

Article

Collision-Free Trajectory Planning Optimization Algorithms for Two-Arm Cascade Combination System

Jingjing Xu ¹, Long Tao ¹, Yanhu Pei ², Qiang Cheng ¹, Hongyan Chu ¹ and Tao Zhang ^{1,*}

¹ Institute of Advanced Manufacturing and Intelligent Technology, Beijing University of Technology, Beijing 100124, China; xujj0908@bjut.edu.cn (J.X.); taolong@emails.bjut.edu.cn (L.T.); chengqiang@bjut.edu.cn (Q.C.); chuhongyan@bjut.edu.cn (H.C.)

² Genertec Machine Tool Engineering Research Institute Co., Ltd., Beijing 100102, China; peiyanhu@gt.cn

* Correspondence: zhangtao@bjut.edu.cn

Abstract: As a kind of space robot, the two-arm cascade combination system (TACCS) has been applied to perform auxiliary operations at different locations outside space cabins. The motion coupling relation of two arms and complex surrounding obstacles make the collision-free trajectory planning optimization of TACCS more difficult, which has become an urgent problem to be solved. For the above problem, this paper proposed collision-free and time–energy–minimum trajectory planning optimization algorithms, considering the motion coupling of two arms. In this method, the screw-based inverse kinematics (IK) model of TACCS is established to provide the basis for the motion planning in joint space by decoupling the whole IK problem into two IK sub-problems of two arms; the minimum distance calculation model is established based on the hybrid geometric enveloping way and basic distance functions, which can provide the efficient and accurate data basis for the obstacle-avoidance constraint condition of the trajectory optimization. Moreover, the single and bi-layer optimization algorithms are presented by taking motion time and energy consumption as objectives and considering obstacle-avoidance and kinematics constraints. Finally, through example cases, the results indicate that the bi-layer optimization has higher convergence efficiency under the premise of ensuring the optimization effect by separating variables and constraint terms. This work can provide theoretical and methodological support for the efficient and intelligent applications of TACCS in the space arena.

Keywords: space robot; trajectory planning; obstacle avoidance; two-arm cascade combination system; bi-layer optimization

MSC: 49M99; 51K05



Citation: Xu, J.; Tao, L.; Pei, Y.; Cheng, Q.; Chu, H.; Zhang, T. Collision-Free Trajectory Planning Optimization Algorithms for Two-Arm Cascade Combination System. *Mathematics* **2024**, *12*, 2245. <https://doi.org/10.3390/math12142245>

Academic Editors: Montserrat Gil-Martinez, Ramón Vilanova Arbós and Marian Barbu

Received: 27 June 2024
Revised: 11 July 2024
Accepted: 15 July 2024
Published: 18 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Background

In the field of modern aerospace, space robots have been used to help people complete certain tasks inside and outside space capsules. With the development of aerospace technology, different kinds of space robots have been studied and applied for on-orbit servicing tasks, such as a 6-DOFs manipulator [1], 7-DOFs manipulator [2], 14-DOFs super redundant manipulator [3], and a dual-arm humanoid robot [4]. The safe and efficient operation of space robots depends on the motion planning, which has always been the focus of researchers in recent studies [5,6]. For example, Shrivastava et al. proposed a jerk-optimized motion planning method for the redundant space robot which used the gray wolf optimization algorithm to search for the optimal result [7]; Xie et al. presented the trajectory planning method for dual-arm free-floating space robot to minimize the final base attitude change and ensure the trajectory smoothness by using an enhanced bidirectional approach [8]. In recent years, the two-arm cascade combination system (TACCS)

shown in Figure 1 has been developed and applied to handling and maintenance outside capsules, and it is obtained through the docking combination of two manipulators (which can largely increase the workable area and the robotic dexterity) and better adapted to the operational tasks on the outer surface of the cylindrical cabin. Two manipulators in TACCS can be regarded as the fixed manipulator (first-level arm) and the mobile manipulator (second-level arm), respectively, and the base coordinate frame (BCF) of the latter can be changed by the joint movement of the former.

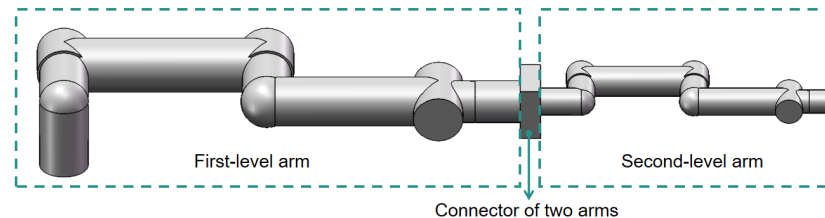


Figure 1. Classic structure of TACCS.

1.2. Literature Review of Motion Planning

In existing studies, the motion planning of manipulators has been the focus of research aimed at finding efficient, accurate and safe modes of application. Actually, the motion planning of TACCS has the same problem with that of the mobile manipulator (composed of a mobile car and a manipulator), which is how to consider the influence of the motion of the first-level arm or the mobile car to the motion performance of the second-level arm or the manipulator. There exist two solving methods for the motion planning of the mobile manipulator: one of them is that the motion planning algorithms of two subsystems are carried separately (called SMP in the following), the other is that the motion of the high-DOFs system is planned as a whole (called WMP in the following).

For SMP, researchers mainly focused on the improvement of the whole planning efficiency of the robotic system in previous studies. For example, Li et al. presented a hierarchical motion planning method for the mobile manipulator, which includes two planning stages: firstly the path of the two-dimensional mobile base was optimized using a hybrid sampling strategy that combined a bridge test and uniform sampling, aiming to improve the planning efficiency; secondly the secure configuration and trajectory of the manipulator was searched [9]. Chen et al. also proposed a hierarchical motion planning method which used the optimization-based A* algorithm and the sampling-based heuristic algorithm for the collision-free motion planning of the mobile base and the manipulator, respectively, which can effectively improve the planning efficiency and success rate [10]. Rastegarpanah et al. used an A* algorithm and rapidly exploring random trees (RRTs) to separately plan the paths for the mobile platform and manipulator and completed the task of picking up an object from one table to one box [11]. In order to consider the coupling relation of configurations of two subsystems, criteria like reachability or manipulability of the manipulator's end effector has been considered to decouple the planning of the two subsystems. For example, Colucci et al. developed a motion planning algorithm based on a manipulability index which considers the mobile base and manipulator separately and provides the closed inverse kinematics algorithm for the redundant manipulator [12,13]. Actually, the motion planning for only the mobile car or manipulator has been popular for quite a long time. Some mature technologies have been proposed. For the motion safety, some distance calculation or collision detection algorithms between the robot and surrounding obstacles based on different envelope ways were developed by scholars. For example, basic geometries like spheres and cylinders [14,15], and hierarchical bounding boxes [16] were used to envelope the robotic arms or obstacles and then applied to solve the collision-free motion planning problem of the manipulator. For the motion stability or trajectory-tracking accuracy, a cubic/quintic polynomial curve [17,18], B-spline curve [19, 20] or segmented function curve [21,22] were utilized to interpolate the motion trajectory, which can eliminate vibrations caused by speed or acceleration mutations when the robot

starts and stops. Moreover, the motion time [23,24], energy consumption [25], angular jerk [26,27] or their combination [28] were always taken as the objectives of the motion planning optimization of the manipulator.

For WMP, the grid-based [29], sampling-based [30] and optimization-based [31] methods were proposed in previous studies. The grid-based method are generally proposed based on the free space represented by a swept volume or a graph-creating algorithm, and since the dimension of space related to the number of robotic joints has a major impact on the planning efficiency, it is more applicable for the path planning of the mobile car or the low-DOFs robot. For example, Liu et al. considered the mobile manipulator (composed of a 2-DOFs manipulator and 3-DOFs mobile device) as a whole system and presented a path planning method based on representation space [29]. The sampling-based methods have been developed well to solve the path planning of mobile manipulators with high DOFs, including RRTs, probabilistic roadmap (PRM) and their modifications. For example, Dai et al. presented a path planning method based on a novel potential bidirectional RRT* for redundant manipulators in joint space [32] which can effectively improve the obstacle-avoidance ability of the robot; Chen et al. proposed an improved PRM based on a new sampling strategy using the virtual force field and the design of a new connection strategy to improve the planning efficiency [33]. However, the above methods are mostly used for the path planning of the robot, which refers to determining a pure geometric description of a series of positions of the end effector or mobile base; additionally, it does not consider the system kinematics, including the change in velocities or accelerations over time, which means the motion stability or motion time cannot be planned or optimized. The optimization-based methods are mostly presented by the modeling and solving of the motion planning optimization problem, wherein constraints for the obstacle avoidance, accuracy, kinematics and dynamics can be considered, and multiple optimization objectives, including the motion time, energy consumption and joint jerks, can be taken to ensure the working efficiency, energy conservation and motion stability [34–36]. In existing works, many intelligent optimization algorithms (like the particle swarm optimization (PSO) algorithm [35], genetic algorithm [36], etc.) have been introduced to solve the above optimization problem. But for the mobile manipulators with high DOFs, the high optimization performance (including the ability of searching for the global optimal result, the convergence efficiency) is difficult to obtain. Therefore, many scholars focus on the modification of the current optimization algorithms or the proposal of the novel optimization algorithm. For example, Jin et al. used the chaotic PSO to solve the motion trajectory planning optimization problem, which improved the premature phenomenon of the traditional PSO algorithm [37]; Li et al. proposed a novel hybrid heuristic algorithm, which combined PSO and a whale optimization algorithm (PSO-WOA) to solve the multi-objective optimization problem of the trajectory planning of the space robot, considering the end-effector pose, base disturbance, motion time and manipulability as objectives [38]; Cao et al. proposed a modified multi-objective PSO algorithm (GMOPSO) by combining a mutation operator, annealing factor and feedback mechanism for the trajectory planning of fruit picking manipulator [39].

The comparison of the existing representative works is shown in Table 1. Since only the path planning was implemented without optimization in many related works, the robot type, DOFs, planning methods, operational performance (like safety, manipulability, stability, efficiency, energy consumption) and the consideration of motion coupling of all works are listed in the comparison table in order to compare them more comprehensively. From the comparison, we can observe that the optimization-based planning method can directly obtain an optimal motion trajectory with the minimum of efficiency or energy consumption as the objective for the robot, not only a collision-free operational path, and it can better control the motion performance of the robot in terms of obstacle avoidance, motion stability, tracking accuracy, efficiency, or energy consumption, etc. However, the low planning optimization efficiency has always been a problem and the focus of the current research.

Table 1. Comparison of some representative works wherein the superscript ^{#F} means the robot type is free-floating space robot.

Works	Robot Type	DOFs	Planning Methods	Safety	Stability	Efficiency	Energy Consumption	Consideration of Motion Coupling
Ref. [9]	Mobile manipulator	3 + 6	A* algorithm; Uniform sampling-based algorithm	Yes	No	No	No	No
Ref. [10]		3 + 6	A* algorithm; Sampling-based heuristic algorithm	Yes	No	No	No	No
Ref. [11]		3 + 6	A* algorithm; RRT	Yes	No	No	No	No
Refs. [12,13]		5 + 6	A* algorithm; RRT	Yes	No	No	No	Yes
Ref. [29]		3 + 2	A* algorithm	Yes	No	No	No	Yes
Ref. [39]		3 + 6	PSO-based optimization	Yes	Yes	Yes	Yes	No
Ref. [30]	High-DOFs robot	14 or 21	Sampling-based algorithm	Yes	No	No	No	/
Ref. [31]		5	PSO-based optimization	No	Yes	Yes	No	/
Ref. [32]		Any	Bidirectional RRT*	Yes	No	No	No	/
Ref. [33]		Any	PRM	Yes	No	No	No	/
Ref. [37]		6 ^{#F}	CPSO-based optimization	No	Yes	Yes	No	/
Ref. [38]		7 ^{#F}	PSO-WOA-based optimization	Yes	Yes	Yes	No	/
The proposed method	TACCS	6 + 6	PSO-based optimization	Yes	Yes	Yes	Yes	Yes

1.3. Problem Formulation

As shown in Figure 1, when one manipulator (DOFs ≥ 6) is mounted as the real end effector of the other manipulator (DOFs ≥ 6), the whole robot will be a super redundant system with DOFs of more than 12 which has been applied to perform on-orbit servicing tasks in the aerospace field, such as assembly, maintenance, the capture of cooperative and non-cooperative objects, deployment and debris removal [40]. Moreover, there exist many protruding structures outside the cylindrical cabin surface used for measurement, external docking, entrance, etc., which means the robotic motion planning needs to consider the obstacle avoidance. Therefore, the high redundancy, the large cylindrical working surface and the complex surrounding obstacles make the collision-free motion planning of TACCS quite difficult. In addition, from the summary of the current motion planning methods, the optimization-based planning method can better control the motion performance of the robot. However, since TACCS has the higher DOFs, as well as faces the special and complex working environment, the motion planning optimization becomes more difficult. The difficulties are mainly reflected in several aspects: (1) the accurate analytical solution of the inverse kinematics of the TACCS is the main basis for its motion planning in joint space, which can better ensure the motion stability of robotic joints, but it has always been a problem due to the high redundancy; (2) the accurate minimum distance between manipulator and its workspace is difficult to calculate quickly, which is very important in obtaining an efficient and reliable motion planning algorithm; (3) the number of variables and constraints increases over the robotic DOFs, which makes the trajectory optimization less efficient and more easily trapped into the local optimum.

In this work, a collision-free and time–energy–minimum (CF-TEM) trajectory planning optimization method is proposed for TACCS considering the dexterity of the second-level arm under the complex and specific working environment outside the capsule. Three problems are solved based on the following contributions: (1) the IK problem of TACCS is decoupled into two IK sub-problems of two arms and then the screw-based IK model is established, wherein the IK solution of second-level arm is related to the end-effector pose of the first-level arm; (2) the minimum distance calculation model is established based on the hybrid geometric enveloping way and basic distance functions, wherein TACCS is enveloped by a convex polyhedrons, slices-based circular/polygonal edges and planes, and the the surrounding obstacles are enveloped by convex polygonal planes; (3) the single and bi-layer optimization algorithms are both proposed by taking motion time and energy

consumption as objectives and considering constraints of obstacle avoidance and robotic kinematics, and they are finally compared in terms of efficiency and effect of optimization, aiming to find an efficient and reliable optimization algorithm for the motion planning problem of the system.

The remainder of this work is organized as the following three sections: Section 2 presents the screw-based inverse kinematics of TACCS, which gives the explicit mathematical mapping from the target pose of the end effector in a base coordinate frame (BCF) of the system to joint angles of two arms. Section 3 gives the minimum distance calculation model based on the hybrid geometric envelope way, considering the characteristics of operation task and obstacles outside the capsule; Section 4 gives the single and bi-layer optimization models and algorithms of the system; Section 5 discusses the results obtained by two optimization algorithms from the efficiency and effect of optimization through a simulation case; Section 6 gives the main conclusions and future works.

2. Screw-Based Inverse Kinematics Model of TACCS

For the inverse kinematics (IK) analysis, the kinematics parameters and three coordinate frames of TACCS are shown in Figure 2, wherein the frame 1 ($o - xyz$) refers to the BCF of the whole system and also the BCF of the first-level arm; the frame 2 ($o_{E1} - x_{E1}y_{E1}z_{E1}$) refers to the end-effector coordinate frame (EECF) of the first-level arm and also the BCF of the second-level arm; the frame 3 ($o_{E2} - x_{E2}y_{E2}z_{E2}$) refers to the EECF of the second-level arm, which is directly related to the operational task. Then, the IK solution problem can be described as the solving of joint angles of two arms by taking the end-effector target pose of the second-level arm in frame 1 as a known condition. Moreover, in Figure 2, q_i represents one point taken from the rotational axis of the i th joint of one arm of TACCS, while q_e is the real end effector of each arm and l_i means the kinematics parameters that will be used in the following IK model. Since the pose gst_{E1} of the end effector of the first-level arm has a major impact on the operation effect of the second-level arm, it will be optimized in the motion planning optimization of the system, which means the IK problem of TACCS can be decoupling into the IK sub-problems of two arms. Through the above decoupling processing, the IK sub-problem 1 aims to solve the joint angles of the first-level arm with the given pose gst_{E1} in frame 1; IK sub-problem 2 aims to solve the joint angles of the second-level arm with the given pose gst_{E2} in frame 1. In the traditional IK problem, the end-effector pose of manipulator is given in its own base coordinate frame, that is, for IK sub-problem 1, the inverse solution can be modeled directly using the screw-based method, but for IK sub-problem 2, the pose gst_{E2} in frame 1 must be transferred in frame 2, as then the solution can be modeled based on the new end-effector pose $gst_{E2,N}$.

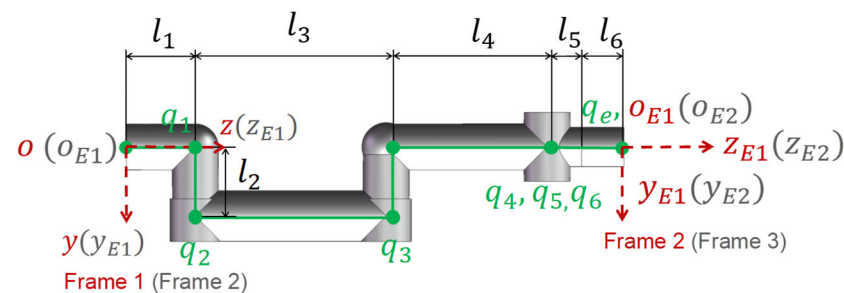


Figure 2. Kinematics parameters of each arm of TACCS.

It is noted that, for the whole IK problem of TACCS, the operation task determines the pose $gst_{E2} \in \mathcal{R}^{4 \times 4}$, which can be expressed by a homogeneous coordinate matrix as follows:

$$gst_{E2} = \begin{bmatrix} R_e & q_e \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

wherein $R_e \in \mathcal{R}^{3 \times 3}$ and $q_e \in \mathcal{R}^3$ denote the orientation and the 3D coordinates of the end effector, respectively.

Moreover, in the following optimization of the pose gst_{E1} , three linear displacements $\Delta x, \Delta y, \Delta z$ along three axes and three rotational angles α, β, γ around three axes of frame 1 are taken as optimization variables. Then, the optimal pose $gst_{E1} \in \mathcal{R}^{4 \times 4}$ can be expressed by the following matrix:

$$gst_{E1} = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma & \Delta x \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & \Delta y \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

wherein $c_\alpha = \cos\alpha, s_\alpha = \sin\alpha; c_\beta = \cos\beta, s_\beta = \sin\beta; c_\gamma = \cos\gamma, s_\gamma = \sin\gamma$.

Since the poses gst_{E2} and $gst_{E2,N}$ have the transfer relation $gst_{E2} = gst_{E1} \cdot gst_{E2,N}$, the pose $gst_{E2,N} \in \mathcal{R}^{4 \times 4}$ can be expressed by

$$gst_{E2,N} = [gst_{E1}]^{-1} \cdot gst_{E2} \tag{3}$$

Based on the Chasles principle, the motion screw of the rotational joint is defined by $\xi = [\omega \ \theta]^T$, and the exponential form of the screw coordinate can be written as follows,

$$\exp(\hat{\xi}\theta) = \begin{bmatrix} \exp(\hat{\omega}\theta) & [I - \exp(\hat{\omega}\theta)](\omega \times v) + \theta\omega\omega^T v \\ 0 & 1 \end{bmatrix} \tag{4}$$

wherein ω is the unit vector of the rotational axis; $\hat{\omega}$ is the anti-symmetric matrix of ω , while

setting $\omega = (\omega_1, \omega_2, \omega_3), \hat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}; \theta$ represents the rotational angle; v

means the linear velocity of the rotational motion, $v = q \times \omega$, where q denotes the 3D coordinate of one point located on the rotational axis; I is a 3×3 identity matrix; $\exp(\hat{\omega}\theta)$ can be calculated by $\exp(\hat{\omega}\theta) = I + \hat{\omega}\sin\theta + \hat{\omega}^2(1 - \cos\theta)$.

Based on the above analysis, two IK sub-problems can be transferred into the general IK problems. Taking the robotic model shown in Figure 2 as the example case, two arms are manipulators with the same configuration but different values to the kinematics parameters listed in Table 2. According to our previous work [41], the IK problem of each arm can be firstly decomposed into three sub-problems: one is for the rotation about three non-intersecting axes; the second is for the rotation about two intersecting axes; and the third is for the rotation about one single axis. For the first two IK sub-problems, the rotational motions are shown as Figure 3, respectively, wherein c_1, c_2, c_3 are the three transition points of two motions, and their 3D coordinates $c_1 = [x_1, y_1, z_1], c_2 = [x_2, y_2, z_2], c_3 = [x_3, y_3, z_3]$ can be obtained by solving a hexagonal quadratic equation system and a ternary quadratic equation system. Based on the above calculations, the IK solution of the arm (the angular displacements of all joints of the arm) can be explicitly modeled and can be expressed as Equation (5). The detailed solving process can be referred to through reference [41].

$$\begin{cases} \theta_1 = \text{atan2}(x_2 y - y_2 x, x_2 x - y_2 y) \\ \theta_2 = \text{atan2}[x_2(z_1 - l_1) - x_1(z_2 - l_1), x_1 x_2 - (z_1 - l_1)(z_2 - l_1)] \\ \theta_3 = \text{atan2}[x_1 l_4, l_4(z_1 - l_1 - l_3)] \\ \theta_4 = \text{atan2}(x_3 y_{e1} - y_3 x_{e1}, x_3 x_{e1} - y_3 y_{e1}) \\ \theta_5 = \text{atan2}[x_3 l_2, l_2(z_3 - l_1 - l_3 - l_4)] \\ \theta_6 = \text{atan2}(y_{e2}, x_{e2}) \end{cases} \tag{5}$$

where, $[x_{e1}, y_{e1}, z_{e1}, 1]^T = [\exp(\hat{\xi}_1\theta_1)\exp(\hat{\xi}_2\theta_2)\exp(\hat{\xi}_3\theta_3)]^{-1}[x_{s1}, y_{s1}, z_{s1}, 1]^T$. In this equation, $[x_{s1}, y_{s1}, z_{s1}]$ is the initial 3D coordinates of one point located on the sixth joint axis but

not on the fourth and fifth axes; $[x_{e2}, y_{e2}, z_{e2}, 1]^T = [\exp(\hat{\xi}_1\theta_1) \cdots \exp(\hat{\xi}_5\theta_5)]^{-1} [x_{s2}, y_{s2}, z_{s2}, 1]^T$, in this equation, $[x_{s2}, y_{s2}, z_{s2}]$ is the initial 3D coordinates of one point located out of the sixth joint axis.

Table 2. Kinematics parameters of two arms.

Parameters	l_1 (mm)	l_2 (mm)	l_3 (mm)	l_4 (mm)	l_5 (mm)	l_6 (mm)
First-level arm	540	540	1494	1260	270	332
Second-level arm	300	300	830	700	150	184.5

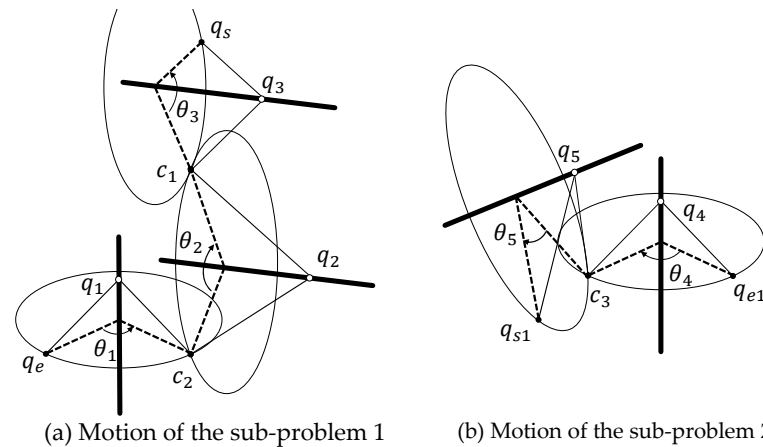


Figure 3. Motions of two IK sub-problems.

Then, IK models can be written as two functions, $\theta_{A1} = fun_IK_A1(gst_{E1})$ and $\theta_{A2} = fun_IK_A2(gst_{E2,N})$, which together form an important basis for the motion planning optimization of the TACCS.

3. Minimum Distance Calculation Based on Hybrid Geometric Envelope

The minimum distance calculation is an important basis for the spatial constraint determination of the motion planning optimization. In real applications, the TACCS will be applied to perform works with targets at different places on the capsule body. Moreover, the outer surface of the capsule may have some raised structures for docking or measuring, which makes the accurate distance calculation more difficult. In order to solve the above problem, this work uses the circular slices and convex polygonal planes to envelop the robotic links and connection structure between two arms, respectively, also using the convex polygonal planes to envelop the capsule and other obstacle structures located at the surface of the capsule, respectively. The envelope structures are shown in Figure 4. From Figure 4, the environmental obstacles include convex obstacles and cylindrical obstacles. Since the distance calculation between the circular slice and the cylinder is difficult, the cylinder is also enveloped by convex polygonal planes, which needs to firstly envelope the end circle using its outer joined regular polygon. For the envelope of a circle, the largest error between the original and new geometry can be calculated by $\epsilon = R[1/\cos(\pi/N_e) - 1]$ (as shown in Figure 5); therefore; if we hope $\epsilon \leq 5$ mm, the number of edges must meet the condition as $N_e \geq \pi/\arccos[R/(5 + R)]$ (based on this condition, when $R = 50$ mm, $N_e \geq 7.3$).

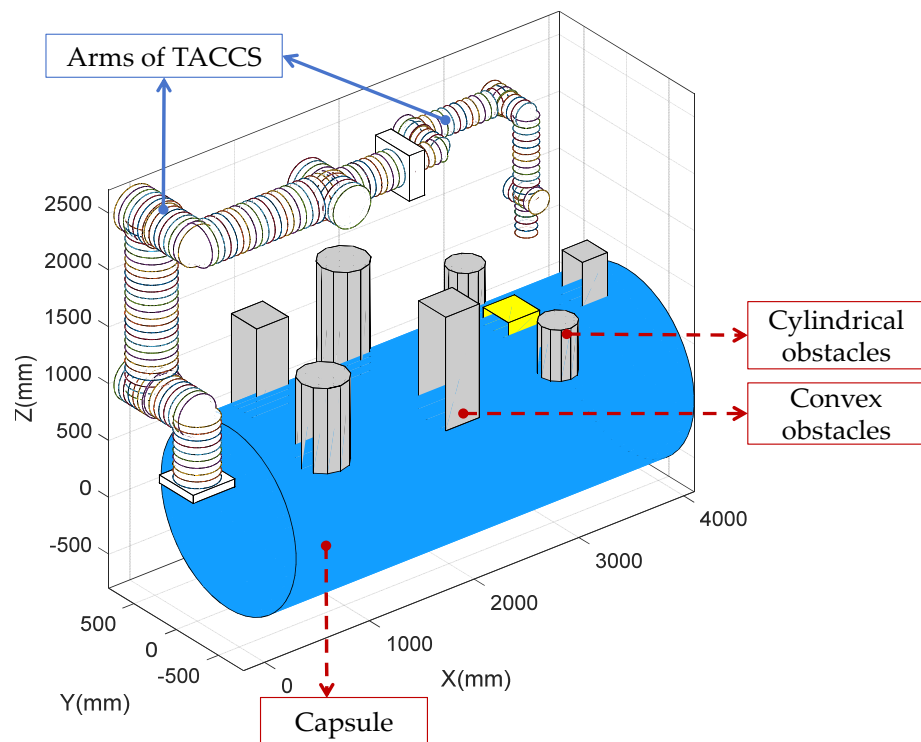


Figure 4. Envelope structures of TACCS and capsule.

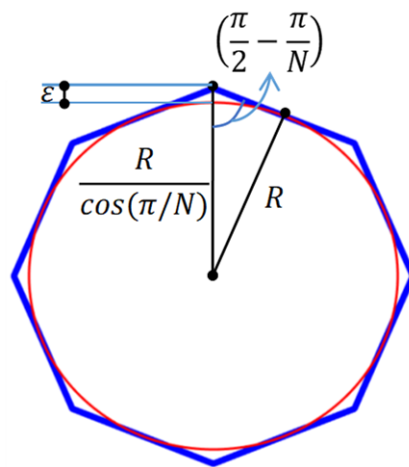


Figure 5. Envelop of a circle.

In our previous study [42], the basic functions of the minimum distance between the circular edge and the convex polygonal plane and that between two convex polygonal planes have been established, which can be represented by $fun_CP(\rho, \zeta)$ and $fun_PP(\zeta_1, \zeta_2)$, wherein ρ represents one circular edge and is determined by the coordinate of the circular center and ζ represents one convex polygonal plane and is determined by the coordinates of polygonal vertices. Based on the envelope ways shown in Figure 3, the structure of the TACCS is finally expressed as a set of geometries, including circular edges and convex polygonal planes in mathematics, and the structure of its workspace obstacles are expressed as a set of convex polygonal planes. Therefore, the minimum distance between TACCS and its workspace obstacles can be written as follows:

$$DI = \min\{fun_CP(\rho_{R,i}, \tau_{O,k}), fun_PP(\rho_{R,j}, \tau_{O,k}) / 1 \leq i \leq I, 1 \leq j \leq J, 1 \leq k \leq K\} \tag{6}$$

wherein $\rho_{R,i}$ and $\rho_{R,j}$ denote the i th circular edge and the j th polygonal plane; $\tau_{O,k}$ means the k th polygonal plane of obstacles; I, J are the numbers of circular edges and polygonal planes of the TACCS, and K is the number of polygonal planes of obstacles.

As shown in Figure 5, the envelope of circle will lead to the largest error, ϵ . In order to further analyze the influence of ϵ on the error of the shortest distance calculation, the distance between the circular plane with different radii ($R = 50, 100, 150, 200$ mm) and an circular edge ($R_e = 150$ mm) are taken as objectives. For the analysis, the center locations of two circles are set by $C = (0, 0, 0)$, $C_e = (200, 300, 400)$, and their normal vectors are set by $n = (0, 0, 1)$, $n_e = \frac{\sqrt{3}}{3} \cdot (1, 1, 1)$, respectively. During the calculation process, taking different values for ϵ , and determining the minimum number of edges by $N_e = \text{ceil}\{\pi / \text{acos}[R / (\epsilon + R)]\}$ ($\text{ceil}\{\}$ represents the function that returns the minimum integer larger than the given real number), the shortest distance between the circular plane and the circular edge can be predicted by the distance between the convex polygonal plane ζ and the circular edge ρ can be calculated by the function $d_p = \text{fun_CP}(\rho, \zeta)$; the real shortest distance d_r can be calculated by taking $N_e = 1000$, which can make the convex polygonal plane close enough to the circular outline; then the error of the shortest distance calculation is determined by $E = d_r - d_p$.

Based on the above strategy, the errors of the shortest distance under different values of ϵ are shown in Figure 6, from which the following conclusions can be obtained: (1) The error basically increases with the increase of the value of ϵ ; this is because the increase of ϵ leads to the lower number of edges of the enveloped polygon (for example, when $\epsilon \leq 5$, the errors for four circular planes are all less than 5 mm, but when $R = 15$ mm, $\epsilon = 17$, the error reaches 11.51 mm). (2) In some cases, even where ϵ takes the large value, the error may be very small. The reasons for that are as follows: ϵ represents the largest error between the original circle and its enveloped polygon, which means some points on their edges coincide or are close together; moreover, the shortest distance between the enveloped polygonal plane and the edge must be the distance between two point elements located on two geometries, and so when the point element at the polygonal plane is located just at the position where the envelope error is small, the predicted shortest distance will be very close to its real value. (3) In order to both ensure the accuracy and efficiency, the reasonable value for ϵ should be taken, since, the smaller the value is, the lower the calculation efficiency and the higher the calculation accuracy. In real applications, the value can be taken according to the safety threshold value, which is set for the absolute safety of the robotic motion.

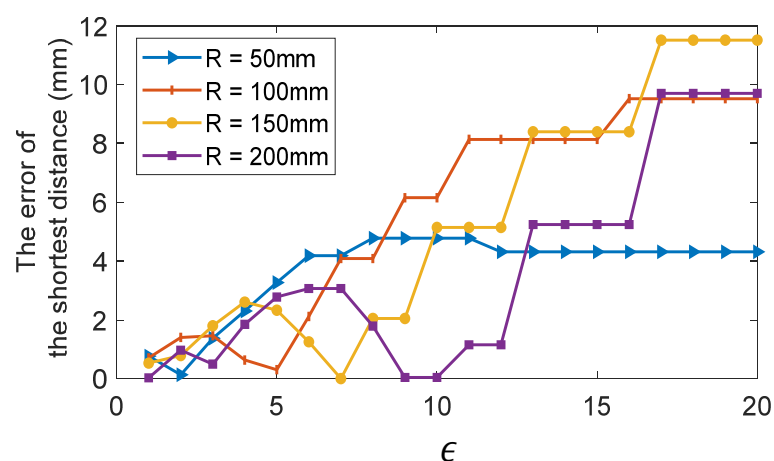


Figure 6. Errors of the shortest distance under different values of ϵ .

Moreover, the pose of the TACCS determines the locations of feature points of the envelope geometries, which can be calculated based on screw-based conversion matrices as follows:

$$q_e' = \left[\prod_{i=1}^N \exp(\hat{\xi}_i \theta_i) \right] q_s' \tag{7}$$

wherein q_s' and q_e' represents the start and final homogeneous coordinates of one point in frame 1, and they are 4D vectors extended from the corresponding 3D coordinate q_s by adding a unit column, i.e., $q_s' = (q_s, 1)^T \in \mathcal{R}^4$; N means the number of joints that affects the pose of this point.

Therefore, since the angular displacement θ_i of the i th joint changes over time during the operational motion, locations of feature points are related to time, which means the minimum distance during motion can be described as a function over time and expressed by $DI(t)$. The above function is a major basis for the following motion planning optimization.

4. Single and Bi-Layer Optimization Models and Algorithms

In the motion planning problem of TACCS, the variables of optimization include the following six motion variables: $\Delta x, \Delta y, \Delta z, \alpha, \beta, \gamma, T_{A1}$ and T_{A2} . T_{A1} and T_{A2} denote the motion times of the first-level arm and the second-level arm, respectively. In order to ensure the operational stability, it is supposed that two arms do not move simultaneously in real applications, which means although the motion dexterity of the second-level arm will be influenced by the pose of the first-level arm, the coupling influence is not dynamic. For this problem, considering both the total motion time and the motion energy consumption as the optimization objective, the objective of the single optimization can be modeled as follows:

$$f_s = w_1(T_{A1} + T_{A2}) + w_2w' \left\{ \sum_{i=1}^{D_{A1}} \int_{t=0}^{T_{A1}} S_{A1,i}(t) + \sum_{i=1}^{D_{A2}} \int_{t=0}^{T_{A2}} S_{A2,i}(t) \right\} \quad (8)$$

wherein w_1 and w_2 are weighting factors for items of the motion and energy consumption, $w_1 + w_2 = 1$; w' denotes the coefficient used to eliminate the influence of the magnitude difference of items; D_{A1} and D_{A2} mean the numbers of DOFs of two arms; $S_{A1,i}(t)$ and $S_{A2,i}(t)$ are time functions of the angular displacement of the i th joint of two arms, respectively, which can be obtained through motion trajectory interpolation. In this work, the quintic polynomial interpolation method is used, and the function of one joint angle can be expressed by

$$S(t) = a_1t^5 + a_2t^4 + a_3t^3 + a_4t^2 + a_5t + a_6 \quad (9)$$

wherein the coefficients of the function can be calculated by $a_1 = 6(\theta_E - \theta_S)/T^5$, $a_2 = -15(\theta_E - \theta_S)/T^4$, $a_3 = 10(\theta_E - \theta_S)/T^3$, $a_4 = a_5 = 0$ and $a_6 = \theta_S$, respectively, which are obtained by the boundary conditions according to requirements of movement stability as $S(0) = \theta_S$, $S(T) = \theta_E$ and $\dot{S}(0) = \dot{S}(T) = \ddot{S}(0) = \ddot{S}(T) = 0$; $\dot{S}(t)$ and $\ddot{S}(t)$ are the functions of angular velocity and acceleration of the joint and can be obtained through one derivative and two derivatives of $S(t)$, respectively.

Then, considering kinematics and obstacle-avoidance constraints, the single optimization problem can be modeled as follows:

$$\left\{ \begin{array}{l} \text{minimize : } f = f_s(\Delta x, \Delta y, \Delta z, \alpha, \beta, \gamma, T_{A1}, T_{A2}) \\ \text{subject to : } \Delta x \in [\Delta x_{min}, \Delta x_{max}] \\ \Delta y \in [\Delta y_{min}, \Delta y_{max}] \\ \Delta z \in [\Delta z_{min}, \Delta z_{max}] \\ \alpha \in [\alpha_{min}, \alpha_{max}] \\ \beta \in [\beta_{min}, \beta_{max}] \\ \gamma \in [\gamma_{min}, \gamma_{max}] \\ S_{A1,i}(t) \in D_{A1,i}; S_{A2,i}(t) \in D_{A2,i} \\ \max \left[\left| \dot{S}_{A1,i}(t) \right| \right] \leq V_{A1,i,max}; \max \left[\left| \ddot{S}_{A1,i}(t) \right| \right] \leq A_{A1,i,max} \\ \max \left[\left| \dot{S}_{A2,i}(t) \right| \right] \leq V_{A2,i,max}; \max \left[\left| \ddot{S}_{A2,i}(t) \right| \right] \leq A_{A2,i,max} \\ \min[DI(t)] \geq DI_{thred} \\ T_{A1} + T_{A2} \leq T_{max} \end{array} \right. \quad (10)$$

where in the subscripts *min* and *max* mean the upper and lower bounds of the parameter, for example Δx_{min} and Δx_{max} , are the upper and lower bounds of the linear displacement of the end effector of the first-level arm, respectively; $D_{A1,i}$ and $D_{A2,i}$ are the allowable range of the angular displacement of the *i*th joints of two arms, respectively; the functions $\max[g]$ and $\min[g]$ refer to taking the maximum value and the minimum value of the function *g* respectively; $V_{A1,i,max}$ and $A_{A1,i,max}$ are the maximum values of the angular velocity and acceleration allowed by the *i*th joints of the first-level arm, respectively; $DI(t)$ is the minimum distance function of TACCS; and DI_{thred} denotes the threshold value taken for the reliable obstacle avoidance.

Therefore, even if the allowable ranges or maximum values of the angular displacements, velocity and acceleration of all joints of each arm are the same, there are 14 constraint conditions in the above model. In order to increase the optimization efficiency, the upper and lower values of variables Δx , Δy , Δz , α , β and γ need to be limited as much as possible. For this purpose, this work proposed a feasible space analysis method based on the IK functions of two arms. In this method, the Monte Carlo method is firstly used to randomly take values for variables Δx , Δy , Δz , α , β , γ ; then the target pose of the end effector of the first-level arm and the base frame of the second-level arm can be determined, based on which the IK functions of two arms can be solved; finally, values of variables with valid solutions in two IK problems can be recorded, based on which the feasible space of frame 2 (as Figure 2) is obtained, and can help to determine the upper and lower bounds of the corresponding variable. The framework of this method is shown in Figure 7.

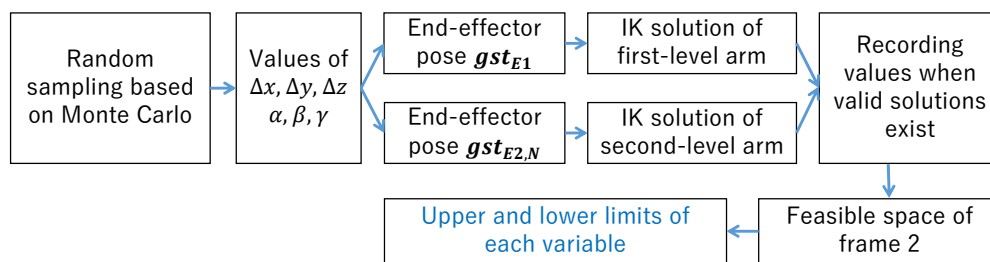


Figure 7. Framework of the feasible space analysis based on IK functions of two arms.

Since the number of variables and constraint conditions have a major impact on the convergence time and effect of the optimization, the bi-layer optimization strategy is also proposed in this work. In this strategy, the optimization problem is decoupled into two layers of sub-optimizations: the first layer of sub-optimization is the end-effector pose optimization of the first-level arm considering the kinematics and obstacle-avoidance constraints, as well as its influence on the operational dexterity of the second-level arm; the second layer includes two sub-optimizations, which are the motion trajectory optimizations of two arms, respectively, based on the optimal result of the first layer. Moreover, the bi-layer optimization problem also considers the motion time and energy consumption of the arms. Therefore, the first sub-optimization takes the sum of Euclidean distances of all joints of two arms as the object, which refers to the equivalent index of the energy consumption; the other sub-optimizations take the motion time and the length of motion path of the arm as the object. Then, the objectives of three sub-optimizations can be expressed by

$$f_{B-1} = \sqrt{\sum_{i=1}^{D_{A1}} \theta_{A1,i}^2 + \sum_{i=1}^{D_{A2}} \theta_{A2,i}^2} \tag{11}$$

$$f_{B-2} = w_1 T_{A1} + w_2 w' \sum_{i=1}^{D_{A1}} \int_{t=0}^{T_{A1}} S_{A1,i}(t) \tag{12}$$

$$f_{B-3} = w_1 T_{A2} + w_2 w' \sum_{i=1}^{D_{A2}} \int_{t=0}^{T_{A2}} S_{A2,i}(t) \tag{13}$$

wherein $\theta_{A1,i}$ and $\theta_{A2,i}$ are the inverse solutions of the i th joint at the target pose of the end effector for two arms, respectively.

The bi-layer optimization framework is shown in Figure 8, wherein there exists a global judgement for obstacle avoidance of the whole robotic motion before the second-layer sub-optimizations. From this framework, the optimal values of the first-layer optimization is the input for sub-optimization of the second-layer optimization, which means two sub-optimizations can perform parallel computing and output optimal trajectories. In actual applications, the parallel computing can largely improve the execution efficiency of the bi-layer optimization algorithm.

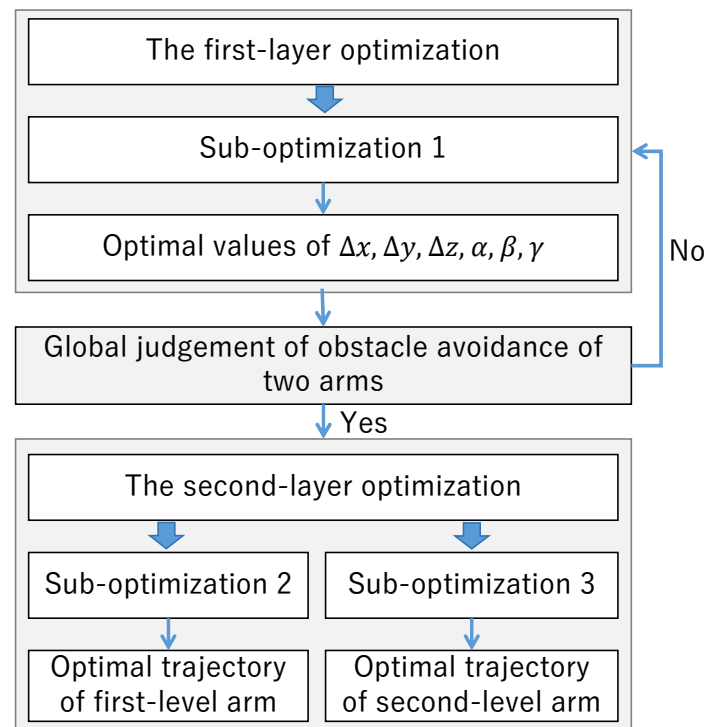


Figure 8. Bi-layer optimization framework.

Moreover, in the global judgement, the motion times of two arms are firstly taken reasonably to determine the overall robotic poses at different motion times, since their values have a minimal influence on the robotic pose at a certain point of the trajectory but largely affect the motion speed and acceleration of each joint; then, the smallest distance between the TACCS and its surrounding obstacles during motion can be calculated based on the model of the minimum distance of Section 3; finally, the smallest distance is compared with its threshold value (taken to ensure the absolutely safety of the system) to complete the judgement, which can detect the possible collisions that may happen during motion accurately and reliably. In the bi-layer optimization framework, if the global judgement condition is not satisfied, it then returns to the first-layer optimization, or else it continues the second-layer optimizations. The above strategy can further help improve the efficiency of the whole optimization, which can be modeled as follows:

$$\left. \begin{aligned}
 & \text{minimize : } f = f_{B-1}(\Delta x, \Delta y, \Delta z, \alpha, \beta, \gamma) \\
 & \text{subject to : } \Delta x \in [\Delta x_{min}, \Delta x_{max}] \\
 & \quad \Delta y \in [\Delta y_{min}, \Delta y_{max}] \\
 & \quad \Delta z \in [\Delta z_{min}, \Delta z_{max}] \\
 & \quad \alpha \in [\alpha_{min}, \alpha_{max}] \\
 & \quad \beta \in [\beta_{min}, \beta_{max}] \\
 & \quad \gamma \in [\gamma_{min}, \gamma_{max}] \\
 & \quad \theta_{A1,i} \in D_{A1,i}; \theta_{A2,i} \in D_{A2,i} \\
 & \quad DI_E \geq DI_{thred} \\
 & \quad \Downarrow \\
 & \text{Judgement : } \min[DI(t)] \geq DI_{thred} \\
 & \quad \Downarrow \\
 & \text{minimize : } f = f_{B-2}(T_{A1}) \\
 & \text{subject to : } S_{A1,i}(t) \in D_{A1,i} \\
 & \quad \max \left[\left| \dot{S}_{A1,i}(t) \right| \right] \leq V_{A1,i,max} \\
 & \quad \max \left[\left| \ddot{S}_{A1,i}(t) \right| \right] \leq A_{A1,i,max} \\
 & \quad T_{A1} \leq T_{A1,max} \\
 & \quad \Downarrow \\
 & \text{minimize : } f = f_{B-3}(T_{A2}) \\
 & \text{subject to : } S_{A2,i}(t) \in D_{A2,i} \\
 & \quad \max \left[\left| \dot{S}_{A2,i}(t) \right| \right] \leq V_{A1,i,max} \\
 & \quad \max \left[\left| \ddot{S}_{A2,i}(t) \right| \right] \leq A_{A2,i,max} \\
 & \quad T_{A2} \leq T_{A2,max}
 \end{aligned} \right\} \tag{14}$$

wherein DI_E means the minimum distance when the TACCS is at its target pose.

In this work, the particle swarm optimization algorithm (PSO) is adopted to solve the optimization models proposed above; the PSO is an intelligent algorithm that searches for the optimal solution in the solution space by a certain number of particles and has advantages of easy implementation and fewer algorithm parameters. Based on the calculation functions of IK and the minimum distance of TACCS, pseudo codes of algorithms based on the single and bi-layer optimizations can be written as Algorithm A1 and Algorithm A2 in Appendix A. For the PSO, parameters include inertia weight ω , maximum flying velocity v_{max} of particles and the number of particles N_p , and they have major impacts on the convergence speed and optimization effect. Among them, the value of v_{max} for each variable can be taken by 10% to 20% of its searching range, and if the ranges for the variables are different, the ratio φ of the value to its corresponding range can be taken as a uniform parameter for values of the maximum velocities of all variables, $0.1 \leq \varphi \leq 0.2$. Therefore, for the motion planning optimization, reasonable values of PSO parameters (ω , φ and N_p) should be taken to obtain the optimal results according to several trails of each optimization. Moreover, in order to ensure obtain the global optimal result, the stopping criterion for iteration is defined by $\max[|f_{k-\tau-1} - f_{k-\tau}|, 0 \leq \tau \leq 14] < \varepsilon$, which means that the change in the objective value during fourteen successive iterations is less than a threshold value $\varepsilon = 0.001$.

5. Verification Based on Simulation Cases

In order to show the performance of two proposed optimization algorithms, this section designed two example cases. The principle of determining initial conditions (including the target pose and locations of obstacles) for the simulations involves ensuring the solvability of the optimization problem, i.e., there is at least one collision-free path during

the conversion from the initial pose to the target pose. The operational environment of case 2 is shown in Figure 4 (the target structure is located on one side of capsule), and for that of case 1, the target structure and its surrounding obstacles are located on the top of the capsule. The target pose and the initial 3D locations of the feature points of the TACCS and obstacles in frame 1 (Table A1) are given for the following simulation analysis based on the above basic principle.

From the proposed method, there also exists one important step before the optimization, which is to determine the upper and lower values of variables Δx , Δy , Δz , α , β and γ according to the framework shown in Figure 8. Figures 9 and 10 show the feasible spaces of variables in two example cases, wherein valid solutions of variables Δx , Δy and Δz and that of variables α , β and γ are plotted as points in a 3D coordinate system, respectively, which finally form point clouds to show the feasible spaces.

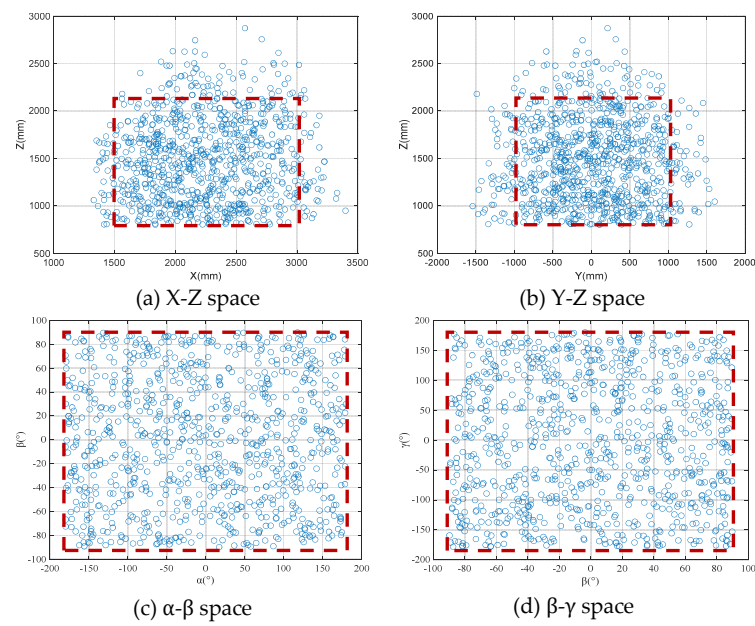


Figure 9. Feasible spaces in example case 1.

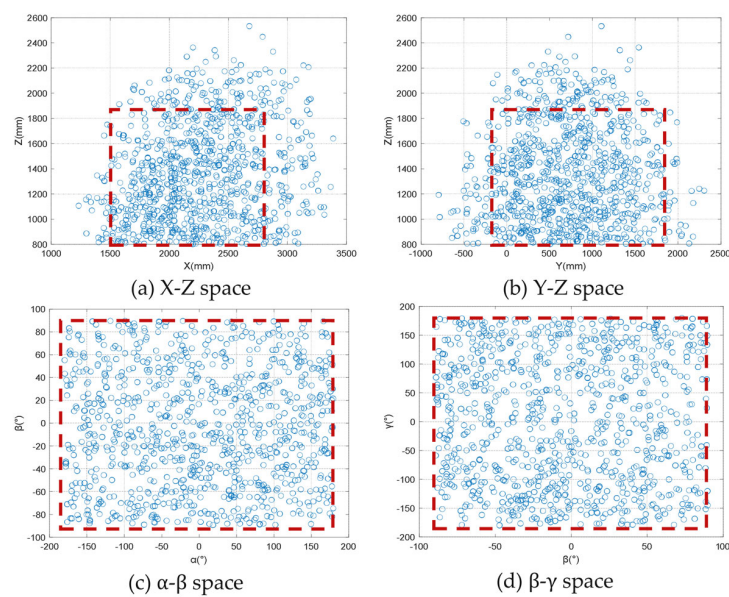


Figure 10. Feasible spaces in example case 2.

In each sub-figure of Figures 9 and 10, the red box is used to show the final space size chosen to further decrease the allowable ranges of variables that can help improve the optimization efficiency. The box selection principles include (1) trying to select regions with dense point clouds, which refers to the point clouds with higher density; (2) ensuring that the proportion of selected points to all points is greater than a reasonable value; and (3) taking the limits of the red box based on the above proportion requirement. From Figures 9 and 10, since the density of the points is uniform, the regions for variables α , β and γ are taken as the whole; the distribution of point clouds for variables Δx , Δy and Δz features sparse edges and a dense middle, and so a reasonable value of the proportion can largely improve the optimization efficiency under the premise of ensuring the optimization effect. In order to show the influence of the proportion of variables Δx , Δy and Δz on the optimization effect, the iterative curves of the sub-optimization 1 (since the size of the allowable range of variables Δx , Δy and Δz has a major impact on sub-optimization 1 of the bi-layer optimization) and the single optimization under different values of the proportion (50%, 70%, 90% and 100%) are shown in Figure 11, respectively, from which, when the proportion is taken by 90%, the optimization efficiency can be largely improved with only a small loss of the objective value. From two figures, the allowable ranges of six variables in two example cases can be finally obtained.

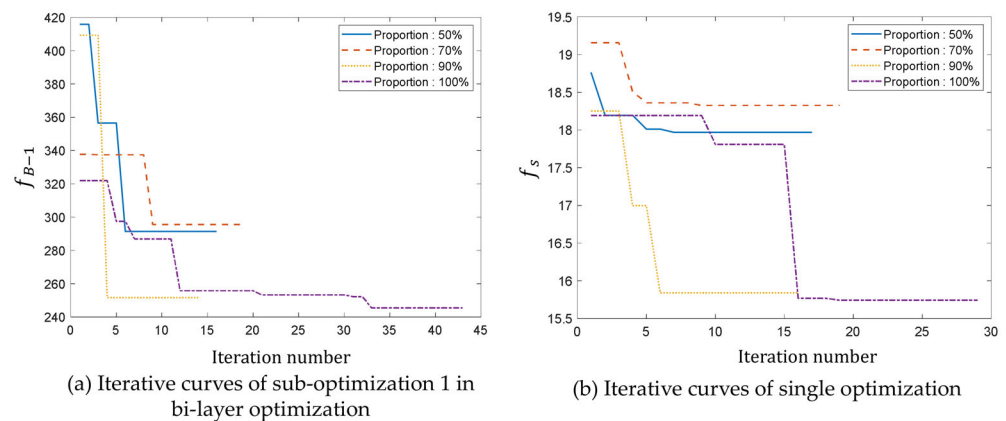


Figure 11. Iterative curves of the sub-optimization 1 in bi-layer optimization and the single optimization under different values of the proportion (50%, 70%, 90% and 100%).

Moreover, as presented in Section 4, PSO parameters (including the inertia weight ω , the velocity ratio φ and the particle number N_p) have a major influence on the optimization effect, and they are always related to ranges of variables and the complexity of the optimization problem. Therefore, different values for these parameters should be taken to perform the optimizations before cases analysis, aiming to determine the reasonable values for PSO parameters and ensure the optimization effect. Iterative curves for the single optimization and three sub-optimizations of the bi-layer optimization with different values of PSO parameters are shown in Figure 12. From this figure, the reasonable values for the PSO parameters can be determined as $[\omega = 0.8, \varphi = 0.15, N_p = 15]$, $[\omega = 1.2, \varphi = 0.15, N_p = 10]$ and $[\omega = 0.8, \varphi = 0.10, N_p = 10]$ for the first-layer optimization (sub-optimization 1), the second-layer sub-optimizations of the bi-layer optimization (sub-optimizations 2 and 3) and the single optimization, respectively, which can ensure the fast convergence speed and global optimization capability.

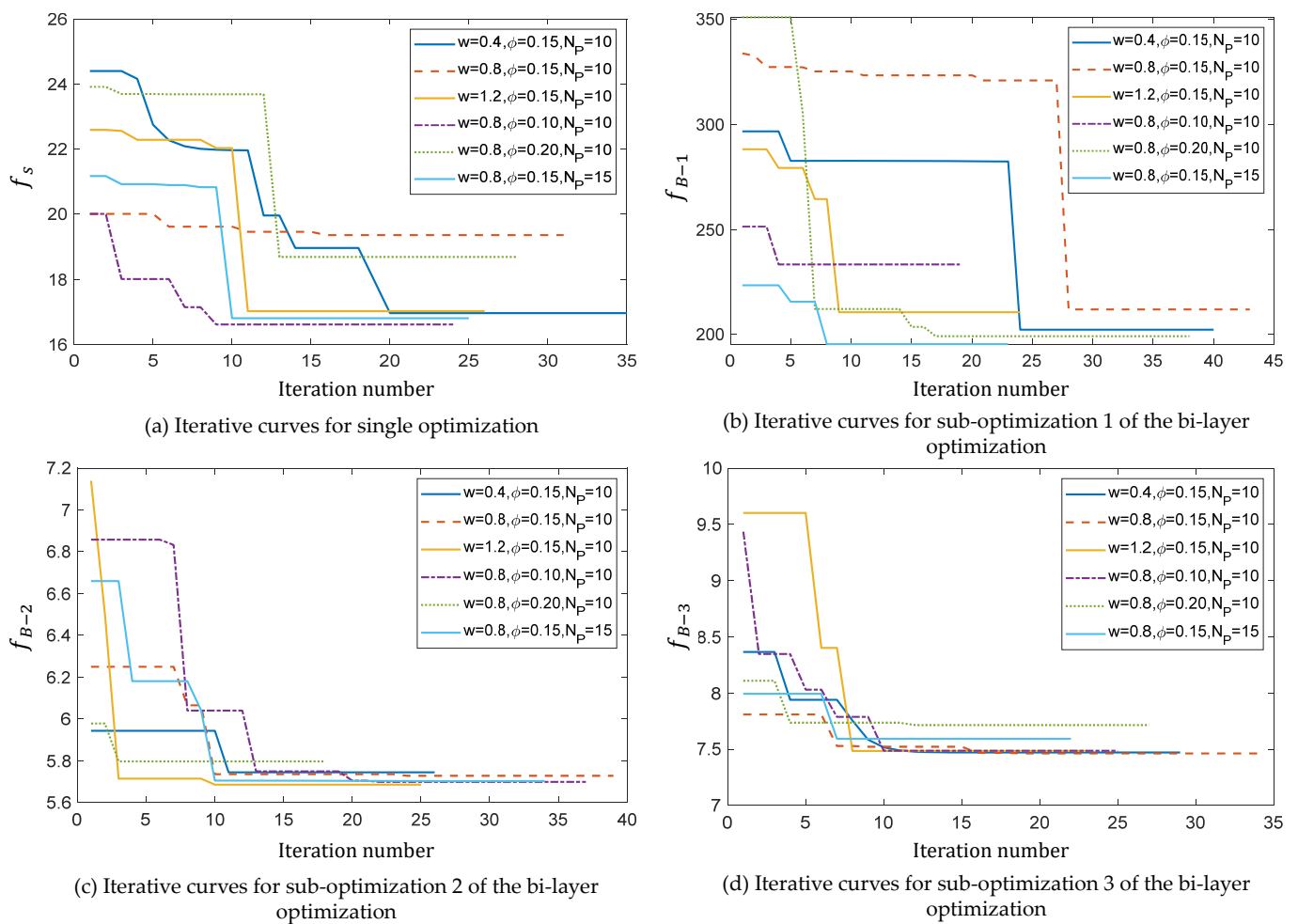


Figure 12. Iterative curves for the single optimization and three sub-optimizations of the bi-layer optimization with different values of PSO parameters.

Since the randomness of PSO iterations has a major impact on the convergence time of optimization, in order to analysis the influence degrees of the randomness for two proposed algorithms, the single and bi-layer optimizations for two example cases are performed 30 times. The comparison of convergence times of two optimization algorithms is shown in Table 3, wherein μ , σ and c_v mean the average convergence time, the standard deviation and the coefficient of variation of 30 optimizations, respectively, while μ is used to evaluate the average optimization efficiency, σ and c_v are used to evaluate the robustness of the algorithms, σ can reflect the dispersion of convergence time, $\sigma = \sqrt{\frac{1}{30} \sum_{i=1}^{30} (t_i - \mu)^2}$ and c_v is the dimensionless representation of σ , $c_v = \sigma / \mu \times 100\%$, (and it can be used to compare the convergence performances of different optimizations). From the result, we can obtain the following conclusions: (1) The average convergence times of the bi-layer optimization algorithm in two cases are reduced by 72.43% and 63.64%, respectively, compared with that of the single optimization algorithm, which are calculated by $(\mu_s - \mu_b) / \mu_s \times 100\%$, wherein μ_s and μ_b are the average convergence time of the single and bi-layer algorithms. The reason is that reducing the number of variables can largely increase the optimization speed; additionally, the collision judgement during the whole motion is placed outside the optimization iteration process. (2) Values of the dispersion of the bi-layer optimization algorithm in two cases are reduced by 79.52% and 82.53%, respectively, and values of the coefficient of variation in two cases are reduced by 25.68% and 51.90%, respectively, compared with that of the single optimization algorithm. That is, the randomness of PSO iterations has a greater impact on the convergence time of the single optimization

algorithm. (3) Although the initial conditions have a major impact on the convergence time of optimizations, the average convergence times of the single optimization in two cases are obviously larger than that of the bi-layer optimization, which indicates that the advantage of bi-layer optimization is confidently certain under different initial conditions.

Table 3. Comparison of convergence times of two optimization algorithms.

Example Case	Optimization	μ/s	σ/s	$c_v/\%$
Case 1	Single	145.54	51.67	35.51
	Bi-layer	40.12	10.58	26.39
	Difference	72.43%	79.52%	25.68%
Case 2	Single	52.48	18.66	35.55
	Bi-layer	19.08	3.26	17.10
	Difference	63.64%	82.53%	51.90%

The constraints and the optimal results of the main parameters based on three algorithms, including the bi-layer optimization, single optimization and traditional optimization, are listed in Table 4, wherein V_{max} and A_{max} mean the maximum values of angular velocity and acceleration of all robotic joints and the values of difference 1 and difference 2 are calculated by $|p_s - p_b|/p_s \times 100\%$ and $|p_t - p_b|/p_t \times 100\%$, while p_s , p_b and p_t represent the parameter values based on single optimization, bi-layer optimization and traditional optimization, respectively. The results show that values for parameters Δx , Δy , Δz , α , β , γ , V_{max} and A_{max} , based on three algorithms, are all less than their corresponding limits given in constraints; compared with the bi-layer optimization, the overall optimization objectives in two cases obtained based on the traditional optimization increase 6.94% and 4.36%, respectively. In particular, the energy consumption increases 11.21% in case 1, and the motion time increases 22.75% in case 2; the differences of the overall optimization objectives based on the bi-layer optimization and the single optimization in two cases are only 3.21% and 1.33%, respectively. The main reasons for the differences include (1) the hierarchical processing of the optimization problem taking into account the end-effector pose of the first-level arm on the motion of the second-level arm, which can rapidly find the optimal result and finally improve the optimization efficiency, but it also sacrifices a small amount of the valid solution space, which leads to some objectives based on single optimization being potentially less than that based on bi-layer optimization (for example, the overall objective based on single optimization is 3.21% smaller than that based on bi-layer optimization in case 1); (2) in spite of the above reasons, it is very difficult for single optimization to find the real global optimal solution, so its objectives may be higher than that of the bi-layer optimization (for example, the overall objective based on single optimization is 1.33% larger than that based on bi-layer optimization). For the constraints based on bi-layer optimization, the maximum velocity and acceleration of robotic joints are 77.71 deg/s and 103.80 deg/s², and they are obviously less than their allowable values 120 deg/s and 200 deg/s², respectively. Also, Figure 13 gives the variation in the shortest distance between the TACCS and its obstacles over motion time based on two algorithms, which indicates that the optimal trajectories of the TACCS based on two algorithms can ensure the motion safety of the system; additionally, in order to clearly show the details of the shortest distance between the TACCS and obstacles, the corresponding structure no. and obstacle no. are also shown in Figure 10 (for example, R2-O1 means the distance between the second robotic structure and the first obstacle is the shortest distance, wherein the no. information can be found in Table A1).

Table 4. Constraints, optimal results of main parameters based on three algorithms, including the bi-layer opt. (bi-layer optimization), single opt. (single optimization) and trad. opt. (traditional optimization).

Example Case	Parameter	Constraints	Bi-Layer Opt.	Single Opt.	Difference 1	Trad. Opt.	Difference 2
Case 1	Δx /(mm)	[1500, 3000]	2016.08	1628.35	-	2500.00	-
	Δy /(mm)	[-1000, 1000]	604.94	675.01	-	815.81	-
	Δz /(mm)	[820, 2200]	2148.84	1797.81	-	2097.21	-
	α /(deg)	[-180, 180]	92.66	-24.45	-	104.74	-
	β /(deg)	[-90, 90]	76.65	64.71	-	5.24	-
	γ /(deg)	[-180, 180]	86.73	2.64	-	90.00	-
	V_{max} /(deg/s)	≤ 120	77.71	64.51	-	55.59	-
	A_{max} /(deg/s ²)	≤ 200	103.80	95.67	-	54.13	-
	T (s)	≤ 20	4.67	4.98	6.22%	4.51	3.55%
	$Path$ (deg)	-	9.82	9.06	8.39%	11.06	11.21%
Case 2	F	-	14.49	14.04	3.21%	15.57	6.94%
	Δx /(mm)	[1500, 2800]	1871.38	1652.65	-	1653.14	-
	Δy /(mm)	[-200, 1800]	1102.91	1246.74	-	1214.89	-
	Δz /(mm)	[800, 1880]	1394.95	1521.54	-	1619.18	-
	α /(deg)	[-180, 180]	22.37	-77.82	-	67.21	-
	β /(deg)	[-90, 90]	72.61	25.74	-	45.72	-
	γ /(deg)	[-180, 180]	17.02	-72.95	-	59.02	-
	V_{max} /(deg/s)	≤ 120	95.97	96.94	-	81.29	-
	A_{max} /(deg/s ²)	≤ 200	136.82	138.06	-	115.61	-
	T (s)	≤ 20	4.21	4.37	3.66%	5.45	22.75%
$Path$ (deg)	-	11.37	11.42	0.44%	10.84	4.89%	
F	-	15.58	15.79	1.33%	16.29	4.36%	

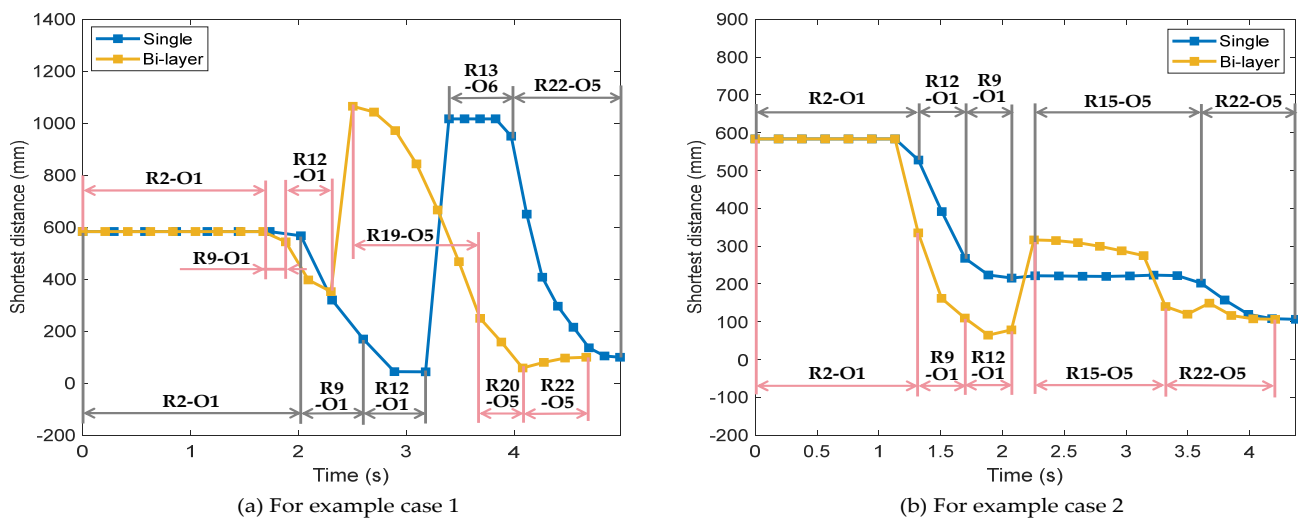


Figure 13. Variation in the shortest distance between TACCS and its obstacles over motion time based on two algorithms.

Moreover, taking example case 1 as the objective, curves of angular parameters of each joint of TACCS and a series of motion poses relative to capsule based on the bi-layer optimization algorithm are shown in Figures 14 and 15, and those based on the single optimization algorithm are shown in Figures A1 and A2, which can further intuitively demonstrate the optimization effect of two algorithms, including the guarantee of the motion stability and safety. Moreover, taking example case 2 as the objective, a series of motion poses relative to the capsule based on the bi-layer and single optimization algorithms are shown in Figures A3 and A4.

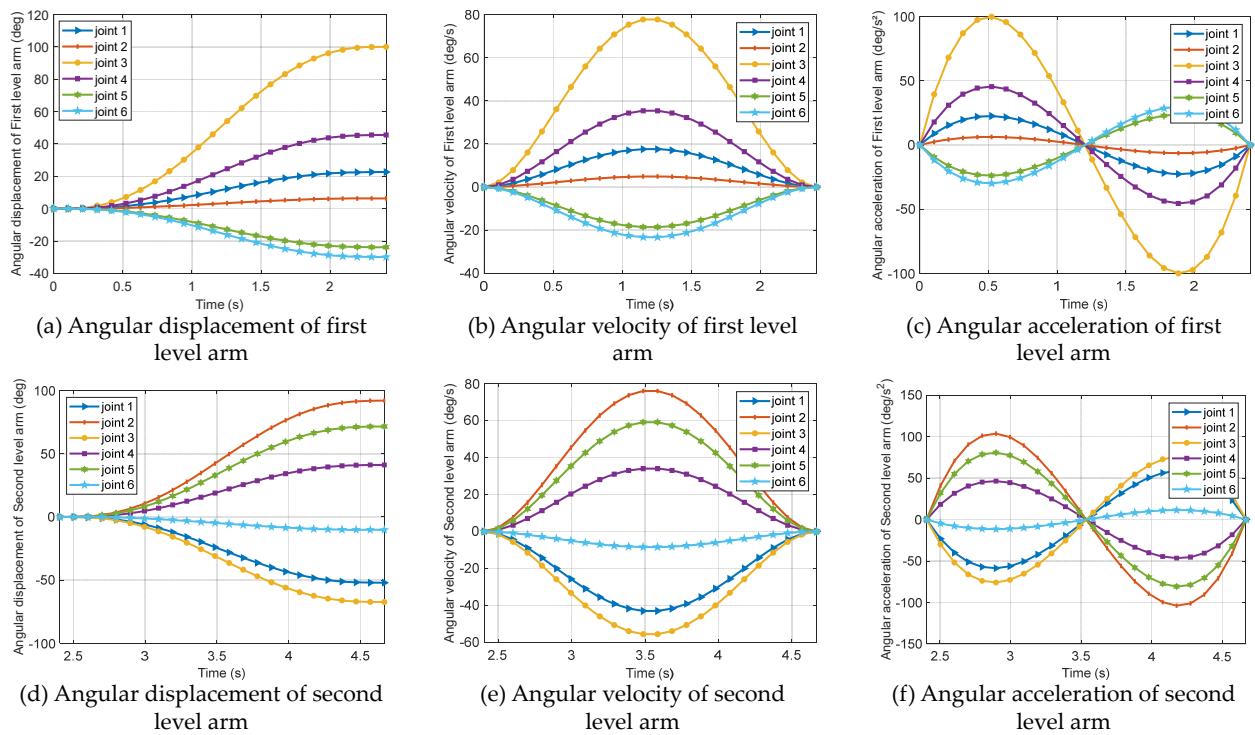


Figure 14. Curves of angular parameters of each joint of TACCS based on bi-layer optimization.

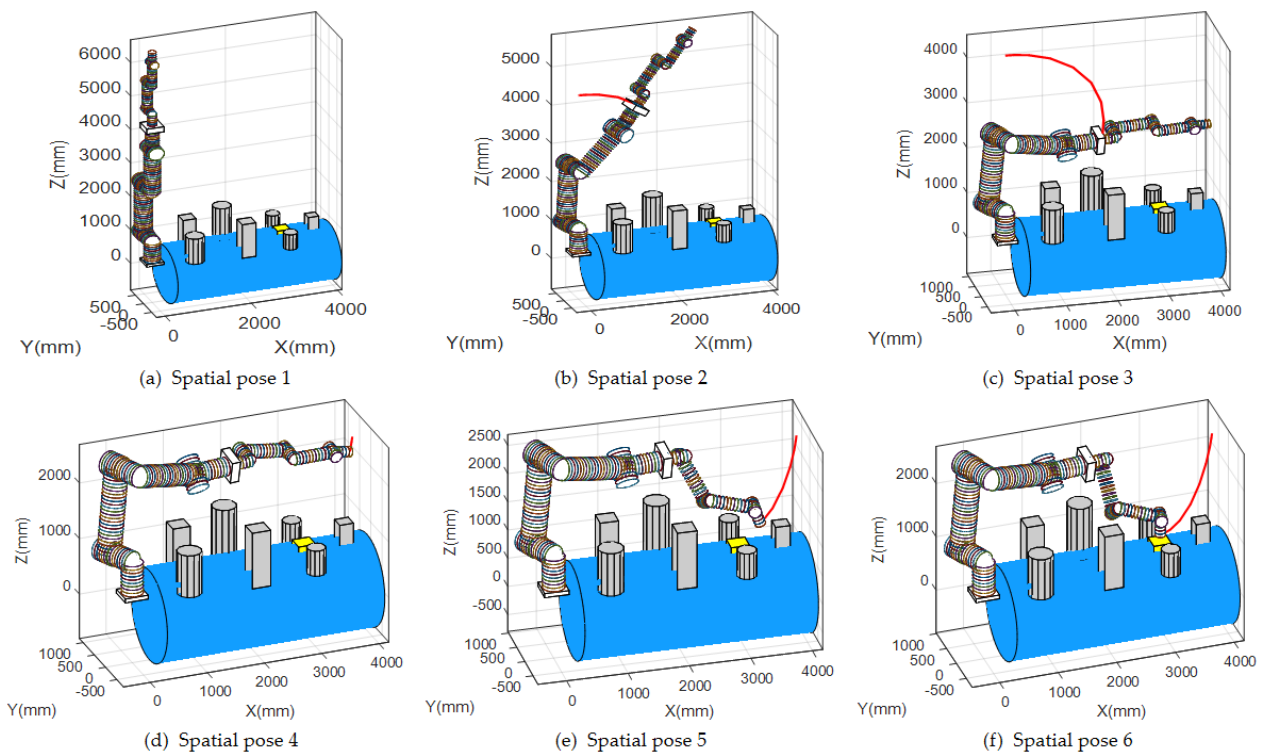


Figure 15. Series of motion poses relative to capsule based on bi-layer optimization.

From the comprehensive analysis results of convergence speed and optimization effect, it can be concluded that the bi-layer optimization has a higher execution efficiency under the premise of ensuring the optimization effect.

6. Conclusions and Future Works

6.1. Conclusions

This work proposed the single and bi-layer optimization algorithms for the collision-free motion planning problem of the TACCS, aiming to optimize the operating time and energy consumption and considering both the kinematics and obstacle-avoidance constraints under the complex operational environment. For the implementation of optimizations, the functions for the inverse kinematics of TACCS and the shortest distance between the TACCS and its obstacles are established based on screw theory and a hybrid geometric envelope, respectively. During the optimization problem modeling, a single optimization problem with more than eight variables is converted into three optimization sub-problems with fewer variables of bi-layer optimization, which aims to improve the convergence efficiency of optimizations. Finally, two algorithms are applied into two example cases. Through experimental simulations, three main conclusions are obtained as follows:

- (1) The reasonable PSO parameters are selected to perform optimizations of two algorithms, both considering their influences on convergence time and optimization effect. Moreover, a box selection principle is designed to determine the allowable ranges of motion variables of the first level arm, which can further improve the execution speed of algorithms by sacrificing very little workspace. The above strategies can help ensure and improve the optimization performances of two algorithms.
- (2) The performance indexes are taken as the average value and dispersion of convergence times of 30 executions, which are obtained in two example cases and based on two algorithms. The comparison of performance indexes indicates that the bi-layer optimization has a higher convergence speed than the single optimization, which can be increased by more than 60%, and the randomness of the PSO iterations has less of an impact on the convergence speed of the bi-layer optimization.
- (3) The optimal results based on two algorithms are compared to show the significance of this work. The results indicate that values of motion parameters and shortest distances during the whole motion are less than their corresponding limits given in constraints; additionally, the overall optimization objective based on the bi-layer optimization is smaller than that based on the single optimization. Combining with the comparison result of convergence speed, it can be concluded that the bi-layer optimization has a higher execution efficiency under the premise of ensuring the optimization effect.

6.2. Future Works

In this work, the proposed algorithms can be used for the collision-free motion planning of the TACCS under the complex structured environment, for which the locations of obstacles and targets are given information, i.e., they are fixed or regularly changed. Therefore, this work cannot solve the real-time collision-free motion planning of the TACCS in randomly dynamic scenarios, which will be one focus of future works. Moreover, the real application of the proposed algorithms will be the other focus of future works, for which we will actively seek cooperation opportunities with aerospace enterprises and apply our result in actual scenarios.

Author Contributions: Conceptualization and methodology, J.X.; validation and writing—original draft preparation, L.T. and Y.P.; writing—review and editing, all authors; supervision, Q.C. and H.C.; project administration, J.X. and T.Z.; funding acquisition, Q.C., H.C. and T.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China (No. 2023YFB3406400), the Joint Funds of the National Natural Science Foundation of China (No. U23B20104), the 2022 industrial technology basic public service platform (No. 2022-232-223-02), and the R&D Program of the Beijing Municipal Education Commission (No. KM202210005031).

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

Conflicts of Interest: Author Yanhu Pei was employed by the Genertec Machine Tool Engineering Research Institute Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The Genertec Machine Tool Engineering Research Institute Co., Ltd. had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Nomenclature

TACCS	Two-arm cascade combination system: a system that one manipulator is mounted at the end effector of the other manipulator.
IK	Inverse kinematics: a mapping from the end-effector pose to the angular displacements of joints for one manipulator.
DOFs	Degrees of freedom.
BCF	Base coordinate frame of the manipulator.
SMP	Motion planning algorithms of two subsystems are carried separately for the mobile manipulator.
WMP	Motion planning of the high-DOFs system is carried in a whole.
A* algorithm	A-star algorithm: a heuristic search algorithm to find the shortest path from the starting point to the target point in a graph.
RRT	Rapidly exploring random trees: build a tree by random sampling and gradually expand the tree to approach the target point, eventually finding a viable path from the starting point to the end point.
PRM	Probabilistic roadmap: the possible motion path of the robot is constructed by probabilistically sampling points in the configuration space and establishing edges between the connectable points.
PSO	Particle swarm optimization.
PSO-WOA	A novel hybrid heuristic algorithm, which combined PSO and whale optimization algorithm.
GMOPSO	A modified multi-objective PSO algorithm.
CF-TEM	A collision-free and time–energy–minimum trajectory planning optimization method.
EECF	End-effector coordinate frame of the manipulator.
3D coordinate	Three-dimensional coordinate.
4D vector	Four-dimensional vector.

Appendix A. Optimization Algorithms

Based on the calculation functions of IK and minimum distance of TACCS, pseudo codes of algorithms based on the single and bi-layer optimizations can be written as Algorithm A1 and Algorithm A2 as follows.

Algorithm A1 Pseudo Codes of the Single Optimization Algorithm

1.	Input: gst
2.	PSO parameters setting $N_p, \omega, c_1, c_2, \varphi$;
3.	Initialize: $[\Delta x, \Delta y, \Delta z, \alpha, \beta, \gamma, T_{A1}, T_{A2}]$
4.	while count ≤ 15 do
5.	for $n = 1$ to N_p do
6.	Calculate $\theta_{A1} \leftarrow fun_IK_{A1}([\Delta x, \Delta y, \Delta z, \alpha, \beta, \gamma])$,
7.	Calculate $\theta_{A2} \leftarrow fun_IK_{A2}(gst, [\Delta x, \Delta y, \Delta z, \alpha, \beta, \gamma])$,
8.	if $\theta_{A1} \neq \emptyset$ && $\theta_{A2} \neq \emptyset$ then
9.	Determine $S_{A1,i}(t_{A1}), \dot{S}_{A1,i}(t_{A1}), \ddot{S}_{A1,i}(t_{A1}), DI(t_{A1})$;
10.	Determine $S_{A2,i}(t_{A2}), \dot{S}_{A2,i}(t_{A2}), \ddot{S}_{A2,i}(t_{A2}), DI(t_{A2})$;
11.	Calculate $\theta_{A1,i,min} = \min\langle S_{A1,i}(t_{A1}) \rangle$ and $\theta_{A1,i,max} = \max\langle S_{A1,i}(t_{A1}) \rangle, 0 \leq t_{A1} \leq T_{A1}$.
12.	Calculate $\theta_{A2,i,min} = \min\langle S_{A2,i}(t_{A2}) \rangle$ and $\theta_{A2,i,max} = \max\langle S_{A2,i}(t_{A2}) \rangle, 0 \leq t_{A2} \leq T_{A2}$
13.	Calculate $\dot{\theta}_{A1,i,min} = \min\langle \dot{S}_{A1,i}(t_{A1}) \rangle$ and $\dot{\theta}_{A1,i,max} = \max\langle \dot{S}_{A1,i}(t_{A1}) \rangle, 0 \leq t_{A1} \leq T_{A1}$.
14.	Calculate $\dot{\theta}_{A2,i,min} = \min\langle \dot{S}_{A2,i}(t_{A2}) \rangle$ and $\dot{\theta}_{A2,i,max} = \max\langle \dot{S}_{A2,i}(t_{A2}) \rangle, 0 \leq t_{A2} \leq T_{A2}$
15.	Calculate $\ddot{\theta}_{A1,i,min} = \min\langle \ddot{S}_{A1,i}(t_{A1}) \rangle$ and $\ddot{\theta}_{A1,i,max} = \max\langle \ddot{S}_{A1,i}(t_{A1}) \rangle, 0 \leq t_{A1} \leq T_{A1}$.

Algorithm A1 Cont.

```

16. Calculate  $\ddot{\theta}_{A2,i,min} = \min\langle \ddot{S}_{A2,i}(t_{A2}) \rangle$  and  $\ddot{\theta}_{A2,i,max} = \max\langle \ddot{S}_{A2,i}(t_{A2}) \rangle$ ,  $0 \leq t_{A2} \leq T_{A2}$ 
17. Calculate  $DI_{A1,min} = \min\langle DI(t_{A1}) \rangle$ ,  $0 \leq t_{A1} \leq T_{A1}$ 
18. Calculate  $DI_{A2,min} = \min\langle DI(t_{A2}) \rangle$ ,  $0 \leq t_{A2} \leq T_{A2}$ 
    if  $\theta_{A1/A2,i,min}, \theta_{A1/A2,i,max} \in D_{A1/A2,i}$  &&  $\dot{\theta}_{A1/A2,i,min}, \dot{\theta}_{A1/A2,i,max} \in \dot{D}_{A1/A2,i}$  &&
19.  $\ddot{\theta}_{A1/A2,i,min}, \ddot{\theta}_{A1/A2,i,max} \in \ddot{D}_{A1/A2,i}$  &&  $DI_{A1/A2,min} \leq DI_{th}$  then
    Calculate  $fit$ ,
    if  $fit_n < f_p$  then
    21.  $f_{p,n} = fit_n; x_{p,n} = x_n$ .
    22.
    23. end if
    24. end if
    25. end if
    26. end for
    27. if  $\min(f_p) < f_g$  then
    28.  $[f_g, num] = \min(f_p); x_g = x_{p,num}$ .
    29. end if
    30. if  $f_{g,k-1} - f_{g,k} \leq 0.001$  then
    31. count = count + 1
    32. else
    33. count = 1
    34. end if
    35. Update the current flying velocities and positions of all particles;
    36. end while

```

Algorithm A2 Pseudo Codes of the Bi-Layer Optimization Algorithm

```

1. Input:  $gst$ 
2. PSO parameters setting  $N_p, \omega, c_1, c_2, \varphi$ ;
3. Initialize:  $[\Delta x, \Delta y, \Delta z, \alpha, \beta, \gamma]$ 
4. while condition  $\leq 0$  do
5. while count1  $\leq 15$  do
6. for  $n = 1$  to  $N_p$  do
7. Calculate  $\theta_{A1} \leftarrow fun\_IK_{A1}([\Delta x, \Delta y, \Delta z, \alpha, \beta, \gamma])$ ,
8. Calculate  $\theta_{A2} \leftarrow fun\_IK_{A2}(gst, [\Delta x, \Delta y, \Delta z, \alpha, \beta, \gamma])$ ,
9. if  $\theta_{A1} \neq \emptyset$  &&  $\theta_{A2} \neq \emptyset$  &&  $\theta_{A1/A2} \in D_{A1/A2}$  then
10. Calculate  $DI_{\theta_{A1}}, DI_{\theta_{A2}}$ 
11. if  $DI_{\theta_{A1/A2}} \leq DI_{th}$  then
12. Calculate  $fit1$ ,
13. if  $fit1_n < f1_p$  then
14.  $f1_{p,n} = fit1_n; x1_{p,n} = x1_n$ .
15. end if
16. end if
17. end if
18. end for
19. if  $\min(f1_p) < f1_g$  then
20.  $[f1_g, num] = \min(f1_p); x1_g = x1_{p,num}$ .
21. end if
22. if  $f1_{g,k-1} - f1_{g,k} \leq 0.001$  then
23. count1 = count1 + 1
24. else
25. count1 = 1
26. end if
27. Update the current flying velocities and positions of all particles;
28. end while
29. Calculate  $DI_{A1,min} = \min\langle DI(t_{A1,C1}) \rangle$ ,  $0 \leq t_{A1,C1} \leq T_{A1,C1}$ 
    Calculate  $DI_{A2,min} = \min\langle DI(t_{A2,C2}) \rangle$ ,  $0 \leq t_{A2,C2} \leq T_{A2,C2}$ 
31. if  $DI_{A1/A2,min} \leq DI_{th}$  then
32. condition = 1
33. end if
34. end while
35. Initialize:  $T_{A1}$ 
36. while count2  $\leq 15$  do
37. for  $n = 1$  to  $N_p$  do
38. Determine  $S_{A1,i}(t_{A1}), \dot{S}_{A1,i}(t_{A1}), \ddot{S}_{A1,i}(t_{A1}), DI(t_{A1});$ 

```

Algorithm A2 Cont.

```

39. Calculate  $\theta_{A1,i,min} = \min\langle \dot{S}_{A1,i}(t_{A1}) \rangle$  and  $\theta_{A1,i,max} = \max\langle \dot{S}_{A1,i}(t_{A1}) \rangle$ ,  $0 \leq t_{A1} \leq T_{A1}$ .
40. Calculate  $\ddot{\theta}_{A1,i,min} = \min\langle \ddot{S}_{A1,i}(t_{A1}) \rangle$  and  $\ddot{\theta}_{A1,i,max} = \max\langle \ddot{S}_{A1,i}(t_{A1}) \rangle$ ,  $0 \leq t_{A1} \leq T_{A1}$ .
41. Calculate  $\ddot{\theta}_{A1,i,min} = \min\langle \ddot{S}_{A1,i}(t_{A1}) \rangle$  and  $\ddot{\theta}_{A1,i,max} = \max\langle \ddot{S}_{A1,i}(t_{A1}) \rangle$ ,  $0 \leq t_{A1} \leq T_{A1}$ .
42. Calculate  $DI_{A1,min} = \min\langle DI(t_{A1}) \rangle$ ,  $0 \leq t_{A1} \leq T_{A1}$ 
    if  $\theta_{A1,i,min}, \theta_{A1,i,max} \in D_{A1,i}$  &&  $\theta_{A1,i,min}, \theta_{A1,i,max} \in \dot{D}_{A1,i}$  &&
43.  $\ddot{\theta}_{A1,i,min}, \ddot{\theta}_{A1,i,max} \in \ddot{D}_{A1,i}$  &&  $DI_{A1,min} \leq DI_{th}$  then
44. Calculate  $fit2$ ,
45. if  $fit2_n < f2_p$  then
46.  $f2_{p,n} = fit2_n$ ;  $x2_{p,n} = x2_n$ .
47. end if
48. end if
49. end for
50. if  $\min(f2_p) < f2_g$  then
51.  $[f2_g, num] = \min(f2_p)$ ;  $x2_g = x2_{p,num}$ .
52. end if
53. if  $f2_{g,k-1} - f2_{g,k} \leq 0.001$  then
54. count2 = count2 + 1
55. else
56. count2 = 1
57. end if
58. Update the current flying velocities and positions of all particles;
59. end while
60. Initialize:  $T_{A2}$ 
61. while count3  $\leq$  15 do
62. for  $n = 1$  to  $N_p$  do
63. Determine  $S_{A2,i}(t_{A2}), \dot{S}_{A2,i}(t_{A2}), \ddot{S}_{A2,i}(t_{A2}), DI(t_{A2})$ ;
64. Calculate  $\theta_{A2,i,min} = \min\langle \dot{S}_{A2,i}(t_{A2}) \rangle$  and  $\theta_{A2,i,max} = \max\langle \dot{S}_{A2,i}(t_{A2}) \rangle$ ,  $0 \leq t_{A2} \leq T_{A2}$ 
65. Calculate  $\dot{\theta}_{A2,i,min} = \min\langle \dot{S}_{A2,i}(t_{A2}) \rangle$  and  $\dot{\theta}_{A2,i,max} = \max\langle \dot{S}_{A2,i}(t_{A2}) \rangle$ ,  $0 \leq t_{A2} \leq T_{A2}$ 
66. Calculate  $\ddot{\theta}_{A2,i,min} = \min\langle \ddot{S}_{A2,i}(t_{A2}) \rangle$  and  $\ddot{\theta}_{A2,i,max} = \max\langle \ddot{S}_{A2,i}(t_{A2}) \rangle$ ,  $0 \leq t_{A2} \leq T_{A2}$ 
67. Calculate  $DI_{A2,min} = \min\langle DI(t_{A2}) \rangle$ ,  $0 \leq t_{A2} \leq T_{A2}$ 
    if  $\theta_{A2,i,min}, \theta_{A2,i,max} \in D_{A2,i}$  &&  $\theta_{A2,i,min}, \theta_{A2,i,max} \in \dot{D}_{A2,i}$  &&
68.  $\ddot{\theta}_{A2,i,min}, \ddot{\theta}_{A2,i,max} \in \ddot{D}_{A2,i}$  &&  $DI_{A2,min} \leq DI_{th}$  then
69. Calculate  $fit3$ ,
70. if  $fit3_n < f3_p$  then
71.  $f3_{p,n} = fit3_n$ ;  $x3_{p,n} = x3_n$ .
72. end if
73. end if
74. end for
75. if  $\min(f3_p) < f3_g$  then
76.  $[f3_g, num] = \min(f3_p)$ ;  $x3_g = x3_{p,num}$ .
77. end if
78. if  $f3_{g,k-1} - f3_{g,k} \leq 0.001$  then
79. count3 = count3 + 1
80. else
81. count3 = 1
82. end if
83. Update the current flying velocities and positions of all particles;
84. end while

```

Appendix B. Experimental Settings

For two example cases, the target position of the end effector (the target orientation is the same as the initial state) and the initial 3D locations of feature points of the TACCS and obstacles in frame 1 are given for the simulation implementation, and they are listed in Table A1 as follows.

Table A1. Target pose and the initial 3D locations of feature points of TACCS and obstacles in frame 1.

Structures	Characteristic Points		Coordinate			Radius/mm (Number of Edges)
	Structure Type	Number	X/mm	Y/mm	Z/mm	
Robot	Polygon 1	1	−200	−200	400	-
		2	200	−200	400	-
		3	200	200	400	-
		4	−200	200	400	-
		5	−200	−200	470	-
		6	200	−200	470	/
		7	200	200	470	/
		8	−200	200	470	/
	Cylinder 2	1	0	0	470	180
		2	0	0	1010	180
	Cylinder 3	1	0	0	1010	180
		2	0	405	1010	180
	Cylinder 4	1	0	405	1010	198
		2	0	740	1010	198
	Cylinder 5	1	0	540	1010	198
		2	0	540	2504	198
	Cylinder 6	1	0	405	2504	198
		2	0	740	2504	198
Cylinder 7	1	0	405	2504	180	
	2	0	0	2504	180	
Cylinder 8	1	0	0	2504	180	
	2	0	0	3764	180	
Cylinder 9	1	0	−270	3764	180	
	2	0	270	3764	180	
Cylinder 10	1	0	0	3764	153	
	2	0	0	4034	153	
Cylinder 11	1	0	0	4034	153	
	2	0	0	4366	153	
Polygon 12	1	−200	−200	4366	/	
	2	200	−200	4366	/	
	3	200	200	4366	/	
	4	−200	200	4366	/	
	5	−200	−200	4516	/	
	6	200	−200	4516	/	
	7	200	200	4516	/	
	8	−200	200	4516	/	
Cylinder 13	1	0	0	4516	100	
	2	0	0	4816	100	
Cylinder 14	1	0	0	4816	100	
	2	0	225	4816	100	
Cylinder 15	1	0	415	4816	110	
	2	0	225	4816	110	
Cylinder 16	1	0	300	4816	110	
	2	0	300	5646	110	
Cylinder 17	1	0	415	5646	110	
	2	0	225	5646	110	
Cylinder 18	1	0	0	5646	100	
	2	0	225	5646	100	

Table A1. Cont.

Structures	Characteristic Points		Coordinate			Radius/mm (Number of Edges)
	Structure Type	Number	X/mm	Y/mm	Z/mm	
Cylinder 19		1	0	0	5646	100
		2	0	0	6346	100
Cylinder 20		1	0	−150	6346	100
		2	0	150	6346	100
Cylinder 21		1	0	0	6346	85
		2	0	0	6496	85
Cylinder 22		1	0	0	6496	85
		2	0	0	6680	85
Polygonal obstacles 1 (Cases 1,2)		1	850	450	500	/
		2	1150	450	500	/
		3	1150	750	500	/
		4	850	750	500	/
		5	850	450	1300	/
		6	1150	450	1300	/
		7	1150	750	1300	/
		8	850	750	1300	/
Polygonal obstacles 2 (Cases 1,2)		1	1840	−660	500	/
		2	2160	−660	500	/
		3	2160	−340	500	/
		4	1840	−340	500	/
		5	1840	−660	1500	/
		6	2160	−660	1500	/
		7	2160	−340	1500	/
		8	1840	−340	1500	/
Cylindrical obstacles 3 (Cases 1,2)		1	800	−500	550	200 ($N_e = 12$)
		2	800	−500	1300	200 ($N_e = 12$)
Cylindrical obstacles 4 (Cases 1,2)		1	1800	500	550	200 ($N_e = 12$)
		2	1800	500	1500	200 ($N_e = 12$)
Polygonal obstacles 5 (Case 1)		1	2850	−150	700	/
		2	3150	−150	700	/
		3	3150	150	700	/
		4	2850	150	700	/
		5	2850	−150	850	/
		6	3150	−150	850	/
		7	3150	150	850	/
		8	2850	150	850	/
Polygonal obstacles 6 (Case 1)		1	3580	−120	700	/
		2	3820	−120	700	/
		3	3820	120	700	/
		4	3580	120	700	/
		5	3580	−120	1100	/
		6	3820	−120	1100	/
		7	3820	120	1100	/
		8	3580	120	1100	/
Cylindrical obstacles 7 (Case 1)		1	3000	−550	550	150 ($N_e = 12$)
		2	3000	−550	1000	150 ($N_e = 12$)
Cylindrical obstacles 8 (Case 1)		1	3000	550	550	150 ($N_e = 12$)
		2	3000	550	1000	150 ($N_e = 12$)

Table A1. Cont.

Structures	Characteristic Points		Coordinate			Radius/mm (Number of Edges)
	Structure Type	Number	X/mm	Y/mm	Z/mm	
Polygonal obstacles 9 (Case 2)		1	2850	388.91	601.04	/
		2	3150	388.91	601.04	/
		3	3150	601.04	388.91	/
		4	2850	601.04	388.91	/
		5	2850	494.97	707.11	/
		6	3150	494.97	707.11	/
		7	3150	707.11	494.97	/
		8	2850	707.11	494.97	/
Polygonal obstacles 10 (Case 2)		1	3580	410.12	579.83	/
		2	3820	410.12	579.83	/
		3	3820	579.83	410.12	/
		4	3580	579.83	410.12	/
		5	3580	692.97	862.67	/
		6	3820	692.97	862.67	/
		7	3580	862.67	692.97	/
		8	3580	862.67	692.97	/
Cylindrical obstacles 11 (Case 2)		1	3000	0	777.82	150 ($N_e = 12$)
		2	3000	318.2	1096.02	150 ($N_e = 12$)
Cylindrical obstacles 12 (Case 2)		1	3000	777.82	0	150 ($N_e = 12$)
		2	3000	1096.02	318.2	150 ($N_e = 12$)
Case	Target point	1	3000	0	950	/
		2	3000	636.4	676.4	/

Appendix C. Experimental Results

Taking example case 1 as the objective, curves of angular parameters of each joint of TACCS and a series of motion poses relative to capsule based on the single optimization algorithm are shown in Figures A1 and A2 as follows.

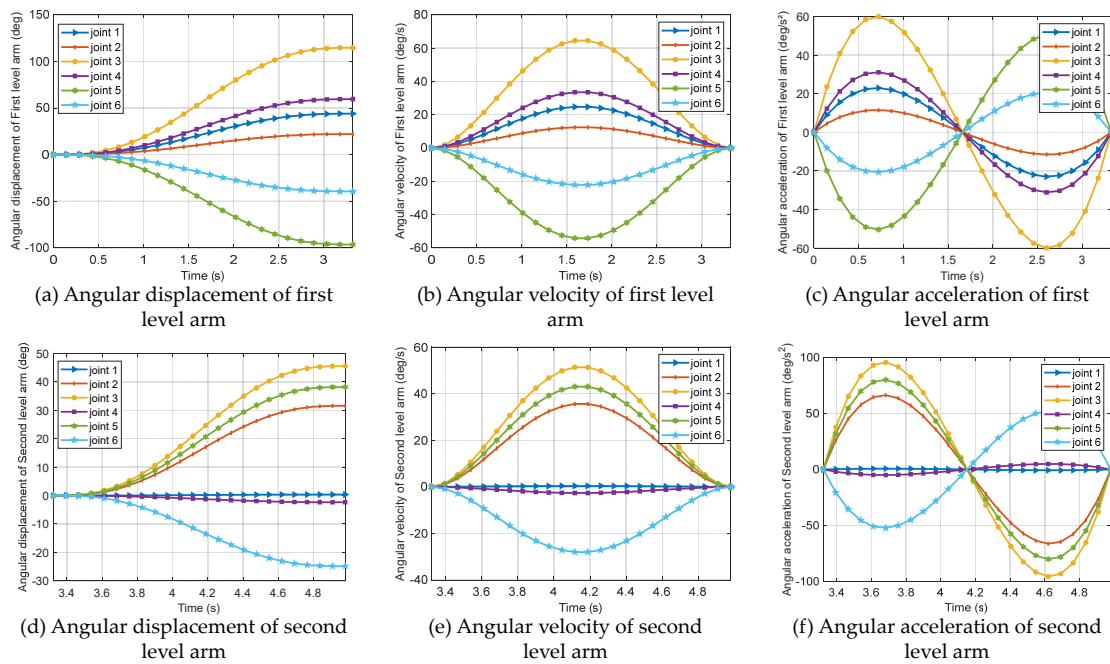


Figure A1. Curves of angular parameters of each joint of TACCS based on single optimization in example case 1.

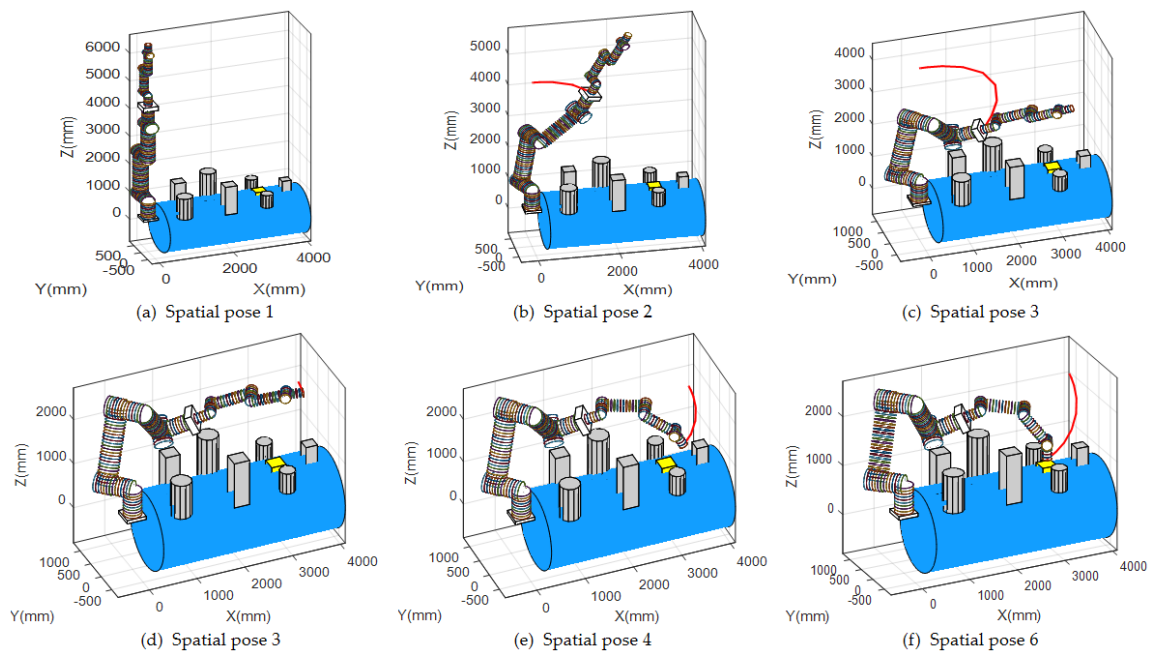


Figure A2. Series of motion poses relative to capsule based on single optimization in example case 1.

Moreover, taking example case 2 as the objective, a series of motion poses relative to capsule based on the bi-layer and single optimization algorithms are shown in Figures A3 and A4 as follows.

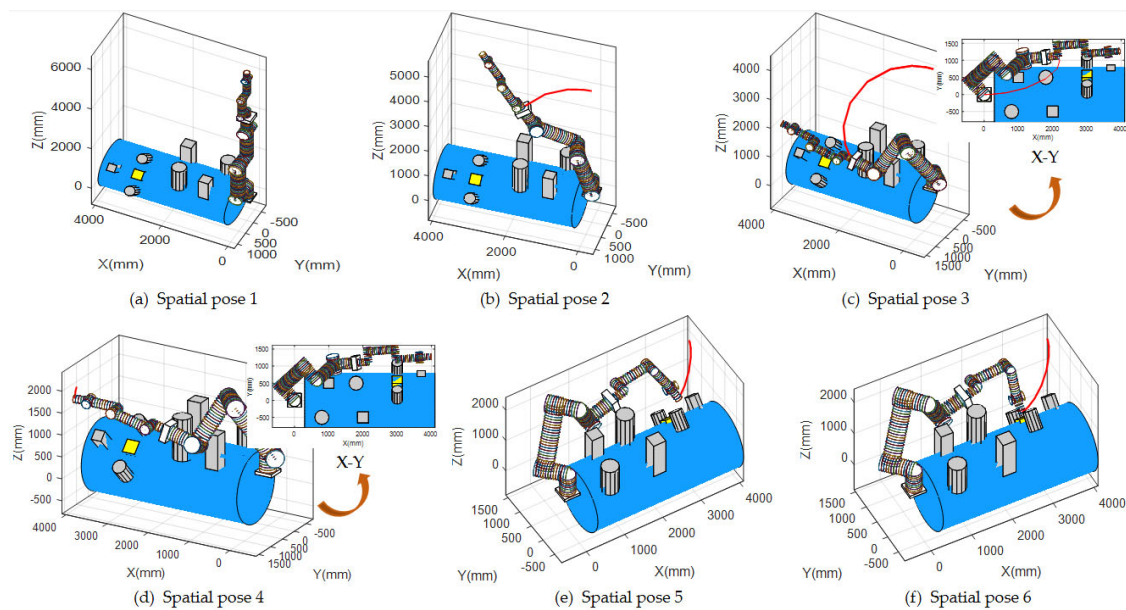


Figure A3. Series of motion poses relative to capsule based on bi-layer optimization in example case 2.

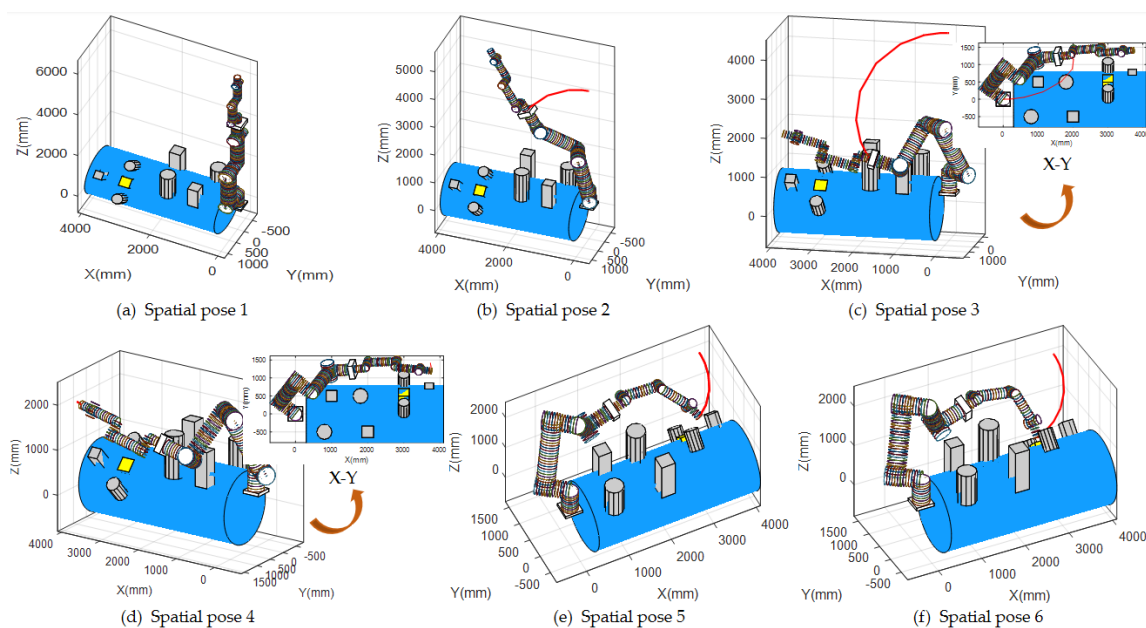


Figure A4. Series of motion poses relative to capsule based on single optimization in example case 2.

References

- Kulakov, F.M. Some Russian research on robotics. *Robot. Auton. Syst.* **1996**, *18*, 365–372. [\[CrossRef\]](#)
- Thronson, H.; Akin, D.; Grunsfeld, J.; Lester, D. The Evolution and Promise of Robotic In-Space Servicing. In Proceedings of the AIAA Space 2009 Conference & Exposition, Pasadena, CA, USA, 14–17 September 2009.
- Jing, Z.; Qiao, L.; Pan, H.; Yang, Y.; Chen, W. An overview of the configuration and manipulation of soft robotics for on-orbit servicing. *Sci. China Inf. Sci.* **2017**, *60*, 050201. [\[CrossRef\]](#)
- Bluethmann, W.; Ambrose, R.; Diftler, M.; Askew, S.; Magruder, D. Robonaut: A Robot Designed to Work with Humans in Space. *Auton. Robot.* **2003**, *14*, 179–197. [\[CrossRef\]](#) [\[PubMed\]](#)
- Al-Kamil, S.J.; Szabolcsi, R. Enhancing Mobile Robot Navigation: Optimization of Trajectories through Machine Learning Techniques for Improved Path Planning Efficiency. *Mathematics* **2024**, *12*, 1787. [\[CrossRef\]](#)
- Dai, Y.; Xiang, C.F.; Zhang, Y.; Jiang, Y.P.; Qu, W.Y.; Zhang, Q.H. A Review of Spatial Robotic Arm Trajectory Planning. *Aerospace* **2022**, *9*, 361. [\[CrossRef\]](#)
- Shrivastava, A.; Dalla, V.K. Jerk Optimized Motion Planning of Redundant Space Robot Based on Grey-Wolf Optimization Approach. *Arab. J. Sci. Eng.* **2022**, *48*, 2687–2699. [\[CrossRef\]](#)
- Xie, Z.; Zhao, X.; Jiang, Z.; Yang, H.; Chongyang, L.I. Trajectory planning and base attitude restoration of dual-arm free-floating space robot by enhanced bidirectional approach. *Front. Mech. Eng. Engl. Ed.* **2022**, *17*, 16. [\[CrossRef\]](#)
- Li, Q.H.; Mu, Y.Q.; You, Y.; Zhang, Z.; Feng, C. A Hierarchical Motion Planning for Mobile Manipulator. *IEEE Trans. Electr. Electron. Eng.* **2020**, *15*, 1390–1399. [\[CrossRef\]](#)
- Chen, H.L.; Zang, X.Z.; Liu, Y.B.; Zhang, X.H.; Zhao, J. A Hierarchical Motion Planning Method for Mobile Manipulator. *Sensors* **2023**, *23*, 6952. [\[CrossRef\]](#)
- Rastegarpanah, A.; Gonzalez, H.C.; Stolkin, R. Semi-Autonomous Behaviour Tree-Based Framework for Sorting Electric Vehicle Batteries Components. *Robotics* **2021**, *10*, 82. [\[CrossRef\]](#)
- Colucci, G.; Botta, A.; Tagliavini, L.; Cavallone, P.; Baglieri, L.; Quaglia, G. Kinematic Modeling and Motion Planning of the Mobile Manipulator Agri.Q for Precision Agriculture. *Machines* **2022**, *10*, 321. [\[CrossRef\]](#)
- Colucci, G.; Tagliavini, L.; Botta, A.; Baglieri, L.; Quaglia, G. Decoupled motion planning of a mobile manipulator for precision agriculture. *Robotica* **2023**, *41*, 1872–1887. [\[CrossRef\]](#)
- NithyaP, K.; Priya, P.S.; Benjamin, G.E.; Venkateswaran, J. Optimal Path Planning and Static Obstacle Avoidance for a Dual Arm Manipulator Used in On-Orbit Satellite Servicing. *IFAC-Pap.* **2020**, *53*, 189–194.
- Yang, H.; Li, D.; Xu, X.R.; Zhang, H. An Obstacle Avoidance and Trajectory Tracking Algorithm for Redundant Manipulator End. *IEEE Access* **2022**, *10*, 52912–52921. [\[CrossRef\]](#)
- Zhang, J.; Zhang, J.; Zhang, Q.; Wei, X. Obstacle Avoidance Path Planning of Space Robot Based on Improved Particle Swarm Optimization. *Symmetry* **2022**, *14*, 938. [\[CrossRef\]](#)
- Cui, L.L.; Wang, H.C.; Chen, W.D. Trajectory planning of a spatial flexible manipulator for vibration suppression. *Robot. Auton. Syst.* **2020**, *123*, 103316. [\[CrossRef\]](#)
- Song, Q.; Li, S.; Bai, Q.; Yang, J.; Zhang, A.; Zhang, X.; Zhe, L. Trajectory Planning of Robot Manipulator Based on RBF Neural Network. *Entropy* **2021**, *23*, 1207. [\[CrossRef\]](#) [\[PubMed\]](#)

19. Liu, X.M.; Qiu, C.R.; Zeng, Q.F.; Li, A.P.; Xie, N. Time-energy Optimal Trajectory Planning for Collaborative Welding Robot with Multiple Manipulators. *Procedia Manuf.* **2020**, *43*, 527–534. [[CrossRef](#)]
20. Cheng, Q.; Hao, X.L.; Wang, Y.; Xu, W.X.; Li, S.J. Trajectory planning of transcranial magnetic stimulation manipulator based on time-safety collision optimization. *Robot. Auton. Syst.* **2022**, *152*, 104039. [[CrossRef](#)]
21. Chen, S.; Zhang, C.; Yi, J. Time-Optimal Trajectory Planning for Woodworking Manipulators Using an Improved PSO Algorithm. *Appl. Sci.* **2023**, *13*, 10482. [[CrossRef](#)]
22. Sun, J.G.; Han, X.Y.; Zuo, Y.M.; Tian, S.Q.; Song, J.W.; Li, S.H. Trajectory Planning in Joint Space for a Pointing Mechanism Based on a Novel Hybrid Interpolation Algorithm and NSGA-II Algorithm. *IEEE Access* **2020**, *8*, 228628–228638. [[CrossRef](#)]
23. Zhang, T.; Cheng, J.; Zou, Y.B. Time-optimal and Smooth Trajectory Planning for Multi-axis Motion Systems Based on ISC Similarity. *Int. J. Control. Autom. Syst.* **2024**, *22*, 1238–1251. [[CrossRef](#)]
24. Ma, J.F.; Gao, S.; Yan, H.Y.; Lv, Q.; Hu, G.Q. A new approach to time-optimal trajectory planning with torque and jerk limits for robot. *Robotics Auton. Syst.* **2021**, *140*, 103744. [[CrossRef](#)]
25. Liu, Z.F.; Xu, J.J.; Yang, C.B.; Zhao, Y.S.; Zhang, T. A TE-E Optimal Planning Technique Based on Screw Theory for Robot Trajectory in Workspace. *J. Intell. Robot. Syst.* **2018**, *91*, 363–375. [[CrossRef](#)]
26. Zhou, Y.; Han, G.; Wei, Z.; Huang, Z.; Chen, X. A Time-Optimal Continuous Jerk Trajectory Planning Algorithm for Manipulators. *Appl. Sci.* **2023**, *13*, 11479. [[CrossRef](#)]
27. Pu, Q.C.; Xu, X.R.; Li, Q.Q.; Zhang, H. Robotic arm time-jerk optimal trajectory based on improved dingo optimization. *J. Braz. Soc. Mech. Sci. Eng.* **2024**, *46*, 198. [[CrossRef](#)]
28. Kashima, T.; Isurugi, Y. Trajectory Planning of Manipulators Based on a Minimum-Energy Criterion and Operating Time. *J. Robot. Soc. Jpn.* **2010**, *15*, 1012–1018. [[CrossRef](#)]
29. Liu, K.P.; Sui, J.L.; Yue, N.; Liu, S.S. Path planning method of mobile manipulator based on the representation space. In Proceedings of the 2016 IEEE International Conference on Mechatronics and Automation, Harbin, China, 7–10 August 2016; pp. 322–326.
30. Luna, R.; Moll, M.; Badger, J.; Kavraki, L. A scalable motion planner for high-dimensional kinematic systems. *Int. J. Robot. Res.* **2020**, *39*, 361–388. [[CrossRef](#)]
31. Zhang, W.M.; Fu, S.X. Time-optimal Trajectory Planning of Dulcimer Music Robot Based on PSO Algorithm. In Proceedings of the Chinese Control and Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 4769–4774.
32. Dai, J.; Zhang, Y.; Deng, H. Novel Potential Guided Bidirectional RRT* with Direct Connection Strategy for Path Planning of Redundant Robot Manipulators in Joint Space. *IEEE Trans. Ind. Electron.* **2024**, *71*, 2737–2747. [[CrossRef](#)]
33. Chen, G.; Luo, N.; Liu, D.; Zhao, Z.H.; Liang, C.C. Path planning for manipulators based on an improved probabilistic roadmap method. *Robot. Comput. Integr. Manuf.* **2021**, *72*, 102196. [[CrossRef](#)]
34. Niu, P.; Cheng, Q.; Chen, C.; Yang, C.; Liu, Z. An approach for crucial geometric error analysis and accuracy enhancement of gantry milling machines based on generalized correlation sensitivity. *J. Manuf. Process.* **2024**, *119*, 401–413. [[CrossRef](#)]
35. Shi, B.H.; Xu, J.X. Time-Optimal Trajectory Planning of Industrial Robot based on Improved Particle Swarm Optimization Algorithm. In Proceedings of the 39th Chinese Control Conference (CCC), Shenyang, China, 27–29 July 2020; pp. 3683–3688.
36. Yu, X.L.; Dong, M.S.; Yin, W. Time-optimal trajectory planning of manipulator with simultaneously searching the optimal path. *Comput. Commun.* **2021**, *181*, 446–453. [[CrossRef](#)]
37. Jin, R.Y.; Rocco, P.; Geng, Y. Cartesian trajectory planning of space robots using a multi-objective optimization. *Aerosp. Sci. Technol.* **2021**, *108*, 106360. [[CrossRef](#)]
38. Li, R.; Liu, M.; Teutsch, J.; Wollherr, D. Constraint trajectory planning for redundant space robot. *Neural Comput. Appl.* **2023**, *35*, 24243–24258. [[CrossRef](#)]
39. Cao, X.; Yan, H.; Huang, Z.; Ai, S.; Xu, Y.; Fu, R.; Zou, X. A Multi-Objective Particle Swarm Optimization for Trajectory Planning of Fruit Picking Manipulator. *Agronomy* **2021**, *11*, 2286. [[CrossRef](#)]
40. Ding, X.L.; Wang, Y.C.; Wang, Y.B.; Xu, K. A review of structures, verification, and calibration technologies of space robotic systems for on-orbit servicing. *Sci. China Technol. Sci.* **2021**, *64*, 462–480. [[CrossRef](#)]
41. Xu, J.J.; Liu, Z.F.; Cheng, Q.; Zhao, Y.S.; Pei, T.H.; Yang, C.B. Models for Three New Screw-based IK Sub-problems Using Geometric Descriptions and Their Applications. *Appl. Math. Model.* **2018**, *67*, 399–412. [[CrossRef](#)]
42. Xu, J.J.; Liu, Z.F.; Zhang, C.X.; Yang, C.B.; Pei, Y.H. Minimal distance calculation between the industrial robot and its workspace based on circle/polygon-slices representation. *Appl. Math. Model.* **2020**, *87*, 691–710. [[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.