Journal of
Marine Science and Engineering

Article

Vibration Suppression Trajectory Planning of Underwater Flexible Manipulators Based on Incremental Kriging-Assisted Optimization Algorithm

Hui Huang, Guoyuan Tang *, Hongxuan Chen, Jianjun Wang, Lijun Han and De Xie

School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; hhuang@hust.edu.cn (H.H.)
* Correspondence: tgyuan@hust.edu.cn

Abstract: It is of great significance to expand the functions of submarines by carrying underwater manipulators with a large working space. To suppress the flexible vibration of underwater manipulators, an improved sparrow search algorithm (ISSA) combining an elite strategy and a sine algorithm is proposed for the trajectory planning of underwater flexible manipulators. In this method, the vibration evaluation function is established based on the precise dynamic model of the underwater flexible manipulator and considering complex motion and vibration constraints. Simulation results show that the ISSA algorithm requires only 1/3.68 of the time of PSO. Compared to PSO, SSA and the opposition-based learning sparrow search algorithm (OBLSSA), the optimization performance is improved by 17.3%, 13.1% and 9.7%, respectively. However, because the complex dynamics model of the underwater flexible manipulator leads to large computational effort and a long optimization time, ISSA is difficult to apply directly in practice. To obtain a large number of optimization results in a shorter time, an incremental Kriging-assisted ISSA (IKA-ISSA) is proposed in this paper. Simulation results show that IKA-ISSA has good nonlinear approximation ability and the optimization time is only 3% of that of the ISSA.

Keywords: flexible manipulator; vibration suppression; trajectory planning; sparrow search algorithm; incremental Kriging

Citation: Huang, H.; Tang, G.; Chen, H.; Wang, J.; Han, L.; Xie, D. Vibration Suppression Trajectory Planning of Underwater Flexible Manipulators Based on Incremental Kriging-Assisted Optimization Algorithm. *J. Mar. Sci. Eng.* **2023**, *11*, 938. https://doi.org/10.3390/jmse11050938

Academic Editors: Mai The Vu and Hyeung-Sik Choi

Received: 28 March 2023
Revised: 23 April 2023
Accepted: 25 April 2023
Published: 27 April 2023

1. Introduction

Nowadays, the extension of the functions of large submersibles using underwater manipulators for the dynamic deployment and recovery of unmanned underwater vehicles (UUV) is a major research direction [1–3]. UUVs have the characteristics of strong concealment, high intelligence, lower construction costs and no casualties. The integration of submarines and UUVs can combine each one's advantages and improve the overall performance of submarines. As new collaborative models between UUVs and submarines are gradually enriched and perfected, the question of how to carry, deploy, dock and recover one or more UUVs on a submarine is a key technical issue in the new collaborative model. To realize the above operations, using a submarine to carry the manipulator with large-scale operation capability is an important solution. The recovery and docking process of UUVs with the underwater manipulator is shown in Figure 1, in which the underwater manipulator can be stored in a small space inside the submarine after the operation is completed. Considering the occupation of the limited collection space of the submarine and the impact on its load capacity, the manipulator should have a large working space, a small storage space and a small weight so that it can be stored in the small space of the submarine. To meet the above technical requirements for large-scale operation and small-space storage of the underwater manipulator, the manipulator is often designed to be very slender. Due to the low stiffness of slender manipulators, the end effector is prone to vibration, and previous control methods that ignore the flexible deformation of manipulators are not able to meet the high-precision requirements of underwater operations [4,5]. In addition, when the manipulator is working, the speed and acceleration at the end of the manipulator should be strictly limited to meet the corresponding constraints. It is also necessary to overcome the interference of hydrodynamic forces. These factors bring great challenges to the control of underwater manipulators. Although extensive research has been carried out on the control of flexible manipulators and underwater manipulators, the vibration suppression of underwater flexible manipulators is still a difficult problem to solve.

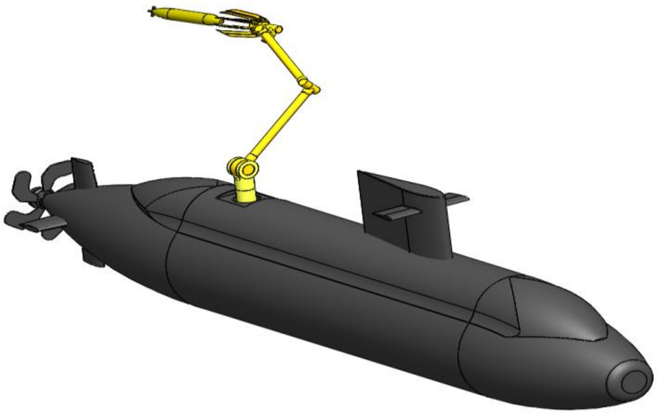

Figure 1. UUV recovery and docking system on a submarine.

At present, the vibration suppression of flexible manipulators is mainly studied from the perspective of control methods and trajectory planning. Sahu and Patra proposed an observer-based backstepping control method for a two-degree-of-freedom flexible manipulator [6]. Yang et al. proposed a hybrid control scheme composed of a trajectory planning approach and adaptive variable structure control to suppress the elastic vibration [7]. Rahmani and Belkheiri designed a neural network adaptive controller of flexible multi-link robots. The adaptive controller has good robustness under uncertain disturbances [8]. Yavuz proposed an improved vibration control method that suppresses residual vibration by shaping the speed input [9]. Guo et al. proposed a residual vibration suppression method for Delta robots based on input shaping technology [10]. Xu et al. proposed a compensation method; this method is based on the analytical solution of a second-order vibration system and designs compensation torque to suppress vibration [11]. Sands designed a whiplash compensator for flexible space robots, which had zero residual vibration at the end of the maneuver [12]. Trajectory planning for vibration suppression was firstly proposed by Park and Park [13]. Their method uses the excitation force, elastic deformation or elastic potential energy as the objective function to find the joint trajectory with the smallest flexible deformation or elastic potential energy, so as to achieve the vibration suppression of the manipulator. The vibration suppression trajectory optimization mainly defines the motion trajectory of the robot joint as polynomial functions or B-spline functions, transforms the discrete dynamic variables into parameter optimization problems and suppresses the flexible vibration of the flexible robot by optimizing the parameters [14–16].

The most commonly used optimization algorithms include traditional optimization algorithms (simplex method and compound shape method) and intelligent optimization algorithms (particle swarm optimization algorithm, genetic algorithm and sparrow search algorithm). The compound shape method is a direct method to solve the constrained nonlinear optimization problem. Lin et al. proposed an optimization method for the shortest total travel time under physical constraints such as joint velocity and acceleration based on the compound shape method [17]. Yin et al. optimized the two joint trajectories based on the Nelder–Mead simplex method in the feedforward control of the flexible manipulator to reduce the residual vibration of the flexible manipulator [18]. Since the selection and replacement of each vertex of the complex shape algorithm needs to meet the requirement of decreasing the value of the objective function and constraints, the search efficiency is relatively low. In processing optimization problems with complex models and constraints, the intelligent optimization algorithms have more advantages than traditional optimization algorithms. Wu et al. used a uniform aperiodic fourth-order B-spline curve to describe the trajectory of the manipulator, and conducted vibration suppression optimization for B-spline control points based on PSO [14]. Li et al. used a polynomial function to describe the trajectory of the manipulator, and used PSO for the optimization of polynomial interpolation points to suppress vibration [19]. Yue et al. and Kazem et al. studied the trajectory planning of robots based on the genetic algorithm (GA) to optimize the time in motion [20,21]. Li et al. used the GA to control the coefficients of cubic spline interpolation to suppress the vibration of the manipulator [22]. Different from GA, PSO does not use hybridization, mutation or replication for individuals, but instead treats each individual as a particle without volume in a multi-dimensional search space. In the process of vibration suppression trajectory optimization, the PSO algorithm has lower computational complexity than the GA. PSO has no special requirements for the objective function, while the GA requires a constant positive fitness function. However, the standard PSO algorithm is an unconstrained optimization algorithm, while the vibration suppression optimization of underwater flexible manipulators is constrained by the joint angle and flexible vibration. For the application of the PSO algorithm in constrained optimization problems, researchers have proposed a series of schemes to solve constrained optimization problems, among which the penalty function method and the constraint-based individual sorting method are the most widely used [23–25]. Cao et al. proposed a PSO algorithm with a compression factor as a penalty function to correct the joint trajectory and suppress the vibration of flexible joints [26]. Wang et al. studied the application of the PSO strategy under constraints of trajectory planning for a free-floating dual-arm robot [27]. Due to the constraints, PSO is more prone to premature convergence in the constrained optimization process. Although various improved PSO algorithms to solve constrained optimization problems have improved the optimization performance to a certain extent, premature convergence still exists. SSA is a new heuristic algorithm for swarm intelligence, proposed by Xue and Shen [28]. Zhang et al. proposed a path planning method for bionic mobile robots based on the sparrow search algorithm [29]. Liu et al. proposed an improved sparrow search algorithm to solve the obstacle avoidance problem of UAV route planning, and achieved good results [30]. Compared with the traditional heuristic search method, the SSA has more diversified position updating strategies, faster convergence speeds and more extensive application scenarios, indicating its great potential in dealing with robot vibration suppression trajectory planning.

Although there are some studies on the vibration suppression of flexible manipulators, in these studies, the suppression optimization effect remains to be further improved due to the relatively simple constraints and largely simplified models of flexible manipulators, and the vibration suppression of underwater manipulators is not involved. In addition, most of the current optimization algorithms suffer from premature convergence and poor accuracy in the constrained optimization process, and the optimization time is too long and costly for the case of complex models, making it difficult to be applied directly to practice.

To solve the problems of the premature convergence and low accuracy of traditional optimization algorithms in the process of constrained optimization, this paper proposes ISSA, which combines the sine algorithm and elite opposition-based learning strategy to optimize the trajectory of constrained underwater flexible manipulators. To obtain a large number of optimization results in a shorter time, this paper proposes IKA-ISSA, based on ISSA, to quickly generate vibration suppression trajectories of underwater flexible manipulators. The main contributions of this paper are summarized as follows. (i) In the past, the research object of vibration suppression trajectory planning was land-based flexible manipulators, while the research object of this paper is underwater flexible manipulators, considering the influence of hydrodynamics. (ii) The previous vibration suppression trajectory planning of flexible manipulators has greatly simplified the model. This paper is based on a more accurate dynamic model, and also considers the velocity, acceleration, maximum flexible displacement and other complex constraints. (iii) Based on the combination of the sine algorithm and OBL strategy, an improved sparrow algorithm is proposed, which is used for the first time to solve the vibration suppression trajectory planning problem for underwater flexible manipulators. (iv) IKA-ISSA is proposed for the problems of large computation demands and long optimization times for complex models. The algorithm has good robustness and nonlinear approximation capability, and can obtain accurate vibration suppression trajectories in a short time.

The rest of this paper is organized as follows. Section 2 presents the dynamic modeling approach for an underwater flexible manipulator. Section 3 describes the motion and vibration constraints, and establishes the manipulator vibration evaluation function. Section 4 designs the vibration suppression trajectory strategy of the underwater flexible manipulator based on PSO and ISSA and verifies the optimization effect of ISSA by simulation. Section 5 proposes IKA-ISSA and verifies its high optimization accuracy and efficiency by simulation. Finally, the conclusions can be found in Section 6.

2. Materials and Methods

The modeling of underwater flexible manipulator systems involves the complex integration of multiple disciplines, such as fluid mechanics and multi-body dynamics. The underwater flexible manipulator is usually disturbed by hydrodynamics, as well as some uncertain factors. These influences and disturbances are highly nonlinear and time-varying, which increases the difficulty of flexible manipulator modeling [31,32]. The authors of this paper fully consider the coupling effect of hydrodynamic force and flexible vibration, and we have established the dynamic equation of rigid–flexible coupling in our previous work [33,34].

The mechanism diagram of a two-link rigid–flexible coupling manipulator can be simplified as in Figure 2, in which arm 1 is abstracted as a rigid link and arm 2 is abstracted as a flexible link. The flexible link adopts the Euler–Bernoulli beam model, and the transverse shear deformation is ignored. θ_i and l_i, respectively, represent the rotation angle and length of the i-th link. w represents the flexible displacement. According to the assumption of small elastic deformation, only the first two-order modes are considered when w is expanded based on the assumed mode method (AMM).

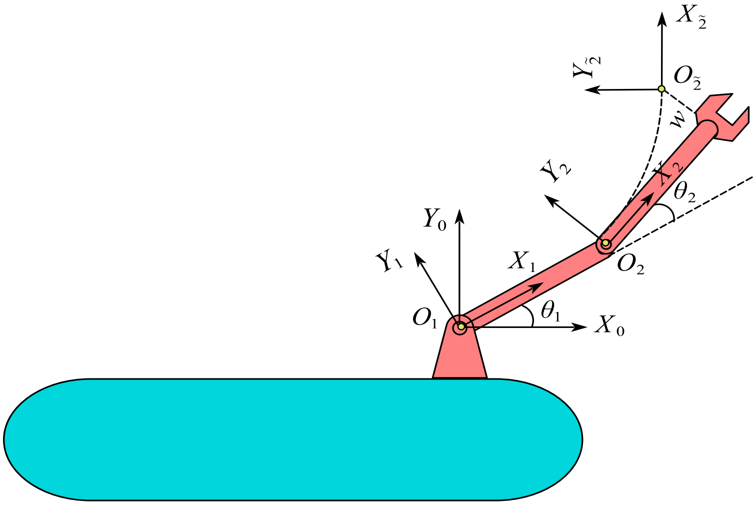

Figure 2. Model of 2-DOF flexible manipulator.

By combining the Morrison formula and Lagrange equation, the dynamic equations of the underwater flexible manipulator can be expressed in the following compact form [33,34]:

$$\begin{pmatrix} \boldsymbol{M}_{\theta\theta} & \boldsymbol{M}_{\theta q} \\ \boldsymbol{M}_{q\theta} & \boldsymbol{M}_{qq} \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{\theta}} \\ \ddot{\boldsymbol{q}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{C}_r \\ \boldsymbol{K}_q \boldsymbol{q} + \boldsymbol{C}_f \end{pmatrix} = \begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{0} \end{pmatrix} + \begin{pmatrix} \boldsymbol{F}_\theta \\ \boldsymbol{F}_q \end{pmatrix} \tag{1}$$

where $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T$ is the joint angle, $\boldsymbol{q} = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^T$ is the first two-order modes, $\boldsymbol{M}_{**}$ is the mass matrices, $\boldsymbol{K}_q$ is the stiffness matrix, $\boldsymbol{C}_r$ and $\boldsymbol{C}_f$ is the terms of centrifugal forces, Coriolis forces, and gravity. $\boldsymbol{\tau}$ is the joint torque, and $\boldsymbol{F}_\theta$ and $\boldsymbol{F}_q$ are the generalized forces related to hydrodynamic force.

3. Vibration Suppression Trajectory Planning

3.1. Cubic Polynomial Trajectory Planning

According to (1), the flexible motion part of the dynamic equation of the underwater flexible manipulator can be expressed as

$$\boldsymbol{M}_{qq}\ddot{\boldsymbol{q}} + \boldsymbol{K}_q \boldsymbol{q} + \boldsymbol{C}_f - \boldsymbol{F}_q = -\boldsymbol{M}_{q\theta}\ddot{\boldsymbol{\theta}} \tag{2}$$

From (2), it can be known that the inertia moment generated during the rotation of the flexible manipulator arouses the elastic vibration of the manipulator, which can be controlled by optimizing the trajectory of the manipulator during the working process to suppress the vibration of the manipulator. In order to improve the optimization efficiency and the search speed, the basic displacement value of the trajectory control point can be obtained by discretely using a reference trajectory curve. In this paper, a quintic polynomial is used as the reference trajectory curve; it is expressed as

$$\theta_d(t) = \left[6(\frac{t}{t_f})^5 - 15(\frac{t}{t_f})^4 + 10(\frac{t}{t_f})^3 \right] \times \left(\theta_f - \theta_0 \right) + \theta_0 \tag{3}$$

where t_f is the ending time; θ_0 and θ_f are the positions of the start and end times of the manipulator, respectively.

After obtaining the reference curve, sufficient nodes are taken in the reference trajectory curve, and then some floating changes to the basic value are made. By adding the interpolation point increment to change the position of each interpolation point, a new set of interpolation point values are obtained. Then, different trajectory curves are obtained by curve fitting. As shown in Figure 3, the dotted line represents the initial trajectory, and the solid line is the optimization trajectory. Since the speed and acceleration of the underwater manipulator are limited by the rated power and working environment, the speed and acceleration constraints need to be satisfied in the optimization process, and the cubic polynomial function has second-order derivability at both the interpolation point and the interpolation interval; it has good stability and it is easy to calculate the velocity and acceleration. Therefore, after the trajectory control points are obtained, the cubic polynomial interpolation function is used to fit the obtained two adjacent control points.

The joint angle at the i-th trajectory control point can be expressed in the following form:

$$\theta_{Bi} = \begin{cases} \theta_0, & i = 1 \\ \widetilde{\theta}_{Bi} + \Delta\theta_{i-1}, & i = 3, 4 \ldots n-2 \\ \theta_f, & i = n \end{cases} \tag{4}$$

where $\widetilde{\theta}_{Bi}$ is the basic value of the trajectory control point and $\Delta\theta_{i-1}$ is its floating value.

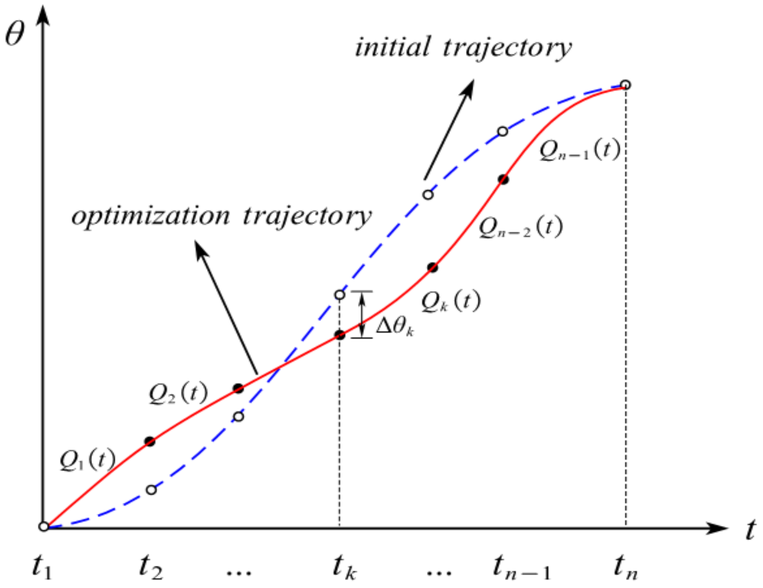

Figure 3. Schematic diagram of trajectory optimization.

Let t_i be the corresponding time series at node i, and $Q_i(t)$ is a cubic polynomial over the time interval $[t_i, t_{i+1}]$. Since the second derivative of $Q_i(t)$ is a linear function, it can be expressed as

$$Q_i{}''(t) = \frac{t_{i+1} - t}{h_i} Q_i{}''(t_i) + \frac{t - t_i}{h_i} Q_i{}''(t_{i+1}), i = 1, 2, \ldots, n-1 \tag{5}$$

where $h_i = t_{i+1} - t_i$, and the interpolation function $Q_i(t)$ can be obtained by integrating $Q_i{}''(t)$ twice and substituting the node conditions

$$\begin{aligned} Q_i(t) = {} & \frac{Q_i{}''(t_i)}{6h_i}(t_{i+1} - t)^3 + \frac{Q_i{}''(t_{i+1})}{6h_i}(t - t_i)^3 + \left[\frac{\theta_{B(i+1)}}{h_i} - \frac{h_i Q_i{}''(t_{i+1})}{6}\right](t - t_i) \\ & + \left[\frac{\theta_{Bi}}{h_i} - \frac{h_i Q_i{}''(t_i)}{6}\right](t_{i+1} - t), i = 1, 2, \ldots, n-1 \end{aligned} \tag{6}$$

In order to determine the undetermined coefficient of the cubic fitting function, each selected interpolation point should satisfy the joint trajectory function over the interpolation point, the joint trajectory function continuity, the joint trajectory first-order derivative function continuity and the joint trajectory second-order derivative function continuity. Coupled with the upper boundary conditions, the following equations in matrix form can be obtained [17]:

$$\boldsymbol{D}\begin{bmatrix} Q_2{}''(t_2) \\ Q_3{}''(t_3) \\ \vdots \\ Q_{n-1}{}''(t_{n-1}) \end{bmatrix} = \boldsymbol{b} \tag{7}$$

In this paper, the trajectory planning adopts uniform interpolation and $h_i = h$. $\boldsymbol{D}$ is the matrix describing the continuous relationship of the interpolation curve, which can be expressed as

$$\boldsymbol{D} = \begin{bmatrix} 6h & h & & & & & \\ 0 & 4h & h & & & & \mathrm{O} \\ & h & 4h & h & & & \\ & & & \vdots & & & \\ & & & & h & 4h & 0 \\ \mathrm{O} & & & & & h & 6h \end{bmatrix} \tag{8}$$

The matrix $\boldsymbol{b}$ can be expressed as

$$\boldsymbol{b}=\begin{bmatrix} \frac{6}{h^2}\left(-\theta_0+\theta_{B3}-2h\omega_0-\frac{5h^2}{6}a_0\right) \\ \frac{6}{h^2}\left(\theta_0+h\omega_0+\frac{h^2}{3}a_0+\theta_{B4}-2\theta_{B3}\right) \\ \frac{6}{h^2}\left(\theta_{B(i+1)}+\theta_{B(i-1)}-2\theta_{Bi}\right) \\ \vdots \\ \frac{6}{h^2}\left(\theta_f-h\omega_f+\frac{h^2}{3}a_f-2\theta_{B(n-2)}+\theta_{B(n-3)}\right) \\ \frac{6}{h^2}\left(-\theta_f+h\omega_f-\frac{5h^2}{6}a_f+\theta_{B(n-2)}\right) \end{bmatrix} \tag{9}$$

where ω_0 and ω_f are the angular velocities at the start time and the end time, and a_0 and a_f are the angular acceleration at the start time and the end time.

In (8), the matrix is an upper triangular matrix, and each element on the diagonal is greater than zero; thus, this is a non-singular matrix. Therefore, the unique solution of $[Q_2{}''(t_2),\ \cdots,\ Q_i{}''(t_i),\ \cdots,\ Q_{n-2}{}''(t_{n-2})]$ can be calculated. By substituting the solution into (21), the values of trajectory control point 2 and point n-2 can be obtained.

$$\theta_{Bi}=\begin{cases} \theta_0+h\times\omega_0+\frac{h^2}{3}a_0+\frac{h^2}{6}Q_2{}''(t_2),\ i=2 \\ \theta_f-h\times\omega_f+\frac{h^2}{3}a_f+\frac{h^2}{6}Q_{n-1}{}''(t_{n-1}),\ i=n-1 \end{cases} \tag{10}$$

The final optimal trajectory is obtained by combining (4), (6) and (10).

3.2. Constraint Condition

When the underwater manipulator is working, the angular velocity and angular acceleration are limited by the rated power and working environment. The joint angular velocity and angular acceleration constraints of the *i*-th trajectory control point of joint j can be expressed as

$$\left|Q'_{ji}(t)\right|\le\omega C_j\ ,\ \ i=1,2,\ldots,n-1 \tag{11}$$

$$\left|Q''_{ji}(t)\right|\le aC_j\ ,i=1,2,\ldots,n-1 \tag{12}$$

where ωC_j is the velocity constraint for joint j, aC_j is the acceleration constraint for joint j, and the joint angular velocity is a quadratic polynomial. The maximum of the absolute value of the angular velocity exists at the endpoint t_i, t_{i+1} or extreme point $\overline{t_i}$. The velocity constraint can be expressed as

$$\max\left|Q'_{ji}(t)\right|=\max\left\{\left|Q'_{ji}(t_i)\right|\left|Q'_{ji}(t_{i+1})\right|\left|Q'_{ji}(\overline{t_i})\right|\right\}\le\omega C_j \tag{13}$$

where

$$\left|Q'_{ji}(t_i)\right|=\left|\frac{a_{ji}}{2}h+\frac{\left(\theta_{B(j,i+1)}-\theta_{B(j,i)}\right)}{h}+\frac{\left(a_{ji}-a_{j,i+1}\right)h}{6}\right| \tag{14}$$

$$\left|Q'_{ji}(t_{i+1})\right|=\left|\frac{a_{j,i+1}}{2}h+\frac{\left(\theta_{B(j,i+1)}-\theta_{B(j,i)}\right)}{h}+\frac{\left(a_{ji}-a_{j,i+1}\right)h}{6}\right| \tag{15}$$

From the condition of the extreme point that $\left|Q''_{ji}(\overline{t_i})\right|=0$, we can derive that

$$|Q'_{ji}(\overline{t_i})|=\begin{cases} \left|\frac{a_{ji}a_{j,i+1}}{2\left(a_{ji}-a_{j,i+1}\right)}h+\frac{\left(\theta_{B(j,i+1)}-\theta_{B(j,i)}\right)}{h}+\frac{\left(a_{ji}-a_{j,i+1}\right)h}{6}\right|, & if\ \ a_{ji}\neq a_{j,i+1}\ \ and\ \ Q''_{ji}(t_i)Q''_{ji}(t_{i+1})\le 0 \\ 0 & if\ \ a_{ji}=a_{j,i+1}\ \ or\ \ Q''_{ji}(t_i)Q''_{ji}(t_{i+1})>0 \end{cases} \tag{16}$$

Since the joint angular acceleration is a first-order polynomial, and the maximum value of the instantaneous velocity can only be obtained at the endpoint, the joint angular acceleration constraint can be expressed as

$$\max\left|Q''_{ji}(t)\right| = \max\{|a_{j1}|, |a_{j2}|, \ldots, |a_{jn}|\} \leq aC_j \tag{17}$$

In addition, in order to improve the safety performance of the flexible manipulator, it is necessary to limit the amplitude of the manipulator, and the vibration constraint can be expressed as

$$w = |\phi_1(L_2)q_1(t) + \phi_2(L_2)q_2(t)| \leq w_{\max} \tag{18}$$

3.3. Objective Function

In order to minimize the vibration of the flexible manipulator, the objective function of vibration suppression trajectory planning based on the accumulation of vibration displacement at the end of the flexible manipulator is proposed:

$$f(\theta) = \alpha_1 \int_0^{t_f} w(\theta, t)dt + \alpha_2 \int_{t_f}^{3t_f} w(\theta, t)dt \tag{19}$$

The first half of the objective function is used to measure the vibration deviation of the flexible manipulator during the movement process. The latter part of the objective function is used to measure the residual vibration of the flexible manipulator in time interval t_f to $3t_f$ after the motion stops. α_1 and α_2 are weight factors.

4. Optimization Algorithm

4.1. PSO Algorithm

For a d-dimensional optimization problem, suppose that the position and velocity of the particle are $\boldsymbol{X}_i = (x_{i,1}\ x_{i,2}\ \ldots\ x_{i,d})$, $\boldsymbol{V}_i = (v_{i,1}\ v_{i,2}\ \ldots\ v_{i,d})$. The particle individual optimal solution is recorded as $\boldsymbol{P}_i = (p_{i,1}\ p_{i,2}\ \ldots\ p_{i,d})$, and the global optimal solution currently found from the whole group is recorded as $\boldsymbol{P}_g = (p_{g,1}\ p_{g,2}\ \ldots\ p_{g,d})$. The particles update their position and velocity according to the following rules [35]:

$$\begin{cases} v_{i,j}(t+1) = \varpi v_{i,j}(t) + c_1\varsigma_1[p_{i,j} - x_{i,j}(t)] + c_2\varsigma_2[p_{g,j} - x_{i,j}(t)] \\ x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \end{cases} \tag{20}$$

where ϖ is the inertia weight, c_1 and c_2 are positive learning factors, and ς_1, ς_2 are random numbers evenly distributed from 0 to 1.

The speed update formula of a PSO algorithm with a shrinkage factor is [36]

$$\begin{cases} v_{i,j}(t+1) = \varphi\{\varpi v_{i,j}(t) + c_1\varsigma_1[p_{i,j} - x_{i,j}(t)] + c_2\varsigma_2[p_{g,j} - x_{i,j}(t)]\} \\ \varphi = \frac{2}{|2-C-\sqrt{C^2-4C}|} \end{cases} \tag{21}$$

where $C = c_1 + c_2$ to ensure the smooth solution of the algorithm.

PSO is an optimization method without variable constraints. For the application of the PSO algorithm in constrained optimization problems, researchers have proposed a series of schemes to solve constrained optimization problems. Among them, the penalty function method is the most widely used because of its simple design and stable calculation results [23,24]. In this paper, the penalty function method is introduced to expand the objective function to deal with the constraints. The new objective function is expressed as

$$F(\theta) = f(\theta) + K_p \sum_{i=1}^{n} [\min(0, g_i(x))]^2 \tag{22}$$

where K_p is the penalty factor, and $g_i(x)$ is the constraint discriminant. When the constraint is satisfied, $g_i(x) = 1$; otherwise, $g_i(x) = -1$.

The steps of the particle swarm optimization algorithm for vibration suppression trajectory planning are as follows. Firstly, when the trajectory of the end actuator of the manipulator is known, the joint motion trajectory can be obtained through inverse kinematics, as shown in the dashed blue frame in Figure 4. The initial value of the joint interpolation point increment is generated by tent chaotic mapping, and the value of the discrete trajectory control point is obtained by (4), (6) and (10). Secondly, the discrete trajectory control points are fitted segmentally by using a cubic polynomial to obtain the angular displacement, angular velocity and angular acceleration of the trajectory to be optimized. Next, the dynamic equation of the system is calculated through (2) to obtain the vibration mode of the flexible manipulator. Then, the objective function is calculated according to (22) and the speed is updated according to (21) until the maximum iterative steps are reached. Finally, the optimal variables are output and the optimal trajectory is obtained by cubic polynomial fitting. The entire process is shown in Figure 4.

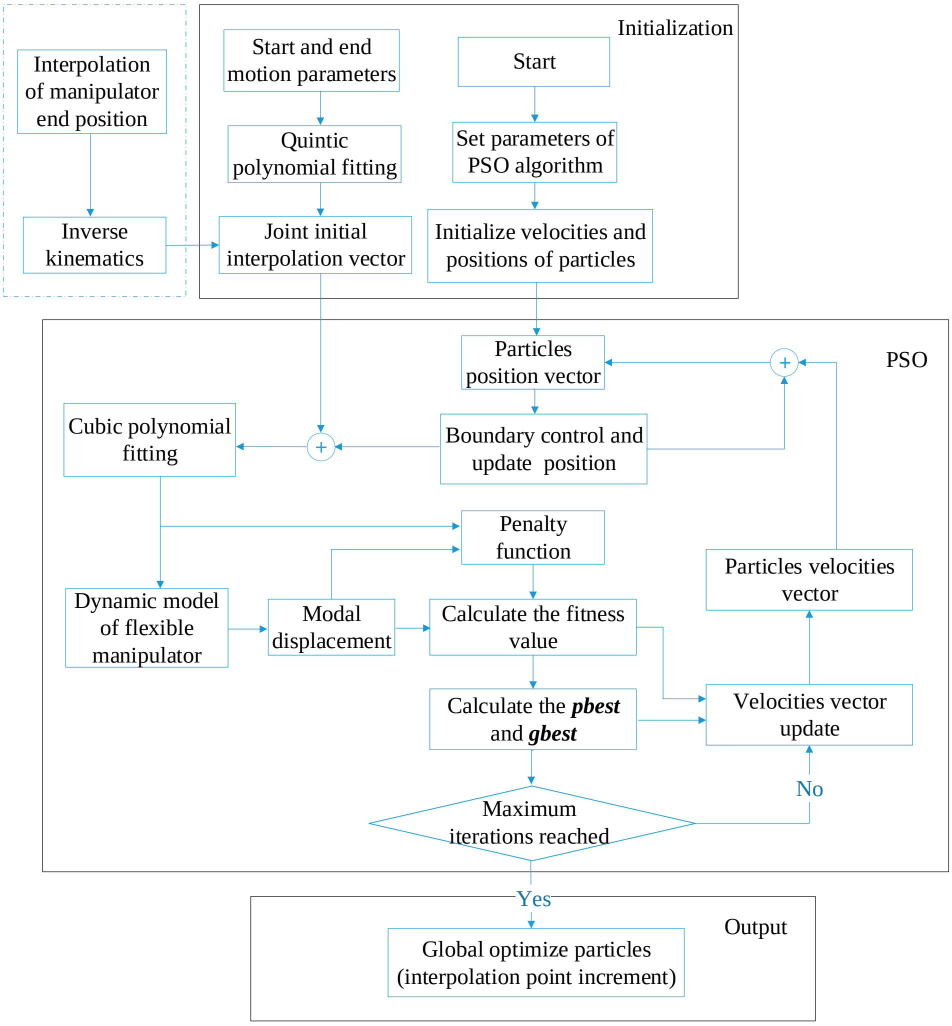
Interpolation of manipulator end position
Inverse kinematics
Initialization
Start and end motion parameters
Quintic polynomial fitting
Joint initial interpolation vector
Start
Set parameters of PSO algorithm
Initialize velocities and positions of particles
PSO
Particles position vector
Boundary control and update position
Cubic polynomial fitting
+
+
Penalty function
Particles velocities vector
Dynamic model of flexible manipulator
Modal displacement
Calculate the fitness value
Calculate the pbest and gbest
Velocities vector update
No
Maximum iterations reached
Yes
Output
Global optimize particles (interpolation point increment)

Figure 4. Vibration suppression trajectory optimization by PSO.

4.2. Sparrow Search Algorithm (SSA)

SSA is a new heuristic optimization algorithm, the position update strategy of the sparrow algorithm is richer than that of PSO and it has better optimization ability in benchmark functions and certain applications [28]. There are two types of sparrows in sparrow algorithms, namely the discoverer and follower, and the proportion of discoverers in the population is set as *PD*. Discoverers with better fitness values will give priority to obtaining food in the search process and providing the direction of foraging for the followers. Therefore, the foraging search range of the discoverer is larger than that of the followers. For a d-dimensional optimization problem, the position information of the *i*-th sparrow can be expressed as $\boldsymbol{X}_i = \begin{bmatrix} x_{i1} & x_{i2} & \ldots & x_{id} \end{bmatrix}$. During each iteration, the discoverer updates the position according to the following rules:

$$\boldsymbol{X}_i^{(t+1)} = \begin{cases} \boldsymbol{X}_i^{(t)} \cdot exp(-\frac{i}{\gamma \cdot itermax}), & if\ R_e < ST \\ \boldsymbol{X}_i^{(t)} + \Psi \cdot \mathbf{L}, & if\ R_e \geq ST \end{cases} \tag{23}$$

where *itermax* is the maximum number of iterations, $\gamma \in [0,1]$ is a random number, $R_e \in [0,1]$ represents the early warning value, $ST \in [0.5,1]$ represents the safety value, Ψ is a random number subject to a normal distribution, and $\mathbf{L}$ stands for a 1 × d-dimensional matrix in which each element is l. When $R_e < ST$, there is no predator in the foraging environment, and the discoverer can conduct an extensive search operation; when $R_e \geq ST$, some sparrows spot predators and send an early warning to others, and all sparrows need to fly to a safe area.

During foraging, some followers will always monitor the discoverer. Once they realize that the discoverer has found better food, they will leave to compete with the discoverer. If they win, they can immediately obtain the food of the discoverer and execute the rule as the discoverer (23); otherwise, they continue to execute the rule (24). The position update of participants is described as follows:

$$\boldsymbol{X}_i^{(t+1)} = \begin{cases} \Psi \cdot exp(\frac{\boldsymbol{X}_w - \boldsymbol{X}_i^{(t)}}{i^2}), & if\ \ i > n/2 \\ \boldsymbol{X}_p + \left| \boldsymbol{X}_i^{(t)} - \boldsymbol{X}_p \right| \cdot \boldsymbol{E}^+ \cdot \mathbf{L}, & if\ \ i \leq n/2 \end{cases} \tag{24}$$

where $\boldsymbol{X}_p$ is the best position of the current discoverer, and $\boldsymbol{X}_w$ is the worst position in the current global situation. $\boldsymbol{E}$ stands for a 1 × d-dimensional matrix in which each element is randomly assigned 1 or -1 and $\boldsymbol{E}^+ = \boldsymbol{E}^T \left(\boldsymbol{E}\boldsymbol{E}^T \right)^{-1}$. When $i > n/2$, it indicates that the *i*-th follower has a low fitness value, and it needs to fly to other places to obtain more energy.

In addition, the sparrow algorithm assumes that some sparrows in the population perceive danger. The proportion of these sparrows is set as *SD*, and their initial positions are randomly generated in the population. If the individual fitness value of these sparrows is greater than the current global optimal fitness value, it means that the sparrow is on the verge of being attacked by predators. When the individual fitness value of these sparrows is equal to the current global optimal fitness value, it means that the sparrow is aware of the danger and needs to be close to other sparrows to minimize the risk of predation. It can be expressed as follows:

$$\boldsymbol{X}_i^{(t+1)} \begin{cases} \boldsymbol{X}_g^{(t)} + \beta \cdot \left| \boldsymbol{X}_i^{(t)} - \boldsymbol{X}_g^{(t)} \right|, & if f_i > f_g \\ \boldsymbol{X}_i^{(t)} + K_r \cdot \left(\frac{\left| \boldsymbol{X}_i^{(t)} - \boldsymbol{X}_w \right|}{(f_i - f_w) + \varepsilon} \right), & if f_i = f_g \end{cases} \tag{25}$$

where $\boldsymbol{X}_g$ is the current global optimal location, β is the step control parameter, $K_r \in [-1\ ,1]$ is a random number indicating the moving direction and step size, and ε is a small constant. f_i is the fitness value of the current sparrow; f_g and f_w are, respectively, the current global best and worst fitness values.

4.3. Improved Sparrow Search Algorithm (ISSA)

In order to improve the global search ability of the sparrow optimization algorithm and reduce the risk of falling into local optima, this paper integrates the sine cosine algorithm (SCA) and elite opposition learning strategy into the sparrow algorithm, and proposes a new, improved sparrow algorithm. The SCA algorithm has a simple structure, few parameter settings and easy implementation, offering certain advantages over PSO and GA in some optimization examples. SCA carries out global exploration and local development according to the oscillation change of the sine cosine model. The position update formula of the standard SCA is as follows [37]:

$$\boldsymbol{X}_i^{(t+1)} = \begin{cases} \boldsymbol{X}_i^{(t)} + \varsigma_3 \times \sin(\varsigma_4) \times \left|\varsigma_5 \boldsymbol{X}_{best} - \boldsymbol{X}_i^{(t)}\right| & \varsigma_6 < P_r \\ \boldsymbol{X}_i^{(t)} + \varsigma_3 \times \cos(\varsigma_4) \times \left|\varsigma_5 \boldsymbol{X}_{best} - \boldsymbol{X}_i^{(t)}\right| & \varsigma_6 \geq P_r \end{cases} \tag{26}$$

where $\boldsymbol{X}_{best}$ is the best individual of the group after the t-th iteration; ς_4, ς_5 and ς_6 are random numbers, and $\varsigma_4 \in [0, 2\pi]$, $\varsigma_5 \in [-2, 2]$, $\varsigma_6 \in [0, 1]$. P_r is a constant indicating the probability of switching between sine and cosine. ς_3 is the conversion parameter.

$$\varsigma_3 = a_r e^{-\frac{t}{itermax}} \tag{27}$$

where a_r is a preset constant, t is the current iteration, and $itermax$ is the maximum number of iterations.

The improved sparrow algorithm introduces the sine algorithm into the position update formula of the discoverer of the standard sparrow search algorithm; it is described as follows:

$$\boldsymbol{X}_i^{(t+1)} = \begin{cases} \boldsymbol{X}_i^{(t)} + \varsigma_3 \times \sin(\varsigma_4) \times \left|\varsigma_5 \boldsymbol{X}_{best} - \boldsymbol{X}_i^{(t)}\right| \ , & if\ R_e < ST \\ \boldsymbol{X}_i^{(t)} + \Psi \cdot \mathbf{L}, \quad if\ R_e \geq ST \end{cases} \tag{28}$$

The elite opposition-based learning strategy seeks to find the reverse solution for some elite individuals, and then liberate the reverse solution of elite individuals into the population to participate in competition [38]. It is proven that this method can effectively improve the search ability of intelligent optimization algorithms. The improved sparrow algorithm uses the inverse solution of elite points to further update the sparrow position after sparrow optimization. The elite group is represented by $\boldsymbol{X}_e$. The proportion of elite groups is ED. $\boldsymbol{X}_{ei}$= $(x_{ei,1}, x_{ei,2}, \ldots, x_{ei,d}) \in [\boldsymbol{\ell}, \boldsymbol{u}]$ is an elite point in d-dimensional space, and its inverse solution $\boldsymbol{X}'_{ei} = \left(x'_{i,1}, x'_{i,2}, \ldots, x'_{i,d}\right)$ can be expressed as

$$\boldsymbol{X}'_{ei} = k \cdot (\boldsymbol{\ell} + \boldsymbol{u}) - \boldsymbol{X}_{ei} \tag{29}$$

where k is a constant. When $f(\boldsymbol{X}_{ei}) > f(\boldsymbol{X}'_{ei})$, we take $\boldsymbol{X}'_{ei}$ as the elite individual of the next iteration to replace $\boldsymbol{X}_{ei}$. Then, we calculate the global optimal position of the group and rearrange the new sparrow population.

The steps of trajectory optimization of flexible manipulators by using ISSA are as follows. Firstly, the initial value is generated by tent chaotic mapping, and then the objective function is calculated according to (19). The positions of the discoverer, follower and early warning sparrow are updated, respectively, according to (28), (24) and (25), and then the position of the sparrow is further updated through the opposition-based learning of the elite solution, until the optimization variable that minimizes the objective function of the flexible manipulator is found. Finally, the optimal variables are output and the optimal trajectory is obtained by cubic polynomial fitting. The entire process is shown in Figure 5.

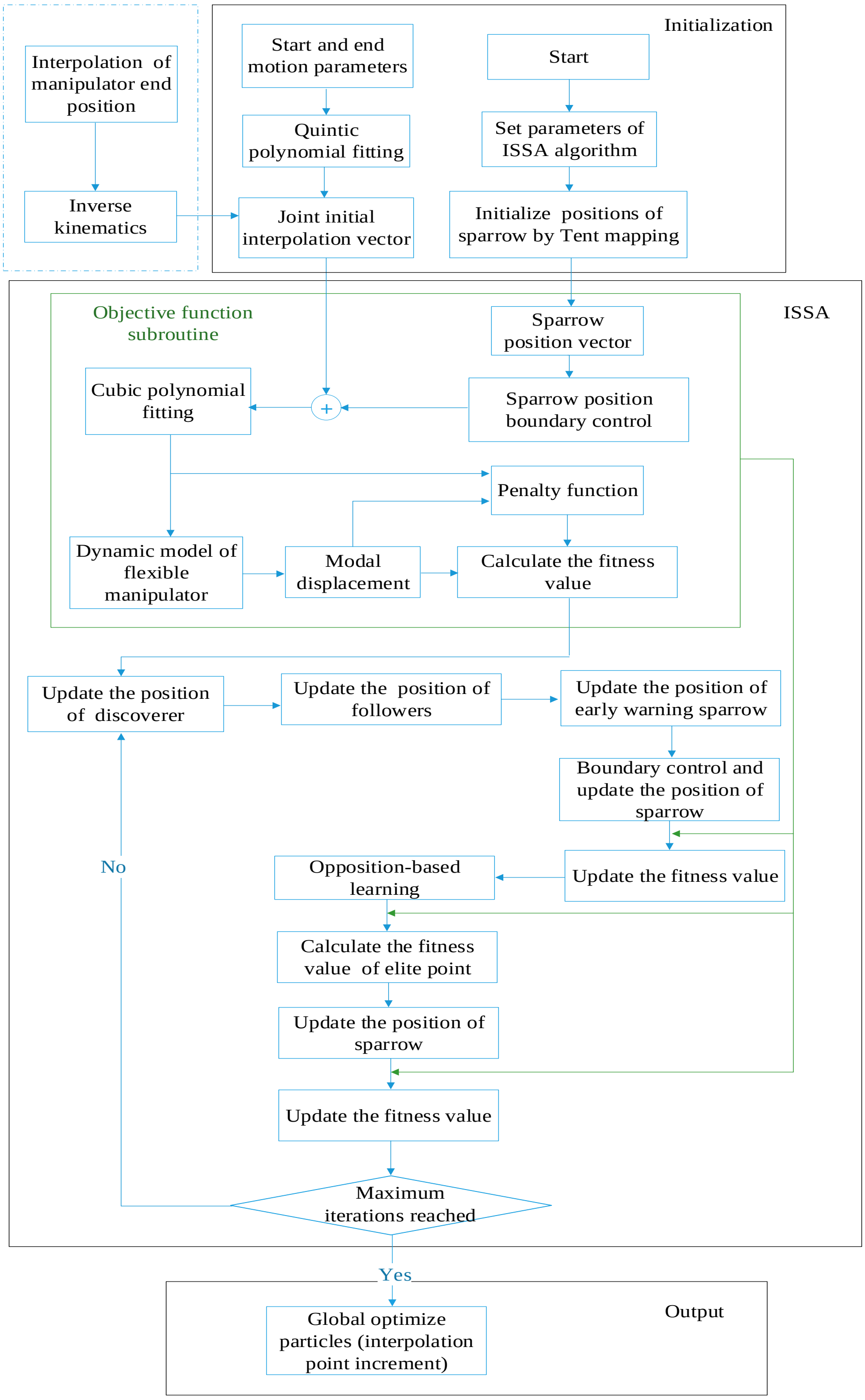
Interpolation of manipulator end position
Inverse kinematics
Initialization
Start and end motion parameters
Quintic polynomial fitting
Joint initial interpolation vector
Start
Set parameters of ISSA algorithm
Initialize positions of sparrow by Tent mapping
ISSA
Objective function subroutine
Sparrow position vector
Cubic polynomial fitting
+
Sparrow position boundary control
Penalty function
Dynamic model of flexible manipulator
Modal displacement
Calculate the fitness value
Update the position of discoverer
Update the position of followers
Update the position of early warning sparrow
Boundary control and update the position of sparrow
No
Opposition-based learning
Update the fitness value
Calculate the fitness value of elite point
Update the position of sparrow
Update the fitness value
Maximum iterations reached
Yes
Global optimize particles (interpolation point increment)
Output

Figure 5. Vibration suppression trajectory optimization by ISSA.

4.4. Simulation

To verify the effectiveness of the proposed ISSA in vibration suppression trajectory optimization, the flexible manipulator shown in Figure 2 is used as an example for optimization simulation, and compared with the traditional PSO algorithm, SSA and OBLSSA. The simulation is carried out in the Matlab2020 environment, using the 10-core parallel operation provided by the National Supercomputing Center in Jinan, China. The size of the manipulators in the simulation is $l_1 = 0.975$ m, $l_2 = 1$ m; the density is $\rho_1 = 13.507$, $\rho_2 = 12.363$; the bending stiffness of the flexible link is $EI = 125\ \mathrm{N \cdot m^2}$. The parameters of the manipulator are shown in Table 1. In the selection of the initial parameters, we refer to some existing methods [19,26] and study the sensitivity of some important parameters. The selected parameters can ensure that the 4 optimization algorithms have good performance.

Table 1. Parameters of underwater flexible manipulator.

Parameter	Symbol	Value
Length (m)	l	$l_1 = 0.975,\ l_2 = 1$
Density (kg·m^{-1})	ρ	$\rho_1 = 13.507,\ \rho_2 = 12.363$
Bending stiffness (N·m^2)	EI	125
Water resistance coefficient	C_D	1.1
Additional mass coefficient	C_M	1

4.4.1. Trajectory Planning with Different Gravity Conditions

Firstly, the trajectory optimization effects of the traditional PSO, SSA, OBLSSA and ISSA for the underwater flexible manipulator are studied under two conditions: considering the influence of gravity (the resultant force of gravity and buoyancy is 0.1 times that of gravity) and not considering the influence of gravity (the resultant force of gravity and buoyancy is zero). In this case, the accelerations at the initial and termination moments are known and the accelerations are zero. The starting position is set as $\boldsymbol{\theta}_0 = [0{,}0]$, and the target position is set as $\boldsymbol{\theta}_f = [\pi/4, \pi/2]$; the weight coefficient of objective function $\alpha_2 = 0$, seven groups of time points are evenly selected, and the optimization parameters are set as $[\Delta\theta_{13},\ \Delta\theta_{14},\ \Delta\theta_{15},\ \Delta\theta_{23},\ \Delta\theta_{24},\ \ \Delta\theta_{25}]$. The dynamic equation of the flexible manipulator is solved by the fourth-order Runge–Kutta method, and the number of data points is taken as *Num* = 500. The corresponding sampling step is taken as $2\pi/499$; the joint angle, velocity and acceleration constraints are $|\theta|_{\max} = \left|\dot{\theta}\right|_{\max} = \left|\ddot{\theta}\right|_{\max} = 2$; the maximum flexible vibration displacement is 0.1 m. The size of the population is N, and the dimension of the optimization problem is recorded as d; the parameters of several optimization algorithms are shown in Table 2.

Table 2. Optimization algorithm parameters.

Case	Algorithm	Optimization Algorithm Parameters
Case 1 (without gravity)	PSO	N = 30, d = 6, $c1$ = 2.05, $c2$ = 2.05, K = 150, $\varpi = 0.9$, Num = 500
	SSA	N = 30, d = 6, ST = 0.7, PD = 0.7, SD = 0.2, K = 150, Num = 500
	OBLSSA	N = 30, d = 6, ST = 0.7, PD = 0.7, SD = 0.2, ED = 0.2, K = 150, Num = 500
	ISSA	N = 30, d = 6, ST = 0.7, PD = 0.7, SD = 0.2, ED = 0.2, a_r = 2, K = 150, Num = 500
Case 1 (with gravity)	PSO	N = 30, d = 6, $c1$ = 2.05, $c2$ = 2.05, K = 150, $\varpi = 0.9$, Num = 500
	SSA	N = 30, d = 6, ST = 0.7, PD = 0.7, SD = 0.2, K = 150, Num = 500
	OBLSSA	N = 30, d = 6, ST = 0.7, PD = 0.7, SD = 0.2, ED = 0.2, K = 150, Num = 500
	ISSA	N = 30, d = 6, ST = 0.7, PD = 0.7, SD = 0.2, ED = 0.2, a_r = 2, K = 150, Num = 500
Case 2 ($\alpha_2 = 1$)	PSO	N = 30, d = 8, $c1$ = 2.05, $c2$ = 2.05, K = 150, $\varpi = 0.9$, Num = 500
	SSA	N = 30, d = 8, ST = 0.7, PD = 0.7, SD = 0.2, K = 150, Num = 500
	OBLSSA	N = 30, d = 8, ST = 0.7, PD = 0.7, SD = 0.2, ED = 0.2, K = 150, Num = 500
	ISSA	N = 30, d = 8, ST = 0.7, PD = 0.7, SD = 0.2, ED = 0.2, a_r = 2, K = 150, Num = 500
Case 2 ($\alpha_2 = 0$)	PSO	N = 30, d = 8, $c1$ = 2.05, $c2$ = 2.05, K = 150, $\varpi = 0.9$, Num = 500
	SSA	N = 30, d = 8, ST = 0.7, PD = 0.7, SD = 0.2, K = 150, Num = 500
	OBLSSA	N = 30, d = 8, ST = 0.7, PD = 0.7, SD = 0.2, ED = 0.2, K = 150, Num = 500
	ISSA	N = 30, d = 8, ST = 0.7, PD = 0.7, SD = 0.2, ED = 0.2, a_r = 2, K = 150, Num = 500

After iterative operation, an optimal trajectory consisting of three cubic polynomial trajectories can be obtained. Figure 6a shows the evolution process of the four optimization algorithms without considering the influence of gravity, and Figure 6b shows the evolution process of the four optimization algorithms considering the influence of gravity. In order to eliminate the influence of accidental factors, each optimization algorithm is implemented three times to take the average. The optimization results are shown in Table 3. It can be seen from Figure 6a and Table 3 that, compared with the other three algorithms, SSA falls into the local optimum earlier; the OBLSSA optimization results are improved compared with SSA; the optimization result of PSO is better than that of SSA and OBLSSA, but the optimization time is much longer than that of the other three algorithms; ISSA improves the global search ability, and the search accuracy and convergence speed are better than for the other three optimization algorithms. Combined with Figure 6b and Table 3, it can be seen that the optimization effect of the SSA optimization algorithm is better than that of the PSO optimization algorithm. Under the set parameters, the best fitness values of the optimization trajectories when using the OBLSSA optimization algorithm are not improved compared with the use of SSA. This may be due to the increase in random factors in the process of generating elite solutions and controlling the boundaries of elite solutions, which reduces the accuracy of the optimization results. The ISSA optimization algorithm combines the advantages of the strong global search ability of the sine and cosine algorithm and opposition elite learning to avoid falling into local optima, and it overcomes the influence of random factors. Figure 7 shows the optimized trajectory without considering the influence of gravity. It can be seen from Figure 7 that the interpolation point increment of the ISSA-optimized trajectory is much larger than that of the SSA optimization algorithm. Figure 8 is the optimized trajectory considering the influence of gravity. Compared with Figure 7, it can be seen that the curvature of the optimized trajectory when gravity's influence is considered is greater than that when gravity's influence is not considered, which is caused by the inherent vibration characteristics of the manipulator. Figure 9a shows the flexible displacement at the end of the manipulator under each optimized trajectory when the influence of gravity is not considered. Compared with the non-optimized trajectory, the flexible vibration at the end of the manipulator based on the SSA- and ISSA-optimized trajectories is significantly suppressed in the first 4 s, and the total vibration is smaller than that of the non-optimized trajectory. The vibration accumulation and maximum vibration displacement of the manipulator optimized by ISSA are obviously lower than those of the one optimized by SSA. Figure 9b shows the flexible displacement at the end of the manipulator generated by the SSA- and ISSA-optimized trajectories when the influence of gravity is considered. Compared with the non-optimized trajectory, the flexible vibration is effectively suppressed at 1.5–4 s. Due to the large flexible displacement caused by gravity, the vibration displacement reduced by trajectory planning is not obvious compared with the total flexible displacement.

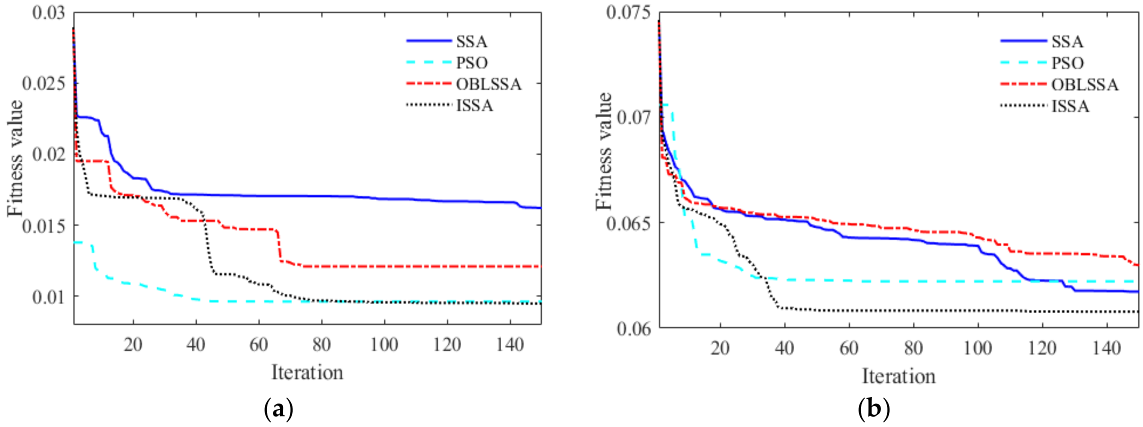

Figure 6. Optimization process of optimization algorithms: (**a**) without gravity effect; (**b**) with gravity effect.

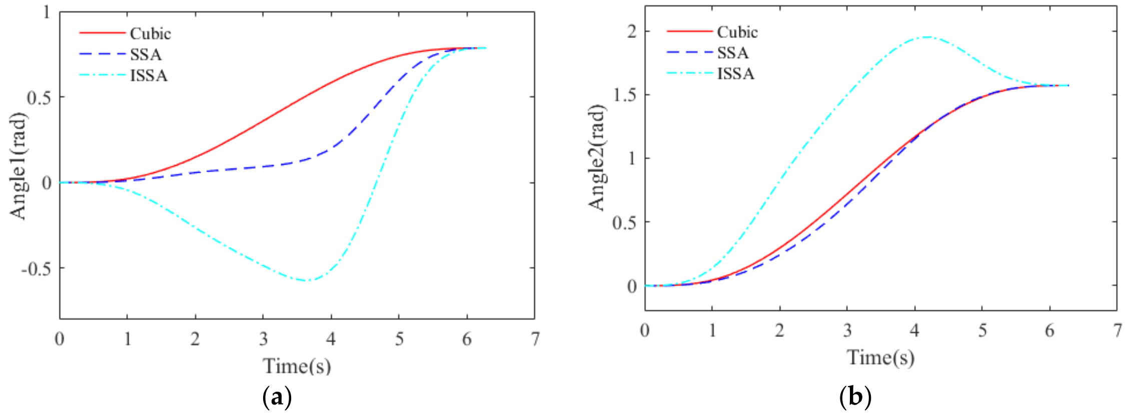

Figure 7. Trajectory of joint without gravity effect: (**a**) joint1; (**b**) joint2.

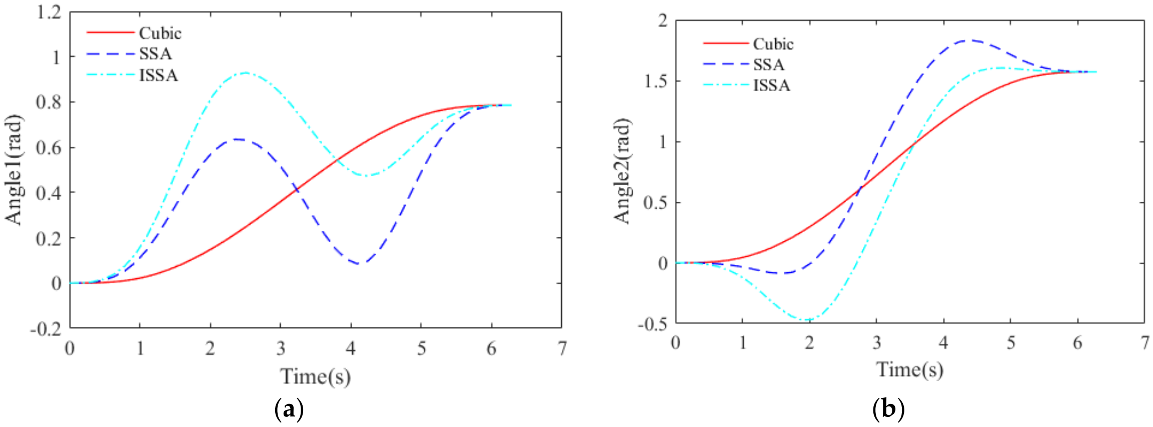

Figure 8. Trajectory of joint with gravity effect: (**a**) joint1; (**b**) joint2.

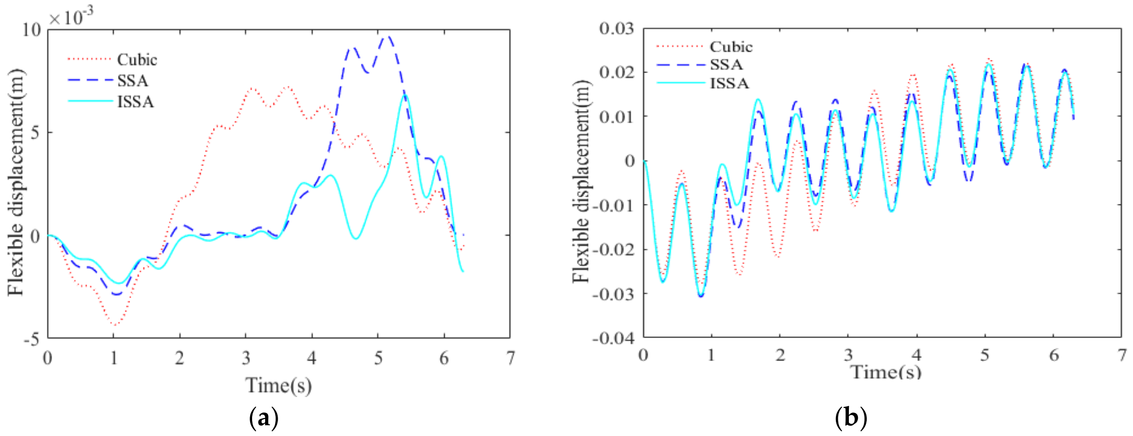

Figure 9. Manipulator endpoint flexible displacement: (**a**) without gravity effect; (**b**) with gravity effect.

Table 3. Optimization results.

Case	Algorithm	Optimal Solution			Running Time/s
		Best	Worst	Mean	
Case 1 (without gravity)	PSO	0.0096	0.0111	0.0101	66,838
	SSA	0.0162	0.0166	0.0163	16,434
	OBLSSA	0.0108	0.0121	0.0110	19,978
	ISSA	0.0095	0.0095	0.0095	17,518
Case 1 (with gravity)	PSO	0.0622	0.0622	0.0622	65,619
	SSA	0.0617	0.0617	0.0617	19,058
	OBLSSA	0.0630	0.0632	0.0631	22,286
	ISSA	0.0608	0.0612	0.0609	18,673
Case 2 ($\alpha_2 = 1$)	PSO	0.0191	0.0191	0.0191	67,459
	SSA	0.0174	0.0174	0.0174	16,893
	OBLSSA	0.0206	0.0208	0.0207	20,974
	ISSA	0.0148	0.0148	0.0148	18,146
Case 2 ($\alpha_2 = 0$)	PSO	0.0217	0.0217	0.0217	82,930
	SSA	0.0137	0.0145	0.0142	15,743
	OBLSSA	0.0132	0.0135	0.0133	19,876
	ISSA	0.0112	0.0113	0.0113	22,495

4.4.2. Trajectory Planning with Different Weight Factors

When the acceleration at the initial and terminal times is not zero and is unknown, the residual vibration will interfere with the control of the manipulator after the joint stops moving. In order to study the influence of residual vibration on the optimization of the vibration suppression trajectory, the paper discuss the trajectory optimization effect of PSO, SSA, OBLSSA and ISSA for underwater flexible manipulators in $\alpha_2 = 1$ and $\alpha_2 = 0$. The starting position is set as $\boldsymbol{\theta}_0 = [0, 0]$, and the target position is set as $\boldsymbol{\theta}_f = [\pi/4, \pi/2]$. The resultant force of gravity and buoyancy is zero. Seven groups of time points are evenly selected, and the joint angle increment at the middle time point and the initial acceleration of joint2 are taken as the parameters to be optimized. The optimization parameters can be expressed as $[\Delta\theta_{13}, \Delta\theta_{14}, \Delta\theta_{15}, \Delta\theta_{23}, \Delta\theta_{24}, \Delta\theta_{25}, \ddot{\theta}_{20}, \ddot{\theta}_{2f}]$; the joint angle, velocity and acceleration constraints are $|\theta|_{\max} = \left|\dot{\theta}\right|_{\max} = \left|\ddot{\theta}\right|_{\max} = 2$; the maximum flexible vibration displacement is 0.1 m.

Figure 10a shows the optimization processes of four optimization algorithms at $\alpha_2 = 1$. Combining these with Table 3, it can be seen that the optimization result of SSA is better than that of the PSO optimization algorithm. The optimization point obtained by the PSO optimization algorithm in the initial stage of optimization exceeds the constraint boundary. Due to the effect of the penalty function, it overcomes the constraints and obtains a good optimization result. The fitness value of the OBLSSA optimization result is lower than that of SSA, which is related to the increase in random factors in the process of generating and controlling the boundary of the elite solution. The ISSA optimization algorithm combines the global search ability of the sine and cosine algorithm and opposition elite learning to avoid falling into the local optimum and overcome the influence of random factors, obtaining better results than other algorithms. Figure 10b shows the optimization processes of four optimization algorithms at $\alpha_2 = 0$, The optimization points obtained by the PSO optimization algorithm in the initial stage of optimization also exceed the constraint boundary. According to Table 3, it can be seen that the optimization results of the three sparrow optimization algorithms are better than those of the PSO. The OBLSSA optimization algorithm reverse elite solution works in overcoming the local optimum. The optimization results are better than those of the SSA optimization algorithm, and ISSA improves the global search ability and avoids falling into the local optimum, and it obtains optimization results that are better than those of other algorithms. By comparing the optimal fitness values obtained by the four optimization algorithms in Table 3, it can be seen that the fitness values of SSA, OBLSSA and ISSA when $\alpha_2 = 1$ are greater than those when $\alpha_2 = 0$. This is because we add the residual vibration after the joint motion stops. The fitness value of the PSO algorithm when $\alpha_2 = 1$ is less than that when $\alpha_2 = 0$; this may be caused by the PSO algorithm falling into the local optimum. Figures 11 and 12, respectively, show the joint optimization trajectories when $\alpha_2 = 1$ and $\alpha_2 = 0$. Figure 13 shows the comparison of the optimization trajectories of ISSA when $\alpha_2 = 1$ and $\alpha_2 = 0$. Compared with the joint optimization trajectory when $\alpha_2 = 0$, the joint optimization trajectory has smaller curvature and a smaller interpolation point increment when $\alpha_2 = 1$. Figure 14 shows the vibration displacement of the manipulator endpoint. It can be seen from Figure 14a that the residual vibration of the trajectory optimized by SSA and ISSA is significantly suppressed at $t = [t_f, 3t_f]$. It can be seen from Figure 14a,b that the vibration suppression is mainly within 1–5 s, and the vibration amount of the optimized trajectory by ISSA is less than that by SSA. Although the amplitude of residual vibration is less than that in the process of motion, the accumulation of residual vibration will also have a great impact on the optimization results.

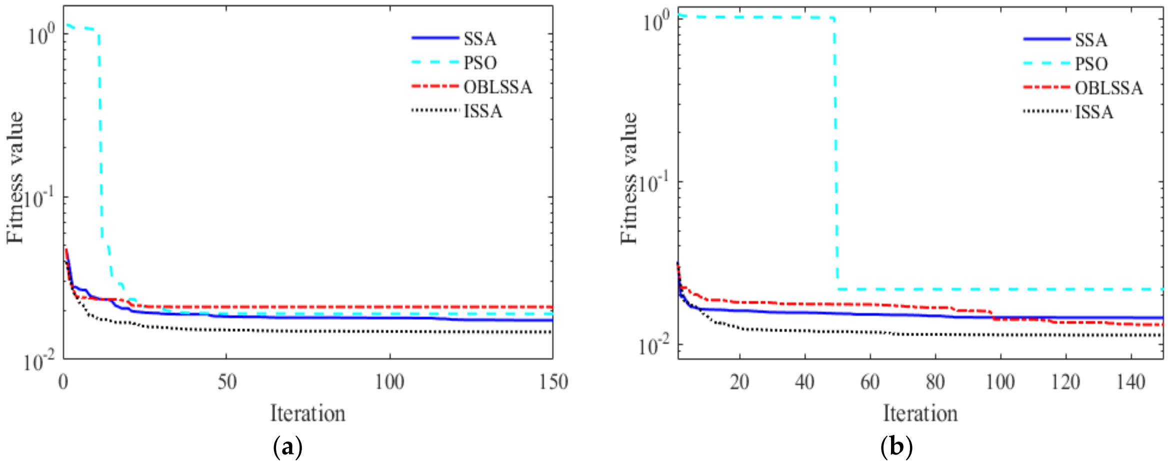

Figure 10. Optimization process of optimization algorithms: (**a**) $\alpha_2 = 1$; (**b**) $\alpha_2 = 0$.

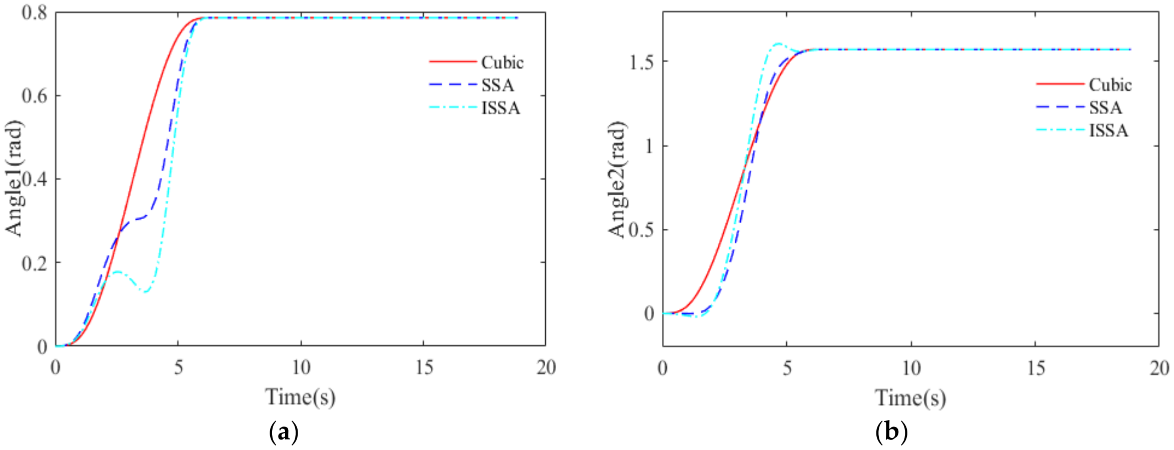

Figure 11. Trajectory of joint at $\alpha_2 = 1$: (**a**) joint1; (**b**) joint2.

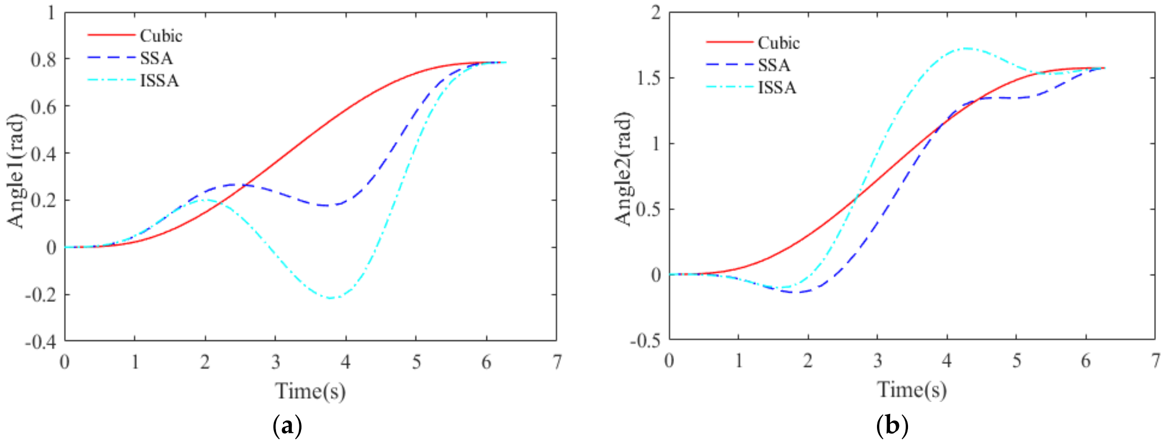

Figure 12. Trajectory of joint at $\alpha_2 = 0$: (**a**) joint1; (**b**) joint2.

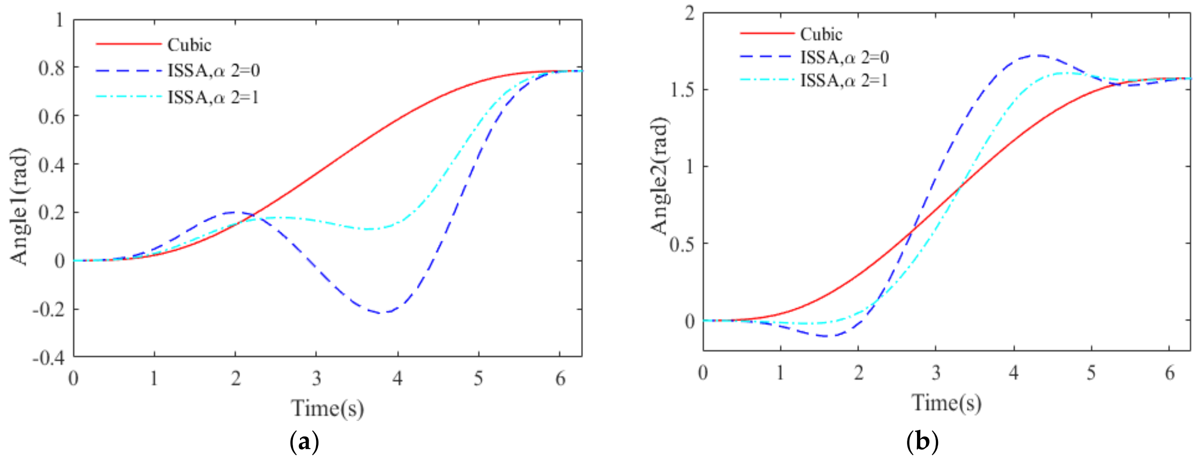

Figure 13. Trajectory of joint at different α_2: (**a**) joint1; (**b**) joint2.

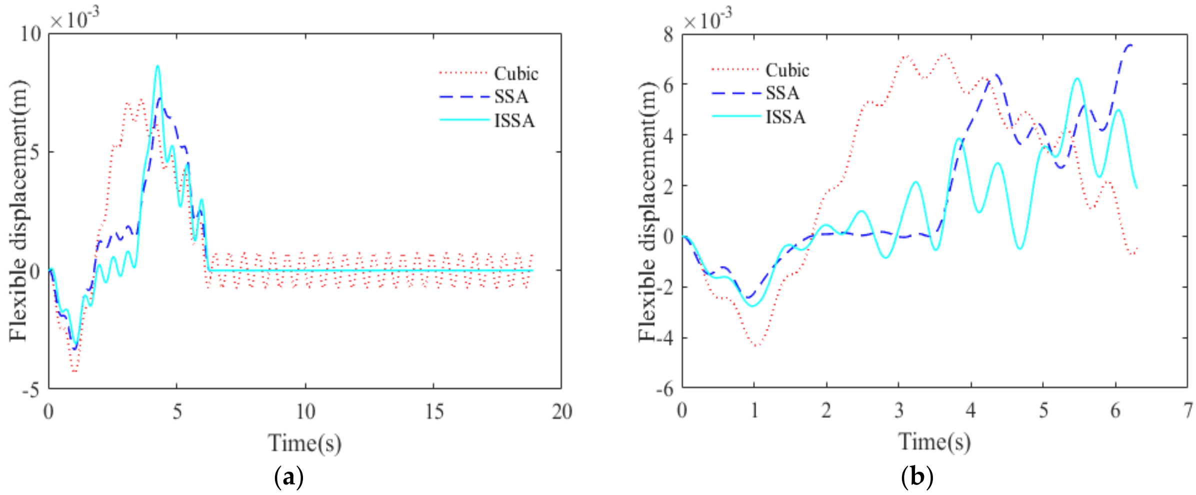

Figure 14. Manipulator endpoint flexible displacement: (**a**) $\alpha_2 = 1$; (**b**) $\alpha_2 = 0$.

Remark 1. *Since the initial value of the joint interpolation point increment is automatically generated by tent chaotic mapping, different optimization algorithms will have different initial fitness values, as shown in Figures 6 and 10. The fitness value of the PSO algorithm in Figure 10 in the early iteration is much higher than that of other optimization algorithms, which is caused by the action of the penalty function.*

Remark 2. *The optimized objective function is the accumulation of vibration displacement in the whole optimization process. In Figure 14b, although the vibration accumulation of SSA is smaller than that of ISSA in the first 4 s, the vibration accumulation of the SSA optimization algorithm increases sharply in the following seconds, resulting in a total vibration accumulation greater than that of ISSA.*

To further analyze the optimization performance of the four optimization algorithms, two optimization performance indicators, the average optimization percentage λ and average relative optimization time κ, are introduced.

$$\lambda = \frac{1}{4}\sum_{i=1}^{4}\frac{F_{n_i} - F_i}{F_{n_i}} \tag{30}$$

$$\kappa = \frac{1}{4}\sum_{i=1}^{4}\frac{t_i}{t_{I_i}} \tag{31}$$

where F_n is the non-optimized objective function value, F is the optimized objective function value, t is the optimization time, and t_I is the optimization time of the ISSA algorithm.

The optimization performance indicators are shown in Table 4. According to Table 4, the ISSA algorithm takes only 1/3.68 of the time of PSO. Compared to PSO, SSA and OBLSSA, the optimization performance is improved by 17.3%, 13.1% and 9.7%, respectively. This is because the PSO algorithm and the SSA algorithm are more likely to fall into the local optimum compared with OBLSSA and ISSA. The introduction of the opposition learning strategy improves the optimization ability of SSA to a certain extent, but random factors are added in the process of generating and controlling the boundary of the elite solution, which will affect the optimization accuracy of OBLSSA. As the application of SA in ISSA overcomes the effects of random factors, compared with the other three optimization algorithms, ISSA has higher optimization accuracy and a faster convergence speed and obtains a good vibration reduction effect.

Table 4. Optimization performance indicators.

Optimization Algorithm	λ	κ
PSO	25.0%	3.68
SSA	29.2%	0.90
OBLSSA	32.6%	1.10
ISSA	42.3%	1.00

5. Incremental Kriging-Assisted ISSA

The numerical simulation in Section 4 verifies that the ISSA-based vibration suppression trajectory planning method can obtain better optimization results. Due to the long optimization time, ISSA is difficult to be directly applied in practice. In order to achieve a fast online search of vibration suppression trajectories, neural networks and intelligent algorithms are used as optimal path generators [19], which can generate vibration suppression trajectories in real time after the start and end positions of the joint angle are given. These methods require a large number of optimization results as sample data. To obtain a large amount of optimized trajectory data in a shorter time, this paper uses an incremental Kriging model based on an improved additive point criterion to construct a global surrogate model, and combines the ISSA optimization algorithm to train the model.

5.1. Incremental Learning Kriging Model

Data-driven technology has great potential in engineering applications [39]. The Kriging model is widely used in engineering optimization because of its good robustness and nonlinear approximation capability [40,41]. The Kriging model introduces statistical assumptions to represent the objective function as a stochastic process as follows.

$$Y(\boldsymbol{x}) = \mu + Z(\boldsymbol{x}) \tag{32}$$

The following Gaussian correlation function is often used for the Kriging model:

$$R(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}, \boldsymbol{\vartheta}) = \exp[-\sum_{i=1}^{d} \vartheta_i (x^{(i)} - x^{(j)})^2] \tag{33}$$

The estimated value of the model is expressed as

$$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{r}^T \boldsymbol{R}^{-1} (\boldsymbol{y} - \mathbf{1}\hat{\mu}) \tag{34}$$

The mean square deviation of the estimated value is expressed as

$$\hat{s}^2(\boldsymbol{x}) = \hat{\sigma}^2 \left[1 - \boldsymbol{r}^T \boldsymbol{R}^{-\mathbf{1}} \boldsymbol{r} + \frac{(1 - \mathbf{1}^T \boldsymbol{R}^{-1} \boldsymbol{r})^2}{\mathbf{1}^T \boldsymbol{R}^{-\mathbf{1}} \mathbf{1}} \right] \tag{35}$$

where $\boldsymbol{\vartheta}$, μ, σ^2 are hyperparameters and $\hat{\boldsymbol{\vartheta}}$, $\hat{\mu}$, $\hat{\sigma}^2$ are the corresponding maximum likelihood estimations.

The training complexity of the Kriging model mainly comes from the optimization of the hyperparameters, which requires solving the correlation matrix $\boldsymbol{R}$ and its inverse. The Cholesky decomposition can improve the accuracy of the numerical solution. The Cholesky decomposition of the correlation matrix $\boldsymbol{R}$ can be expressed as

$$\boldsymbol{R} = \boldsymbol{L}\boldsymbol{L}^T \tag{36}$$

The training and updating of the Kriging model have $O(n^3)$ complexity. Since each new piece of data is a small amount, it is not necessary to completely retrain the current Kriging model, but only to learn the effect of the new data on the current Kriging model. The incremental Kriging model approach proposed by Zhan and Xing can significantly reduce the computational complexity and greatly shorten the surrogate model construction time by chunking the incremental relationship matrix $\widetilde{\boldsymbol{R}}$ with respect to old data and new data [42].

$$\widetilde{\boldsymbol{R}} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{A} \\ \boldsymbol{A}^T & \boldsymbol{B} \end{bmatrix} \tag{37}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$ is the incremental matrix of new data. For the new data with q incremental points, we can obtain

$$\widetilde{\boldsymbol{1}} = [\boldsymbol{1}, 1_1, \cdots, 1_q]^T \tag{38}$$

$$\widetilde{\boldsymbol{y}} = [\boldsymbol{y}, y^{(n+1)}, \cdots, y^{(n+q)}]^T \tag{39}$$

$$\widetilde{\boldsymbol{r}} = [\boldsymbol{r}, R(\boldsymbol{x}, \boldsymbol{x}^{(\mathrm{n+1})}; \hat{\boldsymbol{\vartheta}}), \cdots, R(\boldsymbol{x}, \boldsymbol{x}^{(\mathrm{n+q})}; \hat{\boldsymbol{\vartheta}})]^T \tag{40}$$

$$\widetilde{\boldsymbol{R}} = \widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{L}}^T \tag{41}$$

$$\widetilde{\boldsymbol{L}} = \begin{bmatrix} \boldsymbol{L} & \boldsymbol{0} \\ \boldsymbol{A}^T (\boldsymbol{L}^T)^{-1} & \mathrm{chol}(\boldsymbol{B} - \boldsymbol{A}^T (\boldsymbol{L}^T)^{-1} \boldsymbol{L}^{-1} \boldsymbol{A}) \end{bmatrix} \tag{42}$$

After obtaining the incremental relationship matrix $\boldsymbol{R}$ and the incremental matrices $\boldsymbol{A}$, $\boldsymbol{B}$, the trained Kriging model can be obtained, and then the predicted values and predicted standard deviations of the unknown points can be obtained by the new Kriging model.

5.2. Infill Criteria

At present, the most commonly used infill criteria include the MSP and EI criteria. The MSP criterion aims to find the optimal solution $\boldsymbol{x}^*$ of the objective function directly on the surrogate model, and then obtain the exact solution of $\boldsymbol{x}^*$ as new sample data. The EI criterion finds the point with the highest probability of improvement of the objective function as a new sample point. The probability value of EI improvement can be expressed as

$$\mathrm{EI}(\boldsymbol{x}) = (y_{\min} - \hat{y}(\boldsymbol{x}))\Phi(\frac{y_{\min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}) + \hat{s}(\boldsymbol{x})\varphi(\frac{y_{\min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}) \tag{43}$$

where $y_{\min}$ is the current optimal response function value, Φ is the standard normal cumulative distribution function, and φ is the standard normal probability density function.

Similar methods are used to establish the Kriging model of constraint $g_i(\boldsymbol{x})$, $g_i(\boldsymbol{x}) \in N[\hat{g}_i(\boldsymbol{x}), {s_{gi}}^2(\boldsymbol{x})]$. Then, the probability of satisfying the constraint at any position is

$$P[-\zeta \leq g_i(\boldsymbol{x}) \leq \zeta] = 2\Phi(\frac{\zeta - \hat{g}_i(\boldsymbol{x})}{s_{gi}(\boldsymbol{x})}) - 1 \tag{44}$$

The EI value with constraints is expressed as

$$\mathrm{E_cI}(\boldsymbol{x}) = \mathrm{EI}(\boldsymbol{x}) \cdot \prod_{i=1}^{N_c} P[-\zeta \leq g_i(\boldsymbol{x}) \leq \zeta] \tag{45}$$

Studies have shown that the combination of the MSP criterion and EI criterion has a better optimization effect than a single infill criterion [43]. The IKA-ISSA used in this paper constructs an incremental Kriging model based on the MSP + EI hybrid infill criterion and combines the ISSA optimization algorithm to train the model. The IKA-ISSA optimization process is shown in Algorithm 1.

Algorithm 1. Optimization process of IKA-ISSA.

Parameters: number of sample points *Nt*, maximum number of sample points *Nm*, learning increment step of Kriging model *generation*

```
1. Initial sample set generation: Latin superelevation method to generate N initial design points sample_x. the response value of vibration accumulation sample_y1 and the maximum flexible displacement constraint value sample_y2 obtained by the dynamic model.
2. Generate objective function values: generate objective function y with penalty factors based on response and constraint values and rank y to obtain the optimal value y_min.
3. While Nt < Nm do
4. If generation = = 1
Generating the initial Kriging model: the initial sample set is used to generate the responding Kriging model kriging_model1 and the constrained Kriging model kriging_model2, respectively.
5. Else
Update the Kriging model: generate new Kriging models kriging_model1, kriging_model2 based on the Kriging model from the previous learning increment step generation and the sample set increments infill_x, infill_y1, in fill_y2.
End if
6. ISSA optimization: set the optimal population P, obtain the predicted value of the population sample based on the initial Kriging model and perform K iterations of ISSA optimization to obtain the optimized population P1.
7. Generate candidate increment set: merge similar individuals in population P1 to generate population P2. Remove points in population P2 that are similar to sample set sample_x as candidate increment set P_candi.
8. Generate increment sets: select sample set increments infill_x, infill_y1, in fill_y2 from the candidate increment set P_candi based on MSP + EI infill criterion and Kriging model.
9. Generate objective function values of infill points: obtain the response value infill_y1 and the constraint value in fill_y2 for the sample set increment through the dynamic model and calculate the objective function in fill_y for the increment sample set.
10. Update variables: update the sample sets sample_x, sample_y1, sample_y2;
update Nt and generation;
update the optimal value y_min and the corresponding optimal point x_min.
11. End while
12. Output optimal solution: Output y_min and x_min.
```

5.3. Simulation

Taking the example of case 1 (without gravity), six groups of time points are evenly selected, d $= 4$ and the optimization parameters are set as $[\Delta\theta_{13},\ \Delta\theta_{14},\ \Delta\theta_{23},\ \Delta\theta_{24}]$. The maximum flexible displacement is selected as the constraint, and five groups of different joint angle initial and termination positions $[\theta_{11},\ \theta_{1t},\ \theta_{21},\ \theta_{2t}]$ are used to verify the learning ability of IKA-ISSA under the constraint. The five groups of joint angle initial and termination positions are set as $[0, \pi/4, 0, \pi/2]$, $[0, \pi/2, 0, \pi/4]$, $[0, \pi/4, \pi/6, \pi/2]$, $[\pi/6, \pi/2, 0, \pi/4]$, $[\pi/6, \pi/2, \pi/6, \pi/2]$. In IKA-ISSA, the number of initial sample points $N_K = 100$, the maximum number of samples in the sample set $N_m = 180$, the size of the sparrow population $N = 30$, and the number of sparrow optimization iteration steps $K = 10$. The parameter of normal ISSA is the same as that in case 1.

From Figure 15, it can be seen that the vibration accumulation of the optimized trajectory is significantly smaller than that of the non-optimized trajectory, and the optimized result of IKA-ISSA is slightly different from that of ordinary ISSA after a short learning and training period, and even better than that of ordinary ISSA in some cases. Figure 16 shows the optimization times of the two methods. IKA-ISSA takes less than 3% of the time of ISSA, among which the time to generate the initial sample set accounts for a large proportion, which is caused by the complex dynamic calculation. In addition, the learning and training time of IKA-ISSA is much less than that of ISSA optimization training, which indicates that IKA-ISSA integrates the strong nonlinear approximation ability of the incremental Kriging model and the high search efficiency of ISSA.

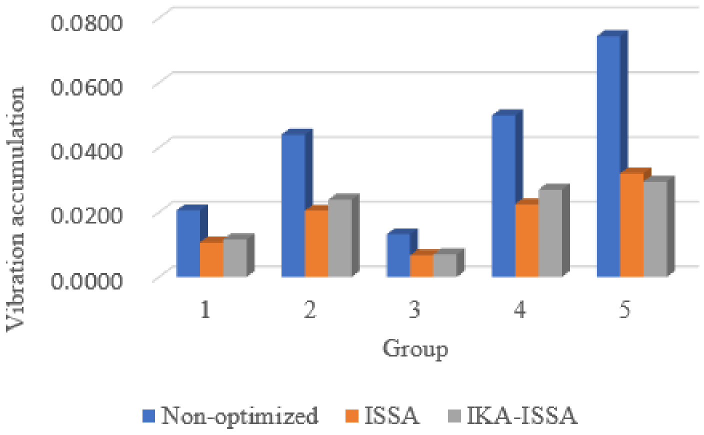

Figure 15. Vibration accumulation of optimized trajectory.

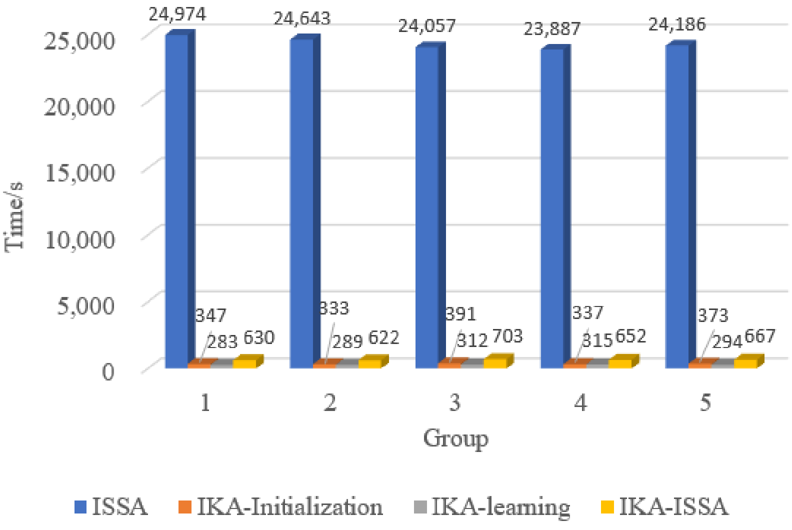

Figure 16. Optimization time of vibration suppression trajectory planning.

6. Conclusions

In this paper, ISSA combining an elite strategy and the sine algorithm is proposed for the trajectory planning of underwater flexible manipulators. The simulation results show that ISSA has the advantages of the global search ability of the sine and cosine algorithm and the reverse elite learning ability to avoid falling into local optima. It is more computationally efficient than the PSO algorithm, and the average time consumption is only 1/3.68 of that of the PSO algorithm. In order to solve the problem of the long optimization time of ISSA, this paper further proposes IKA-ISSA, which integrates the strong nonlinear approximation capability of the incremental Kriging model and the high search efficiency of ISSA to greatly reduce the time required for optimization and ensure the accuracy of optimization. This method can obtain a large number of optimization results in a shorter time to construct real-time optimal path generators.

Author Contributions: Conceptualization, methodology, writing—original draft, writing—review and editing, H.H.; conceptualization, writing—review and editing, funding acquisition, supervision, resources, G.T.; investigation, software, data curation, H.C.; project administration, writing—review and editing, resources, J.W.; writing—review and editing, investigation, validation, L.H.; supervision, investigation, formal analysis, D.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Natural Science Foundation of China (No. 51979116); the HUST Interdisciplinary Innovation Team Project (No. 2020JYCXJJ063); and the Wuhan Science and Technology Project (No. 2020010602012052).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Matlab codes of IKA-ISSA and ISSA can be obtained from the website https://pan.baidu.com/s/1AfesaXXT5hlV8TlgLDQFIA?pwd=zyip (accessed on 27 March 2023) by using the password zyip.

Conflicts of Interest: The authors declare that there are no competing interests.

Nomenclature

A nomenclature table for variables and abbreviations in the paper is provided as follows:

SSA	Sparrow search algorithm	R_e	Early warning value
ISSA	Improved sparrow search algorithm	ST	Safety value
PSO	Particle swarm optimization	SD	The proportion of early warnings
IKA-ISSA	Incremental Kriging-assisted ISSA	PD	The proportion of discoverers
GA	Genetic algorithm	ED	The proportion of elite groups
UUV	Unmanned underwater vehicles	P_r	Sine cosine switching coefficient
AMM	Assumed mode method	X_e	Elite group
$\boldsymbol{\theta}$	Joint angle	K	Optimize iteration steps
$\boldsymbol{q}$	First two-order modes	$c1, c2$	Learning factor
$\boldsymbol{K}_q$	Stiffness matrix	$itermax$	Maximum number of iterations
$\boldsymbol{C}_r, \boldsymbol{C}_f$	Centrifugal forces, Coriolis forces and gravity terms	Ψ	Random number subject to normal distribution
$\boldsymbol{M}_{**}$	Mass matrices, $*$ represents θ or q	$\mathbf{L}$	1 × d-dimensional matrix in which each element is l
$\boldsymbol{F}_\theta, \boldsymbol{F}_q$	Generalized force related to hydrodynamic force	X_p	Best position of the current discoverer
$\boldsymbol{\tau}$	Joint torque	X_w	Worst position in the current global situation
θ_d	Reference joint angle trajectory	X_g	Global optimal position
θ_0	Initial joint angle	β	Step control parameter
θ_f	Joint angle at end time	N	Number of individuals in the population
t_f	End time	d	Optimized dimensionality
θ_{Bi}	Joint angle of the *i*-th trajectory control point	Num	The number of data points for the fourth-order Runge–Kutta model
$\Delta\theta_i$	Floating value of the *i*-th trajectory control point	EI	EI criterion
$\widetilde{\theta}_{Bi}$	Basic value of the *i*-th trajectory control point	$\mathrm{E_cI}$	EI criterion with constraints

Q_i	Cubic polynomial over the time interval $[t_i, t_{i+1}]$	Y	Objective function
ω_0, ω_f	Angular velocities at the start time and the end time	ϑ, μ, σ^2	Hyperparameters
a_0, a_f	Angular acceleration at the start time and the end time	Z	Gaussian process with mean zero
ωC_j	Maximum angular velocity for joint j	$\boldsymbol{R}$	Correlation matrix
aC_j	Maximum angular acceleration for joint j	$\hat{\vartheta}, \hat{\mu}, \hat{\sigma}^2$	Maximum likelihood estimation of hyperparameters
w	Flexible displacement at the end of the manipulator	$\boldsymbol{r}$	Correlation vector
ϕ_1, ϕ_2	First two-order modal shape functions	$\boldsymbol{L}$	Lower triangular matrix
α_1, α_2	Weight factors	A, B	Incremental matrix of new data
X_i	Position of the population	$\boldsymbol{y}$	Objective value of sample points
V_i	Velocity of the particle	$\hat{y}$	Best linear unbiased prediction
P_i	Optimal solution of the particle	$\hat{s}$	Error estimation of the prediction
P_g	Global optimal solution currently found	Φ	Standard normal cumulative distribution function
γ	Random number	φ	Standard normal probability density function
K_r	Random number	$\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{L}}$	Incremental form of $\boldsymbol{y}, \boldsymbol{L}$
X_{best}	Best individual of the group after the t-th iteration	$\widetilde{\boldsymbol{R}}$	Updated correlation matrix

References

1. Irani, R.A.; Limited, R.-R.C.; Kehoe, D.; Spencer, W.W.; Watt, G.D.; Gillis, C.; Carretero, J.A.; Dubay, R. Towards a UUV launch and recovery system on a slowly moving submarine. In Proceedings of the International Conference on Warship, Bath, UK, 15–16 June 2014.
2. Watt, G.D.; Roy, A.R.; Currie, J.; Gillis, C.B.; Giesbrecht, J.; Heard, G.J.; Birsan, M.; Seto, M.L.; Carretero, J.A.; Dubay, R.; et al. A Concept for Docking a UUV With a Slowly Moving Submarine Under Waves. *IEEE J. Ocean. Eng.* **2015**, *41*, 471–498. [CrossRef]
3. Sivčev, S.; Coleman, J.; Omerdić, E.; Dooly, G.; Toal, D. Underwater manipulators: A review. *Ocean Eng.* **2018**, *163*, 431–450. [CrossRef]
4. Vakil, M.; Fotouhi, R.; Nikiforuk, P.N. A new method for dynamic modeling of flexible-link flexible-joint manipulators. *J. Vib. Acoust.-Trans. ASME.* **2011**, *134*, 014503. [CrossRef]
5. Meng, D.; Wang, X.; Xu, W.; Liang, B. Space robots with flexible appendages: Dynamic modeling, coupling measurement, and vibration suppression. *J. Sound Vib.* **2017**, *396*, 30–50. [CrossRef]
6. Sahu, U.K.; Patra, D. Observer based backstepping method for tip tracking control of 2-DOF Serial Flexible Link Manipulator. In Proceedings of the Region 10 Conference, Singapore, 22–25 November 2016.
7. Yang, Y.-L.; Wei, Y.-D.; Lou, J.-Q.; Fu, L.; Fang, S.; Chen, T.-H. Dynamic modeling and adaptive vibration suppression of a high-speed macro-micro manipulator. *J. Sound Vib.* **2018**, *422*, 318–342. [CrossRef]
8. Rahmani, B.; Belkheiri, M. Adaptive Neural Network Output Feedback Control for Flexible Multi-Link Robotic Manipulators. *Int. J. Control.* **2018**, *92*, 2324–2338. [CrossRef]
9. Yavuz, Ş. An improved vibration control method of a flexible non-uniform shaped manipulator. *Simul. Model. Pract. Theory* **2021**, *111*, 102348. [CrossRef]
10. Guo, Z.; Zhang, J.; Zhang, P. Research on the Residual Vibration Suppression of Delta Robots Based on the Dual-Modal Input Shaping Method. *Actuators* **2023**, *12*, 84. [CrossRef]
11. Xu, S.; Cui, N.; Fan, Y.; Guan, Y. A Study on Optimal Compensation Design for Meteorological Satellites in the Presence of Periodic Disturbance. *Appl. Sci.* **2018**, *8*, 1190. [CrossRef]
12. Sands, T. Optimization Provenance of Whiplash Compensation for Flexible Space Robotics. *Aerospace* **2019**, *6*, 93. [CrossRef]
13. Park, K.-J.; Park, Y.-S. Fourier-based optimal design of a flexible manipulator path to reduce residual vibration of the endpoint. *Robotica* **1993**, *11*, 263–272. [CrossRef]
14. Wu, H.; Sun, F.; Sun, Z.; Wu, L. Optimal Trajectory Planning of a Flexible Dual-Arm Space Robot with Vibration Reduction. *J. Intell. Robot. Syst.* **2004**, *40*, 147–163. [CrossRef]
15. Guo, C.; Gao, H.; Ni, F.; Liu, H. A vibration suppression method for flexible joints manipulator based on trajectory optimization. In Proceedings of the IEEE International Conference on Mechatronics & Automation, Harbin, China, 7–10 August 2016. [CrossRef]
16. Cui, L.; Wang, H.; Chen, W. Trajectory planning of a spatial flexible manipulator for vibration suppression. *Robot. Auton. Syst.* **2019**, *123*, 103316. [CrossRef]
17. Lin, C.; Chang, P.; Luh, J. Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots. *IEEE Trans. Autom. Control* **1984**, *28*, 1066–1074. [CrossRef]
18. Yin, H.; Kobayashi, Y.; Hoshino, Y.; Emaru, T. Hybrid sliding mode control with optimization for flexible manipulator under fast motion. In Proceedings of the Robotics and Automation (ICRA), 2011 IEEE International Conference, Shanghai, China, 9–13 May 2011; pp. 458–463. [CrossRef]
19. Li, Y.; Ge, S.S.; Wei, Q.; Gan, T.; Tao, X. An Online Trajectory Planning Method of a Flexible-Link Manipulator Aiming at Vibration Suppression. *IEEE Access* **2020**, *8*, 130616–130632. [CrossRef]
20. Yue, H.S.; Henrich, D.; Xu, W.L. Trajectory planning in joint space for flexible robots with kinematics redundancy. In Proceedings of the IASTED International Conference, Honolulu, HI, USA, 13–16 August 2001; pp. 1–6.

21. Kazem, I.B.; Mahdi, A.I.; Oudah, A.T. Motion Planning for a Robot Arm by Using Genetic Algorithm. *Jordan J. Mech. Ind. Eng.* **2008**, *2*, 131–136.
22. Qingmei, L.; Jia, J.; Yu-An, H. Vibration suppression of manipulator using quantum genetic algorithm. In Proceedings of the Information Technology, Networking, Electronic and Automation Control, Chengdu, China, 15–17 December 2017; pp. 802–807. [CrossRef]
23. Kramer, O. A Review of Constraint-Handling Techniques for Evolution Strategies. *Appl. Comput. Intell. Soft Comput.* **2010**, *2010*, 185063. [CrossRef]
24. Jordehi, A.R. A review on constraint handling strategies in particle swarm optimisation. *Neural Comput. Appl.* **2015**, *26*, 1265–1275. [CrossRef]
25. Kohler, M.; Forero, L.; Vellasco, M.; Tanscheit, R.; Pacheco, M.A. PSO+: A nonlinear constraints-handling particle swarm optimization. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 2518–2523. [CrossRef]
26. Cao, B.; Sun, K.; Li, T.; Gu, Y.; Jin, M.; Liu, H. Trajectory Modified in Joint Space for Vibration Suppression of Manipulator. *IEEE Access* **2018**, *6*, 57969–57980. [CrossRef]
27. Wang, M.; Luo, J.; Yuan, J.; Walter, U. Coordinated trajectory planning of dual-arm space robot using constrained particle swarm optimization. *Acta Astronaut.* **2018**, *146*, 259–272. [CrossRef]
28. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control. Eng.* **2020**, *8*, 22–34. [CrossRef]
29. Zhang, Z.; He, R.; Yang, K. A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm. *Adv. Manuf.* **2022**, *10*, 114–130. [CrossRef]
30. Liu, G.; Shu, C.; Liang, Z.; Peng, B.; Cheng, L. A Modified Sparrow Search Algorithm with Application in 3d Route Planning for UAV. *Sensors* **2021**, *21*, 1224. [CrossRef]
31. Farivarnejad, H.; Moosavian, S.A.A. Multiple Impedance Control for object manipulation by a dual arm underwater vehicle–manipulator system. *Ocean Eng.* **2014**, *89*, 82–98. [CrossRef]
32. Li, J.; Huang, H.; Wan, L.; Zhou, Z.; Xu, Y. Hybrid Strategy-based Coordinate Controller for an Underwater Vehicle Manipulator System Using Nonlinear Disturbance Observer. *Robotica* **2019**, *37*, 1710–1731. [CrossRef]
33. Huang, H.; Tang, G.Y.; Han, L.J.; Cheng, M.L.; Xie, D.; Chen, H.X. Neural network Adaptive Backstepping Control of Multi-link Underwater Flexible Manipulators. In Proceedings of the 31st International Ocean and Polar Engineering Conference, Rhodes, Greece, 20–25 June 2021.
34. Huang, H.; Tang, G.; Chen, H.; Han, L.; Xie, D. Dynamic Modeling and Vibration Suppression for Two-Link Underwater Flexible Manipulators. *IEEE Access* **2022**, *10*, 40181–40196. [CrossRef]
35. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948. [CrossRef]
36. Clerc, M.; Kennedy, J. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [CrossRef]
37. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [CrossRef]
38. Tizhoosh, H. Opposition-based learning: A new scheme for machine intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, Vienna, Austria, 28–30 November 2005; pp. 695–701. [CrossRef]
39. Zhao, C.; Dinar, M.; Melkote, S.N. A data-driven framework for learning the capability of manufacturing process sequences. *J. Manuf. Syst.* **2022**, *64*, 68–80. [CrossRef]
40. Zhang, S.; Liang, P.; Pang, Y.; Li, J.; Song, X. Multi-fidelity surrogate model ensemble based on feasible intervals. *Struct. Multidiscip. Optim.* **2022**, *65*, 1–13. [CrossRef]
41. Yang, Z.; Pak, U.; Yan, Y.; Kwon, C. Reliability-based robust optimization design for vehicle drum brake considering multiple failure modes. *Struct. Multidiscip. Optim.* **2022**, *65*, 1–17. [CrossRef]
42. Zhan, D.; Xing, H. A fast kriging-assisted evolutionary algorithm based on incremental learning. *IEEE Trans. Evol. Comput.* **2021**, *25*, 941–955. [CrossRef]
43. Zhong, J.; Zheng, Y.; Chen, J.; Jing, Z. Variable-stiffness composite cylinder design under combined loadings by using the improved Kriging model. *Acta Mech. Sin.* **2019**, *35*, 201–211. [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.