 December 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/). as a combination of the super-twisting control algorithm and time delay estimation method was proposed for a robot with uncertain motion in order to save energy by minimizing the forces and to reduce the uncertainties and disturbances. In [8], the combination of an adaptation law, sliding mode disturbance observer, and a continuous nonsingular terminal sliding mode control was proposed to compensate for the uncertainty and external disturbance and to maintain the robustness; this combination resulted in the minimization of the forces. In [9], an artificial neuro-fuzzy system with sliding mode control applied to a two-DOF rigid robot manipulator was proposed to generate an adaptation control in order to increase the robustness against disturbances. Moreover, this controller was compared with a proportional integral derivative sliding mode control, and the gains were optimized using a particle swarm optimization algorithm; however, the complexity of the controller design increased due to the several gains that needed to be tuned. In [10], an adaptive neural gain scheduling sliding mode control method was presented to minimize the forces and it was applied to a quadcopter with external disturbances. In [11,12], adaptive robust controllers were introduced for high-precision tracking control problems of robots with output constraints. In [13,14], model-free finite-time terminal sliding mode control schemes of uncertain robots were discussed. In [15], a model-free high-order sliding-mode controller with a time-base generator was implemented to achieve finite-time convergence. Nevertheless, some of these mentioned approaches of the sliding mode control tended to result in big forces.

On the other hand, the Bat algorithm was applied for several control problems. In [16,17], the Bat algorithm was proposed. In [18], the combination of a proportional integral derivative (PID) controller and a Bat algorithm was introduced to obtain pulsewidth modulation for converters. In [19], a fractional order PID controller that was based on the Bat algorithm was implemented to improve the performance of a distillation column process. In [20], the fusion of a modified Bat algorithm and an adaptive fuzzy controller was introduced for autonomous mobile robots. In [21], a fuzzy PID controller with a Bat algorithm was introduced for a wind turbine. In [22], a tracking technique that was based on a fuzzy controller and the Bat algorithm was created to desirably tune the control parameters. In [23], the design of a fuzzy PID controller with a Bat algorithm is presented for a brushless direct current drive. In [24], an adaptive fractional order PID controller using a Bat algorithm was proposed for a caterpillar robot manipulator. In [25], a PID controller using a Bat algorithm was proposed for the control of a MEMS gyroscope. In [26], a system for control of enumeration strategies that was based on a Bat algorithm is discussed. In [27], a fuzzy controller rule base using a Bat algorithm was addressed for a magnetic ball suspension system. The mentioned results presented the combination of a Bat algorithm, fuzzy controller, and PID controller; hence, it would be desirable to present the combination of the Bat algorithm and sliding mode controller.

In this work, the optimization of sliding mode controller gains using the Bat algorithm was proposed in order to save energy by minimizing the forces. This design includes a sliding mode control that is based on the reaching law with a constant rate, which was selected due to its simplicity. The optimization based on the Bat algorithm was proposed due to the small number of gains to be tuned, its flexibility, and its simplicity. The proposed control was designed to overcome the position-tracking problem in a two-DOF SCARA robot. The contributions of the proposed approach are described as follows: (a) the Bat algorithm was modified with four additional steps so that it can be applied to find the best gains to save energy by minimizing the forces of a two-DOF SCARA robot, (b) the Bat algorithm was modified with two additional steps so that its gains remain bounded, and (c) the best gains of the Bat algorithm are used to save energy by minimizing the forces of the sliding mode controller for the position-tracking problem in a two-DOF SCARA robot.

The organization of the paper is as follows: In Section 2, the dynamic model description of a two-DOF SCARA robot is presented, which is described using the Euler–Lagrange formulation and the data were based on a real robot presented in [28]. In Section 3, an SMC design for the position-tracking problem of a two-DOF SCARA robot is presented. In Section 4, the optimization of the SMC and an explanation about how the Bat algorithm works are presented, where the proposed algorithm optimized the gains of SMC in order to save energy by minimizing the forces. Finally, in Section 5, the simulation results of the SMC and SMC with the Bat algorithm are presented.

2. Dynamic Model of a Two-DOF SCARA Robot

According to [29], the dynamics of the robot in Figure 1 can be described using the Euler–Lagrange equation.



Figure 1. A two-DOF SCARA robot.

The Euler–Lagrange equation is shown as follows:

$$\frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}_i} \right] - \frac{\partial L(q, \dot{q})}{\partial \dot{q}_i} = u_i, \tag{1}$$

where u_i are the forces that are related to the energy; q, q, and q are the joint positions, joint velocities, and joint accelerations, respectively; and L(q, q) is the Lagrangian that is defined as the difference between the kinetic energy K(q, q) and the potential energy N(q, q) and is represented by the next equation:

$$L(q, \dot{q}) = K(q, \dot{q}) - N(q, \dot{q}),$$
⁽²⁾

The general dynamic model of a two-DOF SCARA robot can be described as

$$u = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + f(\dot{q}), \tag{3}$$

where M(q) is a 2 × 2 matrix that represents the kinetic energy of system, C(q, q) is a 2 × 2 matrix that includes the Coriolis force and centrifugal force of the system, $f(\dot{q})$ is a 2 × 1 vector that contains the static and Coulomb friction losses, and u is a 2 × 1 vector that contains the forces. Thus, the terms of the dynamic model can be rewritten in matrix form as

$$\begin{bmatrix} u_1\\ u_2 \end{bmatrix} = \begin{bmatrix} M_{11}(q) & M_{12}(q)\\ M_{21}(q) & M_{22}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_1\\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11}(q,\dot{q}) & C_{12}(q,\dot{q})\\ C_{21}(q,\dot{q}) & C_{22}(q,\dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_1\\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} f_1(\dot{q})\\ f_2(\dot{q}) \end{bmatrix},$$
(4)

where

$$\begin{split} M_{11}(q) &= 1.7277 + 0.1908 \cos(q_2), \\ M_{12}(q) &= 0.0918 + 0.0954 \cos(q_2), \\ M_{21}(q) &= 0.3340 + 0.3418 \cos(q_2), \\ M_{22}(q) &= 0.9184, \\ C_{11}(q, q) &= 31.8192 - 0.0954 \sin(q_2)q_2, \\ C_{12}(q, q) &= -0.0954 \sin(q_2)(q_1 + q_2), \\ C_{21}(q, q) &= 0.34128 \sin(q_2)q_1, \\ C_{22}(q, q) &= 12.5783, \\ f_1(q) &= 1.0256 sign(q_1), \\ f_2(q) &= 1.7842 sign(q_2), \end{split}$$
(5)

The values of the dynamic model that is presented in Equations (4) and (5) of the two-DOF SCARA robot were obtained from [28], which presented an approximation from real robot parameters. The dynamic model that is presented in Equations (4) and (5) can be rewritten in terms of acceleration as

$$\begin{bmatrix} \ddot{u}_1\\ \ddot{q}_2\\ \ddot{u}_2 \end{bmatrix} = \begin{bmatrix} M_{11}(q) & M_{12}(q)\\ M_{21}(q) & M_{22}(q) \end{bmatrix}^{-1} \left(\begin{bmatrix} u_1\\ u_2 \end{bmatrix} - \begin{bmatrix} C_{11}(q,\dot{q}) & C_{12}(q,\dot{q})\\ C_{21}(q,\dot{q}) & C_{22}(q,\dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_1\\ \dot{q}_2 \end{bmatrix} - \begin{bmatrix} f_1(\dot{q})\\ f_2(\dot{q}) \end{bmatrix} \right)$$
(6)

3. Sliding Mode Controller Design

The proposed SMC controller for a two-DOF SCARA robot is shown in Figure 2. The design of this section is important because of the following two reasons: (1) the SMC will be used with the Bat algorithm in the next section, and (2) in the simulations, the SMC was compared with the SMC with the Bat algorithm in the next section.



Figure 2. Sliding mode controller for a two-DOF SCARA robot.

Considering the SCARA robot dynamic model that is presented above, the proposed sliding mode controller was designed with the following sliding mode function:

$$s = \beta e + \dot{e},\tag{7}$$

where $\beta > 0$ is a gain that must satisfy the Hurwitz condition. *e* is the position error, which is defined as the difference between the reference position and real position, i.e.,

$$e = q_d - q, \tag{8}$$

e is given as the difference between the derivatives of the reference position and the real position, i.e.,

$$\dot{e} = \dot{q}_d - q, \tag{9}$$

Accordingly to Equations (7)–(9), the derivative of the above sliding mode function is

$$\dot{s} = \beta \dot{e} + \ddot{e}, \tag{10}$$

$$\dot{s} = \beta \left(\dot{q}_d - \dot{q} \right) + \left(\ddot{q}_d - \ddot{q} \right), \tag{11}$$

According to [30], there are several exponential reaching laws that help to save energy by minimizing the forces, but each one requires certain gains. The tuning process of the gains involves the knowledge of certain variables of the robot, or in other cases, this process is based on simulations. In this work, a conventional exponential reaching law was selected since it requires only one gain to be optimized in order to save energy by minimizing the forces and to reach the reference position in the system. Thus, the reaching law is

$$\dot{s} = -\xi \operatorname{sgn}(s),$$
 (12)

where $\xi > 0$. Therefore, relating Equations (11) and (12), the result is

$$-\zeta \operatorname{sgn}(s) = \beta \left(\dot{q}_d - \dot{q} \right) + \left(\ddot{q}_d - \ddot{q} \right), \tag{13}$$

Hence, Equation (3) can also be written as

и

$$u = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + f(\dot{q}), \tag{14}$$

$$M(q)\ddot{q} = u - C(q, \dot{q})\dot{q} - f(\dot{q}), \tag{15}$$

$$\ddot{q} = M(q)^{-1} \Big[u - C(q, \dot{q})\dot{q} - f(\dot{q}) \Big],$$
(16)

where u represents the forces. Substituting Equation (16) into Equation (13), we have

$$-\xi \operatorname{sgn}(s) = \beta \left(\dot{q}_d - \dot{q} \right) + \left(\ddot{q}_d - \ddot{q} \right), \tag{17}$$

$$-\xi \operatorname{sgn}(s) = \beta \left(\dot{q}_d - \dot{q} \right) + \left(\ddot{q}_d - M(q)^{-1} \left(u - C(q, \dot{q})\dot{q} - f(\dot{q}) \right) \right),$$
(18)

Thus, the design of the sliding mode control (SMC) is as follows [31–34]:

$$M(q)^{-1}\left(u - C(q, \dot{q})\dot{q} - f(\dot{q})\right) = \xi \operatorname{sgn}(s) + \beta \left(\dot{q}_d - \dot{q}\right) + \ddot{q}_d, \tag{19}$$

$$u - C(q, \dot{q})\dot{q} - f(\dot{q}) = M(q) \left(\xi \operatorname{sgn}(s) + \beta \left(\dot{q}_d - \dot{q}\right) + \ddot{q}_d\right),$$
(20)

$$= M(q) \left(\xi \operatorname{sgn}(s) + \beta \left(\dot{q}_d - \dot{q} \right) + \ddot{q}_d \right) + C(q, \dot{q})\dot{q} + f(\dot{q}),$$

$$u = u_{eq} + u_{sw},$$

$$u_{eq} = M(q) \left(\beta \left(\dot{q}_d - \dot{q} \right) + \ddot{q}_d \right) + C(q, \dot{q})\dot{q} + f(\dot{q}),$$

$$u_{sw} = M(q) (\xi \operatorname{sgn}(s)),$$
(21)

where u_{eq} is the equivalent control, u_{sw} is the switching control, $\beta = diag(\beta_1, \beta_2)$, and $\xi = diag(\xi_1, \xi_2)$. Moreover, Equation (21) can also be written in matrix terms as:

$$\begin{bmatrix} u_{1} \\ u_{2} \end{bmatrix} = \begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \begin{bmatrix} \xi_{1} \operatorname{sgn}(s) + \beta_{1} \left(\dot{q}_{1d} - \dot{q}_{1} \right) + \ddot{q}_{1d} \\ \xi_{2} \operatorname{sgn}(s) + \beta_{2} \left(\dot{q}_{2d} - \dot{q}_{2} \right) + \ddot{q}_{2d} \end{bmatrix} + \begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_{1} \\ \dot{q}_{2} \end{bmatrix} + \begin{bmatrix} f_{1}(\dot{q}) \\ f_{2}(\dot{q}) \end{bmatrix},$$
(22)

Equation (22) is important because it is used in the design of the next section. Substituting Equation (16) into Equation (11):

$$\dot{s} = \beta \left(\dot{q}_d - \dot{q} \right) + \ddot{q}_d - M(q)^{-1} \left(u - C(q, \dot{q})\dot{q} - f(\dot{q}) \right), \tag{23}$$

Now, substituting Equation (21) into Equation (23):

$$\dot{s} = \beta \left(\dot{q}_d - \dot{q} \right) + \ddot{q}_d$$

$$-M(q)^{-1}M(q) \left(\xi \operatorname{sgn}(s) + \beta \left(\dot{q}_d - \dot{q} \right) + \ddot{q}_d + C(q, \dot{q})\dot{q} + f(\dot{q}) - C(q, \dot{q})\dot{q} - f(\dot{q}) \right),$$
(24)

$$\dot{s} = \beta \left(\dot{q}_d - \dot{q} \right) + \ddot{q}_d - \xi \operatorname{sgn}(s) - \beta \left(\dot{q}_d - \dot{q} \right) - \ddot{q}_d, \tag{25}$$

$$\dot{s} = -\xi \operatorname{sgn}(s),$$
 (26)

This reaching law with the switching variable is used to achieve the switching manifolds. When ξ is too small, this results in big forces that require spending large amounts of energy, or when ξ is too big, this results in intense chattering.

Remark 1. If Equation (12) is replaced with $s = -\xi_1 \text{sgn}(s_1) - \xi_2 \text{sgn}(s_2)$, where s_1 and s_2 are different sliding modes, discontinuous differential equation techniques would be used to frame this problem into more of a dynamical system setting [35,36]. It would be one alternative to show the applicability of the proposed approach to other more general and common real-world problems.

4. Optimization of SMC Using the Bat Algorithm

The SMC controller that was optimized with the Bat algorithm is shown in Figure 3. The algorithm considers the position regulation of a two-DOF SCARA robot in order to save energy by minimizing the forces that are required to move the two-DOF SCARA robot. From the SMC in Equation (21), one Bat algorithm adjusts the gains $\beta = diag(\beta_1, \beta_2)$ and the other Bat algorithm adjusts the gains $\xi = diag(\xi_1, \xi_2)$.



Figure 3. Sliding mode controller with the Bat algorithm for a two-DOF SCARA robot.

Equations (27)–(36) describe the steps of the Bat algorithm that are used twice, one time to adjust the gains $\beta = diag(\beta_1, \beta_2)$ and the other time to adjust the gains $\xi = diag(\xi_1, \xi_2)$.

Suppose there are two virtual bats that correspond to the two DOFs of a two-DOF SCARA robot; the information about bat i(i = 1, 2) is

$$\langle x_{i,t}, v_{i,t}, f_{ri,t}, A_{i,t}, r_{i,t} \rangle$$

$$(27)$$

where $x_{i,t} = (x_{1,t}, x_{2,t})$ and $v_{i,t} = (v_{1,t}, v_{2,t})$ are the position and velocity of bat *i* at time *t*, respectively, and the frequency $f_{ri,t}$, loudness $A_{i,t}$, and emission rate $r_{i,t}$ are three required parameters.

In the next generation, the velocity is updated as follows:

$$v_{i,t+1} = v_{i,t} + (x_{i,t} - p_t) f_{ri,t}$$
(28)

where p_t is the best position that is found using

$$p_t = \frac{\sum_{i=1}^{2} x_{i,t}}{2}$$
(29)

Equation (28) can be viewed as a combination of the inertia part $v_{i,t}$ and the influence of p_t . The frequency $f_{ri,t}$ is changed according to

$$f_{ri, t} = f_{r\min} + (f_{r\max} - f_{r\min}) \operatorname{rand}_{i,1}$$
(30)

where f_{rmax} and f_{rmin} are the maximum and minimum values of the frequency, respectively, and rand_{*i*,1} is a random number that is generated using a uniform distribution within [0, 1].

To reflect the bat's decision, the position is changed with some randomness. Let rand_{*i*,2} be a random number that is generated with a uniform distribution within [0, 1]; if rand_{*i*,2} < $r_{i,t}$ is satisfied, bat *i* will undergo the following global search pattern:

$$\overline{x}_{i, t+1} = x_{i, t} + v_{i, t+1} \tag{31}$$

Otherwise, the following local search pattern is adopted:

$$\overline{x}_{i,t+1} = p_t + \varepsilon_i \overline{A}_t \tag{32}$$

where ε_i is a random number that is generated using a uniform distribution within [-1, 1] and \overline{A}_t is the average loudness of all bats:

$$\bar{A}_{t} = \frac{\sum_{i=1}^{2} A_{i,t}}{2}$$
(33)

After $\overline{x}_{i, t+1} = (\overline{x}_{1i, t}, \overline{x}_{2i, t})$ is obtained with Equations (31) and (32), then $x_{i, t+1}$ will update its position as follows:

$$x_{i, t+1} = \begin{cases} \overline{x}_{i, t+1} & (\text{rand}_{i,3} < A_{i, t}) \text{ and } (g(\overline{x}_{i, t+1}) < g(x_{i, t})) \\ x_{i, t} & \text{otherwise} \end{cases}$$
(34)

Equation (34) implies that the position is updated only when the following two conditions are met: (a) it finds the better position, i.e., $g(\bar{x}_{i,t+1}) < g(x_{i,t})$, (b) with a probability rand_{*i*,3} < $A_{i,t}$, where rand_{*i*,3} is a random number that is generated using a uniform distribution within [0, 1]. If the position of the bat is updated, the corresponding loudness $A_{i,t+1}$ and emission rate $r_{i,t+1}$ are also changed as follows:

$$A_{i, t+1} = \alpha A_{i, t} \tag{35}$$

$$r_{i, t+1} = r_{i, 1} \left(1 - e^{-\gamma t} \right) \tag{36}$$

where $\alpha > 0$ and $\gamma > 0$ are two predefined parameters and $A_{i, 1}$ and $r_{i, 1}$ are the initial values for the loudness and emission rate, respectively.

The following four additional steps are included in the Bat algorithm to obtain the values of the functions $g(\bar{x}_{i,t+1})$ and $g(x_{i,t})$ for $\beta = diag(\beta_1, \beta_2)$ and $\xi = diag(\xi_1, \xi_2)$.

For the gains $\beta = diag(\beta_1, \beta_2)$, these gains are equal to the positions of the Bat algorithm:

$$\begin{array}{l}
x_{1, t} = \beta_{1, t} \\
x_{2, t} = \beta_{2, t}
\end{array}$$
(37)

 $g(\overline{x}_{i, t+1})$ and $g(x_{i, t})$ are updated as follows:

$$\begin{bmatrix} u_{\beta1, t+1} \\ u_{\beta2, t+1} \end{bmatrix} = \begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \begin{bmatrix} \xi_{1, 1} \operatorname{sgn}(s) + \overline{x}_{1, t+1} (\dot{q}_{1d} - \dot{q}_{1}) + \ddot{q}_{1d} \\ \xi_{2, 1} \operatorname{sgn}(s) + \overline{x}_{2, t+1} (\dot{q}_{2d} - \dot{q}_{2}) + \ddot{q}_{2d} \end{bmatrix} \\ + \begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_{1} \\ \dot{q}_{2} \end{bmatrix} + \begin{bmatrix} f_{1}(\dot{q}) \\ f_{2}(\dot{q}) \end{bmatrix},$$
(38)
$$g(\overline{x}_{1, t+1}) = u_{\beta1, t+1}^{2} \\ g(\overline{x}_{2, t+1}) = u_{\beta2, t+1}^{2} \end{bmatrix}$$

$$\begin{bmatrix} u_{\beta 1, t} \\ u_{\beta 2, t} \end{bmatrix} = \begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \begin{bmatrix} \xi_{1, 1} \operatorname{sgn}(s) + x_{1, t} (\dot{q}_{1d} - \dot{q}_{1}) + \ddot{q}_{1d} \\ \xi_{2, 1} \operatorname{sgn}(s) + x_{2, t} (\dot{q}_{2d} - \dot{q}_{2}) + \ddot{q}_{2d} \end{bmatrix} + \begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_{1} \\ \dot{q}_{2} \end{bmatrix} + \begin{bmatrix} f_{1}(\dot{q}) \\ f_{2}(\dot{q}) \end{bmatrix},$$

$$g(x_{1, t}) = u_{\beta 1, t}^{2} \\ g(x_{2, t}) = u_{\beta 2, t}^{2}$$
(39)

where $\xi_{1,1}$ and $\xi_{2,1}$ are the initial conditions of $\xi_{1,t}$ and $\xi_{2,t}$, respectively.

For the gains $\xi = diag(\xi_1, \xi_2)$, these gains are equal to the positions of the Bat algorithm:

$$\begin{aligned} x_{1, t} &= \xi_{1, t} \\ x_{2, t} &= \xi_{2, t} \end{aligned} (40)$$

 $g(\overline{x}_{i, t+1})$ and $g(x_{i, t})$ are updated as follows:

$$\begin{bmatrix} u_{\xi_{1},t+1} \\ u_{\xi_{2},t+1} \end{bmatrix} = \begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \begin{bmatrix} \overline{x}_{1,t+1}\operatorname{sgn}(s) + \beta_{1,1}(\dot{q}_{1d} - \dot{q}_{1}) + \ddot{q}_{1d} \\ \overline{x}_{2,t+1}\operatorname{sgn}(s) + \beta_{2,1}(\dot{q}_{2d} - \dot{q}_{2}) + \ddot{q}_{2d} \end{bmatrix} \\ + \begin{bmatrix} C_{11}(q,\dot{q}) & C_{12}(q,\dot{q}) \\ C_{21}(q,\dot{q}) & C_{22}(q,\dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_{1} \\ \dot{q}_{2} \end{bmatrix} + \begin{bmatrix} f_{1}(\dot{q}) \\ f_{2}(\dot{q}) \end{bmatrix},$$
(41)
$$g(\overline{x}_{1,t+1}) = u_{\xi_{1,t+1}}^{2} \\ g(\overline{x}_{2,t+1}) = u_{\xi_{2,t+1}}^{2} \\ \begin{bmatrix} u_{\xi_{1,t}} \\ u_{\xi_{2,t}} \end{bmatrix} = \begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \begin{bmatrix} x_{1,t}\operatorname{sgn}(s) + \beta_{1,1}(\dot{q}_{1d} - \dot{q}_{1}) + \ddot{q}_{1d} \\ x_{2,t}\operatorname{sgn}(s) + \beta_{2,1}(\dot{q}_{2d} - \dot{q}_{2}) + \ddot{q}_{2d} \end{bmatrix} \\ + \begin{bmatrix} C_{11}(q,\dot{q}) & C_{12}(q,\dot{q}) \\ C_{21}(q,\dot{q}) & C_{22}(q,\dot{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_{1} \\ \dot{q}_{2} \end{bmatrix} + \begin{bmatrix} f_{1}(\dot{q}) \\ f_{2}(\dot{q}) \end{bmatrix},$$
(42)
$$g(x_{1,t}) = u_{\xi_{1,t}}^{2} \\ g(x_{2,t}) = u_{\xi_{2,t}}^{2} \end{bmatrix}$$

where $\beta_{1, 1}$ and $\beta_{2, 1}$ are the initial conditions of $\beta_{1, t}$ and $\beta_{2, t}$, respectively.

The following two additional steps that are included in the Bat algorithm maintain the bounds of the values for $\beta = diag(\beta_1, \beta_2)$ and $\xi = diag(\xi_1, \xi_2)$.

After $x_{i, t+1} = (x_{1, t}, x_{2, t})$ is obtained with Equation (34) and using Equation (37) for the gains $\beta = diag(\beta_1, \beta_2), x_{i, t+1}$ will update its position as follows:

$$x_{i, t+1} = \begin{cases} x_{i, t} & (\beta_{i, t} < \beta_{i\min}) \text{ or } (\beta_{i, t} > \beta_{i\max}) \\ x_{i, t+1} & otherwise \end{cases}$$
(43)

where β_{imax} and β_{imin} are the maximum and minimum values that $\beta_{i, t}$ can take.

After $x_{i, t+1} = (x_{1, t}, x_{2, t})$ is obtained with Equation (34) and using Equation (40) for the gains $\xi = diag(\xi_1, \xi_2), x_{i, t+1}$ will update its position as follows:

$$x_{i, t+1} = \begin{cases} x_{i, t} & (\xi_{i, t} < \xi_{i\min}) \text{ or } (\xi_{i, t} > \xi_{i\max}) \\ x_{i, t+1} & otherwise \end{cases}$$
(44)

where ξ_{imax} and ξ_{imin} are the maximum and minimum values that $\xi_{i, t}$ can take.

The steps of the Bat algorithm that are used to obtain the best values of the gains $\beta = diag(\beta_1, \beta_2)$ are described by Equations (27)–(39), and (43).

The steps of the Bat algorithm that are used to obtain the best values of the gains $\xi = diag(\xi_1, \xi_2)$ are described by Equations (27)–(36), (40)–(42), and (44).

The pseudocode of the Bat algorithmis as follows:

- 1. Generate the bat population $x_{i, 1}$ and initial velocity $v_{i, 1}$, i = (1, 2).
- 2. Define the pulse frequency $f_{ri, 1}$ in $x_{i, 1}$.
- 3. Initialize the values for the pulse rate $r_{i,1}$ and loudness $A_{i,1}$.
- 4. While (*t* < maximum iteration number):
- 5. Update frequency $(f_{ri, t+1})$ and velocity $(v_{i, t+1})$ according to Equations (28) and (30), respectively.
- 6. If $(rand_{i,2} < r_{i,t})$:
- 7. Update position ($\overline{x}_{i, t+1}$) according to Equation (31).
- 8. Otherwise, update position ($\overline{x}_{i, t+1}$) according to Equations (29), (32), and (33).
- 9. End If.
- 10. Determine the target functions $g(\bar{x}_{i, t+1})$ and $g(x_{i, t})$ for $\beta = diag(\beta_1, \beta_2)$ with Equations (37)–(39) or determine the target functions $g(\bar{x}_{i, t+1})$ and $g(x_{i, t})$ for $\xi = diag(\xi_1, \xi_2)$ with Equations (40)–(42).
- 11. If $((\operatorname{rand}_{i,3} < A_{i,t}) \text{ and } (g(\overline{x}_{i,t+1}) < g(x_{i,t})))$:
- 12. Accept the new result $x_{i, t+1}$ with Equation (34).
- Decrease A_{i, t+1} and increase r_{i, t+1} according to Equations (35) and (36), respectively.
 End If
- 15. If $((\beta_{i,t} < \beta_{i\min}) \text{ or } (\beta_{i,t} > \beta_{i\max}))$ or If $((\xi_{i,t} < \xi_{i\min}) \text{ or } (\xi_{i,t} > \xi_{i\max}))$:
- 16. Accept the past value of $x_{i, t+1}$ with Equation (43) or (44).
- 17. End If

The objective function g_0 that is used in the Bat algorithm to achieve this objective is given by

$$\min g_0 = \min(u_1^2 + u_2^2)$$

subject to
$$\beta_{\min} \le \beta_t \le \beta_{\max}$$

$$\xi_{\min} \le \xi_t \le \xi_{\max}$$
(45)

As a comparison [37,38], the objective function g_0 that is considered in the Simplex algorithm to achieve this objective is given by

$$\min g_{0} = \min(u_{1} + u_{2})$$

subject to
$$\beta_{\min} \leq \beta_{t} \leq \beta_{\max}$$

$$\xi_{\min} \leq \xi_{t} \leq \xi_{\max}$$
(46)

Remark 2. The proposed technique is applied to the SCARA robot with two DOFs; however, this technique could be applied to other more complicated robots, such as the Schoenflies-Motion Generator with four DOFs [39,40] or parallel robots with three DOFs by using the following steps: (1) the dynamic model of the SCARA robot (Equation (6)) would be changed by the dynamic model of the other robot, (2) the changed dynamic model would be used in the control design of Equations (18)–(26) to obtain a different SMC, and (3) the different SMC would be used in the Bat design of Equations (38)–(42).

5. Simulations

In order to demonstrate the effectiveness of the proposed controller, a position-tracking problem of the two-DOF SCARA robot that is described Equations (3)–(6) was considered. The purpose of the controllers was to save energy by minimizing the forces that are required to move the two-DOF SCARA robot. Please note that in all the simulations, the sat function was considered. This implies that the reaching law (Equation (12)) used the sat function instead of the sign function.

Under this scenario, the initial positions of articulations 1 and 2 were 0 and 0, respectively. The reference trajectory was proposed as a periodic form, as is shown in the next equation:

$$q_d = \begin{bmatrix} q_{1d} \\ q_{2d} \end{bmatrix} = \begin{bmatrix} 0.5\sin(1.25t) \\ 0.5\sin(1.25t) \end{bmatrix}$$
(47)

The physical restrictions should be considered at this point to select the β and ξ since they are critical for the speed and stability of system response, i.e., $u \le u_{\text{max}}$. The reference trajectory (Equation (47)) was evaluated because it was evaluated by many previous studies and other reference trajectories that are different from Equation (47) could be evaluated with similar results.

5.1. Optimization of a Sliding Mode Controller for a Two-DOF SCARA Robot Using the Simplex Algorithm

As a comparison, the SMC with the Simplex algorithm of [37,38] is considered. In order to save energy by minimizing the forces in the proposed controller, optimization was carried out to find the values of the gains $\beta = diag(\beta_1, \beta_2)$ and $\xi = diag(\xi_1, \xi_2)$ using the Simplex algorithm. The optimization process for the proposed sliding mode controller was carried out offline. According to the description of the Simplex algorithm, the optimization started with the initial values of the gains as $\beta_1 = diag(10, 10)$ and $\xi_1 = diag(10, 10)$ and the parameters described in Table 1, which were evaluated using the SMC that is described by Equations (21) and (22). Hence, if the objective function was able to save energy by minimizing the forces with the Simplex algorithm that is described by Equation (46) for the gains $\beta = diag(\beta_1, \beta_2) = diag(x_{b1}, x_{b2})$ and with the Simplex algorithm that is described by Equation (46) for the gains $\xi = diag(\xi_1, \xi_2) = diag(x_{p1}, x_{p2})$, then the best values were selected and stored. After using the Simplex algorithm, the best values of the gains $\beta = diag(\beta_1, \beta_2) = diag(x_{b1}, x_{b2})$ and $\xi = diag(\xi_1, \xi_2) = diag(x_{p1}, x_{p2})$ were used in the SMC that is described by Equations (21) and (22) for the position-tracking problem of the two-DOF SCARA robot that is described by Equations (3)–(6).

Parameter	Symbol	Value
Minimum gain value	β_{\min}	0
Maximum gain value	β_{max}	100
Minimum gain value	ξ _{min}	0
Maximum gain value	ξmax	100

Table 1. Simplex algorithm parameters.

5.2. Optimization of a Sliding Mode Controller for a Two-DOF SCARA Robot Using the Bat Algorithm

The SMC with the Bat algorithm of this study is shown in Figure 3. In order to save energy by minimizing the forces in the proposed controller, optimization was carried to find the values of the gains $\beta = diag(\beta_1, \beta_2)$ and $\xi = diag(\xi_1, \xi_2)$ using the Bat algorithm. The optimization process for the proposed sliding mode controller was carried out offline. According to the description of the Bat algorithm, the optimization started with the initial values of the gains as $\beta_1 = diag(10, 10)$ and $\xi_1 = diag(10, 10)$ and the parameters described in Table 2, which were evaluated in the SMC that is described by Equations (21) and (22). Hence, if the objective function was able to save energy by minimizing the forces with the Bat algorithm that is described by Equations (27)–(39), and (43) for the gains $\beta = diag(\beta_1, \beta_2) = diag(x_{b1}, x_{b2})$ and with the Bat algorithm that is described by Equations (27)–(36), (40)–(42), and (44) for the gains $\xi = diag(\xi_1, \xi_2) = diag(x_{p1}, x_{p2})$, then the best values were selected and stored. After using the Bat algorithm, the best values of the gains $\beta = diag(\beta_1, \beta_2) = diag(x_{b1}, x_{b2})$ and $\xi = diag(\xi_1, \xi_2) = diag(x_{p1}, x_{p2})$ were used in the SMC that is described by Equations (21) and (22) for the position-tracking problem of the two-DOF SCARA robot that is described by Equations (3)–(6).

Table 2. Bat algorithm parameters.

Parameter	Symbol	Value
Loudness	α	0.9
Pulse rate	γ	0.9
Minimum frequency	f_{rmin}	0.0
Maximum frequency	f_{rmax}	100
Minimum gain value	β_{\min}	0
Maximum gain value	β_{\max}	100
Minimum gain value	ξ _{min}	0
Maximum gain value	ξmax	100

6. Discussion

The evolution of the gain values of the SMC with the Simplex algorithm is shown in Figure 4, where the end values of the gains were $\beta = diag(x_{b1}, x_{b2}) = diag(42.1275, 26.0464)$ and $\xi = diag(x_{p1}, x_{p2}) = diag(40.0146, 69.9986)$. The gains that were obtained from the optimization process of the Simplex algorithm were implemented in the SMC for a two-DOF SCARA robot. The response regarding the position-tracking problem of the SMC with the Simplex algorithm in Equation (46) for a two-DOF SCARA robot that is described by Equations (3)–(6), which is denoted as Simplex, is shown in Figures 5–7 for the evolution of the forces u_1 and u_2 , the joint positions q_1 and q_2 , and the joint velocities q_{d1} and q_{d2} , respectively.



Figure 4. Evolution of the gain values.



Figure 5. Evolution of the forces.



Figure 6. Evolution of the joint positions.





The evolution of the gain values of the SMC with the Bat algorithm is shown in Figure 4, where the end values of the gains were $\beta = diag(x_{b1}, x_{b2}) = diag(16.5605, 17.4366)$ and $\xi = diag(x_{p1}, x_{p2}) = diag(10.0112, 10.3736)$. The gains that were obtained from the optimization process of the Bat algorithm were implemented in the SMC for a two-DOF SCARA robot. The response regarding the position-tracking problem of the SMC with the Bat algorithm that is described in Equations (21), (22), and (27)–(44) for a two-DOF SCARA robot

that is described in Equations (3)–(6), which was denoted as Bat, is shown in Figures 5–7 for the evolution of the forces u_1 and u_2 , the joint positions q_1 and q_2 , and the joint velocities q_{d1} and q_{d2} , respectively.

The RMSEs of the position errors e_1 and e_2 of Equation (48) and the forces u_1 and u_2 in Equation (48) for the position-tracking problem of the two-DOF SCARA robot that was solved using the SMC with the Simplex algorithm that is described in Equations (21), (22), and (46), which is denoted as Simplex, and by using the SMC with the Bat algorithm in Equations (21), (22), and (27)–(44), which is denoted as Bat, are shown in Table 3.

$$RMSEe_{1} = \sqrt{\int_{1}^{10} e_{1}^{2} dt}$$

$$RMSEe_{2} = \sqrt{\int_{1}^{10} e_{2}^{2} dt}$$

$$RMSEu_{1} = \sqrt{\int_{1}^{10} u_{1}^{2} dt}$$

$$RMSEu_{2} = \sqrt{\int_{1}^{10} u_{2}^{2} dt}$$
(48)

Table 3. Performance of the SMC.

	RMSEu ₁	RMSEu ₂	RMSEe ₁	RMSEe ₂
Simplex	214.2696	85.3328	2.1358×10^{-5}	2.1445×10^{-5}
Bat	203.8097	78.6126	2.0201×10^{-5}	2.0223×10^{-5}

In some cases, the gains for the sliding mode controller were acquired according to the experience of controller designers and simulations. However, the tuning process is sometimes hard and complex due to certain processes, systems, or particular situations. Recent works that are discussed at the beginning of this work presented several techniques to save energy by minimizing the forces; however, they require the tuning of several gains. Thus, other advances proposed adaptation laws as a response to save energy by minimizing the forces in complicated processes with multiple conditions; however, they require that the process or system present motion or load changes throughout their operation since they start from certain initial gains. The magnitudes of the forces depend on a correct selection of sliding mode controller gains. Hence, the methodology that is presented in this paper helps to find the gains using the Bat algorithm, which requires few iterations and a small population, i.e., the required forces are minimized and energy is saved; moreover, it is simple, flexible, and easy to implement. The results show an improvement in the minimization of the forces. Although there were favorable results when using this methodology, it should be applied offline since this requires a minimum number of interactions to optimize. Finally, this feature can be considered as a disadvantage for some systems but an attractive solution for other systems.

7. Conclusions

The design of a sliding mode controller with a reaching law was proposed for a two-DOF SCARA robot. Sliding mode control is a robust control technique thatis used due to its resistance against parametric variation and external disturbances. However, the resulting big forces require high amounts of energy. In order to save energy by minimizing the forces, optimization of the sliding mode controller gains using the Bat algorithm was implemented. The SMC was simulated and the phase plane for the errors and their derivative positions of each link of a two-DOF SCARA robot was presented. In the simulation, the behavior of the forces was foundfor the position-tracking problem. Then, a simulation of the SMC that used the Bat algorithm for optimizationwas designed and implemented. Lastly, the simulation of the proposed controller showed an improvement with regard to saving energy by minimizing the forces in the position-tracking problem for a two-DOF SCARA robot. In the future, differential evolution or genetic algorithms will be compared with the Bat algorithm and experimental results will be shown.

Author Contributions: Investigation and formal analysis, L.A.S., J.d.J.R., E.O., and D.A.C.; software and validation, G.O., R.B., and D.R.C.; writing—original draft, review, and editing, J.A.M.-C., A.Z., and G.J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This article received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors are grateful to the guest editor and reviewers for their valuable comments and insightful suggestions, which helped to significantly improve this article. The authors also thank the Instituto Politécnico Nacional, Secretaría de Investigación y Posgrado, Comisión de Operación y Fomento de Actividades Académicas, and Consejo Nacional de Ciencia y Tecnología for their support.

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

References

- 1. Benbouhenni, H.; Bizon, N. Improved Rotor Flux and Torque Control Based on the Third-Order Sliding Mode Scheme Applied to the Asynchronous Generator for the Single-RotorWind Turbine. *Mathematics* **2021**, *9*, 2297. [CrossRef]
- 2. Solodusha, S.; Bulatov, M. Integral Equations Related to Volterra Series and Inverse Problems: Elements of Theory and Applications in Heat Power Engineering. *Mathematics* **2021**, *9*, 1905. [CrossRef]
- 3. Le, T.D.; Kang, H.-J.; Suh, Y.-S. Chattering-Free Neuro-Sliding Mode Control of 2-DOF Planar Parallel Manipulators. *Int. J. Adv. Robot. Syst.* 2013, *10*, 22. [CrossRef]
- 4. Dal, M.; Teodorescu, R. Sliding Mode Controller Gain Adaptation and Chattering Reduction Techniques for DSP-Based PM DC Motor Drives. *Turk. J. Electr. Eng. Comput. Sci.* **2011**, *19*, 532–549.
- 5. Swathi, K.; Kumar, G.V.N. Design of intelligent controller for reduction of chattering phenomenon in robotic arm: A rapid prototyping. *Comput. Electr. Eng.* **2019**, *74*, 483–497. [CrossRef]
- 6. Ferrara, A.; Incremona, G.P.; Sangiovanni, B. Tracking control via switched Integral Sliding Mode with application to robot manipulators. *Control. Eng. Pr.* **2019**, *90*, 257–266. [CrossRef]
- Kali, Y.; Saad, M.; Benjelloun, K.; Khairallah, C. Super-twisting algorithm with time delay estimation for uncertain robot manipulators. *Nonlinear Dyn.* 2018, 93, 557–569. [CrossRef]
- 8. Jing, C.; Xu, H.; Niu, X. Adaptive sliding mode disturbance rejection control with prescribed performance for robotic manipulators. *ISA Trans.* **2019**, *91*, 41–51. [CrossRef]
- 9. Vijay, M.; Jena, D. PSO based neuro fuzzy sliding mode control for a robot manipulator. J. Electr. Syst. Inf. Technol. 2017, 4, 243–256. [CrossRef]
- Cibiraj, N.; Varatharajan, M. Chattering reduction in sliding mode control of quadcopters using neural networks. *Energy Procedia* 2017, 117, 885–892. [CrossRef]
- 11. Ba, D.X.; Bae, J. A Precise Neural-Disturbance Learning Controller of Constrained Robotic Manipulators. *IEEE Access* **2021**, *9*, 50381–50390.
- Mobayen, S.; Mofid, O.; Din, S.U.; Bartoszewicz, A. Finite-Time Tracking Controller Design of Perturbed Robotic Manipulator Based on Adaptive Second-Order Sliding Mode Control Method. *IEEE Access* 2021, 9, 71159–71169. [CrossRef]
- 13. Song, T.; Fang, L.; Wang, H. Model-free finite-time terminal sliding mode control with a novel adaptive sliding mode observer of uncertain robot systems. *Asian J. Control* 2021, in press. [CrossRef]
- 14. Xu, Z.; Huang, W.; Li, Z.; Hu, L.; Lu, P. Nonlinear Nonsingular Fast Terminal Sliding Mode Control Using Deep Deterministic Policy Gradient. *Appl. Sci.* **2021**, *11*, 4685. [CrossRef]
- González-García, J.; Narcizo-Nuci, N.; García-Valdovinos, L.; Salgado-Jiménez, T.; Gómez-Espinosa, A.; Cuan-Urquizo, E.; Cabello, J. Model-Free High Order Sliding Mode Control with Finite-Time Tracking for Unmanned Underwater Vehicles. *Appl. Sci.* 2021, *11*, 1836. [CrossRef]
- 16. Yang, X.-S. A New Metaheuristic Bat-Inspired Algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010);* Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74.
- 17. Yang, X.-S.; He, X. Bat algorithm: Literature review and applications. Int. J. Bio-Inspired Comput. 2013, 5, 141–149. [CrossRef]

- 18. Chen, S.-Y.; Yang, B.-C.; Pu, T.-A.; Chang, C.-H.; Lin, R.-C. Active Current Sharing of a Parallel DC-DC Converters System Using Bat Algorithm Optimized Two-DOF PID Control. *IEEE Access* 2019, *7*, 84757–84769. [CrossRef]
- 19. Haji, V.H.; Monje, C.A. Fractional-order PID control of a MIMO distillation column process using improved bat algorithm. *Soft Comput.* **2018**, *23*, 8887–8906. [CrossRef]
- 20. 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]
- 21. Maroufi, O.; Choucha, A.; Chaib, L. Hybrid fractional fuzzy PID design for MPPT-pitch control of wind turbine-based bat algorithm. *Electr. Eng.* 2020, *102*, 2149–2160. [CrossRef]
- 22. Pan, Z.; Quynh, N.V.; Ali, Z.; Dadfar, S.; Kashiwagi, T. Enhancement of maximum power point tracking technique based on PV-Battery system using hybrid BAT algorithm and fuzzy controller. *J. Clean. Prod.* **2020**, 274, 123719. [CrossRef]
- 23. Premkumar, K.; Manikandan, B. Bat algorithm optimized fuzzy PD based speed controller for brushless direct current motor. *Eng. Sci. Technol. Int. J.* **2016**, *19*, 818–840. [CrossRef]
- 24. Rahmani, M.; Ghanbari, A.; Ettefagh, M.M. Robust adaptive control of a bio-inspired robot manipulator using bat algorithm. *Expert Syst. Appl.* **2016**, *56*, 164–176. [CrossRef]
- 25. Rahmani, M.; Komijani, H.; Ghanbari, A.; Ettefagh, M.M. Optimal novel super-twisting PID sliding mode control of a MEMS gyroscope based on multi-objective bat algorithm. *Microsyst. Technol.* **2018**, *24*, 2835–2846. [CrossRef]
- 26. Soto, R.; Crawford, B.; Olivares, R.; Niklander, S.; Johnson, F.; Paredes, F.; Olguín, E. Online control of enumeration strategies via bat algorithm and black hole optimization. *Nat. Comput.* **2017**, *16*, 241–257. [CrossRef]
- 27. Talbi, N. Design of Fuzzy Controller rule base using Bat Algorithm. Energy Procedia 2019, 162, 241–250. [CrossRef]
- 28. Rossomando, F.G.; Soria, C.M. Adaptive Neural Sliding Mode Control in Discrete Time for a SCARA robot arm. *IEEE Lat. Am. Trans.* **2016**, *14*, 2556–2564. [CrossRef]
- 29. Kelly, R.; Santibáñez, V.; Loría, A. Control of Robot Manipulators in Joint Space. In *Advanced Textbooks in Control and Signal Processing*; Springer: London, UK, 2005; ISBN 978-1-85233-994-4.
- 30. Hung, J.Y.; Gao, W.; Hung, J.C. Variable structure control: A survey. IEEE Trans. Ind. Electron. 1993, 40, 2–22. [CrossRef]
- 31. Arciuolo, T.F.; Faezipour, M. PID++: A Computationally Lightweight Humanoid Motion Control Algorithm. *Sensors* **2021**, *21*, 456. [CrossRef]
- 32. Li, Y.; Wang, D.; Zhou, S.; Wang, X. Intelligent Parameter Identification for Robot Servo Controller Based on Improved Integration Method. *Sensors* 2021, 21, 4177. [CrossRef]
- 33. Martinez, D.I.; De Rubio, J.J.; Vargas, T.M.; Garcia, V.; Ochoa, G.; Balcazar, R.; Cruz, D.R.; Aguilar, A.; Novoa, J.F.; Aguilar-Ibanez, C. Stabilization of Robots with a Regulator Containing the Sigmoid Mapping. *IEEE Access* **2020**, *8*, 89479–89488. [CrossRef]
- Martinez, D.; Rubio, J.; Garcia, V.; Vargas, T.; Islas, M.; Pacheco, J.; Gutierrez, G.; Meda-Campaña, J.; Mujica-Vargas, D.; Aguilar-Ibañez, C. Transformed Structural Properties Method to Determine the Controllability and Observability of Robots. *Appl. Sci.* 2021, 11, 3082. [CrossRef]
- 35. Berardi, M.; D'Abbicco, M.A. Critical Case for the Spiral Stability for 2 × 2 Discontinuous Systems and an Application to Recursive Neural Networks. *Mediter. J. Math.* **2016**, *13*, 4829–4844. [CrossRef]
- 36. Difonzo, F.V. A note on attractivity for the intersection of two discontinuity manifolds. Opusc. Math. 2020, 40, 685–702. [CrossRef]
- 37. Lee, Y.; Resiga, A.; Yi, S.; Wern, C. The Optimization of Machining Parameters for Milling Operations by Using the Nelder–Mead Simplex Method. *J. Manuf. Mater. Process.* 2020, *4*, 66. [CrossRef]
- 38. Visuthirattanamanee, R.; Sinapiromsaran, K.; Boonperm, A.-A. Self-Regulating Artificial-Free Linear Programming Solver Using a Jump and Simplex Method. *Mathematics* **2020**, *8*, 356. [CrossRef]
- Angeles, J.; Caro, S.; Khan, W.; Morozov, A. Kinetostatic Design of an Innovative Schönflies-Motion Generator. Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci. 2006, 220, 935–943. [CrossRef]
- Di Gregorio, R.; Cattai, M.; Simas, H. Performance-Based Design of the CRS-RRC Schoenflies-Motion Generator. *Robotics* 2018, 7, 55. [CrossRef]