mathematics

MDPI

Article

A Comparative Analysis of Metaheuristic Algorithms for Enhanced Parameter Estimation on Inverted Pendulum System Dynamics

Daniel Sanin-Villa [1], Miguel Angel Rodriguez-Cabal [1], Luis Fernando Grisales-Noreña [2], Mario Ramirez-Neria [3,*] and Juan C. Tejada [4]

[1] Department of Mechatronics and Electromechanics, Instituto Tecnológico Metropolitano, Medellín 050034, Colombia; danielsanin@itm.edu.co (D.S.-V.); miguelrodriguez@itm.edu.co (M.A.R.-C.)
[2] Department of Electrical Engineering, Faculty of Engineering, Universidad de Talca, Curicó 3340000, Chile; luis.grisales@utalca.cl
[3] InIAT Instituto de Investigación Aplicada y Tecnología, Universidad Iberoamericana, Ciudad de México 01219, Mexico
[4] Artificial Intelligence and Robotics Research Group (IAR), Universidad EIA, Envigado 055428, Colombia; juan.tejada@eia.edu.co
* Correspondence: mario.ramirez@ibero.mx

Abstract: This research explores the application of metaheuristic algorithms to refine parameter estimation in dynamic systems, with a focus on the inverted pendulum model. Three optimization techniques, Particle Swarm Optimization (PSO), Continuous Genetic Algorithm (CGA), and Salp Swarm Algorithm (SSA), are introduced to solve this problem. Through a thorough statistical evaluation, the optimal performance of each technique within the dynamic methodology is determined. Furthermore, the efficacy of these algorithms is demonstrated through experimental validation on a real prototype, providing practical insights into their performance. The outcomes of this study contribute to the advancement of control strategies by integrating precisely estimated physical parameters into various control algorithms, including PID controllers, fuzzy logic controllers, and model predictive controllers. Each algorithm ran 1000 times, and the SSA algorithm achieved the best performance, with the most accurate parameter estimation with a minimum error of 0.015 01 N m and a mean solution error of 0.015 06 N m. This precision was further underscored by its lowest standard deviation in RMSE ($1.443\,99 \times 10^{-6}$ N m), indicating remarkable consistency across evaluations. The 95% confidence interval for error corroborated the algorithm's reliability in deriving optimal solutions.

Keywords: parameter estimation; metaheuristic algorithms; dynamic model; inverted pendulum; particle swarm optimization; continuous genetic algorithm; salp swarm algorithm

MSC: 68T20

check for updates

Citation: Sanin-Villa, D.; Rodriguez-Cabal, M.A.; Grisales-Noreña, L.F.; Ramirez-Neria, M.; Tejada, J.C. A Comparative Analysis of Metaheuristic Algorithms for Enhanced Parameter Estimation on Inverted Pendulum System Dynamics. *Mathematics* **2024**, *12*, 1625. https://doi.org/10.3390/math12111625

Academic Editors: Francisco Beltran-Carbajal, Julio Cesar Rosas Caro, Juan M Ramirez and Jonathan C. Mayo-Maldonado

Received: 24 April 2024
Revised: 11 May 2024
Accepted: 15 May 2024
Published: 22 May 2024

1. Introduction

The inverted pendulum symbolizes the equilibrium between stability and instability within control systems. It finds real-world uses in robotics, transportation, and automation industries [1]. However, effectively managing inverted pendulum systems entails gauging their characteristics, a task made difficult by their intricate and nonlinear nature. Conventional approaches to parameter estimation may not always yield results [2].

1.1. Metaheuristic Algorithms

Over the years, these algorithms have emerged to address this problem. Inspired by phenomena and social dynamics, these algorithms offer search methods that cover solution spaces. Through applying optimization principles, metaheuristic algorithms support parameter estimation for managing inverted pendulums, leading to stability, robustness, and effectiveness. Exploring the domain of algorithms for parameter estimation in controlling inverted pendulums requires a combination of control theory, optimization tactics, and computational intelligence [3].

Different kinds of algorithms can be found that are centered on the optimization of mechanical and engineering design problems. Due to the nature of these types of problems, which are nonlinear non-convex mathematical models, the main kinds of algorithms used are the metaheuristic techniques such as the particle swarm optimization (PSO) [4], genetic algorithm (GA) [5], and salp swarm algorithms (SSA) [6], among others. For instance, the work developed by Moraes et al. uses an enhanced PSO algorithm for parameter estimation of water in oil emulsification in a duct. The methodology implemented can obtain good-quality solutions, where the objective function describes multiple variables in a non-convex mathematical model—showing the PSO as an efficient tool [7]. Other works, in which the PSO was used, have been carried out by Liu et al. [4], where the authors present a novel variation of the PSO using the Chebyshev functions to improve the algorithm's efficiency. The PSO was tested with EC2017 and CEC2022 benchmark functions, where the objective is the dynamic parameter estimation of different mathematical problems related to control operations. Additionally, Rodriguez et al. implemented a CGA algorithm to minimize the weight of a drive shaft, solving the mathematical model of the nonlinear nonconvex type [8]. On the other hand, Ab Wahab et al. implemented a CGA algorithm for mobile robot global route planning, where the authors developed a set of constrictions related to blockages in the routes to allow the algorithm to find the optimal path for each route [9].

1.2. Applications and Challenges in Control Systems

Inverted pendulum systems find applications across automotive, aerospace, manufacturing, and robotics sectors. Within the field of robotics, these systems play roles in navigation, manipulation, and surveillance tasks. Utilizing the dynamics of inverted pendulums, mobile robots and humanoid platforms can attain agility, maneuverability, and adaptability when operating in environments [10]. Inverted pendulum control systems are pivotal in maintaining vehicle stability, ensuring safe navigation, and improving overall performance and safety in autonomous vehicles, especially in challenging terrains or adverse weather conditions. Accurate parameter estimation techniques such as mass distribution, moment of inertia, and friction coefficients enable precise control and trajectory tracking [11]. In industrial automation, inverted pendulum systems optimize the operation of conveyor belts, robotic arms, and other automated machinery to improve efficiency, throughput, and reliability in production processes [12]. The aerospace industry also utilizes this type of control system in designing and operating unmanned aerial vehicles (UAVs) and spacecraft, enhancing UAVs' maneuverability, stability, and payload capacity for tasks such as surveillance, reconnaissance, and aerial photography. In rehabilitation robotics, inverted pendulum-based systems assist individuals with mobility impairments or neurological disorders. By providing dynamic balance support and gait assistance, these systems aid in rehabilitation therapy, helping patients regain mobility, improve their motor control, and enhance their overall quality of life. Accurate parameter estimation is crucial for achieving optimal performance and functionality across diverse applications of inverted pendulum systems [13,14].

1.3. Relevance of Torque Prediction in Mechanical Systems

Torque is a fundamental parameter in determining a mechanical system's dynamic behavior and stability. An accurate understanding of torque through dynamic modeling is essential for developing effective control strategies. An inadequate torque prediction can compromise a system's efficiency, reliability, and safety. Therefore, dynamic modeling enables engineers to understand how a system responds to external inputs and disturbances. Accurately predicting the torque on mechanical systems is extremely important in control engineering. This prediction significantly affects the control system's design, operation, and performance in various applications [15]. However, without precise torque predictions, the dynamic model may mischaracterize the system's behavior, leading to ineffective control strategies. Control algorithms such as PID (Proportional–Integral–Derivative) controllers, model predictive controllers, and adaptive control schemes rely heavily on accurate torque predictions [16]. These predictions help generate control actions that stabilize the system and achieve the desired performance objectives. Estimating the torque allows controllers to efficiently manage the system performance, adjusting for disruptions and uncertainties as they occur. This involves reducing energy usage, lessening strain, and optimizing effectiveness [17]. Engineers can optimize control strategies by accurately estimating the torque, ensuring the system operates safely while attaining the desired performance metrics. Where energy-efficient control schemes can be implemented, they prolong the mechanical components' lifespan [18]. In safety-critical applications such as automotive systems, robotics, and industrial machinery, accurate torque prediction is imperative for ensuring the safety and reliability of the controlled system. Overestimation or underestimation of the torque can lead to unexpected system behavior, potentially resulting in equipment failure, operational accidents, or injury to personnel. Control systems can implement appropriate safety measures and ensure reliable operation by accurately predicting the torque under varying conditions. Optimizing performance metrics like speed, accuracy, and precision requires a deep understanding of torque dynamics. Accurate torque prediction equips control systems to achieve optimal performance while adhering to operational constraints and performance specifications [19].

Researchers in control engineering continually strive to enhance the accuracy and efficiency of torque prediction in mechanical systems for control purposes. Advanced modeling techniques and optimization algorithms have emerged as promising avenues for improving torque prediction accuracy and optimizing control strategies. However, existing studies have identified limitations and challenges associated with current approaches, necessitating further exploration and refinement of methodologies. A torque-based nonlinear predictive control approach has been proposed for automotive powertrains, highlighting the significance of dynamic torque control [20]. This approach underscores the importance of accurate torque estimation for optimizing powertrain performance and efficiency. Similarly, sensorless control techniques have been developed for synchronous motor drives to achieve resilient torque control, particularly in the presence of parameter errors [21,22]. However, limitations such as applicability under large parameter errors and accuracy constraints at lower speeds have been noted, underscoring the need for robust torque estimation methods. Nonlinear model predictive control (NMPC) has garnered attention for internal combustion engines, offering enhanced performance through advanced modeling and optimization [23]. Yet, challenges remain, including potential tracking errors induced by predictive model inaccuracies. Similarly, real-time model predictive control (MPC) of drive systems with elastic transmission has shown promise but faces limitations in addressing severe torsional vibrations and load disturbances [24]. Metaheuristic optimization techniques have been explored for tuning proportional–integral controllers in drive systems, demonstrating their potential to improve the control efficiency [25]. However, further investigation is necessary to explore their applicability across different control scenarios and systems. Furthermore, the optimization-based motion prediction of mechanical systems has been investigated, highlighting the benefits of sensitivity analysis for optimizing lifting motion. The ongoing quest to enhance torque prediction accuracy and control efficiency underscores the interdisciplinary nature of control engineering. By addressing the limitations and challenges identified in the current research, there is a way to develop more robust and effective control strategies for various mechanical systems. This study aims to contribute to this topic by investigating novel torque prediction using optimization methodologies for parameter estimations, drawing upon insights from advanced modeling techniques, optimization algorithms, and practical applications in various domains. In the pursuit of advancing torque prediction and control methodologies, several seminal studies have laid the groundwork for current research. For instance, a torque-based nonlinear predictive control approach has been proposed for automotive powertrains, emphasizing the importance of dynamic torque control. Despite its potential, challenges remain, including the risk of tracking errors due to predictive model inaccuracies. Innovative approaches leveraging metaheuristic optimization techniques have been explored for optimizing control strategies in drive systems. While these approaches can improve control efficiency, their applicability across different scenarios and systems warrants further investigation [26]. Integrating advanced modeling techniques, optimization algorithms, and practical applications represents a fertile ground for advancing torque prediction and control strategies. Using these strategies, engineers can enhance performance and efficiency across various mechanical systems [27].

1.4. Contributions and Research Objective

The findings of this study have implications for advancing control systems engineering by providing solutions to improve the performance and reliability of inverted pendulum-based applications across different fields. Using algorithms, we aim to address the challenges related to parameter estimation in a mathematical model of inverted pendulum systems, for more resilient and adaptable control strategies in real-world settings. This research is based on a range of literature exploring methods and approaches to find the physical parameters to estimate the torque in the mechanism, and three optimization techniques are proposed: the PSO, CGA, and SSA, where a statistical evaluation was carried out to obtain the best evaluation and response of each technique attained to the dynamic methodology.

2. Methodology

This section presents the methodological strategies employed in estimating the model parameters for a mechanism using experimental data. The model parameters were effectively ascertained through the experimental measurements from an actual implementation of the mechanism with a metaheuristic algorithm.

2.1. Inverted Pendulum System

The inverted pendulum system comprises a planar double inverted pendulum. A DC motor drives the initial link, whereas the subsequent link is an underactuated simple pendulum. θ_1 and θ_2 represent the angular positions of the links, u stands for the control torque input, while m_1 and m_2 are the masses of the links. The lengths of the links are denoted by l_1 and l_2, and l_{c_1} and l_{c_2} indicate the distances to the center of masses. Finally, I_1 and I_2 stand for the inertia of the links. The system is shown in Figure 1.

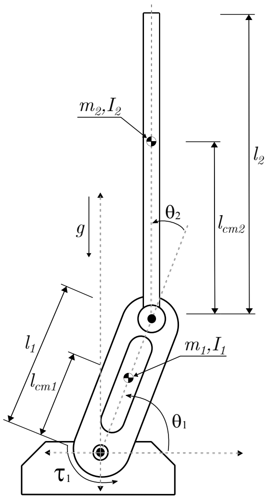

Figure 1. Schematics of the Pendubot system.

Mathematical Model

Underactuated Euler–Lagrange systems of fourth order can be generally represented as [28]

$$P(q(t))u(t) = M(q(t))\ddot{q}(t) + C(q(t), \dot{q}(t))\dot{q}(t) + g(q(t)). \tag{1}$$

Here, $q(t) = \begin{bmatrix} q_1(t) & q_2(t) \end{bmatrix}^{\mathsf{T}}$ denotes the generalized coordinates, $M(q(t)) \in \mathbb{R}^{2\times 2}$ represents the inertia matrix, and $C(q(t), \dot{q}(t)) \in \mathbb{R}^{2\times 2}$ describes the Coriolis and centrifugal forces. The vector $g(q(t)) \in \mathbb{R}^2$ represents the gravitational forces, and $P(q(t)) \in \mathbb{R}^2$ maps the external forces. Additionally, $u(t) \in \mathbb{R}$ signifies the control input.

Such systems, described by (1), can be alternatively expressed in state space as

$$\begin{aligned} \dot{x}_a(t) &= x_b(t) \\ \dot{x}_b(t) &= f(x_a(t), x_b(t)) + \zeta(x_a(t))u(t), \end{aligned} \tag{2}$$

where $x_a(t)$ represents the joint positions vector $\begin{bmatrix} q_1(t) & q_2(t) \end{bmatrix}^{\mathsf{T}}$, $x_b(t)$ denotes the articular velocities vector $\begin{bmatrix} \dot{q}_1(t) & \dot{q}_2(t) \end{bmatrix}^{\mathsf{T}}$, and

$$f(x_a(t), x_b(t)) = -M(q(t))^{-1}[C(q(t), \dot{q}(t))\dot{q}(t) + g(q(t))] \tag{3}$$

$$\zeta(x_a(t)) = M(q(t))^{-1}P(q(t)). \tag{4}$$

Utilizing the Euler–Lagrange formalism, the dynamic model of the system can be represented as

$$M(q(t))\ddot{q}(t) + C(q(t), \dot{q}(t))\dot{q}(t) + g(q(t)) = P\tau(t), \tag{5}$$

where

$$q(t) = \begin{bmatrix} \theta_1(t) & \theta_2(t) \end{bmatrix}^{\mathsf{T}} \tag{6}$$

$$P = \begin{bmatrix} 1 & 0 \end{bmatrix}^{\mathsf{T}} \tag{7}$$

$$M(q) = \begin{bmatrix} \beta_1 + \beta_2 + 2\beta_3 \cos(\theta_2(t)) & M_{1,2}(\theta) \\ \beta_2 + \beta_3 \cos(\theta_2(t)) & \beta_2 \end{bmatrix} \tag{8}$$

$$C(q, \dot{q}) = \begin{bmatrix} -\beta_3\dot{\theta}_1(t)\sin(\theta_2(t)) & \beta_3 C_{1,2}(\theta) \\ \beta_3\dot{\theta}_1(t)\sin(\theta_2(t)) & 0 \end{bmatrix} \tag{9}$$

$$g(q) = \begin{bmatrix} \beta_4 g\cos(\theta_1(t)) + \beta_5 g\cos(\theta_1(t) + \theta_2(t)) \\ \beta_5 g\cos(\theta_1(t) + \theta_2(t)) \end{bmatrix}, \tag{10}$$

where

$$\beta_1 = m_1 l_{c_1}^2 + m_2 l_{c_2}^2 + I_1 \tag{11}$$

$$\beta_2 = m_2 l_{c_2}^2 + I_2 \tag{12}$$

$$\beta_3 = m_2 l_1 l_{c_2} \tag{13}$$

$$\beta_4 = m_1 l_{c_1} + m_1 l_1 \tag{14}$$

$$\beta_5 = m_2 l_{c_2}. \tag{15}$$

2.2. Experimental Setup

The experimental pendulous prototype is displayed in Figure 2. It has a DC motor model NC5475, manufactured by NISCA, Tokio, Japan; to drive the first link. The angular positions of both links are monitored with incremental encoders of 10,000 counts per revolution, consistent with the earlier example. Power amplification is facilitated by the amplifier model VoltPAQ-X2, manufactured by Quanser, Markham, Canada. The control strategy is implemented in the Matlab–Simulink platform, with a sampling time of 0.001 s.

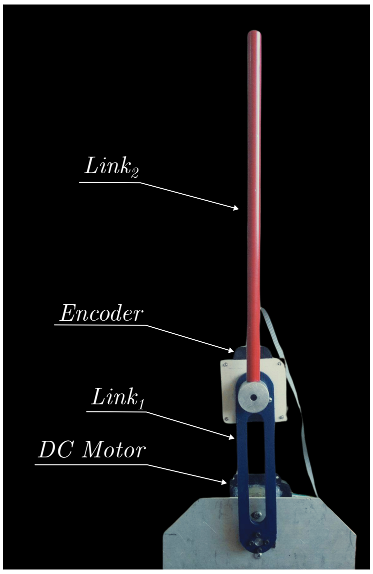

Figure 2. Experimental pendulum prototype.

The actual physical parameters of the prototype are outlined below and were measured before the assembly:

- Link inertias: $I_1 = 0.00053$ [kg·m^2] and $I_2 = 0.00077$ [kg·m^2].
- Masses of the links: $m_1 = 0.210$ [kg] and $m_2 = 0.1$ [kg].
- Lengths of the links: $l_1 = 0.15$ [m] and $l_2 = 0.3$ [m].
- Distances to the center of mass: $l_{c_1} = 0.12$ [m] and $l_{c_2} = 0.15$ [m].
- Armature resistance: $k_\tau = 0.0724$ [Ω].
- Torque constant: $\tau_m = 2.983$ [Ω].

The provided values do not encompass any joints, glue, screws, or any accessories that are not part of the model. To refine the model, it is necessary to compute the equivalent values of mass, center of mass location, and inertia for each link. To assess the dynamic model, a trajectory is suggested with the initial conditions set as $x_1(0) = \frac{\pi}{2}$ [rad] and $x_3(0) = 0$ [rad]. The trajectory involves a seamless transition from rest to rest.

$$
\begin{aligned}
y_f^*(t) &= \frac{\beta_2 + \beta_3}{\beta_2} x_{\delta 1}^*(t) + x_{\delta 3}^*(t) \\
y_f^*(t) &= \frac{\beta_2 + \beta_3}{\beta_2} (\theta_1^*(t) - \frac{\pi}{2}) + \theta_2^*(t)
\end{aligned} \tag{16}
$$

Initially, at $t = 0$, the trajectory starts from $y_f^*(0) = 0$, setting $\theta_1^*(t) = \frac{\pi}{2}$ and $\theta_2^*(t) = 0$. At $t = 4.5$, it moves to $y_f^*(6) = \frac{\beta_2+\beta_3}{\beta_2}\left(\frac{\pi}{6}\right) - \left(\frac{\pi}{6}\right)$ with $\theta_1^*(t) = \frac{3}{2}\pi$ and $\theta_2^*(t) = -\frac{\pi}{6}$ over a duration of 2.5 seconds. Subsequently, at $t = 9.5$ [s], it transitions to $y_f^*(13.5) = -\frac{\beta_2+\beta_3}{\beta_2}\left(\frac{\pi}{6}\right) + \left(\frac{\pi}{6}\right)$ with $\theta_1^*(t) = \frac{\pi}{3}$ and $\theta_2^*(t) = \frac{\pi}{6}$ within 4 s. Finally, at $t = 17.5$ [s], it returns to the initial position until the test concludes. Figure 3 illustrates the rest-to-rest positions of the pendulum.

The velocity and angular acceleration for each link will be estimated from the position values derived from the smooth trajectory. These kinematic features will inform the evaluation of the prototype's dynamic behavior, facilitating the acquisition of the experimental torque data necessary to proceed with the parametric estimation.

The experimental torque is estimated based on the motor's angular velocity ($\dot{\phi(t)}$) and the supply voltage ($V(t)$), expressed as

$$
\tau = \frac{k_1}{R_a} V(t) - \frac{k_1 k_2}{R_a} \dot{\phi}. \tag{17}
$$

Here, k_1 represents the torque constant, k_2 denotes the back-EMF constant, and R_a stands for the winding resistance.

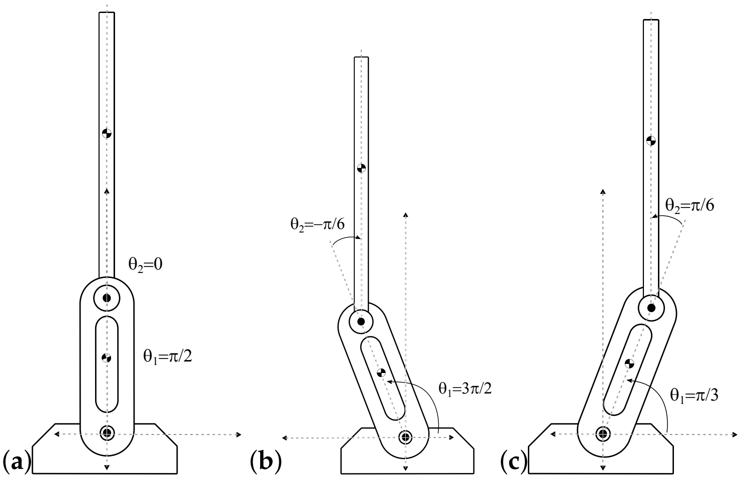

Figure 3. Inverted pendulum trajectory. (**a**) $t = 0$ s. (**b**) $t = 7$ s. (**c**) $t = 13.5$ s.

2.3. Algorithm Implementation

To solve the optimization problem addressed, it is necessary to determine the limits of the optimization parameters, which define the solution space to the algorithms. These upper and lower limits ensure the algorithm produces accurate and relevant results. By defining these limits, the algorithm can effectively narrow down the range of potential solutions, resulting in faster and more efficient problem solving. The data presented in Table 1 provide the upper and lower limits for the model parameters to be estimated, with consideration of the physical constraints. These limits were based on the estimated actual parameters and were determined by ensuring that the center of mass distance was below the length of the link.

Table 1. Upper and lower limits for the parameters to be estimated on the model.

Parameter	m_1 $[kg]$	lc_1 $[m]$	I_1 $[kgm^2]$	m_2 $[kg]$	lc_2 $[m]$	I_2 $[kgm^2]$
Upper	0.4	0.15	0.001	0.4	0.3	0.001
Lower	0.001	0.001	0.0001	0.001	0.1	0.0001

To realize the optimization problem, the root mean square error is used as a fitness function (Equation (18)); this calculates the square root over the average of the squared differences between the predicted values of the torque and the actual (experimental) values. This allows evaluation of how closely the model's predictions match the real-world data [29], where the main objective is to minimize this function, which means that the mathematical model with the estimated parameters matches the experimental values. To quantify the results obtained by the algorithms, the uncertainty associated with the mean solution errors is evaluated, where the confidence interval of 95% is calculated for each algorithm's mean error across the 1000 simulation runs. The confidence interval provides a range in which the true mean error is expected with a 95% probability.

The mean solution is calculated by a set of RMSE measurements (Equation (18)) for each algorithm. n is the number of observations, $\hat{\tau}_i$ is the predicted torque value for the i-th observation, and τ_i is the experimental value for the i-th observation.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{\tau}_i - \tau_i)^2} \tag{18}$$

The RMSE is a useful fitness function for assessing the effectiveness of a developed model with the estimated parameters. A lower RMSE implies that the model's predictions are closer to the actual values, which is what we desire. Conversely, a higher RMSE indicates significant discrepancies between the predictions and actual values, resulting in poor parameter estimation on the model performance [30]. Therefore, a lower RMSE denotes more accurate predictions, reflecting a better parameter estimation.

The standard deviation s, is calculated with Equation (19):

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(RMSE_i - R\bar{M}SE)^2}. \tag{19}$$

The standard error of the mean (SEM) is then given by Equation (20):

$$SEM = \frac{s}{\sqrt{n}}. \tag{20}$$

We use the standard error and the Z-score corresponding to the desired confidence level to calculate the 95% confidence interval for the mean error. The Z-score is a statistical measure that indicates how many standard deviations an element is away from the mean of a distribution. For a two-tailed test at a 95% confidence level, the Z-score is approximately 1.96.

Thus, the confidence interval (CI) is calculated as follows:

$$CI = R\bar{M}SE \pm (Z \cdot SEM). \tag{21}$$

In this study, we applied this methodology to compute the confidence intervals for the mean error obtained after applying the three proposed algorithms: SSA, CGA, and PSO. The computed intervals provide insight into the precision of the mean error estimates and allow us to ascertain the statistical significance of the differences observed between the algorithms' performances in determining the parameters. The algorithms that determine the parameters of the model are presented in Table 1. Their objective is to minimize the RMSE, finding the specific parameters that close the gap between the predictive torque and the experimental measurement, as shown in Equation (18). Each metaheuristic algorithm proposed in this work is described below. Each algorithm setting was optimized using a stochastic PSO algorithm to obtain the best parameters and improve the performance for solving the optimization problem.

2.4. Particle Swarm Optimization

The PSO is a metaheuristic technique for optimizing non-convex nonlinear mathematical models. It is a bio-inspired algorithm that simulates the way flocks of birds sweep a terrain in search of food, where each animal is modeled as a particle. Each particle in the swarm has a cognitive and social component, allowing the swarm to follow the best particle that finds the best solution in the current iteration. Adding a velocity that changes in each iteration depending on its position allows the exploration of the entire solution space [4,31]. Algorithm 1 shows the process to solve an optimization problem. The RMSE is often employed as a fitness function in optimization scenarios to reduce the disparity between the predicted and observed values. The selection of the RMSE as the fitness metric does not directly impact how the optimization techniques perform; rather, it consistently gauges the error these algorithms strive to diminish. For instance, the PSO algorithm utilizing the RMSE offers a continuous error gauge, steering the particles toward the possible solution. The algorithm fine-tunes particles' positions to minimize the RMSE. This choice does not inherently influence how well these optimization techniques function; instead, they leverage the RMSE to direct their quest for a solution. The efficacy of each technique in minimizing the RMSE depends on its mechanisms and parameter configurations.

Algorithm 1: Particle swarm optimization procedure.

```
Read Parameters of the PSO, mathematical model, experimental data
for $iter = 1 : iter_{\max}$ do
    if $iter == 1$ then
        Generate the particle swarm;
        Evaluate the mathematical model Equations (1) to (17);
        Evaluate the fitness function Equation (18);
        Select the minimal solutions and its positions;
    else
        Update the velocities;
        Update particle swarm positions;
        Evaluate the mathematical model;
        Evaluate the fitness function (18);
        Update the best solutions and its positions;
        if $iter == iter_{max}$ then
            Break
        else
Result: Best solution found.
```

2.5. Continuous Genetic Algorithm

The genetic algorithm is a classic optimization technique widely used to solve continuous problems with nonlinear mathematical models. The algorithm starts by creating an initial population of individuals, each evaluated based on the objective function and a set of constraints representing the problem's physics [5]. These enable the evaluation of the response viability, allowing the algorithm to consider non-feasible points as potential solutions. This facilitates better exploration and exploitation of the solution space, preventing the algorithm from getting stuck in local optima. Subsequently, descendant populations are generated using recombination, selection, and mutation techniques. The objective function and set of restrictions are evaluated in each iteration, allowing for advancement through the solution space to find an adequate and viable solution to the analyzed problem [8]. Algorithm 2 briefly details the CGA process to solve an optimization problem.

Algorithm 2: Continuous genetic algorithm procedure.

```
Read Parameters of the CGA, mathematical model, and experimental data
for $iter = 1 : iter_{\max}$ do
    if $iter == 1$ then
        Generate the initial population;
        for $i = 1 : a$ do
            Evaluate the mathematical model Equations (1) to (17);
            Evaluate the fitness function Equation (18);
            Determine the best solution;
    else
        Generate the descending population;
        for $i = 1 : a$ do
            Evaluate the mathematical model Equations (1) to (17);
            Evaluate the fitness function Equation (18);
            Update the best solution;
        Determine the new population;
        if $iter == iter_{\max})$ then
            Break
Result: Best solution found.
```

2.6. Salp Swarm Algorithm

The salp swarm algorithm is a bio-inspired metaheuristic technique with key factors to avoid being trapped in local optima, allowing the exploration and exploitation of the solution space [6,30]. This algorithm is based on swarm intelligence, which simulates the behavior of salps in the ocean, which move in nature in the form of a chain, where the first individual is taken as the leader particle, and the rest of the salps are the followers, where in each iteration, the search is carried out for the area in which the food is found, which represents the optimal zone; then the movement of the leader particle towards the zone is carried out, solving the mathematical model of the various points where the followers are located, to update the positions of the leader particle, where the objective is that all the particles approach the area of the best food, finding the best solution to the problem [32]. Algorithm 3 briefly details the SSA process to solve an optimization problem.

Algorithm 3: Salp swarm algorithm procedure.

```
Read Parameters of the SSA, mathematical model, and experimental data
for $iter = 1 : iter_{\max}$ do
    Determine the movement of the salp chain
    if $iter == 1$ then
        Generate the initial population;
        Evaluate the mathematical model Equations (1) to (17);
        Evaluate the fitness function Equation (18);
        Select the best solution as the salp chain leader;
        Update the position of the leader in the solution space;
    else
        Determine the movement of the salp chain;
        Evaluate the mathematical model Equations (1) to (17);
        Evaluate the fitness function Equation (18);
        Select the best solution as the salp chain leader;
        Update the position of the leader in the solution space;
        if $iter == iter_{max}$ then
            Break
        else
            —
Result: Best solution found.
```

3. Results

Figure 4 presents the experimental torque results correlated with the position, velocity, and acceleration data outlined in Section 2.2.

To provide a clear and direct comparison of the results on the parameter estimation for the mechanism, Figure 5 shows the plots of the torque predicted using the mathematical model with the parameters found by the different algorithms. It also compares the experimentally measured torque with the theoretical model alongside the predictions made by the SSA, CGA, and PSO algorithms. A zoomed view box gives a detailed comparison for a specific time interval between 5 s and 6.6 s, allowing for closer scrutiny of the algorithms' performance in relation to the experimental data.

Table 2 shows the parameter estimation found by each algorithm; those values correspond to the minimum RMSE over the 1000 runs.

The proposed mathematical model was validated by comparing it with the experimental torques. The overlapping lines show that the model closely aligns with the experimental observations, which serves as a benchmark for the algorithms. The estimated torques of the SSA, CGA, and PSO algorithms appear as distinct lines. The SSA estimation adheres most closely to the experimental and theoretical benchmarks, indicating that the SSA's parameter optimization can accurately calculate the system dynamics over the observed time. The

CGA and PSO estimations generally follow the same pattern; however, their divergence from the experimental and theoretical lines in different points indicates less precision in the model fitting.

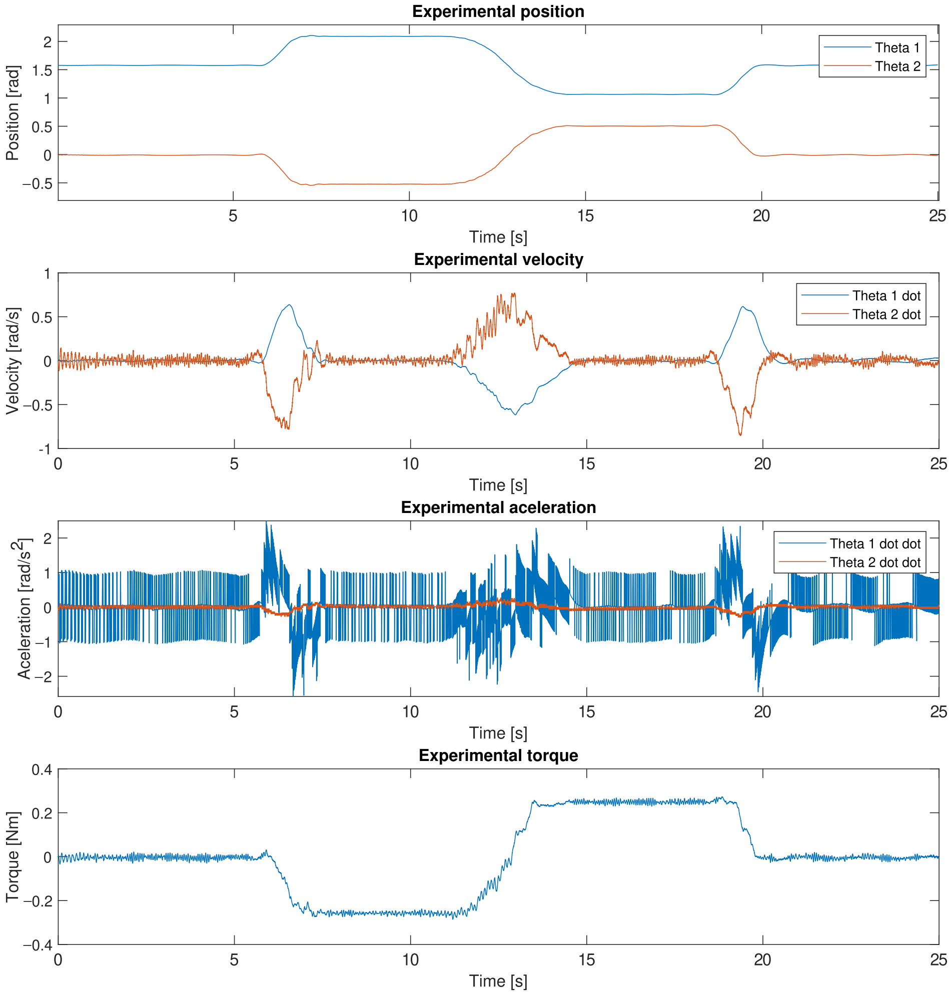

Figure 4. Experimental results for inverted pendulum.

Table 2. Comparative results of parameter estimation.

Parameters	**Experimental**	**SSA**	**CGA**	**PSO**
m_1	0.20000	0.20579	0.14958	0.25497
lc_1	0.12000	0.15000	0.01194	0.15000
I_1	0.00053	0.00059	0.00073	0.00100
m_2	0.14000	0.13496	0.20461	0.13557
lc_2	0.15000	0.12001	0.10402	0.10000
I_2	0.00077	0.00051	0.00012	0.00010

The performance of the SSA algorithm is consistent, as indicated by its proximity to the experimental and theoretical lines shown in Figure 5. This aligns with the low mean RMSE and small confidence interval mentioned earlier, which confirms that it is a robust optimization tool. On the other hand, the CGA and PSO lines exhibit greater variance from the expected torque values, particularly in areas where sharp changes in torque occur. This behavior could be attributed to these algorithms' exploration and exploitation mechanisms, which may not be as well-tuned as the SSA algorithm to capture the intricacies of the modeled system.

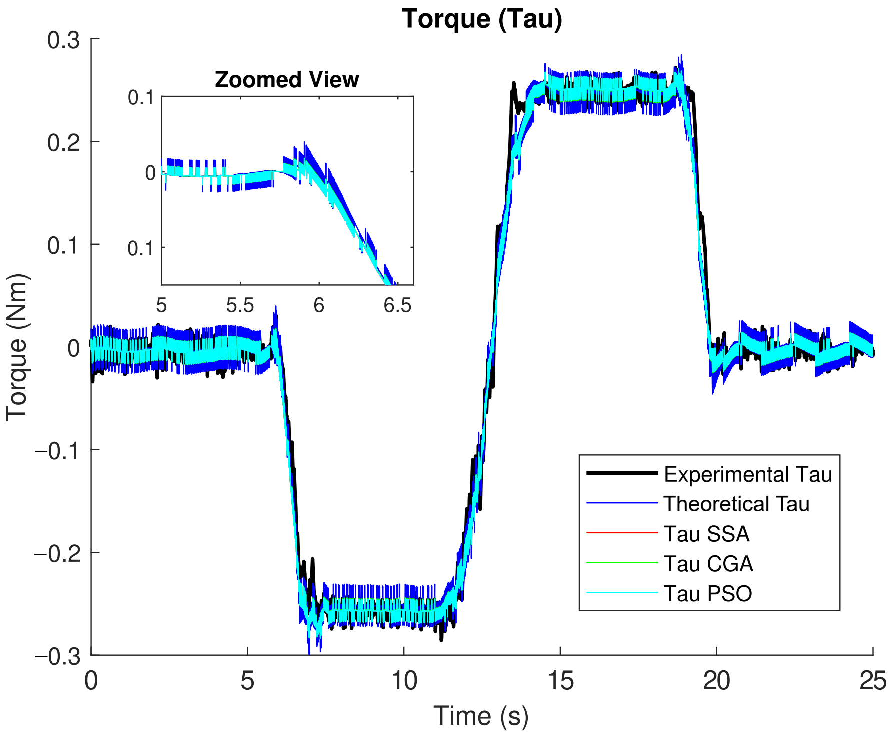

Figure 5. Comparison of average RMSE for algorithms.

A comparative result for the performance of the proposed algorithms, with value numbers, is presented in Tables 3 and 4. The SSA algorithm achieved the best parameter estimation for the mechanism with a minimum error at 0.015 015 147 N m compared to the experimental results. This algorithm achieved a mean solution error of 0.015 066 207 N m, the lowest among the tested algorithms. The standard deviation RMSE for SSA was also the lowest, $1.443\,99 \times 10^{-6}$ N m, indicating highly consistent results across the 1000 runs. The 95% confidence interval for the error was narrow, further confirming the reliability of the SSA in finding the optimal solutions. This algorithm demonstrated a moderate mean computing time of 916.5614 s, with a standard deviation of 18.7228 s and a closed confidence interval that suggests consistent performance in terms of time efficiency.

The CGA also offered interesting results, with the best solution RMSE recorded at 0.015 039 607 N m. However, the error of the mean solution was slightly higher at 0.015 490 795 N m, with the standard deviation of error being more considerable, implying less consistency in obtaining the best-fit parameters compared to the SSA. The error's confidence interval was wider than the SSA's, implying a larger CGA performance variation. The algorithm also had a higher mean computing time of 954.1851 s, with an identical standard deviation to the SSA (18.7228 s). The PSO, although the fastest with a mean computing time of 888.4771 s and a relatively low standard deviation of 17.4958 s, had a higher mean RMSE solution, the largest among the three algorithms, and the confidence interval for the mean RMSE was notably broad. These values indicate a significant variation in PSO's results' accuracy, making it less reliable for parameter estimation despite its computational speed.

Table 3. Comparative results of metaheuristic algorithms for parameter estimation—error metrics.

Algorithms	Best Solution Error	Mean Solutions Error	Standard Deviation of Error	Confidence Interval for Error
SSA	0.015015147	0.015066207	1.44399×10^{-6}	[0.01506, 0.01506]
CGA	0.015039607	0.015490795	7.34505×10^{-6}	[0.01548, 0.01549]
PSO	0.015045076	0.016549584	1.47122×10^{-5}	[0.01654, 0.01655]

Table 4. Comparative results of metaheuristic algorithms for parameter estimation—time metrics.

Algorithms	Mean Time	Standard Deviation of Time	Confidence Interval for Time
SSA	916.5614	18.7228	[912.89167, 920.23103]
CGA	954.1851	18.7228	[950.51546, 957.85481]
PSO	888.4771	17.4958	[885.04796, 891.90631]

The bar chart in Figure 6 provides a clear visual comparison of the average computing time required by the three metaheuristic algorithms. The PSO demanded the least amount of computing time on average, indicating a more time-efficient approach in parameter estimation, with the average time being noticeably less than that of the other two algorithms. The confidence limits in the bars suggest consideration of the variability in computing time, indicating the reliability and consistency of each algorithm's performance over the 1000 runs. While the PSO led in terms of time efficiency, the SSA algorithm followed as a close second, displaying a marginally higher average computing time. According to the analysis, the average CGA algorithm took the longest time to compute, which may not be ideal when time is critical. Moreover, the analysis indicates an inverse relationship between the precision of the algorithms discussed in detail in Table 4. This implies that there may be a trade-off situation where the CGA algorithm, despite its longer computation time, may not necessarily produce better solutions, as demonstrated by its mean solution error compared to the SSA and PSO algorithms.

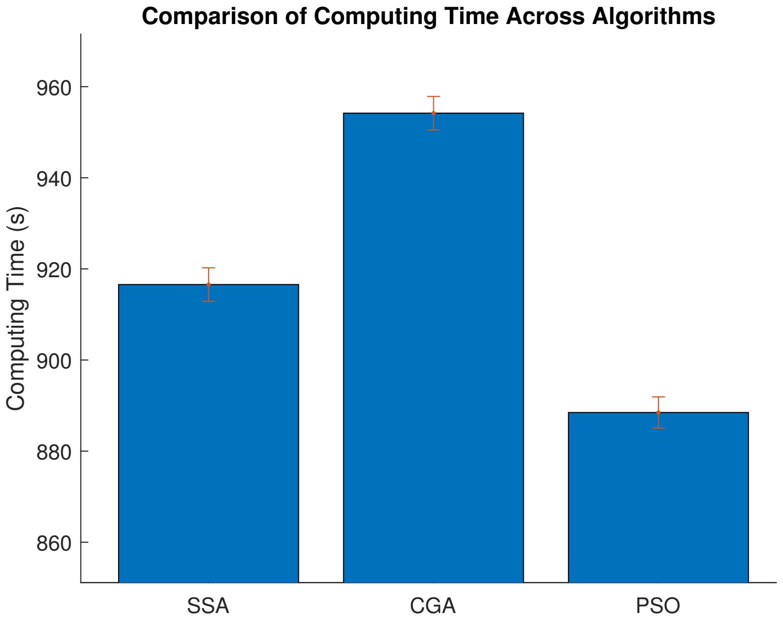

Figure 6. Comparison of average computing times for algorithms.

Figure 7 presents a comparative analysis of the average RMSE for the applied metaheuristic algorithms. The bar chart reveals that the SSA algorithm has the lowest average RMSE, indicating the highest level of precision in fitting the experimental measurements to the mathematical model; it also has a comparatively lower height of the SSA confidence interval limits with a confidence of 95 %. These error bars indicate variability in the accuracy of the algorithms from one run to another, with the SSA showing the least variability, followed by the CGA and PSO. The CGA algorithm, while surpassing the performance of the PSO, exhibits a higher average RMSE than the SSA, as visualized by the taller height of its corresponding bar. This suggests the CGA is less precise than the SSA but more so than the PSO. The error bars for the CGA and PSO are noticeably longer, particularly for the PSO, which signals a greater spread in the RMSE values and, consequently, less consistent performance. The PSO algorithm's bar, which is the tallest and the longest error bar, indicates that it has the poorest response for this problem.

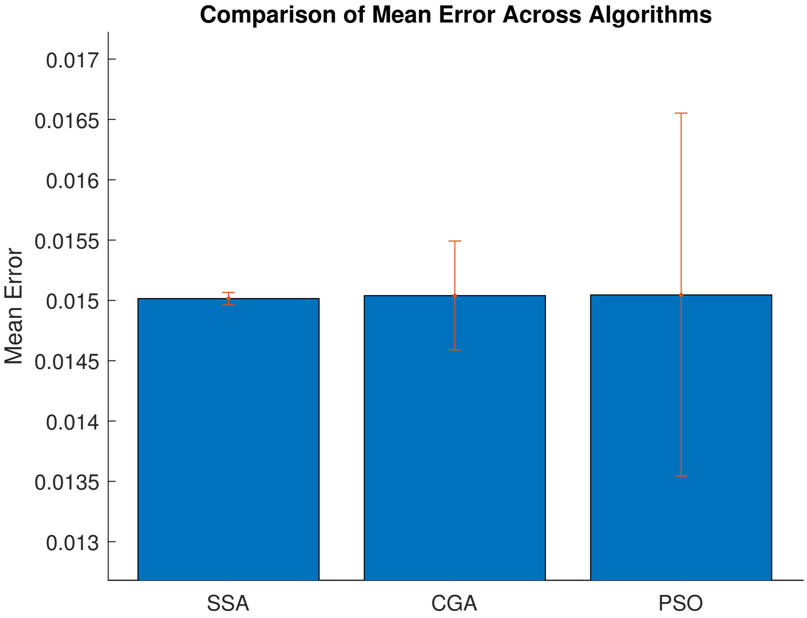

Figure 7. Comparison of average RMSE for algorithms.

4. Conclusions

A parameter estimation via dynamic modeling of an inverted pendulum system was carried out, where the comparison between the mathematical model and the experimental measurements of the torque express a correlation that shows the accuracy of the methodology described.

To solve the mathematical model for the parameter estimation, three algorithms were implemented, where the best solution was achieved by the SSA, which was 0.163% better than the CGA and 0.2% than the PSO. Regarding the repeatability of the solution, the SSA also presented the best standard deviation, reflecting a better quality of solution compared with the CGA and the PSO. Finally, in terms of the computing time, the PSO algorithm took 7.013% less time to obtain the response compared with the SSA and CGA. The SSA was selected as the best algorithm, since, for the problem addressed, the quality of the response has more influence.

This research introduces an innovative methodology and empirically validates dynamic modeling for parameter estimation in inverted pendulum systems. By integrating metaheuristic algorithm techniques and a comparative analysis of their performance, this study improves the precision of the modeling of complex dynamic systems. In future work, it is proposed that these parametric estimation methods be used to determine gains and control parameters of various linear and nonlinear techniques to control dynamic systems like mobile robots, drones and other robotic systems. Furthermore, a comparison of the effectiveness of metaheuristic methodologies with other algorithms to obtain nonlinear models will be studied, such as in MPC.

Author Contributions: The authors contributions to the achievement of the research are as follows: Conceptualization, D.S.-V.; methodology, L.F.G.-N.; software, M.A.R.-C. and J.C.T.; validation, M.R.-N., J.C.T., and M.A.R.-C.; formal analysis, D.S.-V.; investigation, D.S.-V., M.A.R.-C., and J.C.T.; resources, M.R.-N.; data curation, J.C.T.; writing—original draft preparation, J.C.T. and M.A.R.-C.; writing—review and editing, D.S.-V. and L.F.G.-N.; visualization, D.S.-V.; supervision, M.A.R.-C.; project administration, D.S.-V.; funding acquisition, M.R.-N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by InIAT Instituto de Investigación Aplicada y Tecnología Universidad Iberoamericana Ciudad de México under Project DINVP-051 and Universidad EIA under Project INV-CO-049-2023.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, M. Ramirez-Neria (mario.ramirez@ibero.mx), upon reasonable request.

Acknowledgments: Thanks to Instituto Tecnológico Metropolitano, Universidad de Talca, InIAT Instituto de Investigación Aplicada y Tecnología-Universidad Iberoamericana and IAR Research Group-Universidad EIA.

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

References

1. Nayak, A.; Banavar, R.N. Almost-global tracking of the unactuated joint in a pendubot. *IFAC Pap. OnLine* **2018**, *51*, 137–142. 6th IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control LHMNC 2018. [CrossRef]
2. Yan, Z.; Lai, X.; Meng, Q.; Wu, M.; She, J.; Iwasaki, M. Modeling, analysis, and adaptive neural modified-backstepping control of an uncertain horizontal pendubot with double flexible joints. *Control Eng. Pract.* **2023**, *139*, 105647. [CrossRef]
3. Anh, H.P.H.; Son, N.N.; Van Kien, C.; Ho-Huu, V. Parameter identification using adaptive differential evolution algorithm applied to robust control of uncertain nonlinear systems. *Appl. Soft Comput.* **2018**, *71*, 672–684. [CrossRef]
4. Liu, H.; Wang, W.; Cheng, X.; Zheng, H. Particle swarm optimization with Chebychev functional-link network model for engineering design problems. *Appl. Soft Comput.* **2022**, *131*, 109819. [CrossRef]
5. Milenković, B.; Jovanović, D.; Krstić, M. Application of Particle Swarm Optimization for Classical Engineering Problems. *IJEEC Int. J. Electr. Eng. Comput.* **2020**, *5*, 42–49. [CrossRef]
6. Medjahed, S.A.; Ouali, M. A new hybrid SSA-TA: Salp Swarm Algorithm with threshold accepting for band selection in hyperspectral images. *Appl. Soft Comput.* **2020**, *95*, 106534. [CrossRef]
7. Moraes, A.O.; Mitre, J.F.; Lage, P.L.; Secchi, A.R. A robust parallel algorithm of the particle swarm optimization method for large dimensional engineering problems. *Appl. Math. Model.* **2015**, *39*, 4223–4241. [CrossRef]
8. Rodriguez-Cabal, M.A.; Grisales-Norea, L.F.; Maŕn, J.A.; Montoya, O.D. Optimal design of transmission shafts: A continuous genetic algorithm approach. *Stat. Optim. Inf. Comput.* **2019**, *7*, 802–815. [CrossRef]
9. Ab Wahab, M.N.; Nazir, A.; Khalil, A.; Ho, W.J.; Akbar, M.F.; Noor, M.H.M.; Mohamed, A.S.A. Improved genetic algorithm for mobile robot path planning in static environments. *Expert Syst. Appl.* **2024**, *249*, 123762. [CrossRef]
10. Turrisi, G.; Carlos, B.B.; Cefalo, M.; Modugno, V.; Lanari, L.; Oriolo, G. Enforcing Constraints over Learned Policies via Nonlinear MPC: Application to the Pendubot. *IFAC Pap. OnLine* **2020**, *53*, 9502–9507. 21st IFAC World Congress. [CrossRef]
11. Cychowski, M.; Szabat, K. Model predictive speed control with optimal torque constraints handling of drive systems with elastic transmission. In Proceedings of the 2009 IEEE International Electric Machines and Drives Conference, Miami, FL, USA, 3–6 May 2009; pp. 251–258. [CrossRef]
12. Khalil, W.; Boyer, F. An efficient calculation of computed torque control of flexible manipulators. In Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, 21–27 May 1995; IEEE: New York, NY, USA; pp. 609–614. [CrossRef]
13. Du, C.; Yan, F.; Yan, Y.B.; Yang, P.L. Methods of engine torque estimation for control algorithms. *Trans. Csice* **2008**, *26*, 446–451.
14. Leuer, M.; Bocker, J. Self-optimizing Model Predictive Direct Torque Control for electrical drives. In Proccedings of the 2015 IEEE 24th International Symposium on Industrial Electronics (ISIE), Buzios, Brazil, 3–5 June 2015; Volume 6, pp. 1046–1051. [CrossRef]
15. Miranda, H.; Cortes, P.; Yuz, J.I.; Rodriguez, J. Predictive Torque Control of Induction Machines Based on State-Space Models. *IEEE Trans. Ind. Electron.* **2009**, *56*, 1916–1924. [CrossRef]
16. Bilal, A.; Waheed, A.; Shah, M.H. A Comparative Study of Machine Learning Algorithms for Controlling Torque of Permanent Magnet Synchronous Motors through a Closed Loop System. In Proccedings of the 2019 Second International Conference on Latest Trends in Electrical Engineering and Computing Technologies (INTELLECT), Karachi, Pakistan, 13–14 November 2019; pp. 1–6. [CrossRef]
17. Novak, M.; Xie, H.; Dragicevic, T.; Wang, F.; Rodriguez, J.; Blaabjerg, F. Optimal Cost Function Parameter Design in Predictive Torque Control (PTC) Using Artificial Neural Networks (ANN). *IEEE Trans. Ind. Electron.* **2021**, *68*, 7309–7319. [CrossRef]
18. Xiang, M.S.Y.; Mutschler, S.; Brix, D.I.N.; Brach, C.; Geimer, M. Optimization of Hydrostatic-Mechanical Transmission Control Strategy by Means of Torque Control. In Proccedings of the 12th International Fluid Power Conference, Dresden, Germany, 9–11 March 2020.
19. Chang, S.; Gordon, T.J. Model-based predictive control of vehicle dynamics. *Int. J. Veh. Auton. Syst.* **2007**, *5*, 3. [CrossRef]
20. Atabay, O.; Ötkür, M.; Ereke, İ.M. Model based predictive engine torque control for improved drivability. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2018**, *232*, 1654–1666. [CrossRef]
21. Varatharajan, A.; Pellegrino, G.; Armando, E.; Hinkkanen, M. Sensorless Control of Synchronous Motor Drives: Accurate Torque Estimation and Control Under Parameter Errors. *IEEE J. Emerg. Sel. Top. Power Electron.* **2021**, *9*, 5367–5376. [CrossRef]

22. Hajian, M.; Soltani, J.; Markadeh, G.; Hosseinnia, S. Adaptive Nonlinear Direct Torque Control of Sensorless IM Drives with Efficiency Optimization. *IEEE Trans. Ind. Electron.* **2010**, *57*, 975–985. [CrossRef]
23. Kang, M.; Shen, T. Nonlinear model predictive torque control for IC engines. In Proceedings of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 29 June–4 July 2014; pp. 804–809. [CrossRef]
24. Szabat, K. Efficient real-time model predictive control of the drive system with elastic transmission. *IET Control Theory Appl.* **2010**, *4*, 37–49.
25. Costa, B.L.G.; Graciola, C.L.; Angélico, B.A.; Goedtel, A.; Castoldi, M.F. Metaheuristics optimization applied to PI controllers tuning of a DTC-SVM drive for three-phase induction motors. *Appl. Soft Comput.* **2018**, *62*, 776–788. [CrossRef]
26. Elaziz, M.A.; Elsheikh, A.H.; Oliva, D.; Abualigah, L.; Lu, S.; Ewees, A.A. Advanced Metaheuristic Techniques for Mechanical Design Problems: Review. *Arch. Comput. Methods Eng.* **2022**, *29*, 695–716. [CrossRef]
27. Oladipo, S.; Sun, Y.; Wang, Z. Optimization of PID Controller with Metaheuristic Algorithms for DC Motor Drives: Review. *Int. Rev. Electr. Eng. (IREE)* **2020**, *15*, 352. [CrossRef]
28. Ochoa-Ortega, G.; Villafuerte-Segura, R.; Luviano-Juárez, A.; Ramírez-Neria, M.; Lozada-Castillo, N. Cascade Delayed Controller Design for a Class of Underactuated Systems. *Complexity* **2020**, *2020*, 2160743. [CrossRef]
29. Sanin-Villa, D.; Montoya, O.D.; Grisales-Noreña, L.F. Material property characterization and parameter estimation of thermoelectric generator by using a master–slave strategy based on metaheuristics techniques. *Mathematics* **2023**, *11*, 1326. [CrossRef]
30. Sanin-Villa, D.; Montoya, O.D.; Gil-González, W.; Grisales-Noreña, L.F.; Perea-Moreno, A.J. Parameter Estimation of a Thermoelectric Generator by Using Salps Search Algorithm. *Energies* **2023**, *16*, 4304. [CrossRef]
31. Grisales-Noreña, L.; Cortés-Caicedo, B.; Montoya, O.D.; Sanin-Villa, D.; Gil-González, W. Integration of BESS in grid connected networks for reducing the power losses and CO2 emissions: A parallel master-stage methodology based on PDVSA and PSO. *J. Energy Storage* **2024**, *87*, 111355. [CrossRef]
32. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.