processes MDPI

Article

Time-Jerk optimal Trajectory Planning of Industrial Robots Based on a Hybrid WOA-GA Algorithm

Fang Wang, Zhijun Wu * and Tingting Bao

Automobile School, Zhejiang Institute of Communications, Hangzhou 311112, China;
wangfang@zjvtit.edu.cn (F.W.); tingting@zjvtit.edu.cn (T.B.)
* Correspondence: zjwu@zjvtit.edu.cn; Tel.: +86-158-6901-7950

Abstract: An optimal and smooth trajectory for industrial robots has a positive impact on reducing the execution time in an operation and the vibration in their joints. In this paper, a methodology for the time-optimal and jerk-continuous trajectory planning of industrial robots is proposed. The entire trajectory is interpolated in the joint space utilizing fifth-order B-splines and then optimized by a hybrid whale optimization algorithm and genetic algorithm (WOA-GA). Two objective functions, including the integral of the squared jerk along the entire trajectory and the total execution time, are minimized to obtain the optimal entire trajectory. A fifth-order B-spline interpolation technique enables the achievement of a jerk-continuous trajectory, while respecting the kinematic limits of jerk, acceleration and velocity. WOA-GA is utilized to solve the time-jerk optimal trajectory planning problem with nonlinear constraints. The proposed hybrid optimization algorithm yielded good results and achieved the time-jerk optimal trajectory better under kinematic constraints compared to the genetic algorithm, whale optimization algorithm, improved whale optimization algorithm with particle swarm optimization and adaptive cuckoo search algorithm. The numerical results show the competent performances of the proposed methodology to generate trajectories with high smooth curves and short total execution time.

Keywords: industrial robots; optimal trajectory; WOA-GA; B-splines

check for updates

Citation: Wang, F.; Wu, Z.; Bao, T. Time-Jerk optimal Trajectory Planning of Industrial Robots Based on a Hybrid WOA-GA Algorithm. *Processes* **2022**, *10*, 1014. http://doi.org/10.3390/pr10051014

Academic Editor: Blaž Likozar

Received: 27 April 2022
Accepted: 18 May 2022
Published: 19 May 2022

1. Introduction

Due to their outstanding performance in accuracy, repeatability and efficiency, industrial robots have become one of the most important equipment to replace human resources in the machinery, automobile, electronics, metal, and other industries [1–3]. Generally speaking, an industrial robot is a nonlinear dynamic mechanism, which needs to follow a specific trajectory to avoid any possible collision. Therefore, trajectory planning is an essential technique in automated industrial activities using industrial robots. A reasonable trajectory can reduce the vibration and noise of robots and ensure a high trajectory tracking accuracy and motion stability of robots [4–6].

In the relevant literature, trajectories are mainly constructed by polynomial, spline, Bezier, NURBS and other interpolation functions. In order to obtain a smooth trajectory with jerk boundary, Jond et al. used polynomials of the orders 3, 4 and 5 to generate an acceleration-continuous trajectory, in which the acceleration values at the start and end motions are non-zero [7]. Sonja Macfarlane et al. employed a series of fifth-order polynomials to plan a smooth trajectory between two directional points [8]. Similarly, Bureerat et al. used quintic polynomials to connect the initial and intermediate points, to finally the end point [9]. Lin et al. proposed a method of a 7-segment acceleration profile motion model based on a time-optimal trajectory, which divided all waypoints into continuous batches, and each batch had two overlaps of waypoints [10]. Wu et al. proposed a novel point-to-point trajectory planning algorithm (PTPA) to improve the motion efficiency of industrial robots [11]. Moreover, spline functions have been used as trajectory primitives to ensure acceleration continuity or jerk continuity. Since each control point on the B-spline curve will only affect the parameters within a range of the curve, various B-splines were used to construct joint trajectory. For example, the minimum time trajectory is generated by cubic splines for the robotic manipulator [12–14]. Muhammad Zeeshan et al. interpolated the entire fifth-order trajectory with B-spline interpolation method to obtain the minimum time trajectory [15,16].

Trajectory optimization is a highly complicated problem due to the nonlinear characteristics of trajectories and the need of motion efficiency, smooth operation and low noise of industrial robots. Many studies take the minimum time [17–21], the minimum acceleration [9,16] and the minimum energy [22,23] as the trajectory planning optimization objectives. In addition, the hybrid optimization methods have become the focus of robot research in recent years. For example, Zhang et al. proposed a multi-objective optimization method for robot trajectory planning under obstacles, including time optimization and acceleration optimization. This method uses quintic B-splines to generate trajectory configurations in the joint space with motion constraints and obstacles constraints [24]. Wang et al. proposed a new algorithm to define the objective function from two aspects of time and speed [14]. Zhang et al. proposed an adaptive cuckoo search (ACS) algorithm to optimize the trajectory of quintic B-spline programming [25]. Ameer adopted an RNN based meta-heuristic approach (RNN-MA) to achieve real-time implementation [26]. Cheng proposed beetle antennae search algorithm (BAS) to tackle the formulated optimization function [27]. Hybrid algorithms, such as an improved whale optimization algorithm with differential evolution algorithm or particle swarm optimization (IWOA-DE and IWOA-PSO), and new genetic algorithm and adaptive elite genetic algorithm (AEGA-SA) have been proposed to improve the exploration and exploitation capacities of the original algorithm [21,28,29]. Several optimization methods with their characteristics are shown in Table 1.

Table 1. Several optimization methods with their characteristics.

Name	Characteristics
IWOA-DE [21]	Good global optimization ability, strong robustness and slow convergence speed.
ACS [25]	Faster convergence degree, higher accuracy and better global search ability, but the final accuracy needs to be improved.
RNN-MA [26]	Fast convergence speed and suitable for real-time implementation.
BAS [27]	Fast convergence speed with low searching precision and convergence speed, and sometimes it is prone to local optimum.
AEGA-SA [28]	Superior to the original genetic algorithm in terms of trajectory smoothness and trajectory efficiency, but it is highly dependent on population size, and is prone to low convergence speed or fall into local optimum.
IWOA-PSO [29]	Good global exploration ability and slow convergence speed.

This paper proposes a time-optimal and jerk-continuous trajectory planning technique for industrial robots. This technique is based on fifth-order B-spline and hybrid WOA-GA. Fifth-order B-spline is utilized to interpolate the multi-point trajectory. The WOA-GA combining the advantage of GA and WOA is proposed to solve the time-jerk trajectory optimization problem with nonlinear constraints. The rest of this paper is organized as follows. Section 2 defines the problems of trajectory optimization. In Section 3, the fifth-order B-spline is chosen to interpolate the joint trajectory, and the hybrid WOA-GA is presented. In Section 4, we compare the simulation results from different optimization algorithms, and Section 5 is the conclusion.

2. Problem Formulation

Trajectory planning is mostly carried out in the joint space, which can ensure the stability of joint motion and avoid motion jump of the manipulator, while planning trajectories in the operation space. Thus, this paper focus on planning trajectories in the joint space.

We define the optimal planning problem similarly to Ref. [12]. Two objective functions, including the integral of the squared jerk along the whole trajectory and the total execution time, are minimized to obtain the optimal whole trajectory. These two terms have opposite effects. This implies that reducing the total execution time will lead to trajectories with large values of the kinematic quantities, while reducing the integral of the squared jerk along the whole trajectory will lead to a smoother trajectory with low efficiency. The trade-off between these two objectives can be achieved by legitimately adjusting the weights of the two objectives. Therefore, the optimization problem can be formulated as:

$$\begin{cases} \text{find :} \\ \text{min} w_T N \sum\limits_{i=1}^{N_p-1} T_i + w_J \sum\limits_{j=1}^{N} \int_0^{T_f} (J_j(t))^2 dt \\ \ subject\ to: \\ \ P_j^{\text{lower}} \le P_j(t) \le P_j^{\text{upper}},\ j = 1, \ldots, N \\ \quad |V_j(t)| \le VL_j,\ j = 1, \ldots, N \\ \quad |A_j(t)| \le AL_j,\ j = 1, \ldots, N \\ \quad |J_j(t)| \le JL_j,\ j = 1, \ldots, N \end{cases} . \tag{1}$$

The meaning of the symbols in Equation (1) is described in Table 2.

Table 2. Meaning of the symbols in Equation (1).

Symbol	Meaning
N	Number of robot joints
w_T	Weight of the term proportional to the execution time
N_p	Number of via-points
T_i	Time interval between two via-points
$P_j(t)$	Position of the *j*th joint
$V_j(t)$	Velocity of the *j*th joint
$A_j(t)$	Acceleration of the jth joint
$J_j(t)$	Jerk of the jth joint
w_J	Weight of the term proportional to the jerk
T_f	Total execution time of the trajectory
$P_j{}^{\text{upper}}$	Upper-bound position of the *j*th joint
$P_j{}^{\text{lower}}$	Lower-bound position of the *j*th joint
VL_j	Velocity limit for the *j*th joint (symmetrical)
AL_j	Acceleration limit for the *j*th joint (symmetrical)
JL_j	Jerk limit for the *j*th joint (symmetrical)

3. Proposed Method

3.1. Multiple B-Splines

Multiple B-spline curves are widely used in the trajectory planning algorithms of robots with good local support and continuous smoothness. A B-spline curve can be expressed mathematically as follows:

$$p(u) = \sum_{i=0}^{n} d_i N_{i,k}(u). \tag{2}$$

where $p(u)$ is the joint position, d_i is the *i*th control points of the B-spline, k is the degree of the B-spline, $N_{i,k}(u)$ is the *i*th basis function of the *k*th order B-spline, n is the total number of the control points and u is the normalized knot vector variable. $N_{i,k}(u)$ can be obtained using the Cox–de Boor recursion formula.

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, \text{if} \quad t_i \le t < t_{i+1} \\ 0, \text{others} \end{cases} \\ N_{j,k}(u) = \frac{t-t_i}{t_{i+k}-t_i} N_{j,k-1}(u) + \frac{t_{i+k-1}-t}{t_{i+k-1}-t_{i+1}} N_{j+1,k-1}(u) \\ define \frac{0}{0} = 0 \end{cases} . \tag{3}$$

The trajectory of an industrial robot is discretized to some designed points on the time scale. Two virtual points are adopted at the second and the second–last positions of the designed point sequence, due to the consideration of zero jerk at the initial and final movements. Normalized knot vector u is generated from the sequence of $\boldsymbol{t}$ corresponding to the designed points [16].

$$\boldsymbol{u} = [u_0, u_1, \ldots, u_{n+2k}] \tag{4}$$

where

$$\begin{cases} u_0 = u_1 = \ldots = u_k = 0 \\ u_i = u_{i-1} + |h_{i-k-1}| / \sum\limits_{j=0}^{n-1} |h_j| \\ u_{n+k} = u_{n+k+1} = \ldots = u_{n+2k} = 1 \end{cases} . \tag{5}$$

The lth order derivatives can be described as

$$p^l(u) = \sum_{j=i-k+r}^{i} d_j^r N_{i,k-r}(u), \tag{6}$$

where

$$d_j^l = \begin{cases} d_j, l = 0 \\ (k+1-l)\frac{d_j^{l-1} - d_{j-1}^{l-1}}{u_{j+k+1-l} - u_j}, l = 1, 2, \ldots, r; j = i-k+l, \ldots, i \end{cases} . \tag{7}$$

3.2. Whale Optimization Algorithm

Whale Optimization Algorithm (WOA) is a recently proposed population-based optimization algorithm inspired by the hunting behavior of humpback whales [30]. The WOA algorithm starts with a set of random solutions, including three kinds of hunting behaviors: encircling prey, exploration and exploitation. At each iteration, the location of individuals in the population is updated according to the most promising solution [31].

3.2.1. Encircling Prey

Since the prey in the search space is not known, the WOA assumes that the current best-so-far solution is the target prey. Then, the other individuals will try to update their positions towards the target prey. This behavior is represented by the following equations:

$$\boldsymbol{X}(g+1) = \boldsymbol{X}^{\text{best}}(g) - \boldsymbol{A}\left|\boldsymbol{C}\boldsymbol{X}^{\text{best}}(g) - \boldsymbol{X}(g)\right|. \tag{8}$$

where $\boldsymbol{A}$ and $\boldsymbol{C}$ are coefficient vectors, $\boldsymbol{X}^{\text{best}}$ is the current best-so-far solution, $\boldsymbol{X}$ is an individual, g is the current iteration and | | denotes the absolute value.

The vectors $\boldsymbol{A}$ and $\boldsymbol{C}$ are calculated as follows.

$$\boldsymbol{A} = (2r-1)\boldsymbol{a} \tag{9}$$

$$\boldsymbol{C} = 2r \tag{10}$$

where $\boldsymbol{a}$ is linearly decreased from 2 to 0 over the iterations and $\boldsymbol{r}$ is a random set in [0, 1].

The selection of this operator depends on the coefficient vector $\boldsymbol{A}$ and a random number Po, where Po is a random number in [0, 1]. For each individual, if $Po < 0.5$ and $|\boldsymbol{A}| < 1$, the position is updated by encircling prey.

3.2.2. Bubble-Net Attacking Phase

If $Po > 0.5$, the bubble-net predation stage begins. The mathematical model of the search is as follows.

$$\boldsymbol{X}(g+1) = \left|\boldsymbol{X}^{\text{best}}(g) - \boldsymbol{X}(g)\right| e^{bm} \cos(2\pi m) + \mathrm{X}^{\text{best}}(g) \tag{11}$$

where b is a constant for defining the shape of the logarithmic spiral and m is a random number in $[-1, 1]$.

3.2.3. Searching for Prey

If $Po < 0.5$ and $|\boldsymbol{A}| \geq 1$, the individuals need to expand their exploration areas. This phase uses a randomly generated set of solutions $\boldsymbol{X}^{\text{rand}}$ to update the individuals' locations, which allow the WOA algorithm to perform a global search. The rules of the update solution are as follows.

$$\boldsymbol{X}(g+1) = \boldsymbol{X}^{\text{rand}}(g) - \boldsymbol{A}\left|\boldsymbol{C}\boldsymbol{X}^{\text{rand}}(g) - \mathrm{X}(g)\right|. \tag{12}$$

3.3. Genetic Algorithm

Genetic algorithm (GA) is a random global search and optimization method developed by imitating the biological evolution mechanism of nature [32]. It can automatically obtain and accumulate knowledge about the search space in the search process, and adaptively control the search process to obtain the best solution. Each individual in the population represents a possible solution to the optimization problem. The adaptation ability of individual is judged by the fitness function. The individuals with poor fitness are eliminated, and the individuals with good fitness can continue to reproduce. In the process of reproduction, it needs selection, crossover and variation to form a new population. Then, the better solution will finally be obtained.

3.4. Proposed Hybrid WOA-GA Algorithm

The flowchart of the proposed WOA-GA algorithm is shown in Figure 1, and detailed introductions are as follows:

Step 1: Define the initial parameters, such as *Pop_quantity*, *MaxGen*, *Pc*, *Pm* and *Er*. Then, an initial population is obtained randomly.
Step 2: Calculate the fitness value of each individual using Equation (1) and find $\boldsymbol{X}^{\text{global\_best}}$ and $\boldsymbol{X}^{\text{local\_best}}$.
Step 3: Update the populations by WOA. If $Po < 0.5$ and $|\boldsymbol{A}| < 1$, individuals are updated by encircling prey using Equations (9)–(11). If $Po < 0.5$ and $|\boldsymbol{A}| \geq 1$, individuals are updated by searching for prey using Equation (13). If $Po > 0.5$, individuals are updated by bubble-net attacking using Equation (12).
Step 4: Update the binary-encoded population by GA. Roulette wheel selection method, single-point crossover and bitwise mutation are utilized in the corresponding operators.
Step 5: Calculate the fitness value of each individual and update $\boldsymbol{X}^{\text{global\_best}}$ and $\boldsymbol{X}^{\text{local\_best}}$.
Step 6: If $t < MaxGen$, then $t = t + 1$, update $\boldsymbol{a}$, $\boldsymbol{A}$, $\boldsymbol{C}$, m, b and Po, and return to step 3. If $t = MaxGen$, then the best fitness and $\boldsymbol{X}^{\text{global\_best}}$ are obtained.

Since finding a suitable constraint handling method for WOA-GA is out of the scope of this work, a constraint handling technique named static penalty is employed [33].

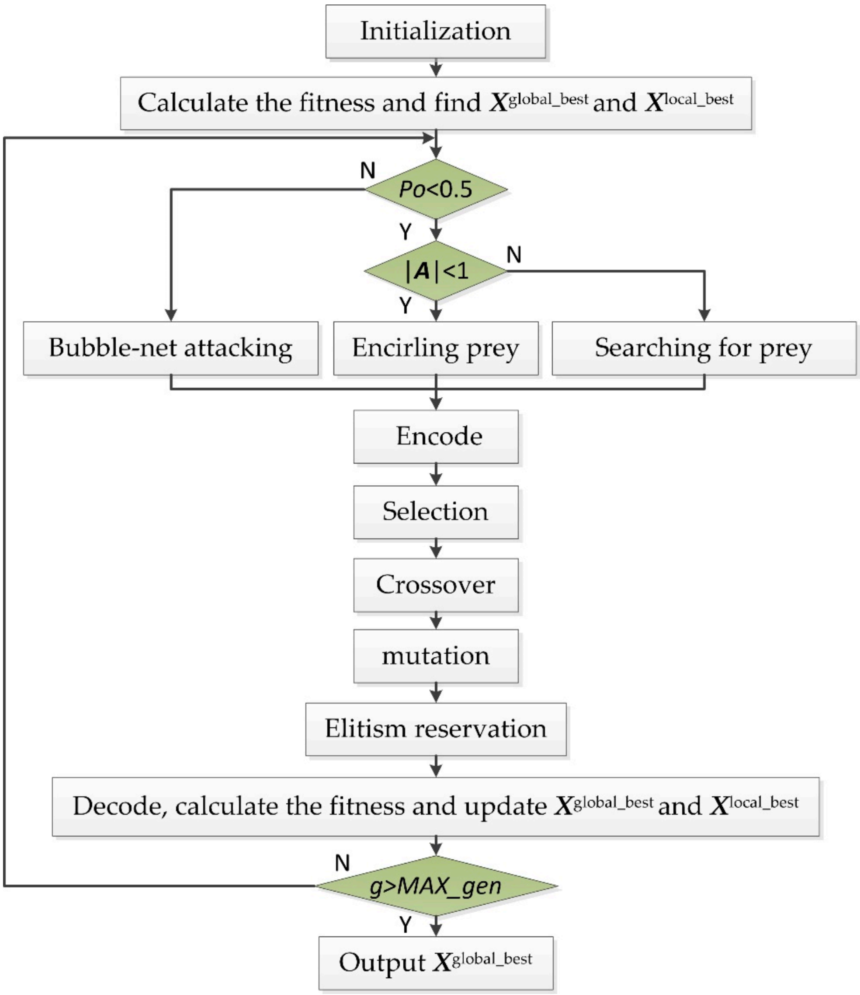
Initialization
Calculate the fitness and find $X^{\text{global\_best}}$ and $X^{\text{local\_best}}$
N
Po<0.5
Y
|***A***|<1
N
Y
Bubble-net attacking
Encirling prey
Searching for prey
Encode
Selection
Crossover
mutation
Elitism reservation
Decode, calculate the fitness and update $X^{\text{global\_best}}$ and $X^{\text{local\_best}}$
N
g>MAX_gen
Y
Output $X^{\text{global\_best}}$

Figure 1. Flowchart of the WOA-GA optimization algorithm.

4. Results and Discussion

In order to verify the feasibility of the proposed method, the numerical simulations were tested by solving six unconstrainted optimization problems and a time-optimal trajectory planning problem using 5th-order B-splines. WOA-GA was programed in MATLAB R2018b, as well as WOA, GA, IWOA-PSO [29] and ACS [25], which were utilized for comparative analysis. The configurations of the personal computer were a Core i5-1035G1 CPU with 16 GB of RAM. The parameter configuration of each algorithm are presented in Table 3. All the algorithms were initialized with the same population for each run. For WOA-GA and GA, we utilized 14 bits to code each time interval. The best-so-far feasible solution in the population obtained at the end of generations was utilized to evaluate WOA-GA on effectiveness and stability. The mean ('MEAN') and standard deviations ('STD') of the best-so-far solutions are utilized as evaluation indices, formulated as follows:

$$\text{MEAN} = \frac{1}{Nr}\sum_{i=1}^{Nr} f_i^{\text{best}} \tag{13}$$

$$\text{STD} = \sqrt{\frac{1}{Nr}\sum_{i=1}^{Nr} (f_i^{\text{best}} - MEAN)^2} \tag{14}$$

where f_i^{best} is the best-so-far feasible solution obtained in the ith run and Nr is the number of runs. Note that the smaller the values of MEAN and STD, the more reliable and stable the solutions that the algorithm can provide [34].

Table 3. Parameter configuration of each algorithm.

Algorithm	Parameter Configuration
WOA-GA	Pc = 0.85, Pm = 0.01, Er = 0.05
GA	Pc = 0.85, Pm = 0.01, Er = 0.05
IWOA-PSO [29]	Pc = 0.85, k = 0.4, c_1 = 2.0, c_2 = 2.0
ACS [25]	a = 0.7, b = 0.5, Pa = 0.25

4.1. Application 1: WOA-GA for Test Functions

The first six unconstrainted optimization problems are classical benchmark functions shown in Table 4. These functions include unimodal (F_1–F_2), variable-dimension multimodal (F_3–F_4) and fixed-dimension multimodal (F_5–F_6) benchmarks. Table 4 shows the details of the test functions. Note that only one optimal solution exists in the unimodal functions to evaluate the convergence rate and exploitation capacity of the algorithm, and a global optimal solution with several local ones exist in the multimodal functions to evaluate the exploration capacity of the algorithm. For each test function, WOA-GA was performed 20 times as well as WOA, GA, IWOA-PSO and ACS. For all algorithm, the test population size and the maximum iteration were set to 30 and 500, respectively.

Table 4. Details of the test functions.

Function	Dim	Range	Min
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100, 100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10, 10]	0
$F_3(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n} (y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1}) + (y_n - 1)^2]\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i+1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a < x_i < a \\ k(-x_i - a)^m, x_i < -a \end{cases}$	30	[−50, 50]	0
$F_4(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)2[1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[0, π]	0
$F_5(x) = 4x_1^2 + -2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5, 5]	−1.0316
$F_6(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[−2, 2]	3

Table 5 reports the statistical results for the unimodal function and shows that WOA-GA can obtain the global optimal solution for F_1 and F_2. Table 6 reports the statistical results for the variable-dimension multimodal and fixed-dimension multimodal functions. WOA-GA can obtain the near-optimal solution for F_3 and F_4, and the global optimal solution for F_5 and F_6. Tables 5 and 6 indicate that WOA-GA has advantages in global searching and can prevent falling into a local optimum, and demonstrate that WOA-GA is competitive with other algorithms. Hence, although the time consumption is high, the proposed WOA-GA can provide good exploration and exploitation capabilities.

Table 5. Comparisons of the optimization results for the unimodal functions.

Function		WOA-GA	WOA	GA	IWOA-PSO [29]	ACS [25]
F_1	BEST	0	1.21×10^{-80}	7.3354	2.83×10^{-29}	36.8232
	WORST	3.61×10^{-4}	6.08×10^{-70}	1.22×10^{3}	7.1957×10^{-17}	71.4722
	MEAN	3.45×10^{-5}	3.91×10^{-71}	434.7116	4.84×10^{-18}	52.3215
	STD	7.76×10^{-5}	1.36×10^{-70}	395.9814	1.62×10^{-17}	11.4114
	Time consumption	60.3291	0.1556	35.0099	0.4467	0.8789
F_2	BEST	0	8.11×10^{-59}	3.1050	7.54×10^{-32}	214.1185
	WORST	0.0060	9.9×10^{-51}	16.5300	1.79×10^{-20}	1.00×10^{10}
	MEAN	3.00×10^{-4}	3.36×10^{-52}	6.4972	9.83×10^{-22}	9.50×10^{9}
	STD	0.0013	2.16×10^{-51}	3.3679	3.90×10^{-21}	2.18×10^{9}
	Time consumption	60.8924	0.1868	36.1690	0.5520	1.0567

Table 6. Comparisons of the optimization results for the multimodal functions.

Function		WOA-GA	WOA	GA	IWOA-PSO [29]	ACS [25]
F_3	BEST	4.88×10^{-5}	0.0075	1.2974	0.1139	2.7302
	WORST	1.72×10^{-4}	0.0576	2.77×10^{6}	0.7058	7.7002
	MEAN	1.14×10^{-4}	0.0211	1.88×10^{5}	0.2740	4.8696
	STD	3.39×10^{-5}	0.0135	6.16×10^{6}	0.1528	1.3589
	Time consumption	60.8924	0.1868	36.1690	0.5520	1.0567
F_4	BEST	7.13×10^{-4}	0.1161	0.5866	1.3246	1.00×10^{10}
	WORST	0.0136	1.0580	1.07×10^{7}	2.3390	1.00×10^{10}
	MEAN	0.0047	0.5327	1.05×10^{6}	1.8482	1.00×10^{10}
	STD	0.0049	0.2221	2.95×10^{6}	0.2569	0
	Time consumption	56.5546	0.7357	35.9564	2.4473	2.0157
F_5	BEST	−1.0316	−1.0316	−1.0299	−1.0316	−1.0316
	WORST	−1.0316	−1.0316	−0.8077	−1.0316	−1.0316
	MEAN	−1.0316	−1.0316	−0.9549	−1.0316	−1.0316
	STD	2.22×10^{-16}	2.83×10^{-9}	0.0734	5.03×10^{-9}	4.55×10^{-16}
	Time consumption	20.6915	0.1751	18.0697	0.4982	0.6485
F_6	BEST	3	3	3	3	3
	WORST	3	3.0025	44.9507	30	3
	MEAN	3	3.0001	20.6778	11.1041	3
	STD	0	5.34×10^{-4}	14.2922	12.3703	3.12×10^{-15}
	Time consumption	7.9300	0.0662	7.0914	0.1689	0.2644

Figure 2 shows the convergence curves from WOA-GA, WOA, GA, IWOA-PSO and ACS. WOA, IWOA-PSO and ACS have an advantage on the local space search and can converge rapidly, but they are easily to trap in a local optimum in low iterations. GA has the advantage on global space search, but it converges slowly and needs more generations to obtain the best fitness. In each iteration of WOA-GA, WOA follows several generations of GA. Thus, the global search capacity of WOA is enhanced by GA, and the local search ability of GA is improved by WOA. With the positive interaction between WOA and GA, the exploration and exploitation abilities of WOA-GA are improved. The proposed WOA-GA can converge rapidly and obtain the best fitness compared with WOA, GA, IWOA-PSO and ACS.

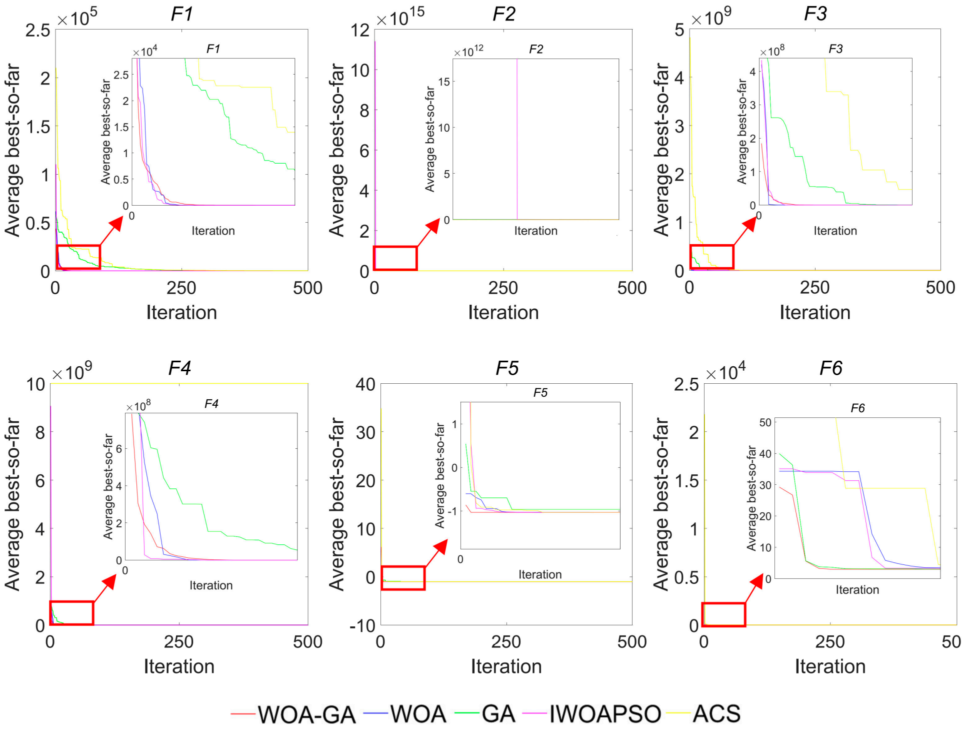

Figure 2. Convergence curves from WOA-GA, WOA, GA, IWOA-PSO and ACS.

4.2. Application 2: WOA-GA for Time-Optimal Trajectory Planning Problem

A typical multi-point trajectory task for a 6-DOF industrial robot was employed [12]. The desired positions and kinematic constraints of test task are listed in Tables 7 and 8, respectively. Since the order of magnitudes of the integral of the squared jerk is much greater than that of the total execution time, w_T and w_J were set as 0.9999 and 0.0001 to achieve the trade-off between two objectives. For all algorithms, a test population size of 30 and the maximum iteration of 100 and 500 were utilized to solve this problem. WOA-GA was performed 10 times as well as WOA, GA, IWOA-PSO and ACS.

Table 7. Desired positions of the test trajectory planning task.

Position No.	Joint No. (°)					
	1	2	3	4	5	6
1	−10	20	15	150	30	120
2	Virtual point					
3	60	50	100	100	110	60
4	20	120	−10	40	90	100
5	Virtual point					
6	55	35	30	10	70	25

Table 8. Kinematic constraints of the test trajectory planning task.

Kinematic Constraints	Joint No.					
	1	2	3	4	5	6
Velocity (°/s)	100	95	100	150	130	110
Acceleration (°/s^2)	60	60	75	70	90	80
Jerk (°/s^3)	60	66	85	70	75	70

Table 9 shows the comparisons of the optimization results obtained using WOA-GA, WOA, GA, IWOA-PSO and ACS. WOA-GA is able to converge best in the test task, and the mean and standard deviations are also the smallest, which implies that WOA-GA has a good stability. Figure 3 shows the distributions of the best-so-far solutions from WOA-GA, WOA, GA, IWOA-PSO and ACS. Additionally, the percentage of performance rank in Table 9 indicates that WOA-GA mostly achieves a better performance than other benchmark algorithms.

Table 9. Comparisons of the optimization results for the test trajectory planning task.

Algorithm	BEST	WORST	MEAN	STD	Rank 1st	Rank 2nd	Rank 3rd	Rank 4th	Rank 5th
WOA-GA	39.9778	39.9837	39.9809	0.0020	80%	20%	/	/	/
WOA	39.9865	42.6031	40.9722	0.8536	/	/	20%	80%	/
GA	40.0018	41.6566	40.4576	0.5605	/	/	80%	20%	/
IWOA-PSO [29]	41.0080	46.0309	43.1747	1.5323	/	/	/	/	100%
ACS [25]	93.9780	39.9952	39.9861	0.0064	20%	80%	/	/	/

The travelling time of the best solution from WOA-GA is 29.9930 s, where T_1 = 0.0030 s, T_2 = 10.8240 s, T_3 = 8.9690 s, T_4 = 10.1850 s and T_5 = 0.0120 s. Figures 4–7 illustrate the trajectory curves of the industrial robot and their derivatives, including jerk, acceleration and velocity. As shown in Figure 7, the jerk curves of all joints were continuous, which resulted in the smoothness of the optimized trajectories. Furthermore, the values of jerk, acceleration and velocity were zero at the start and end movement. The motion curves were all bound under the kinematic constraints all the time. Therefore, the proposed method is an effective time-jerk optimal trajectory planning tool for industrial robots.

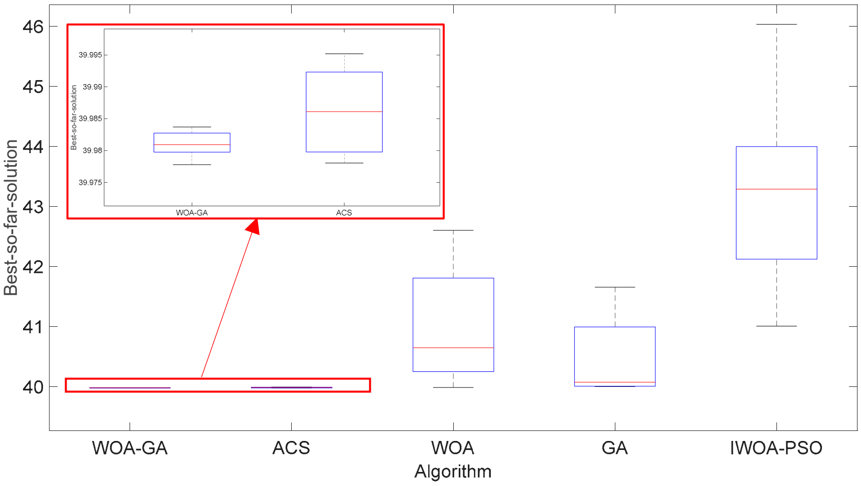

Figure 3. Best-so-far feasible solution from WOA-GA, WOA, GA, IWOA-PSO and ACS.

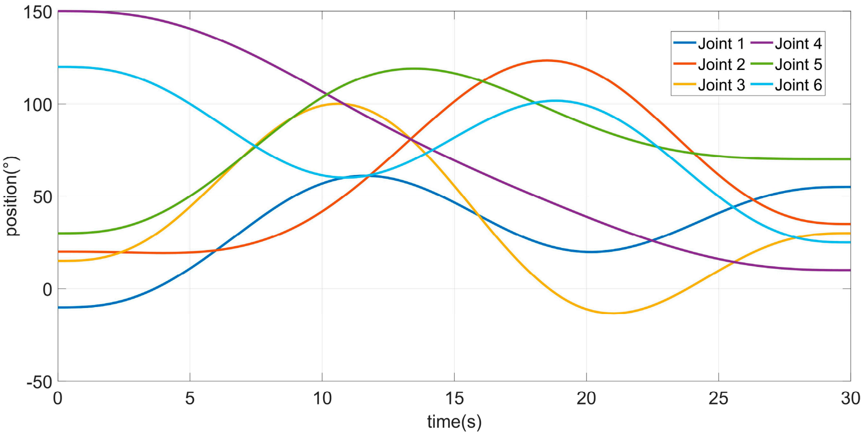

Figure 4. Position curves of all joints by the proposed technique (T_f = 29.9930 s).

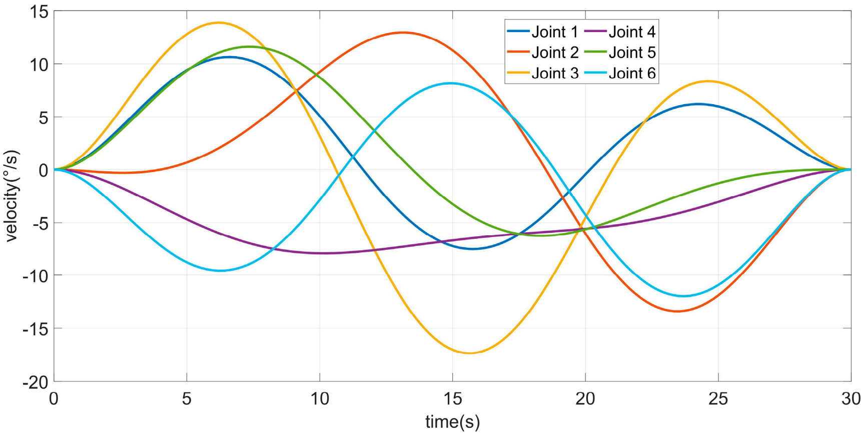

Figure 5. Velocity curves of all joints by the proposed technique (T_f = 29.9930 s).

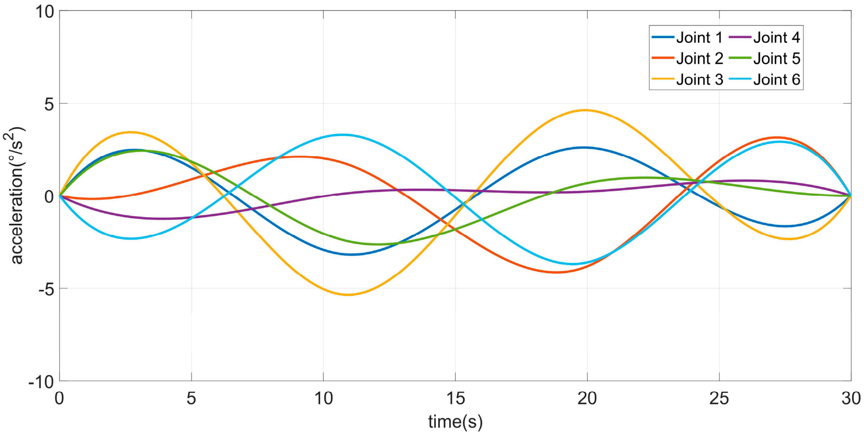

Figure 6. Acceleration curves of all joints by the proposed technique (T_f = 29.9930 s).

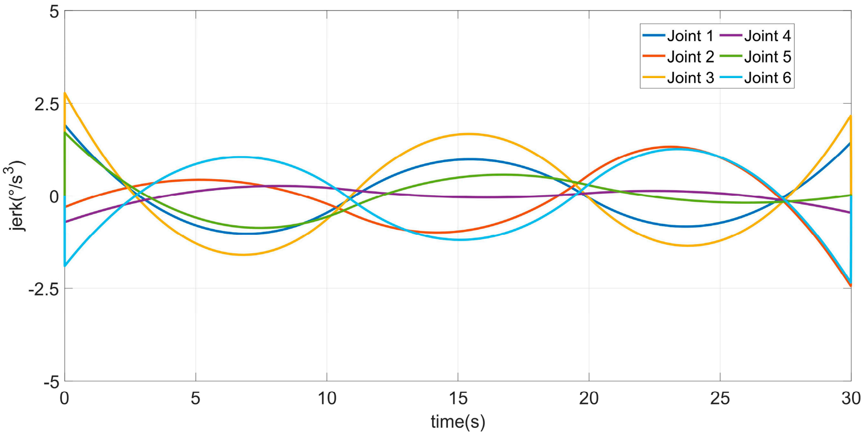

Figure 7. Jerk curves of all joints by the proposed technique (T_f = 29.9930 s).

5. Conclusions

In this article, we proposed a new methodology for time-optimal and jerk-continuous trajectory planning of industrial robots. Fifth-order B-spline was adopted to interpolate the multi-point trajectory in the joint space. The proposed WOA-GA combining the advantages of WOA and GA is utilized to solve the time-jerk optimal trajectory planning problem with nonlinear constraints. Simulations were carried out to verify the effectiveness of the proposed methodology. The comparison results of the best-so-far feasible solutions showed that WOA-GA provides the best results compared with WOA, GA, IWOA-PSO and ACS. Furthermore, the proposed WOA-GA algorithm has good stability. The optimized motion curves obtained by the proposed methodology were all bound under the kinematic constraints all the time. Therefore, these results demonstrate that the proposed methodology is an effective tool for industrial robots to generate time-jerk optimal trajectories.

Future work will be devoted to extending WOA-GA to optimize other trajectory models based on polynomial and NURBS. The effectiveness of the proposed technique may further be investigated by experiments.

Author Contributions: Conceptualization, F.W. and Z.W.; methodology, F.W. and Z.W.; writing—original draft preparation, F.W., Z.W. and T.B.; writing—review and editing, F.W., Z.W. and T.B.; supervision, T.B.; project administration, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by [General Scientific Research Project of Education Department of Zhejiang Province] grant number [Y202147343] and [Research on Public Welfare Technology Application Projects of Zhejiang Province] grant number [GG19E050005].

Acknowledgments: The authors acknowledge the support and inspiration of the research on public welfare technology application projects of Zhejiang province and the general scientific research project of the education department of Zhejiang province.

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

References

1. Paryanto; Brossog, M.; Bornschleg, M.; Franke, J. Reducing the energy consumption of industrial robots in manufacturing systems. *Int. J. Adv. Manuf. Technol.* **2015**, *78*, 1315–1328. [CrossRef]
2. Lin, W.Y.; Ren, X.Y.; Zhou, T.T.; Cheng, X.J.; Tong, M.S. A novel robust algorithm for position and orientation detection based on cascaded deep neural network. *Neurocomputing* **2018**, *308*, 138–146. [CrossRef]
3. Yin, G.T.; Zhu, Z.H.; Gong, H.; Lu, Z.F.; Yong, H.S.; Liu, L.; He, W. Flexible punching system using industrial robots for automotive panels. *Robot. Comput. Integr. Manuf.* **2018**, *52*, 92–99. [CrossRef]
4. Rubioa, F.; Llopis-Albert, C.; Valero, F.; Suñer, J.L. Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory. *Robot. Auton. Syst.* **2016**, *86*, 106–112. [CrossRef]
5. Ha, C.W.; Lee, D. Analysis of embedded prefilters in motion profiles. *IEEE Trans. Ind. Electron.* **2018**, *65*, 1481–1489. [CrossRef]
6. Gasparetto, A.; Zanotto, V. A new method for smooth trajectory planning of robot manipulators. *Mech. Mach. Theory* **2007**, *42*, 455–471. [CrossRef]
7. Jond, H.B.; Nabiyev, V.V.; Benveniste, R. Trajectory planning using high-order polynomials under acceleration constraint. *J. Optim. Ind. Eng.* **2017**, *21*, 1–6. [CrossRef]
8. Macfarlane, S.; Croft, E.A. Jerk-Bounded Manipulator Trajectory Planning: Design for Real-Time Applications. *IEEE Trans. Robot. Autom.* **2003**, *19*, 42–52. [CrossRef]
9. Bureerat, S.; Pholdee, N.; Radpukdee, T.; Jaroenapibal, P. Self-adaptive MRPBIL-DE for 6D robot multi-objective trajectory planning. *Expert Syst. Appl.* **2019**, *136*, 133–144. [CrossRef]
10. Lin, J.; Somani, N.; Hu, B.; Rickert, M.; Knoll, A. An Efficient and Time-Optimal Trajectory Generation Approach for Waypoints under Kinematic Constraints and Error Bounds. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018. [CrossRef]
11. Wu, Z.; Chen, J.; Bao, T.; Wang, J.; Zhang, L.; Xu, F. A Novel Point-to-Point Trajectory Planning Algorithm for Industrial Robots Based on a Locally Asymmetrical Jerk Motion Profile. *Processes* **2022**, *10*, 728. [CrossRef]
12. Gasparetto, A.; Zanotto, V. A technique for time-jerk optimal planning of robot trajectories. *Robot. Comput. Integr. Manuf.* **2008**, *24*, 415–426. [CrossRef]
13. Chettibi, T.; Lehtihet, H.E.; Haddad, M.; Hanchi, S. Minimum cost trajectory planning for industrial robots. *Eur. J. Mech. A Solids* **2004**, *23*, 703–715. [CrossRef]
14. Wang, T.; Xin, Z.; Miao, H.; Zhang, H.; Chen, Z.; Du, Y. Optimal Trajectory Planning of Grinding Robot Based on Improved Whale Optimization Algorithm. *Math. Probl. Eng.* **2020**, *4*, 3424313. [CrossRef]
15. Zeeshan, M.; Xu, H. Jerk-Bounded Trajectory Planning of Industrial Manipulators. In Proceedings of the IEEE International Conference on Robotics and Biomimetics, Dali, China, 6–8 December 2019. [CrossRef]
16. Huang, J.S.; Hu, P.F.; Wu, K.Y.; Zeng, M. Optimal time-jerk trajectory planning for industrial robots. *Mech. Mach. Theory* **2018**, *121*, 530–544. [CrossRef]
17. Fang, Y.; Liu, W.; Shao, Q.; Qi, J.; Peng, Y.H. Smooth and time-optimal s-curve trajectory planning for automated robots and machines. *Mech. Mach. Theory* **2019**, *137*, 127–153. [CrossRef]
18. Fang, Y.; Hu, J.; Qi, J.; Liu, W.H.; Wang, W.M.; Peng, Y.H. Planning trigonometric frequency central pattern generator trajectory for cyclic tasks of robot manipulators. *Arch. Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2019**, *233*, 4014–4031. [CrossRef]
19. Rymansaib, Z.; Iravani, P.; Sahinkaya, M.N. Exponential trajectory generation for point to point motions. In Proceedings of the 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Wollongong, Australia, 9–12 July 2013. [CrossRef]
20. Boryga, M.; Graboś, A. Planning of manipulator motion trajectory with higher-degree polynomials use. *Mech. Mach. Theory* **2009**, *44*, 1400–1419. [CrossRef]
21. Abu-Dakka, F.J.; Valero, F.J.; Suner, J.L.; Mata, V. A direct approach to solving trajectory planning problems using genetic algorithms with dynamics considerations in complex environments. *Robotica* **2015**, *33*, 669–683. [CrossRef]
22. Bailón, W.P.; Cardiel, E.B.; Campos, I.J.; Paz, A.R. Mechanical energy optimization in trajectory planning for six DOF robot manipulators based on eighth-degree polynomial functions and a genetic algorithm. In Proceedings of the 2010 7th International Conference on Electrical Engineering Computing Science and Automatic Control, Tuxtla Gutierrez, Mexico, 8–10 September 2010. [CrossRef]
23. Fung, R.F.; Cheng, Y.H. Trajectory planning based on minimum absolute input energy for an LCD glass-handling robot. *Appl. Math. Model.* **2014**, *38*, 2837–2847. [CrossRef]
24. Zhang, X.; Shi, G. Multi-objective optimal trajectory planning for manipulators in the presence of obstacles. *Robotica* **2022**, *40*, 888–906. [CrossRef]

25. Liu, Y.; Guo, C.; Weng, Y. Online time-optimal trajectory planning for robotic manipulators using adaptive elite genetic algorithm with singularity avoidance. *IEEE Access* **2019**, *7*, 146301–146308. [CrossRef]
26. Zhang, L.; Wang, Y.; Zhao, X. Time-optimal trajectory planning of serial manipulator based on adaptive cuckoo search algorithm. *J. Mech. Sci. Technol.* **2021**, *35*, 3171–3181. [CrossRef]
27. Zhao, J.; Zhu, X.; Song, T. Serial Manipulator Time-Jerk Optimal Trajectory Planning Based on Hybrid IWOA-PSO Algorithm. *IEEE Access* **2022**, *10*, 6592–6604. [CrossRef]
28. Khan, A.H.; Li, S.; Chen, D.; Liao, L. Tracking control of redundant mobile manipulator: An RNN based metaheuristic approach. *Neurocomputing* **2020**, *400*, 272–284. [CrossRef]
29. Cheng, Y.; Li, C.; Li, S.; Li, Z. Motion Planning of Redundant Manipulator with Variable Joint Velocity Limit Based on Beetle Antennae Search Algorithm. *IEEE Access* **2020**, *8*, 138788–138799. [CrossRef]
30. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
31. Zhang, L.; Wang, J.; Chen, J.; Chen, K.; Lin, B.; Xu, F. Dynamic modeling for a 6-DOF robot manipulator based on a centrosymmetric static friction model and whale genetic optimization algorithm. *Adv. Eng. Softw.* **2019**, *135*, 102684. [CrossRef]
32. Holland, J.H. *Adaptation in Natural and Artificial Systems*; MIT Press: London, UK, 1992.
33. Coello, A.C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Comput. Method. Appl. Mech. Eng.* **2002**, *191*, 1245–1287. [CrossRef]
34. Zhao, W.; Wang, L.; Mirjalili, S. Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Comput. Method. Appl. Mech. Eng.* **2022**, *388*, 114194. [CrossRef]