

Article

A New Method for Identifying Kinetic Parameters of Industrial Robots

Bin Kou ^{1,2} , Shijie Guo ^{1,2,*} and Dongcheng Ren ^{1,2}

¹ Academy for Engineering and Technology, Fudan University, Shanghai 200433, China; 18110860041@fudan.edu.cn (B.K.); 18110860025@fudan.edu.cn (D.R.)

² Guanghua Lingang Engineering Application and Technology R & D (Shanghai) Co., Ltd., Shanghai 201306, China

* Correspondence: guoshijie@fudan.edu.cn

Abstract: Identifying the kinetic parameters of an industrial robot is the basis for designing a controller for it. To solve the problems of the poor accuracy and easy premature convergence of common bionic algorithms for identifying the dynamic parameters of such robots, this study proposed simulated annealing with similar exponential changes based on the beetle swarm optimization (SEDSABSO) algorithm. Expressions for the dynamics of the industrial robot were first obtained through the SymPyBotics toolkit in Python, and the required trajectories of excitation were then designed to identify its dynamic parameters. Following this, the search pattern of the global optimal solution for the beetle swarm optimization algorithm was improved in the context of solving for these parameters. The global convergence of the algorithm was improved by improving the iterative form of the number N of skinks in it by considering random perturbations and the simulated annealing algorithm, whereas its accuracy of convergence was improved through the class exponential change model. The improved beetle swarm optimization algorithm was used to identify the kinetic parameters of the Zhichang Kawasaki RS010N industrial robot. The results of experiments showed that the proposed algorithm was fast and highly accurate in identifying the kinetic parameters of the industrial robot.

Keywords: industrial robot; kinetic parameter identification; beetle swarm optimization algorithm; stochastic perturbation



Citation: Kou, B.; Guo, S.; Ren, D. A New Method for Identifying Kinetic Parameters of Industrial Robots.

Actuators **2022**, *11*, 2. <https://doi.org/10.3390/act11010002>

Academic Editors: Marco Carricato and Edoardo Ida

Received: 29 November 2021

Accepted: 21 December 2021

Published: 23 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/>).

1. Introduction

Kinetic parameters are the main factor influencing the control of fast and highly precise movements of industrial robots [1]. The process of identifying of their kinetic parameters is usually divided into a number of steps, such as kinetic modeling, designing the excitation trajectory, data acquisition, and identifying and verifying the kinetic parameters [2]. Gautier et al. used a two degree-of-freedom (DOF) robot as the object of study and applied the extended Kalman filter and the least-squares method to identify its parameters. Memar used the SCHUNK Powerball LWA 4P as an experimental object, constructed a dynamics model for it, and implemented the least-squares method to identify the dynamic parameters of the industrial robot [3]. However, the least-squares method is susceptible to measurement noise that lowers its accuracy [4]. Fu et al. [5] used the particle swarm optimization (PSO) algorithm with least squares to identify the kinetic parameters of a seven-DOF collaborative robot in Xinsong, but PSO can easily fall into the local optimum owing to poor population diversity in the late stage of processing that reduces the accuracy of identification of the parameters. Ding et al. [6] identified the dynamics of the robot by using the genetic algorithm, but the process of coding of the algorithm is cumbersome.

Summarizing the existing research, it is found that the identification of robot dynamics parameters is a high-dimensional function problem. Common algorithms are either the PSO accuracy is not high enough, or the genetic algorithm coding is more complicated, so a simple programming, strong anti-interference ability, and convergence accuracy are needed. High algorithm. BSO is a bionic algorithm recently proposed, which mainly uses the principles of PSO and the Beetle Antenna Search (BAS) algorithm, so it has the

advantage of simple programming [7–9]. This article proposes SEDSABSO on the basis of BSO. SEDSABSO combined random perturbation-based behavior and the simulated annealing (SA) algorithm with the global optimal solution of the BSO [10,11] to improve its ability to search for global optimal particles as well as the manner of changes in N particles through an exponential decay model. This improved the convergence of BSO without affecting its computational complexity. Therefore, using the SEDSABSO algorithm for robot dynamic parameter identification will have the advantages of simple programming, high convergence accuracy, and fast iteration speed.

The rest of the paper is organized as follows. Section 2 describes the D-H and the dynamic parameters of the RS010N robot and then linearized them through the SymPyBotics toolkit in Python. In Section 3, the genetic algorithm toolbox in MATLAB was used to design the excitation trajectories required to identify the dynamic parameters of the robot. Section 4 describes the principle of the SEDSABSO algorithm and its process for the identification of robot dynamics parameters. In Section 5, the minimum set of parameters for the dynamics of the RS010N robot was identified and was used to compare the performance of the SEDSABSO, BSO, and LDWPSO algorithms. The results verified the accuracy and effectiveness of the proposed algorithm [12].

2. Robotic arm Dynamics Model

2.1. RS010N Industrial Robots

In this paper, we used the Zhichang Kawasaki RS010N robot as the research object. Figure 1 shows its structural configuration, and Figure 2 shows the configuration of its DH coordinates. The RS010N robot is a typical six-DOF industrial robot. Because the kinetic parameters of its three rear joints are much smaller than those of its three front joints, the former have a smaller influence on the accuracy of control of the robot's motion. Thus, we considered only the first three joints of the RS010N industrial robot here [13].

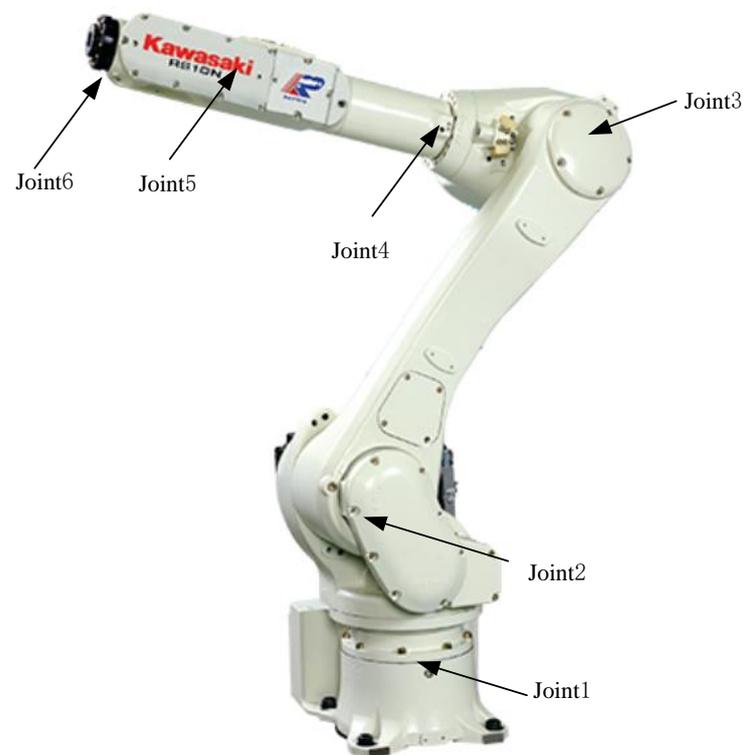


Figure 1. RS010N robot.

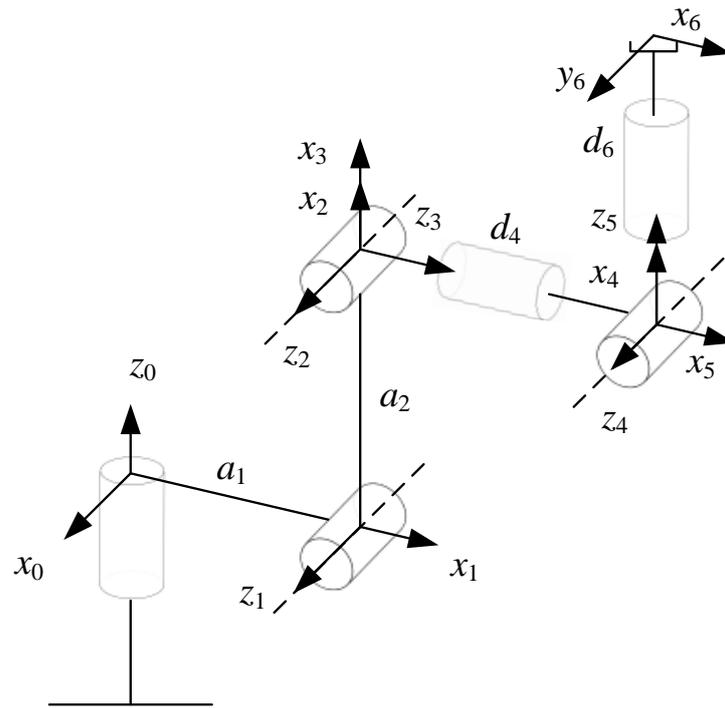


Figure 2. RS010N Robot DH coordinate system.

2.2. Dynamics Modeling

For an n -link robotic arm, the dynamics as modeled by the Newton-Euler method can be described as

$$\tau = D(q) \cdot \ddot{q} + C(q, \dot{q}) + G(q) \tag{1}$$

In Equation (1), τ denotes the driving moment, q denotes the vector of the position of the joints, and \dot{q} and \ddot{q} denote vectors of the velocity and acceleration of the joints, respectively. $D(q)$ denotes the inertial matrix, $C(q, \dot{q})$ denotes the Koch and centrifugal force terms, respectively, and $G(q)$ denotes the gravitational force term [14].

According to the improved Newtonian–Eulerian dynamics, the above equation can be transformed into

$$\tau = \Phi(q, \dot{q}, \ddot{q})P \tag{2}$$

where Φ is the observation matrix with a size of $n * 12n$, and P denotes the vector of the inertial parameters of the robot arm.

2.3. Minimum Set of Parameters for the Dynamical Model

The parameters of the first three joint linkages of the RS010N six-freedom robot are shown in Table 1 below. $a_0 = 350$ mm, $a_1 = 1035$ mm.

Table 1. RS010N connecting rod parameters.

i	α_{i-1}/rad	a_{i-1}/mm	d_i/mm	θ_i/rad
1	$\text{pi}/2$	a_0	0	θ_1
2	$-\text{pi}/2$	a_1	0	θ_2
3	0	0	0	θ_3

The kinetic parameters are shown in Table 2.

Table 2. Actual kinetic parameters.

Member Number	1	2	3
$I_{xx}/(\text{kg}\cdot\text{m}^2)$	36.33	-10.02	7.74
$I_{yy}/(\text{kg}\cdot\text{m}^2)$	40.61	7.05	1.02
$I_{zz}/(\text{kg}\cdot\text{m}^2)$	36.33	11.61	9.81
$I_{xy}/(\text{kg}\cdot\text{m}^2)$	0.00	6.14	3.34
$I_{xz}/(\text{kg}\cdot\text{m}^2)$	0.00	5.93	-3.42
$I_{yz}/(\text{kg}\cdot\text{m}^2)$	0.00	-7.82	-0.29
$m/(\text{kg})$	18.33	25.18	20.47
$x/(\text{mm})$	0.00	122.41	152.34
$y/(\text{mm})$	110.11	163.81	90.24
$z/(\text{mm})$	0.00	73.36	-34.21
$F_c (\text{N}\cdot\text{m})$	0.00	0.00	0.00
$F_v (\text{N}\cdot\text{m})$	0.00	0.00	0.00

We obtained the minimum set of parameters and related expressions for them through the SymPyBotics toolkit in Python.

$$\begin{array}{l}
 \left[\begin{array}{l}
 I_{1xx} \\
 I_{1yy} \\
 I_{1zz} \\
 I_{1xy} \\
 I_{1xz} \\
 I_{1yz} \\
 m_1 \\
 x_1 \\
 y_1 \\
 z_1 \\
 F_{1c} \\
 F_{1v} \\
 I_{2xx} \\
 I_{2yy} \\
 I_{2zz} \\
 I_{2xy} \\
 I_{2xz} \\
 I_{2yz} \\
 m_2 \\
 x_2 \\
 y_2 \\
 z_2 \\
 F_{2c} \\
 F_{2v} \\
 I_{3xx} \\
 I_{3yy} \\
 I_{3zz} \\
 I_{3xy} \\
 I_{3xz} \\
 I_{3yz} \\
 m_3 \\
 x_3 \\
 y_3 \\
 z_3 \\
 F_{3c} \\
 F_{3v}
 \end{array} \right]
 \end{array}
 \xrightarrow{\text{Linearization}}
 \begin{array}{l}
 \left[\begin{array}{l}
 I_{1zz} + I_{2zz} + I_{3zz} + 2 * a_1 * I_{1x} + m_1 * a_1^2 - \\
 (m_2 + m_3) * (a_2^2 - a_1^2) \\
 F_{1c} \\
 F_{1v} \\
 I_{2xx} - I_{2yy} - (m_2 + m_3) * a_2^2 \\
 I_{2xy} \\
 I_{2x} - a^2 * (I_{2z} - I_{3y}) \\
 I_{2yz} \\
 I_{2zz} - (m_2 + m_3) * a_2^2 \\
 I_{2x} + (m_2 + m_3) * a_2 \\
 I_{2y} \\
 F_{2c} \\
 F_{2v} \\
 I_{3xx} - I_{3zz} \\
 I_{3xy} \\
 I_{3xz} \\
 I_{3yy} \\
 I_{3yz} \\
 I_{3x} \\
 I_{3z} \\
 F_{3c} \\
 F_{3v}
 \end{array} \right]
 \end{array}
 \tag{3}$$

According to Equation (3) above, the 36 kinetic parameters of the robot were linearized to 21, and the kinetic model can be written in the following form

$$\tau = \Phi_b(q, \dot{q}, \ddot{q}) p_b \quad (4)$$

where Φ_b is the full-rank observation matrix and P_b denotes the underlying parameter vector of the robotic arm.

3. Incentive Track Design

The commonly used excitation trajectory when identifying the kinetic parameters of industrial robots is the finite-term Fourier series [15,16]. It can be expressed in the following form:

$$\ddot{\theta}_i(t) = -\sum_{n=1}^N a_n n \omega_f \sin(n \omega_f t) - b_n n \omega_f \cos(n \omega_f t) \quad (5)$$

$$\dot{\theta}_i(t) = \sum_{n=1}^N a_n \cos(n \omega_f t) - b_n \sin(n \omega_f t) \quad (6)$$

$$\theta_i(t) = \theta_0 + \sum_{n=1}^N \frac{a_n}{n \omega_f} \sin(n \omega_f t) - \frac{b_n}{n \omega_f} \cos(n \omega_f t) \quad (7)$$

ω_f denotes the fundamental frequency of the Fourier series. Each Fourier series contains a_n , b_n , and θ_0 . N represents the number of harmonic terms of the Fourier series. Because each Fourier series has $2N + 1$ parameters, n represents the number of $1 \dots 2N + 1$.

The constraints on the RS010N robot are shown in Table 3.

Table 3. RS010N robot joint constraints.

Parameter	Joint i	Min	Max
Angle	1	−180	180
	2	−60	140
	3	−180	80
Angle velocity	1	−125	125
	2	−100	100
	3	−165	165
Angle acceleration	1	−45	45
	2	−40	40
	3	−75	75

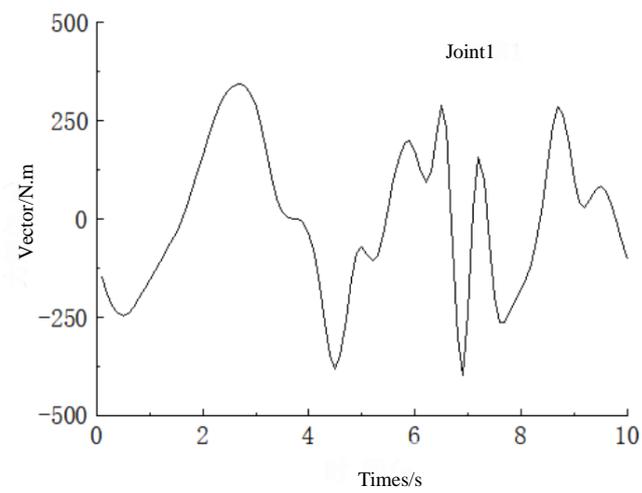
When the robot moved, the observation matrix w was obtained by recording the angle, velocity, and acceleration of the joints of the robot in Φ_b . n denotes the n th sample.

$$w = \begin{bmatrix} \Phi_b(q_1, \dot{q}_1, \ddot{q}_1) \\ \vdots \\ \Phi_b(q_n, \dot{q}_n, \ddot{q}_n) \end{bmatrix} \quad (8)$$

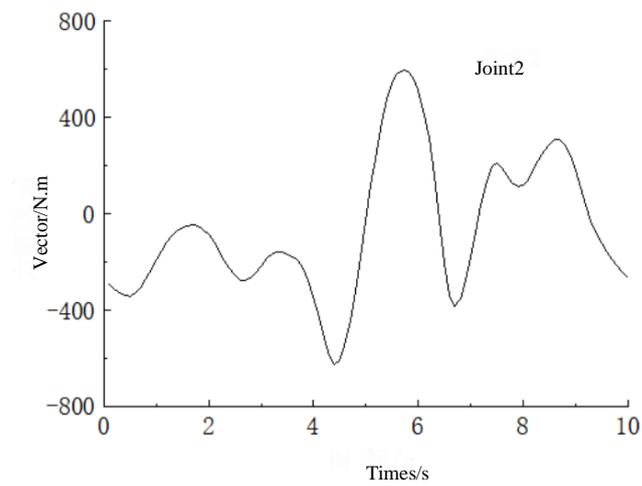
Adaptation function of the incentive trajectory.

$$y = \text{Cond}(w) + P \quad (9)$$

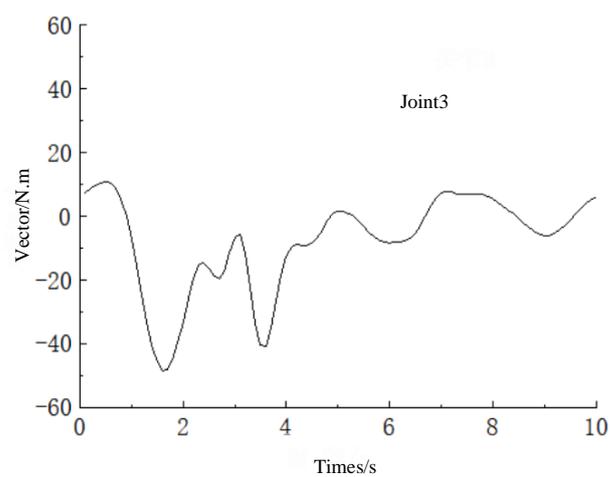
where P denotes a penalty function and Cond denotes the number of conditions of the acquisition matrix. When the trajectory of the industrial robot satisfies the above constraint, $P = 0$, and $P = 10^8$ otherwise. The excitation of the first three joints of the RS010N was obtained by the genetic algorithm toolbox in MATLAB as shown in Figure 3.



(a)



(b)



(c)

Figure 3. The excitation trajectory of the first three joints of the RS010N robot. (a) First joint motivation. (b) Second joint motivation. (c) Third joint motivation.

4. SEDSABSO for the Identification of Industrial Robot Dynamics Parameters

4.1. Improving Global Optimal Search

Owing to the large number of dynamic parameters of the industrial robot that needed to be identified, their solution was a high-dimensional function optimization problem with certain constraints. The BSO used to obtain the kinetic parameter identification of industrial robots easily falls into the premature phenomenon, so the global optimal solution search mode of BSO and the iterative change form of the number N of tennies in the iterative process of BSO are improved in this paper.

BSO

BSO was proposed by Wang et al. based on the PSO and the Aspen whisker algorithm. The position of a body in BSO represents a feasible solution to the optimization problem. The formula to update BSO is:

$$x_{ij}(t+1) = x_{ij}(t) + \lambda v_{ij}(t+1) + (1-\lambda)\xi_{ij}(t) \quad (10)$$

where t denotes the number of iterations of the skink, i denotes the number of skinks in the skink population, j denotes the dimensionality of the problem to be solved, λ denotes $x_{ij}(t+1)$ denotes the position of skink i in the skink population at $t+1$ iterations, $\xi_{ij}(t)$ denotes the position of the i th skink moving autonomously, and $v_{ij}(t+1)$ denotes the velocity of the i th skink. It is expressed as:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_1(p_{ij}(t) - x_{ij}(t)) + c_2r_2(g_{ij}(t) - x_{ij}(t)) \quad (11)$$

w denotes inertial weight, $p_{ij}(t)$ denotes the optimal position of individual i after t iterations of the Aspen population, $x_{ij}(t)$ denotes the given position of i after t iterations, $g_{ij}(t)$ denotes the optimal position of the Aspen population after t iterations, c_1 and c_2 denote adjustment factors, and r_1 and r_2 denote random numbers within the interval $[0,1]$

$$\xi_{ij}(t+1) = \delta(t) * V_{ij}(t) * \text{sign}\left(f\left(x_{lj}(t)\right) - f\left(x_{rj}(t)\right)\right) \quad (1)$$

In Equation (12), $\text{sign}(\cdot)$ denotes the sign function, $\xi_{ij}(t+1)$ denotes the $t+1$ th motion position searched by Aspen autonomously, $\delta(t)$ denotes the step size of the Aspen individual after t iterations, and $f(x_{lj}(t))$ and $f(x_{rj}(t))$ denote the fitness values of the left and right whiskers of the Aspen, respectively.

The positions of the left and right whiskers of the Aspen are indicated by:

$$\begin{aligned} x_{lj}(t+1) &= x_{lj}(t) + V_{ij}(t)d_0/2 \\ x_{rj}(t+1) &= x_{rj}(t) - V_{ij}(t)d_0/2 \end{aligned} \quad (13)$$

$x_{lj}(t+1)$ denotes the position of the left whisker in iteration $t+1$, $x_{rj}(t+1)$ is the $t+1$ right whisker position, and d_0 denotes the distance between the whiskers.

4.2. Improving BSO

4.2.1. Improved Global Optimal Solution

The global optimal particle P_g of BSO is randomly perturbed. Then, the new solution after perturbation is selected by the metropolis criterion of SA to improve the global convergence of BSO.

(1) Simulated annealing algorithm

The simulated annealing algorithm controls the probability of jumping out of the local optimum by setting the temperature. The algorithm evaluates whether to jump out of the worse solution by the metropolis criterion. The procedure is as follows:

- a. The algorithm is initialized, and an initial solution x is randomly generated as the optimal solution.

- b. A new solution x_t is obtained in the vicinity of the initial solution, denoted by $\Delta f = f(x_t) - f(x)$.
- c. The new solution x_t is accepted according to $\min\{1, \exp(-\Delta f/T_k)\} > \text{random}$. T_k denotes the temperature and \exp denotes the exponential function, with natural number e as the base.

(2) Random behavior

Stochastic behavior is a search behavior that can improve the diversity of algorithmic populations and is widely used in a variety of intelligent algorithms, such as in the foraging behavior within the fish swarm algorithm to search for food and the wandering behavior of the wolf pack algorithm to sense the scent of prey in the air in a random pattern [17,18]. Inspired by this, we incorporated stochastic behavior into BSO, and the optimal particles in it were randomly perturbed during each iteration to enhance its ability to escape from the local optimal solution to improve global convergence:

$$P_g^{new} = P_g * (1 + rand) \quad (14)$$

where P_g^{new} denotes the new solution after the random perturbation of the optimal particle of the Aspen swarm, and $rand$ represents a random number from zero to one. The fitness value of the optimal particle p_g of BSO was compared with that of the global optimal particle P_g^{new} for random perturbation; then, the new solution was accepted according to the metropolis criterion of SA. BSO performed q random perturbations of its global optimal solution in each iteration.

4.2.2. Improved Iterative Approach to N Particles of the Aspen Swarm

Each skyline of BSO represents a potential solution to the problem. Assuming that the number of skylines in each iteration of BSO is N , the maximum number of its iterations is $Maxdt$, and the duration of the iteration of a single skyline is t , the total complexity of iterations of BSO is $N * Maxdt * t$ [19]. In solving a high-dimensional function problem, similar to determining the dynamic parameters of the industrial robot, the number of particles N at the moment of each iteration of BSO is constant. BSO is based on PSO, and a larger number of particles in PSO yields a higher accuracy of the convergence, but the time needed for convergence also increases.

For example, in previous work, we proposed a two-stage dynamic PSO that changed the number of particles in the swarm by linearly reducing the number of iterations. The experimental results showed that the accuracy of convergence of the algorithm was similar to that of the classical PSO algorithm, which also reflects a side-effect of the latter. The algorithm proposed here also had this property of a high initial accuracy of convergence [20]. Inspired by this, the number N of skinks in the skink swarm was changed dynamically. The number was larger in the early stage of the skink swarm algorithm and smaller in the later stage, such that the total number of skinks did not differ by much and the gap in the durations of their iterations was not large. We fully exploited the high efficiency of the early iterations of the skink swarm algorithm for the accuracy of its solution. The exponentially decreasing rate of curtailment was used for the number of cows N of the Aspen swarm algorithm by drawing on the idea proposed by Wang et al. [21]. That is, the particle swarm algorithm used a larger number of particles to search in the initial stage. With the increase of the number of iterations, the exponentially decreasing number of particle was used to reduce the Aspen number, so as to improve the search efficiency, and the improved iterative formula for the Aspen number is as follows:

$$N = \text{round}(e^\alpha * N_{start}) \quad (15)$$

$$\alpha = t \frac{Inf_{max} - Inf_{min}}{Maxdt} - Inf_{max} \quad (16)$$

where N_{start} denotes the initial number of skinks of the improved skink herd, i denotes the number of iterations of the skink herd algorithm, $Maxdt$ denotes the maximum number of iterations, and f_{max} and f_{min} are the respective maximum and minimum values of the search factor set used to control the number of skinks N .

4.3. Robot Dynamic Parameter Identification Based on SEDSABSO

The SEDSABSO process is illustrated below:

From Figure 4, the flow of SEDSABSO for industrial robot dynamics parameter identification is:

- (1) Initialization of the algorithm.
- (2) Obtain SEDSABSO individual and group best-fit values.
- (3) Update the position, velocity, and number of skinks N
- (4) Perform a random perturbation search for the global optimal solution and then accept the searched solution with SA's Metropolis criterion and cycle through q searches.
- (5) Compare with the global optimal solution obtained in step 3 after passing q times of search and proceed to the next step of the search by merit.
- (6) Determine whether the algorithm ends, and if the termination condition is not satisfied, return to step 3. If it is satisfied, the global optimal solution is output. In turn, the parameters related to industrial robot dynamics are obtained.

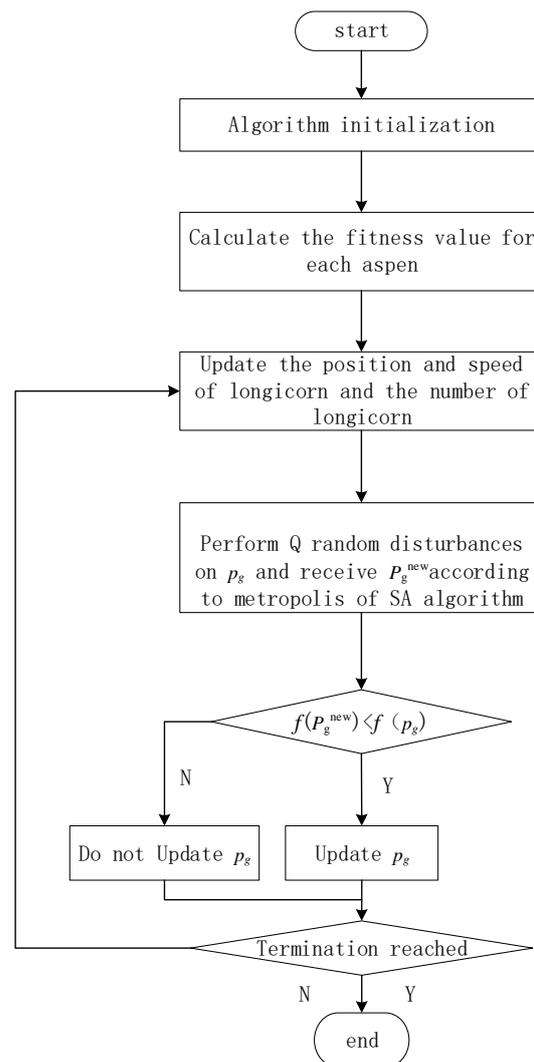


Figure 4. SEDSABSO algorithm flow chart.

5. Simulation Experiments and Results Analysis

5.1. Adaptation Function

The sum of the absolute values of the difference between the torque of joint recognition and the theoretical torque of the RS010N robot when sampled i times was used as the objective function, as in Equation (17)

$$\sum_{i=1}^N abs(f_{k1}(i) - f1(i)) + abs(f_{k2}(i) - f2(i)) + abs(f_{k3}(i) - f3(i)) \quad (17)$$

$i = 1, 2, \dots, N$ denotes the i th sample of the robot, $f_{k1}(i)$, $f_{k2}(i)$, and $f_{k3}(i)$ denote values of the first, second, and third moments of the joints values for the i robot recognition $f1(i)$, $f2(i)$, $f3(i)$ denotes the theoretical value of the three joint moments sampled by the robot.

5.2. Analysis of Experimental Results

Experimental Parameter Setting

To improve the accuracy of identification of the kinetic parameters, the individual algorithms were initialized uniformly by using a small interval [22]. Then, the dynamic parameters of the robot were solved for by using the SEDSABSO algorithm, BSO algorithm, and linearly decreasing particle swarm algorithm (LDWPSO), where the number of particles N was set to 50 for the BSO and LDWPSO algorithms. The initial population N_{start} for SEDSABSO was set to 90, the learning factor was $c_1 = c_2 = 2.0$, and the inertial weight w was 0.8. The maximum number of iterations was set to 600. The initial temperature T for simulated annealing was set to 10,000, the decay scale of the annealing coefficient was set to 0.93, the step $\delta(t)$ was set to a constant value of 0.05, the adjustment factor λ was 0.9, and the ratio of the Aspen step to the distance between the whiskers, c , was set to two. After many trials, the optimal number of particle perturbations for the population was set to $q = 3$, f_{max} was 0.9, and f_{min} was 0.3. The iteration curves of the Aspen number for SEDSABSO and BSO are shown below:

Figure 5 shows that the number of particles N was constant during iterations of the BSO algorithm. Assuming that the maximum number of iterations was $Maxdt$ and the duration of iteration of each particle was t , the total duration of iterations of BSO was $N * Maxdt * t$, and $N * Maxdt$ is equal to $Area1 + Area2$ in the figure. For the SEDSABSO algorithm, the total iteration time was approximately the product of Figure 5. For the SEDSABSO algorithm, the total iteration time was approximately the product of the area of the class trapezoid enclosed by the dash and the horizontal axis and the time t , i.e., $(Area3 + Area1) * t$, and since the area of $Area3$ was not much different from that of $Area2$, the computational complexity of the improved Amanita group algorithm was approximately the same as that of the standard Amanita group algorithm.

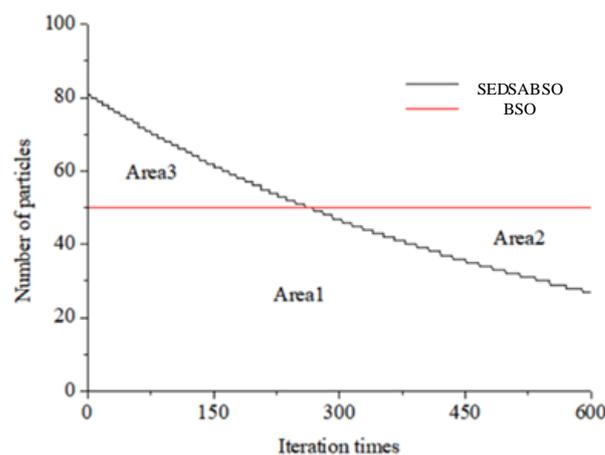


Figure 5. SEDSABSO and BSO Variation curve of number N of longicorn beetles.

Figure 6 below shows the adaptation iteration curves of each algorithm on an Intel(R) Core(TM) i7-8550U main frequency 4.00 GHz computer with 600 iterations through Matlab 9.1:

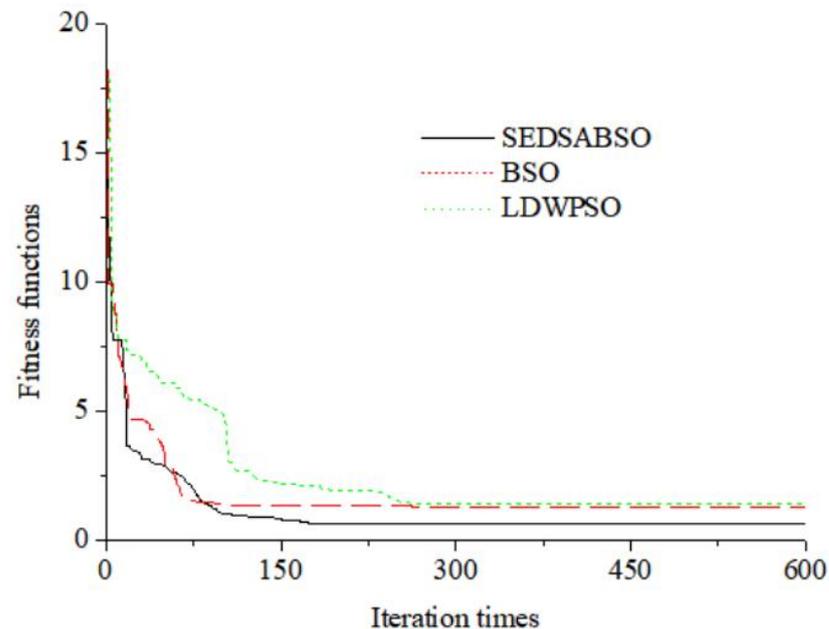


Figure 6. Iterative curve of the algorithm.

From Figure 6, we can see that the SEDSABSO algorithm proposed in this paper converges faster and has smaller fitness values than the BSO and LDWPSO algorithms. It is clear that the SEDSABSO algorithm proposed in this paper performs best.

In order to further verify the advanced level of the SEDSABSO algorithm, this paper ran each of the above three algorithms ten times and found the average fitness and average time as follows in Table 4:

Table 4. Comparison of average fitness values.

Algorithm	Average Fitness	Average Time/s
SDESABSO	0.95	16.93
BSO	1.12	17.31
LDWPSO	1.18	12.73

From Table 4, we can see that the average fitness values of the three algorithms after ten runs were only 0.95 for SEDSABSO, 1.12 for BSO, and 1.18 for LDWPSO, further confirming the stability and effectiveness of the proposed SEDSABSO algorithm. It can be seen that the average time of SEDSABSO was better than that of BSO, further proving the analysis in Figure 5. The difference with LDWPSO was not significant, and combined with the average fitness value, it can be seen that SEDSABSO performed optimally.

The linearized kinetic parameters of the RS010N industrial robot had a total of 21 minimum parameter sets, and the results of the kinetic parameters identified by SEDSABSO, BSO, and LDWPSO are shown in Table 5 below; the absolute errors between the identified kinetic parameters and the actual parameters were also obtained.

Table 5. Identification of dynamic parameters.

Dynamic Minimum Parameters	Theoretical Value	SEDSABSO Identification Value	BSO Identification Value	LDWPSO Identification Value	SEDSABSO Absolute Error	BSO Absolute Error	LDWPSO Absolute Error
$I_{1yy} + I_{2yy} + I_{3zz} + 2 * a_1 * I_{1x} + m_1 * a_1^2 - (m_2 + m_3) * a_2^2$	16.41	16.80	16.89	16.82	-0.39	-0.48	-0.41
F_{1c}	0	0.01	0.35	-0.20	-0.01	-0.35	0.2
F_{1v}	0	-0.07	-0.21	0.22	0.07	0.21	-0.22
$I_{2xx} - I_{2yy} + (m_2 + m_3) * a_2^2$	31.83	31.77	31.92	32.21	0.06	-0.09	-0.38
I_{2xy}	6.14	5.78	5.12	6.12	0.36	1.02	0.02
$I_{2x} - a_2 * (I_{2z} - I_{3y})$	5.76	5.54	6.22	5.83	0.22	-0.46	-0.07
I_{2yz}	-7.82	-7.84	-7.72	-8.10	0.02	-0.1	0.28
$I_{2zz} - (m_2 + m_3) * a_2^2$	-37.29	-37.11	-37.30	-37.94	-0.18	0.01	0.65
$I_{2x} + (m_2 + m_3) * a_2^2$	47.37	47.29	47.52	47.27	0.08	-0.15	0.1
I_{2y}	0.16	0.13	0.09	0.27	0.03	0.07	-0.11
F_{2c}	0	0.14	-0.02	-0.08	-0.14	0.02	0.08
F_{2v}	0	-0.21	-0.23	-0.03	0.21	0.23	0.03
$I_{3xx} - I_{3zz}$	-2.07	-2.20	-2.40	-1.51	0.13	0.33	-0.56
I_{3xy}	3.34	3.24	3.25	4.57	0.10	0.09	-1.23
I_{3xz}	-3.42	-3.34	-2.95	-4.10	-0.08	-0.47	0.68
I_{3yy}	1.02	0.77	1.12	-0.85	0.25	-0.1	1.87
I_{3yz}	-0.29	0.10	-0.34	-1.42	-0.39	0.05	1.13
I_{3x}	0.15	0.11	0.12	-0.04	0.04	0.03	0.19
I_{3z}	-0.034	0.00	-0.16	0.04	-0.034	0.126	-0.074
F_{3c}	0	0.18	-0.13	-1.20	-0.18	0.13	1.2
F_{3v}	0	0.06	-0.34	-0.1	-0.06	0.34	0.1

Taking the second parameter F_{1c} of the set of minimum parameters of the industrial robot in Table 5 as an example, the absolute errors identified by BSO and LDWPSO were -0.35 and 0.2 , respectively, while the absolute error of SEDSABSO was -0.01 . On the whole, the SEDSABSO algorithm yielded the smallest error in identifying the kinetic parameters of the industrial robot, while BSO and LDWPSO The kinetic parameters identified by the SEDSABSO algorithm deviated from the theoretical kinetic parameter values, which further reflects the superiority of the SEDSABSO algorithm. The maximum absolute error in the kinetic parameters as identified by the SEDSABSO algorithm was -0.39 . The overall error in identification was thus small. The SEDSABSO algorithm can thus identify the kinetic parameters of the robot.

6. Conclusions

- (1) In this paper, a new improved Beetle Antennae Search-QEDSABSO is proposed, which makes a class exponential change to the number of skinks in the iterative process of Beetle Antennae Search and effectively improved the utilization rate of Beetle Antennae Search skinks while the total number of skinks was basically unchanged. Simulation experiments showed that the proposed algorithm is more accurate and faster than the common particle swarm and Beetle Antennae Search in identifying the dynamics parameters of robots. The simulations showed that the proposed algorithm can identify the dynamical parameters of the robot with higher accuracy and faster speed than the common particle swarm and Beetle Antennae Search.
- (2) The difficulty in identifying the kinetic parameters of industrial robots lies in the sheer number of variables that need to be determined and the selection of reasonable excitation trajectories. This paper designed the relevant excitation trajectories by using the genetic algorithm and linearized the kinetic parameters of the industrial robot to improve the accuracy of their recognition.
- (3) The work provided the foundation for experiments compensating for the kinetic moments of the industrial robot. The minimum set of parameters of the kinetics could first be obtained by SymPyBotics. Then, the excitation trajectory of the industrial robot was designed by using the genetic algorithm, the data on its kinetic moments were collected, and the moments were identified by the SEDSABSO algorithm. Following this, the theoretical kinetic moments of the robot were calculated and compared with empirically sampled moments to obtain the error. Finally, this error was used to compensate for the kinetic moment of the robot.

Author Contributions: Methodology, B.K.; Software, B.K.; Validation, B.K.; Writing—original draft, B.K.; Writing—review & editing, B.K., S.G. article chart editor, B.K. and D.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the national key R & D plan project (2017YFB1301000).

Data Availability Statement: The data used in this study were self-tested and self-collected during the test. As the control method in this paper is still being further optimized, the data cannot be shared at present. Therefore, data sharing is not applicable to this article.

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

References

1. Liu, Z.; Zhao, B.; Zhu, H. Sorting Experimental Platform Research on Six-DOF Manipulator. *Mach. Des.Manuf.* **2013**, *27*, 210–213.
2. Wu, J.; Wang, J.; You, Z. An overview of dynamic parameter identification of robots. *Robot. Comput. Integr. Manuf.* **2010**, *26*, 414–419. [[CrossRef](#)]
3. Gautier, M.; Janot, A.; Vandanjon, P.O. A new closed-loop output error method for parameter identification of robot dynamics. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 428–444. [[CrossRef](#)]
4. Memar, A.H.; Esfahani, E.T. Modeling and Dynamic Parameter Identification of the SCHUNK Powerball Robotic Arm. In Proceedings of the ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Boston, MA, USA, 2–5 August 2015; American Society of Mechanical Engineers: New York, NY, USA, 2015; p. VOSCT08A024.
5. Fu, X.; Yuan, J.; Wang, S.; Wang, N.; Zhang, W. Nonlinear Dynamic Identification of Robotic Manipulators Based on Particle Swarm Optimization Method. *Mech.-Electr. Integr.* **2017**, *023*, 3–8.
6. Ding, L.; Wu, H.; Yao, Y.; Li, Y.; Xie, B.H.; Chen, B. Parameters Identification of Industrial Robots Based on WLS-ABC Algorithm. *J. South China Unive. Technol. (Natl. Sci. Ed.)* **2016**, *44*, 90–95.
7. Jiang, X.; Li, S. BAS: Beetle antennae search algorithm for optimization problems. *arXiv* **2017**, arXiv:1710.10724. [[CrossRef](#)]
8. Chen, T.; Yin, H.; Jiang, H. Particle Swarm Optimization Algorithm Based on Beetle Antennae Search for Solving Portfolio Problem. *Comput. Syst. Appl.* **2019**, *28*, 171–176.
9. Wang, T.; Long, Y.; Qiang, L. Beetle Swarm Optimization Algorithm: Theory and Application. *arXiv* **2018**, arXiv:1808.00206. [[CrossRef](#)]
10. Zhou, T.; Qian, Q.; Fu, Y. Fusion Simulated Annealing and Adaptive Beetle Antennae Search Algorithm. *Commun. Technol.* **2019**, *52*, 1626–1631.
11. Khan, A.H.; Cao, X.; Li, S.; Katsikis, V.N.; Liao, L. BAS-ADAM:an ADAM based approach to improve the performance of beetle antennae search optimizer. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 461–471. [[CrossRef](#)]
12. Xu, C. *Research on Dynamic Parameter Identification And Feedforward Control of Articulated Robots*; Southeast University: Nanjing, China, 2017.
13. Zhang, J.; Duan, J. Robot Dynamic Parameter Identification Based on Improved Differential Evolution Algorithm. *J. Beijing Union Univ.* **2020**, *1*, 49–55.
14. Craig, J.J. *Introduction to Robotics Mechanics and Control*; Pearson Prentice Hall: Upper Saddle River, NJ, USA, 2005.
15. Sun, Y.; Zhou, B.; Meng, Z. Dynamic Parameter Identification of Industrial Robot Based on Genetic Algorithm. *Ind. Control Comput.* **2017**, *9*, 4–6.
16. Swevers, J.; Ganseman, C.; De Schutter, J.; Van Brussel, H. Experimental robot identification using optimised periodic trajectories. *Mech. Syst. Signal Process.* **1996**, *10*, 561–577. [[CrossRef](#)]
17. Jiang, M.; Yuan, D. Wavelet Threshold Optimization with Artificial Fish Swarm Algorithm. In Proceedings of the 2005 International Conference on Neural Networks, Beijing, China, 13–15 October 2005; IEEE Press: Piscataway, NJ, USA, 2005; pp. 569–572.
18. Liu, C.; Yan, X.; Liu, C.; Wu, H. The wolf colony algorithm and applications. *Chin. J. Electron.* **2011**, *20*, 212–216.
19. Wang, Q.; Li, L.; Lu, C.; Sun, F. Average Computational Time Complexity Optimized Dynamic Particle Swarm Optimization Algorithm. *Comput. Sci.* **2010**, *37*, 191.
20. Kou, B.; Guo, S.; Ren, D. Geometric parameter calibration of industrial robot based on improved particle swarm optimization. *J. Harbin Inst. Technol.* **2021**, *49*, 61–64.
21. Wang, Y.G.; Qu, T.T.; Li, S. Disruption particle swarm optimization algorithm based on exponential decay weight. *Appl. Res. Comput.* **2020**, *37*, 1020–1024.
22. Liu, Y.; Li, G.X.; Xia, D.; Xu, W.F. Identifying Dynamic parameters of a space robot based on improved genetic algorithm. *J. Harbin Inst. Technol.* **2010**, *42*, 1734–1739.