micromachines

MDPI

Technical Note

Space Detumbling Robot Arm Deployment Path Planning Based on Bi-FMT* Algorithm

Ning Chen [1], Yasheng Zhang [2] and Wenhua Cheng [2,*]

[1] Department of Graduate Management, Space Engineering University, Beijing 101416, China; 18811021510@163.com

[2] Department of Space Command, Space Engineering University, Beijing 101416, China; zyspublic@163.com

[*] Correspondence: cwhcn@foxmail.com

Abstract: In order to avoid damage to service satellites and targets during space missions and improve safety and reliability, it is necessary to study how to eliminate or reduce the rotation of targets. This paper focused on a space detumbling robot and studied the space detumbling robot dynamics and robot arm deployment path planning. Firstly, a certain space detumbling robot with a 'platform + manipulator + end effector' configuration is proposed. By considering the end effector as a translational joint, the entire space detumbling robot is equivalent to a link system containing six rotating joints and three translational joints, and the detailed derivation process of the kinematic and dynamic model is presented. Then, ADAMS and MATLAB were used to simulate the model, and the MATLAB results were compared with the ADAMS results to verify the correctness of the model. After that, the robot arm deployment problem was analyzed in detail from the aspects of problem description, constraint analysis and algorithm implementation. An algorithm of robot arm deployment path planning based on the Bi-FMT* algorithm is proposed, and the effectiveness of the algorithm is verified by simulation.

Keywords: space detumbling robot; path planning; ADAMS; Bi-FMT*

check for updates

Citation: Chen, N.; Zhang, Y.; Cheng, W. Space Detumbling Robot Arm Deployment Path Planning Based on Bi-FMT* Algorithm. *Micromachines* **2021**, *12*, 1231. https://doi.org/10.3390/mi12101231

Academic Editor: Marco Ciarcia

Received: 3 September 2021
Accepted: 4 October 2021
Published: 10 October 2021

1. Introduction

Space operation and control refer to the on-orbit activity for specific targets with or without people's participation to achieve proximity detection, auxiliary orbit maneuvers, fault maintenance, fuel filling, system upgrades, assembly, construction, rescue and space debris removal [1]. From the Lunakod/Luna project of the Soviet Union and the SAINT (Satellite Inspector) project of the US to Phoenix and SMART-OLEV, the development of space operation and control has always been promoted by the Space Age. A series of experiments have been carried out to develop and verify relevant technologies by the space powers of the world. Space operation and control have become an important indicator of a country's space force. A review of the development of space operation and control projects around the world is summarized in [2]. Space robot motion and control are the core of space operation and control.

Whether it is on-orbit service (OOS), on-orbit assembly or space debris removal, the approach to non-cooperative targets is important. These non-cooperative targets usually have complex attitude movements, including spins, precession and tumbling, which greatly affect the approach process. In order to avoid adverse damage to service satellites and targets during operation and improve safety and reliability, it is necessary to study how to eliminate or reduce the rotation of the target. Generally speaking, as long as the relative state of the service satellite and target meets certain requirements, effective acquisition can be achieved. From the perspective of detumbling, detumbling can be divided into relative detumbling and absolute detumbling. Relative detumbling means that it does not change the target's motion state but uses the service satellite's own adjustment capabilities to change its own motion state to meet relative state constraints. For example, approaching from the target spin axis direction [3] is a typical relative detumbling strategy.

This article focused on absolute detumbling, that is, through the direct or indirect interaction between the service satellite and target, the target state is changed to satisfy the capture condition. In principle, the main operation to make the target detumble is to apply additional torque to the target. According to the different torque sources, absolute detumbling can be divided into contact detumbling and non-contact detumbling. A series of technical verification tests was conducted by the space powers, proposing numerous detumbling methods. These methods are shown in Table 1.

Table 1. Summary of the detumbling methods in recent years.

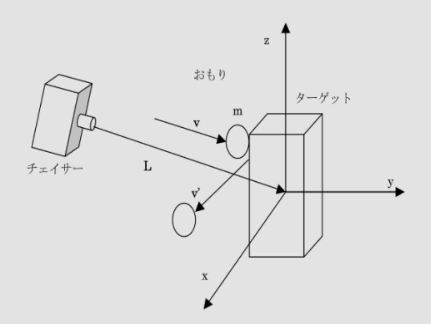
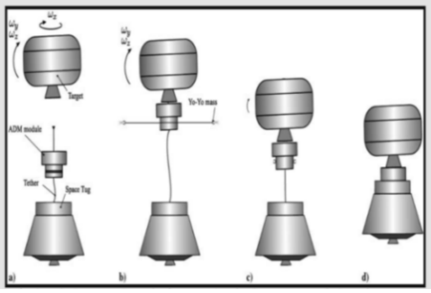
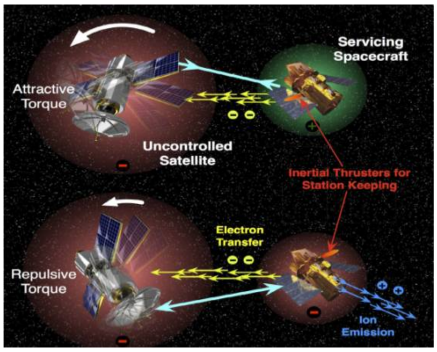
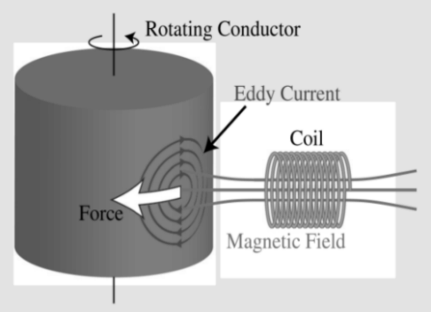
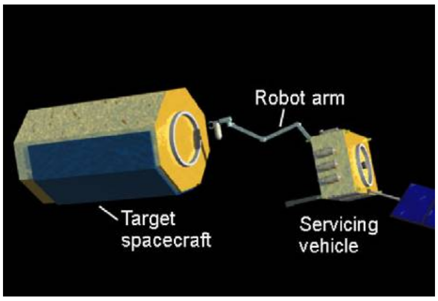
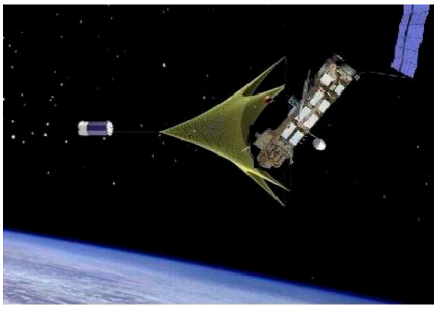
Category	Schematic Diagram	Brief Description
Injection [4–8]	 z おもり ターゲット m v チェイサー L y v' x	The service satellite injects substances such as a gas, ion beam or laser into the target, changing the quality characteristics of the target, including mass and inertia. Thus, it is known from the angular momentum conservation law that the target will be detumbling. On the other hand, the injection could hinder the movement of the target, thereby achieving the purpose of eliminating the target rotation. Using this method, it is necessary to carry an additional end effector and substances for the purpose of detumbling, except for gas injection which can be injected through its own engine but needs more fuel.
Auxiliary Device [9–11]	 ω_z ω_y ω_z Target ADM module Tether Space Tug Yo-Yo mass a) b) c) d)	Attaching an auxiliary device to the target through the service satellite and using the auxiliary device to eliminate the target rotation. The service satellite can avoid contact with the target, and the detumbling mode can flexibly adopt various means according to the target situation. However, similar to the above, the detumbling mode needs to be an additional device that is dedicated to the service satellite and has certain maneuverability and controllability which increases the system complexity.
Electrostatic [12–15]	 Servicing Spacecraft Attractive Torque Uncontrolled Satellite Inertial Thrusters for Station Keeping Electron Transfer Repulsive Torque Ion Emission	Electrons are continuously emitted to the target through an electron-emitting device carried on the service satellite, charging the target. By doing this, the target rotation is detumbled by Coulomb electrostatic force generated by the voltage difference between the service satellite and the target. This also avoids contact between the service satellite and target. However, in this method, it is necessary to continuously charge and discharge to change the potential of the service satellite and target. In addition, this method needs to be further studied in the identification of target charge and discharge characteristics, formation maintenance and charge and discharge control algorithms.

Table 1. *Cont.*

Category	Schematic Diagram	Brief Description
Electromagnetic [16–20]	 Rotating Conductor Eddy Current Coil Force Magnetic Field	Space debris mostly contains conductive materials such as aluminum alloys and titanium alloys. Therefore, when the target is in an external magnetic field, eddy currents are internally induced to hinder the relative motion. By using a superconducting coil to construct an external magnetic field, the target can be detumbled. The electromagnetic damping effect passively eliminates the component angular rate perpendicular to the component of the external magnetic field but does not affect the angular rate component parallel to the magnetic field direction. Thus, the relative position between the magnetic field source and the target is to be changed. In addition, the use of superconducting coils to construct a wide range of electromagnetic fields requires a corresponding power supply and cooling system. How to superimpose the superconducting magnetic field source system with service satellites requires further study.
Robotic Contact [21–23]		In this method, the service satellite touches the target intermittently by using the elastic deceleration device attached to the end of the arm. The target rotation is detumbled by the friction. Robotic contact detumbling can actively adjust the direction, size and time of the control force and provide a higher braking efficiency with an accurate torque control model. However, this type of detumbling mode needs a service satellite to perform a complex orbit maneuver before implementation, located at a position very close to the target, and the collision risk is also increased. In addition, it is suitable for a target with a lower speed considering the on-orbit identification efficiency and the manipulator control precision.
Net or Tether [24–28]		When the net or tether catches the target, the target rotational speed is reduced by the tension and damping force of the tether. This method is only used for debris. In addition, how to avoid failure in catching and preventing the entanglement of the rope also needs further research.

Considering technical maturity and energy consumption, among these methods, the robotic contact method is the most feasible to implement and verify. Additionally, this method combines capture and detumbling together which is very suitable for OOS. As one of the key technologies in space robot control, path planning generates a motion sequence to guide the robot from the initial state to the goal state safely. Path planning is widely used in the field of robotics and has accumulated a wide range of research results [29,30]. Roughly, path planning can be divided into two categories: complete planning and sampling-based planning.

Complete path planning is usually planned directly in the state space, with the Depth First Search (DFS), Breadth First Search (BFS) and Dijstra algorithms representing the original algorithms, and the Astar algorithm representing the most commonly used algorithm. The advantage of this method is that it can completely obtain the solution, but the cost is that the algorithm will become very complicated. This cost is not obvious in path planning in low-dimensional spaces but becomes very prominent in high-dimensional spaces. Since the actual work space-to-state space mapping is non-linear, it is very troublesome to represent obstacles and constraints in the state space. The usual approach is to discretize the space and detect the discrete parts. However, as mentioned earlier, this type of discretization is fine in low-dimensional spaces, but it will bring unimaginable complex calculations in high-dimensional spaces, which directly promotes the generation of sampling-based path planning algorithms.

Sampling-based path planning generally does not plan directly in the state space but randomly arranges a certain density of the sample space to approximate the state space. Sampling-based path planning is also divided into two types: One is graph-based, which scatters sampling points in the original state space and extracts the path by connecting those points with consideration of constraints, such as the probabilistic road map (PRM) algorithm and its improvement. The other is tree-based, which randomly arranges a point in the state space and iteratively grows the tree with the purpose of connecting the starting and ending points, such as the rapid exploration random tree (RRT) algorithm and its improvement. Whether graph-based or tree-based, these algorithms do not need to consider the distribution of obstacles in space but only need to perform collision detection on random sampling points. The planning speed is quite fast and can be used in any dimensional space, and, in particular, path planning has been widely used in high-dimensional spaces.

Due to the complexity of characterizing obstacles and constraints in the state space, complete planning is usually limited to handling low-dimensional problems with simple-shaped obstacles. Sampling-based planning does not need to express obstacles and constraints explicitly but instead combines search-based sampling and performs safety verification through a collision detection algorithm. By separating the planning problem from the actual physical and geometric problems, sampling-based planning greatly accelerates the speed of planning, especially in high-dimensional problems with complex-shaped obstacles.

Space detumbling is a multi-disciplinary complex system engineering problem involving basic disciplines such as mathematics, physics and materials and combining technical disciplines such as control, computer and simulation. Measurement noise, actuator noise, high-order dynamics and orbital perturbations all contribute to the complexity and uncertainty of space detumbling. Considering the uncertainty in space robot motion and control, robot platforms need to have near real-time planning ability in order to handle various uncertainties quickly and safely. Now, the solution for handling uncertainty is mainly divided into three categories. One is to optimize the design of a new spatial structure, as in [31,32]; another is to change the way of thinking and decompose the problem reasonably and abstractly, as explored by Kumar et al., where they decomposed any 3D motion into a 3D translation and three rotations about specific axes related to the object, which allows planning for 3D dexterous in-hand manipulation with a moderate increase in complexity in just a few seconds [33]; the third is to use probabilistic analysis methods. Sampling-based path planning achieves an optimal solution under probability analysis through a reduction in constraints and backward detection and evaluation, which can not only ensure the calculation efficiency but also deal with various constraints well.

Commonly used sampling-based path planning algorithms include PRM [34], RRT [35] and EST [36]. These algorithms can quickly find a feasible path, especially in high-dimensional spaces. However, when the sampling points are too few or the distribution is unreasonable, sampling-based path planning only obtains a feasible path, not the optimal path. In order to solve this problem, scholars have proposed asymptotically optimal versions, PRM* [37] and RRT* [38], where, as the number of samples increases, the solution path obtained will inevitably converge to a global optimum, as with BIT* [39] and RRT# [40]. It is particularly worth noting that the fast marching tree (FMT*) algorithm proposed by Janson et al. [41] is a conceptually novel sampling-based path planning algorithm, and numerical simulation experiments have shown that the FMT* algorithm can converge to the optimal solution faster than PRM* and RRT* in the face of a high-dimensional state space and complex collision detection.

Although sampling-based path planning has not been applied in space missions, its effects and advantages for solving problems with high dynamics and uncertain environments have been verified in ground practical systems. In the Urban Challenge held by the Defense Advanced Research Projects Agency (DARPA), almost all of the winners adopted sampling-based path planning [42–45]. Since the path planning framework is universal, it seems that those research results can be applied to space path planning in theory. However, spacecraft motion is very different from ground robots, especially in space mapping and the C-space [46–49], meaning these planners cannot be directly applied to space missions without modification. Some scholars have studied the feasibility of sampling-based path planning in space missions [50–53], especially the studies by Starek et al. [54–57], in which the real-time implementability, safety and propellant efficiency of path planning by using FMT* or Bi-FMT* were thoroughly discussed in detail.

In the early stage, an improved sampling-based approach for spacecraft proximity operation path planning under Clohessy–Wiltshire–Hill dynamics based on a modified version of the FMT* algorithm with a safety strategy was proposed and analyzed in [58]. In this work, the dynamics and robot arm deployment path planning problem of a certain space detumbling robot were analyzed. Section 2 introduces the design and structure of the space detumbling robot. The kinematics and dynamics of the robot are also analyzed in this section. Then, the detumbling robot arm deployment path planning by using the Bi-FMT* algorithm is described based on the prevention model in detail from the aspects of problem description, constraint analysis and algorithm implementation in Section 3. Additionally, the proposed approach is illustrated by using two numerical experiments in Section 4. Finally, the conclusion and future work directions are provided in Section 5.

2. Space Detumbling Robot Modeling

The space detumbling robot designed in this paper is shown in Figure 1. As shown in the figure, the robot is divided into three parts: satellite platform, manipulator and end effector. Among them, the satellite platform has a 'central rigid body + solar panel' configuration. The arm adopts the configuration of 'elbow-shaped mechanical arm + spherical wrist'. Additionally, the end effector is designed as a flexible brush, which can be considered equivalent to a translational joint with a fixed length.

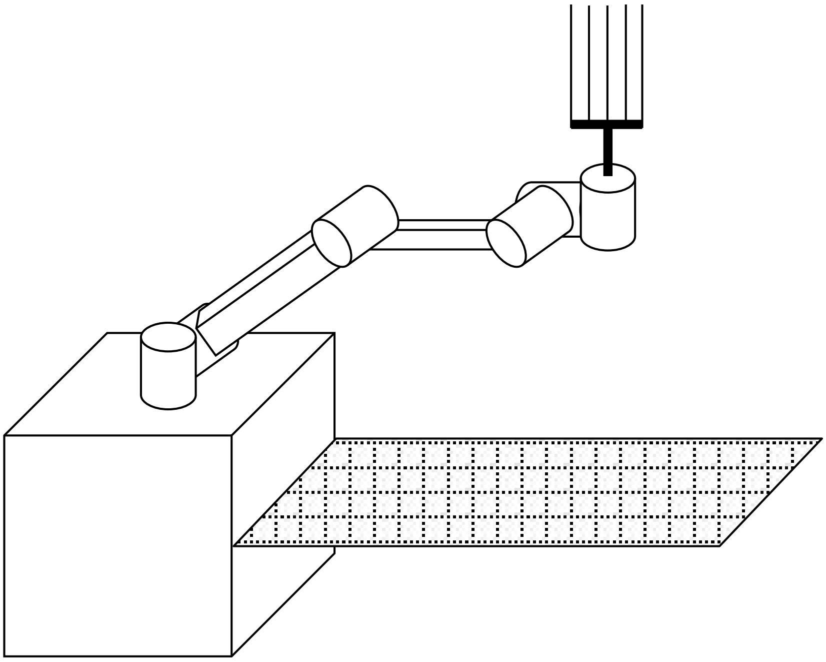


Figure 1. The space detumbling robot designed in this paper.

2.1. Kinematics

Generally, a robot arm system can be treated as a series of links connected by joints. These joints can be divided into single-degree-of-freedom joints and multi-degree-of-freedom joints. In fact, multi-degree-of-freedom joints can also be considered as continuous single-degree-of-freedom joints connected by a zero-length link. Therefore, in this study, it is assumed all joints are single-degree-of-freedom joints. Each coordinate system satisfies the Denavit–Hartenberg (DH) convention, that is,

(1) $\mathrm{O_{i+1}X_{i+1}}$ is perpendicular to $\mathrm{O_iZ_i}$;
(2) $\mathrm{O_{i+1}X_{i+1}}$ intersects with $\mathrm{O_iZ_i}$.

Under the condition of the DH convention, the transformation matrix can be expressed as the product of four basic matrices [59]:

$$\begin{aligned} A &= R(z,\theta)T(z,d)T(x,a)R(x,\alpha) \\ &= \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta & \sin\theta\cos\alpha & \sin\theta\sin\alpha & a\cos\theta \\ -\sin\theta & \cos\theta\cos\alpha & \cos\theta\sin\alpha & -a\sin\theta \\ 0 & -\sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{1}$$

where θ represents the angle from $\mathrm{O_iX_i}$ to $\mathrm{O_{i+1}X_{i+1}}$ measured in a plane perpendicular to $\mathrm{O_iZ_i}$; d represents the distance from the origin $\mathrm{O_i}$ to the intersection of $\mathrm{O_{i+1}X_{i+1}}$ and $\mathrm{O_iZ_i}$; a represents the distance between $\mathrm{O_iZ_i}$ and $\mathrm{O_{i+1}Z_{i+1}}$ measured along $\mathrm{O_{i+1}X_{i+1}}$; α represents the angle measured from $\mathrm{O_iZ_i}$ to $\mathrm{O_{i+1}Z_{i+1}}$ in the plane perpendicular to $\mathrm{O_{i+1}X_{i+1}}$.

Then, the transformation matrix of $\mathrm{O_{n+1}\text{-}X_{n+1}Y_{n+1}Z_{n+1}}$ relative to $\mathrm{O_1\text{-}X_1Y_1Z_1}$ is

$$A_{n+1}^1 = A_1A_2\cdots A_n = \prod_{i=1}^{n} \begin{bmatrix} \cos\theta_i & \sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ -\sin\theta_i & \cos\theta_i\cos\alpha_i & \cos\theta_i\sin\alpha_i & -a_i\sin\theta_i \\ 0 & -\sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{n+1}^1 & T_{n+1}^1 \\ 0 & 1 \end{bmatrix} \tag{2}$$

Suppose the angular velocity of end effector is ω and the linear velocity is v, and let

$$\begin{gathered} [\omega_{n+1}^1\times] = \dot{R}_{n+1}^1\left(R_{n+1}^1\right)^T \\ v_{n+1}^1 = \dot{T}_{n+1}^1 \end{gathered} \tag{3}$$

From the transformation matrix, we can obtain

$$\begin{gathered} R_{i+1}^1 = R_i^1R_{i+1}^i \\ T_{i+1}^1 = R_i^1T_{i+1}^i + T_i^1 \end{gathered} \tag{4}$$

Taking the joint variable q_i as variable, by the chain rule, we can obtain

$$\begin{aligned}
[\omega^1_{n+1}\times] &= \dot{R}^{1}_{n+1}\left(R^1_{n+1}\right)^T = \left(\sum_{i=1}^{n}\frac{\partial R^1_{n+1}}{\partial q_i}\dot{q}_i\right)\left(R^1_{n+1}\right)^T = \left(\sum_{i=1}^{n}\frac{\partial\left(R^1_n R^n_{n+1}\right)}{\partial q_i}\dot{q}_i\right)\left(R^1_n R^n_{n+1}\right)^T \\
&= \left(\sum_{i=1}^{n}\left(\frac{\partial\left(R^1_n\right)}{\partial q_i}R^n_{n+1}\dot{q}_i + R^1_n\frac{\partial\left(R^n_{n+1}\right)}{\partial q_i}\dot{q}_i\right)\right)\left(R^n_{n+1}\right)^T\left(R^1_n\right)^T \\
&= \sum_{i=1}^{n}\left(\frac{\partial\left(R^1_n\right)}{\partial q_i}\dot{q}_i\right)R^n_{n+1}\left(R^n_{n+1}\right)^T\left(R^1_n\right)^T + R^1_n\sum_{i=1}^{n}\left(\frac{\partial\left(R^n_{n+1}\right)}{\partial q_i}\dot{q}_i\right)\left(R^n_{n+1}\right)^T\left(R^1_n\right)^T \\
&= \left[\omega^1_n\times\right] + R^1_n\left[\omega^n_{n+1}\times\right]\left(R^1_n\right)^T = \cdots \\
&= \sum_{i=1}^{n}R^1_i\left[\omega^i_{i+1}\times\right]\left(R^1_i\right)^T \\
&= \sum_{i=1}^{n}\left[\left(R^1_i\omega^i_{i+1}\right)\times\right] = \left[\left(\sum_{i=1}^{n}R^1_i\omega^i_{i+1}\right)\times\right] = \left[\left(\sum_{i=1}^{n}R^1_i\omega_i\right)\times\right]
\end{aligned} \tag{5}$$

Therefore, the angular velocity of the end effector coordinate system relative to the fixed coordinate system is

$$\omega^1_{n+1} = \sum_{i=1}^{n}R^1_i\omega_i \tag{6}$$

Similarly, the velocity of the end effector coordinate system relative to the fixed coordinate system is

$$\begin{aligned}
v^1_{n+1} &= \dot{T}^{1}_{n+1} = \sum_{i=1}^{n}\frac{\partial T^1_{n+1}}{\partial q_i}\dot{q}_i = \sum_{i=1}^{n}\frac{\partial\left(R^1_n T^n_{n+1}+T^1_n\right)}{\partial q_i}\dot{q}_i = \sum_{i=1}^{n}\left(\frac{\partial\left(R^1_n T^n_{n+1}\right)}{\partial q_i}\dot{q}_i + \frac{\partial\left(T^1_n\right)}{\partial q_i}\dot{q}_i\right) \\
&= \sum_{i=1}^{n}\left(\frac{\partial\left(R^1_n\right)}{\partial q_i}T^n_{n+1}\dot{q}_i + R^1_n\frac{\partial\left(T^n_{n+1}\right)}{\partial q_i}\dot{q}_i\right) + \sum_{i=1}^{n}\left(\frac{\partial\left(T^1_n\right)}{\partial q_i}\dot{q}_i\right) \\
&= \sum_{i=1}^{n}\left(\frac{\partial\left(R^1_n\right)}{\partial q_i}\dot{q}_i\right)T^n_{n+1} + R^1_n\sum_{i=1}^{n}\left(\frac{\partial\left(T^n_{n+1}\right)}{\partial q_i}\dot{q}_i\right) + \sum_{i=1}^{n}\left(\frac{\partial\left(T^1_n\right)}{\partial q_i}\dot{q}_i\right) = \cdots \\
&= \sum_{j=1}^{n}\left(\left[\omega^1_j\times\right]R^1_j T^j_{j+1}\right) + \sum_{j=1}^{n}\left(R^1_j\sum_{i=1}^{j}\left(\frac{\partial\left(T^j_{j+1}\right)}{\partial q_i}\dot{q}_i\right)\right) \\
&= \sum_{i=1}^{n}\left(\left[\omega^1_i\times\right]R^1_i T^i_{i+1}\right) + \sum_{i=1}^{n}\left(R^1_i v_i\right) \\
&= \sum_{i=2}^{n}\left(\left(\sum_{j=1}^{i-1}R^1_j\left[\omega^j_{j+1}\times\right]\left(R^1_j\right)^T\right)R^1_i T^i_{i+1}\right) + \sum_{i=1}^{n}\left(R^1_i v_i\right) \\
&= \sum_{i=1}^{n}\left(\left(R^1_i[k\times]\left(R^1_i\right)^T\right)\left(R^1_{i+1}T^i_{i+1} + \cdots + R^1_n T^n_{n+1}\right)\omega_i\right) + \sum_{i=1}^{n}\left(R^1_i v_i\right) \\
&= \sum_{i=1}^{n}\left(\left[\left(R^1_i k\right)\times\right]\left(T^1_{n+1} - T^1_i\right)\omega_i\right) + \sum_{i=1}^{n}\left(R^1_i v_i\right)
\end{aligned} \tag{7}$$

which is

$$\begin{aligned}
\begin{pmatrix} v^1_{n+1} \\ \omega^1_{n+1} \end{pmatrix} &= \begin{pmatrix} \sum\limits_{i=1}^{n}\left(\left[\left(R^1_i k\right)\times\right]\left(T^1_{n+1} - T^1_i\right)\omega_i\right) + \sum\limits_{i=1}^{n}\left(R^1_i v_i\right) \\ \sum\limits_{i=1}^{n}R^1_i\omega_i \end{pmatrix} \\
&= \begin{pmatrix} J^{n+1}_v \\ J^{n+1}_\omega \end{pmatrix}\dot{q} = J^1_{n+1}\dot{q}
\end{aligned} \tag{8}$$

where J is the Jacobian matrix of the robot arm. When the joint is a revolute joint, v_i is 0, and ω_i is the angular velocity of the joint; when the joint is a translational joint, ω_i is 0, and v_i is the translational velocity of the joint.

By considering the end effector as a translational joint, the entire space detumbling robot is equivalent to a link system containing six rotating joints and three translational joints, as shown in Figure 2.

O_i-$X_iY_iZ_i$ (i = 0, 1, . . . , 8) is the coordinate system of each link, which is fixed at the center of mass (CM) of the joint and meets the DH convention. Combined with the structure diagram presented in Figure 2, the DH parameters of the robot system can be obtained, as shown in Table 2.

d_0 is the distance from the platform CM to the center of Joint #1; L1 and L2 are the lengths of the two links of the elbow manipulator; d_6 is the distance from the end effector to the center of the mounting flange of Joint #6; d_7 and a_8 are the vertical and horizontal displacements of the flexible brush.

Table 2. DH parameters of the robot.

Joint	a_i	α_i	d_i	θ_i
0	0	0	d_0	0
1	0	90°	0	θ_1 *
2	L_1	0	0	θ_2 *
3	L_2	0	0	θ_3 *
4	0	90°	0	θ_4 *
5	0	90°	0	θ_5 *
6	0	0	d_6	θ_6 *
7	0	0	d_7 *	0
8	a_8 *	0	0	0

* represents system variables.

As each joint of the robot arm is a rotating joint,

$$\omega_i = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega_i \tag{9}$$

Therefore, the velocity and angular velocity of the flexible brush end coordinate system relative to the platform coordinate system are

$$\begin{aligned} \begin{pmatrix} v_9^0 \\ \omega_9^0 \end{pmatrix} &= \begin{pmatrix} \sum\limits_{i=0}^{8}\left(\left[\left(R_i^0 k\right)\times\right]\left(T_9^0 - T_i^0\right)\omega_i\right) \\ \sum\limits_{i=0}^{8} R_i^0 \omega_i \end{pmatrix} \\ &= \begin{pmatrix} \left[\left(R_0^0 k\right)\times\right]\left(T_9^0 - T_0^0\right) & \cdots & \left[\left(R_8^0 k\right)\times\right]\left(T_9^0 - T_8^0\right) \\ R_0^0 k & \cdots & R_8^0 k \end{pmatrix} \begin{pmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_8 \end{pmatrix} \\ &= \begin{pmatrix} J_v^9 \\ J_\omega^9 \end{pmatrix} \dot{q} = J\dot{q} \end{aligned} \tag{10}$$

2.2. Dynamics

Due to the limitation of the launching mass, the mass of the space robot will be relatively light, and the joint and its accessories will be relatively heavy, meaning the mass center of the joint and the connecting link can be placed at the joint. The Lagrange equation is used to derive the dynamic model. For a space robot, its kinetic energy is the total energy.

Kinetic energy is divided into rotational kinetic energy and translational kinetic energy. For link *i*, its kinetic energy is

$$T_i = \frac{1}{2}m_i\left(v_i^1\right)^T v_i^1 + \frac{1}{2}\left(\omega_i^1\right)^T I_i\omega_i^1 \tag{11}$$

It can be seen from the previous kinematic analysis that

$$\begin{aligned} v_i^1 &= J_v^i\dot{q} \\ \omega_i^1 &= J_\omega^i\dot{q} \end{aligned} \tag{12}$$

which is

$$\begin{aligned} T_i &= \tfrac{1}{2}m_i\left(J_v^i\dot{q}\right)^T J_v^i\dot{q} + \tfrac{1}{2}\left(J_\omega^i\dot{q}\right)^T I_i J_\omega^i\dot{q} \\ &= \tfrac{1}{2}\dot{q}^T\left(m_i\left(J_v^i\right)^T J_v^i + \left(J_\omega^i\right)^T I_i J_\omega^i\right)\dot{q} \\ &= \tfrac{1}{2}\dot{q}^T\left(m_i\left(J_v^i\right)^T J_v^i + \left(J_\omega^i\right)^T R_i^1 I_i^i\left(R_i^1\right)^T J_\omega^i\right)\dot{q} \end{aligned} \tag{13}$$

where I_{ii} is the inertia matrix of the connecting link in this system. Then, the total kinetic energy of the robot arm is

$$\begin{aligned} &T = \tfrac{1}{2}\dot{q}^T\left(\sum m_i\left(J_v^i\right)^T J_v^i + \left(J_\omega^i\right)^T R_i^1 I_i^i\left(R_i^1\right)^T J_\omega^i\right)\dot{q} = \tfrac{1}{2}\dot{q}^T M\dot{q} \\ &= \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}\dot{q}_i\dot{q}_j \end{aligned} \tag{14}$$

where M is the generalized mass matrix of the system and a positive definite symmetric matrix; m_{ij} is an element in M.

From the Lagrange equation, we can obtain

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{q}_k}\right) - \frac{\partial L}{\partial q_k} = \frac{d}{dt}\left(\frac{\partial T}{\partial\dot{q}_k}\right) - \frac{\partial T}{\partial q_k} = Q_k \tag{15}$$

Then,

$$\begin{aligned} &\tfrac{d}{dt}\left(\tfrac{\partial T}{\partial\dot{q}_k}\right) = \tfrac{d}{dt}\left(\sum_{j=1}^{n} m_{kj}\dot{q}_j\right) = \left(\sum_{j=1}^{n} m_{kj}\tfrac{d\dot{q}_j}{dt}\right) + \left(\sum_{j=1}^{n}\tfrac{dm_{kj}}{dt}\dot{q}_j\right) \\ &= \sum_{j=1}^{n} m_{kj}\ddot{q}_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\tfrac{\partial m_{kj}}{\partial q_i}\dot{q}_i\dot{q}_j \\ &\tfrac{\partial T}{\partial q_k} = \tfrac{1}{2}\frac{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} m_{ij}\dot{q}_i\dot{q}_j}{\partial q_k} = \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\tfrac{\partial m_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j \end{aligned} \tag{16}$$

Thus,

$$\sum_{j=1}^{n} m_{kj}\ddot{q}_j + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\right)\dot{q}_i\dot{q}_j = Q_k \tag{17}$$

Therefore, the dynamic equation of the robot arm is

$$\begin{pmatrix} m_{k1} & m_{k2} & \cdots & m_{kn} \end{pmatrix}\begin{pmatrix} \ddot{q}_1 \\ \vdots \\ \ddot{q}_n \end{pmatrix} + \frac{1}{2}\begin{pmatrix} c_{k1} & c_{k2} & \cdots & c_{kn} \end{pmatrix}\begin{pmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{pmatrix} = Q_k \tag{18}$$

where

$$c_{kj} = \sum_{i=1}^{n} \left(\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k} \right) \dot{q}_i \tag{19}$$

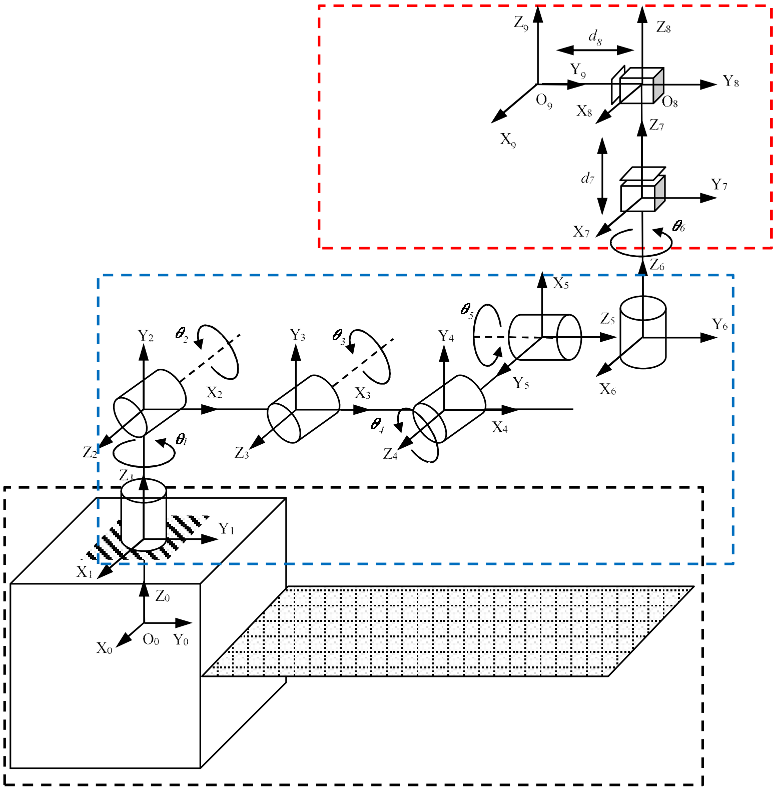


Figure 2. Space detumbling robot modeling.

3. Space Detumbling Robot Arm Path Planning Based on Bi-FMT* Algorithm

3.1. Bi-FMT Algorithm*

The input of the FMT* algorithm is the set S of the initial state X_{initial}, the target state X_{goal} and all sampling points X_{S} in the free state space X_{free}. Assume that when the Euclidean distance of two sampling points satisfies Equation (20), we state that these two sampling points are adjacent.

$$\Delta x < r_n = (2+\eta)\left(\frac{1}{d}\right)^{\frac{1}{d}} \left(\frac{\mu\left(X_{free}\right)}{V_d}\right)^{\frac{1}{d}} \left(\frac{\log(n)}{n}\right)^{\frac{1}{d}} \tag{20}$$

Among them, n is the number of sampling points, d is the dimension of the state space, η is the neighborhood radius coefficient, which is generally greater than 0, $\mu(\cdot)$ is the Lebesgue measure and V_d represents the unit sphere volume in a d-dimensional space. The basic description of the FMT* algorithm is shown in Figure 3.

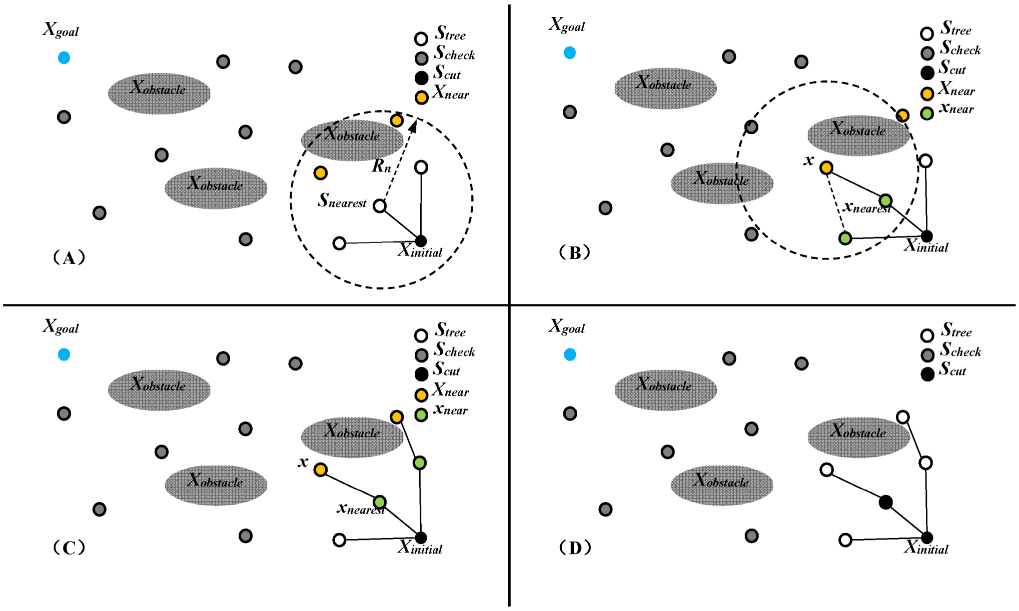


Figure 3. Schematic diagram of the iterative process of the FMT* algorithm. Sub-figure (**A**) shows the neighborhood point X_{near}. Sub-figure (**B**) shows the connection of a new branch. Sub-figure (**C**) shows the all branches added based on X_{near}. Sub-figure (**D**) shows the one step result of FMT*.

In the FMT* algorithm, the set S is divided into three subsets: the node set S_{tree} of the tree, the candidate point set S_{check} and the pruning set S_{cut}. S_{tree} includes sampling points that have been added to the path tree and continue to participate in the next step of the path tree growth; S_{check} includes all sampling points that have not been tested; S_{cut} includes sampling points that have been added to the path tree but are pruned during the growth of the tree The next sampling point no longer participates in the further growth of the path tree. At the beginning, the FMT* algorithm puts $X_{initial}$ in S_{tree} and all other sampling points in S_{check}, while S_{cut} is an empty set; then, the algorithm finds the shortest point in S_{tree} from the $X_{initial}$ path $S_{nearest}$ and finds $S_{nearest}$ in S_{check}. The neighborhood point X_{near} is shown in Figure 3A; for each sampling point x in X_{near}, in turn, find its neighborhood point $\mathrm{x}_{\mathrm{near}}$ in S_{tree}, evaluate the path cost of each connection and find the node with the lowest path cost $x_{nearest}$; if this connection path does not conflict with the obstacle area, it is added as a branch of the tree, as shown in Figure 3B; when each x in X_{near} has completed the above operation, as shown in Figure 3C, move the sampling points successfully connected to the tree from S_{check} to S_{tree}, and at the same time, move $S_{nearest}$ from S_{tree} to S_{cut}, and no longer participate in the growth of the tree, as shown in Figure 3D; keep repeating the above steps to let the tree grow until S_{tree} includes X_{goal} or S_{tree} is an empty set, and the algorithm ends.

It can be seen that the FMT* algorithm synchronizes the construction and search of the path graph. By transforming the collision detection into a local optimal connection problem, a large number of collision detections are avoided. While reducing the computational complexity, it can also ensure the result of path planning. In order to improve the convergence speed of the algorithm, some scholars [60] applied the bidirectional search idea to path planning, trying to search from the initial state to the target state and from the target state back to the initial state. Studies have found that the convergence speed can be greatly improved through bidirectional planning, and this idea can basically be applied to any path planning problem [61,62]. The basic description of the Bi-FMT* algorithm is shown in Figure 4.

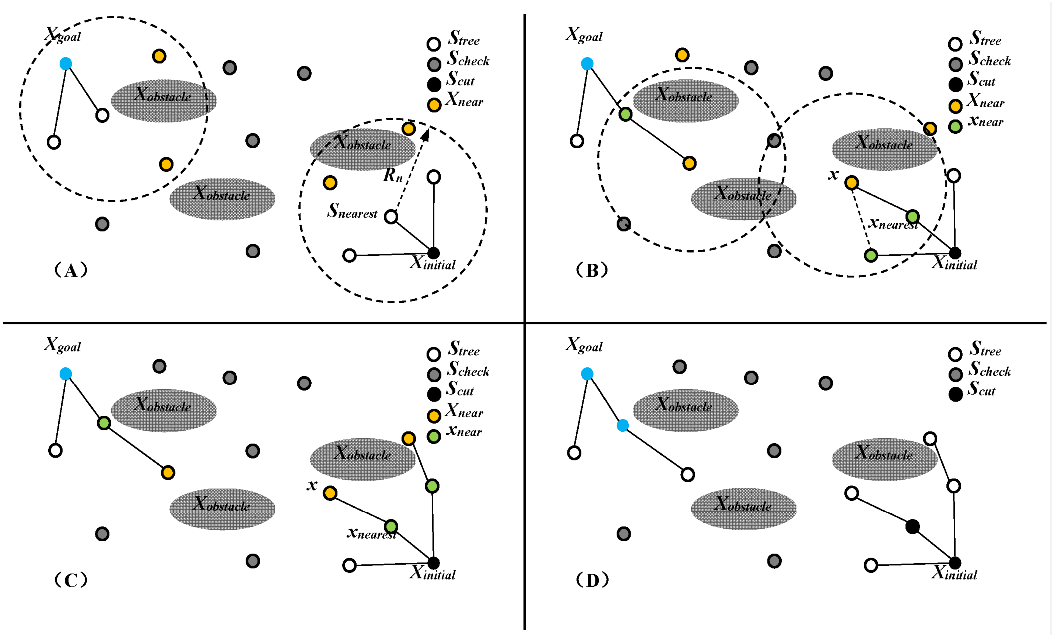


Figure 4. Schematic diagram of the iterative process of the Bi-FMT* algorithm. Sub-figure (**A**) shows the neighborhood point X_{near}. Sub-figure (**B**) shows the connection of a new branch. Sub-figure (**C**) shows the all branches added based on X_{near}. Sub-figure (**D**) shows the one step result of Bi-FMT*.

The core of the Bi-FMT* algorithm is bidirectional. Except for two growing trees, the Bi-FMT* algorithm is basically the same as the FMT* algorithm. Although the basic structure of the Bi-FMT* algorithm is the same as that of the FMT* algorithm, its calculation efficiency on a given sample is much higher than that of the FMT* algorithm. When the dimension of the state space is d, the speed of the Bi-FMT* algorithm can be increased by $2^{\mathrm{d}-1}$ times compared with the FMT* algorithm.

3.2. Problem Definition

The state space Θ is defined as the rotation angles $\{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\}$ of the manipulator joints. $\Theta_{initial}$ denotes the initial configuration of the robot arm, and Θ_{goal} denotes the goal configuration. The manipulator path planning is expressed as follows.

Given:	$\Theta_{initial}, t_0, \Theta_{goal}, \Theta_{free}$
Cost function:	$J(\Theta(t)) = tr\left(JJ^T\right)$
Constraints:	$\Theta(t_0) = \Theta_{initial}$ $\Theta\left(t_f\right) = \Theta_{goal}$ $t_0 < t_f$ $g(\Theta(t), \tau(t), t) \le 0$ $h(\Theta(t), \tau(t), t) = 0$

3.3. Constraint Analysis

3.3.1. Time Constraints

This paper assumes that the running time between adjacent path points is equal, that is,

$$t_f - t_N = t_N - t_{N-1} = \cdots t_2 - t_1 = t_1 - t_0 = \Delta t \tag{21}$$

In addition, the total time the robot moves from the initial state to the target state should be within the time required by the task, i.e.,

$$t_f - t_0 \le T_{plan} \tag{22}$$

The longer the response time, the smaller the required angular acceleration, the more stable the movement of the robot arm and the smoother the trajectory, the longer the travel time. DELTA.t between adjacent path points can be lengthened as much as possible during planning. From Formulas (21) and (22), the maximum value of Δt is found:

$$\Delta t_{\max} = \frac{T_{plan}}{N+1} \tag{23}$$

In this paper, Δt is taken as the maximum value $\Delta t_{\max}$, which is involved in subsequent path planning.

3.3.2. Stationary Constraints

The motion of the manipulator should be smooth, and unsteady motion will intensify the relative motion between the components and cause system vibration and impact. Therefore, in addition to describing the continuity of the function of the motion trajectory of the manipulator, its velocity and acceleration should be continuous.

At this time, the path sampling point $\Theta(t_i)$ satisfies

$$\begin{array}{l} \Theta(t_i-) = \Theta(t_i) = \Theta(t_i+) \\ \dot{\Theta}(t_i-) = \dot{\Theta}(t_i+) \\ \ddot{\Theta}(t_i-) = \ddot{\Theta}(t_i+) \end{array} \tag{24}$$

where $\Theta(t_i-)$ is a local planning trajectory before time t_i, and $\Theta(t_i+)$ is the local planning trajectory after time t_i.

3.3.3. Dynamic Characteristic Constraints

The dynamic characteristic constraint mainly means that the angular velocity and angular acceleration of the joint satisfy the bounded condition

$$\begin{array}{l} \left|\dot{\Theta}(t)\right| \le \omega_{\max} \\ \left|\ddot{\Theta}(t)\right| \le \dot{\omega}_{\max} \end{array} \tag{25}$$

3.4. Path Planning Algorithm Description

The robot path planning algorithm is described in Table 3.

Table 3. Manipulator path planning based on the Bi-FMT* algorithm.

```
given initial state $\Theta_{initial}$, $\Theta_{goal}$, task time $T_{plan}$, sampling point number $N$, maximum angular velocity $\omega_{max}$ and maximum angular acceleration $\dot{\omega}_{\max}$
Calculating single step response time $\Delta t$ by using (23)
The state space is sampled by means of Halton sampling method, and the set of sampling points $\Theta_S$ is obtained.
The path trees $\{S_{tree}, S_{check}, S_{cut}\}$ and $\{S'_{tree}, S'_{check}, S'_{cut}\}$ which are based on $\Theta_{initial}$ and $\Theta_{goal}$ is generated
While Do
Finding the intersection $S_{meet}$ of $S_{tree}$ and $S'_{tree}$
  $S_{meet}$ is not empty
    Calculate the path cost J of each point
    Find the sampling point $\vartheta_{meet}$ with the smallest J
    By connecting $S_{tree}$ and $S'_{tree}$ with $\vartheta_{meet}$ as the connection point, the path between $\Theta_{initial}$ and $\Theta_{goal}$ is obtained
  $S_{meet}$ is empty
    performing FMT*algorithms on $\{S_{tree}, S_{check}, S_{cut}\}$ and $\{S'_{tree}, S'_{check}, S'_{cut}\}$, respectively, and updating $S_{tree}$ and $S'_{tree}$
While Done
The local cubic polynomial interpolation is used to generate trajectory between sampling points in path
```

4. Simulation

4.1. Robot Model

ADAMS and MATLAB were used to simulate the model, and the MATLAB results were compared with the ADAMS results to verify the correctness of the model. Combining common components and their characteristics, the simulation parameter settings of the space detumbling robot are shown in Table 4.

Table 4. List of simulation parameters of the space detumbling robot.

Type	Subtype	Value
Satellite platform	Central body	$m = 1000$ kg $I = diag\{400, 400, 400\}\text{kg} \cdot \text{m}^2$ $V = 1.6\ \text{m} \times 1.6\ \text{m} \times 1.6\ \text{m}$
	Solar panel	$m = 20$ kg $I = diag\{20, 1, 20\}\text{kg} \cdot \text{m}^2$ $V = 2\ \text{m} \times 1.6\ \text{m} \times 0.015\ \text{m}$
Robot arm	Joint1, Joint4~6	$m = 10$ kg $I = diag\{1, 1, 1\}\text{kg} \cdot \text{m}^2$ $V = 0.3\ \text{m} \times (\pi \times 0.15\ \text{m} \times 0.15\ \text{m})$
	Joint2~3	$m = 10$ kg $I = diag\{1, 1, 1\}\text{kg} \cdot \text{m}^2$ $V = 0.3\ \text{m} \times (\pi \times 0.15\ \text{m} \times 0.15\ \text{m})$ $L = 1$ m
Flexible brush	Flexible brush	L= 0.75 m
Simulation parameters	Initial joint state	$\begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 \end{bmatrix} = \begin{bmatrix} 0° & 0° & 0° & 90° & 90° & 0° \end{bmatrix}$
	Joint torque	$\begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 & \tau_5 & \tau_6 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix} N \cdot m$

DH parameters are shown in Table 5.

Table 5. DH parameters of the space detumbling robot.

Joint	a_i	α_i	d_i	θ_i
0	0	0	0.95	0
1	0	90°	0	90°
2	1	0	0	0
3	1	0	0	0
4	0	90°	0	90°
5	0	90°	0	90°
6	0	0	0	0
7	0	0	0.75	0
8	0	0	0	0

DH parameters are shown in Table 5.

We constructed the space detumbling robot model in ADAMS, as shown in Figure 5. Under the torque applied by the joints, the robot simulation results are shown in Figures 6 and 7.

According to the simulation parameters, the kinematics and dynamics equations of the robot are specifically expanded (see Appendix B of [63] for details). We used MATLAB to build the model, import the simulation parameters and obtain the simulation results, as shown in Figure 8.

We compared the ADAMS calculation results with the MATLAB mathematical model simulation results, as shown in Figure 9. It can be seen from the figure that the simulation results' deviation is small, which verifies the correctness of the robot model.

4.2. Path Planning

The initial and goal states of the robot are shown in Figure 10.

$$\Theta_{initial} = \begin{pmatrix} \frac{\pi}{4} & 0 & \pi & 0 & \frac{\pi}{2} & 0 \end{pmatrix}$$

$$\Theta_{goal} = \begin{pmatrix} 0 & \frac{\pi}{2} & 0 & 0 & \frac{\pi}{2} & 0 \end{pmatrix}$$

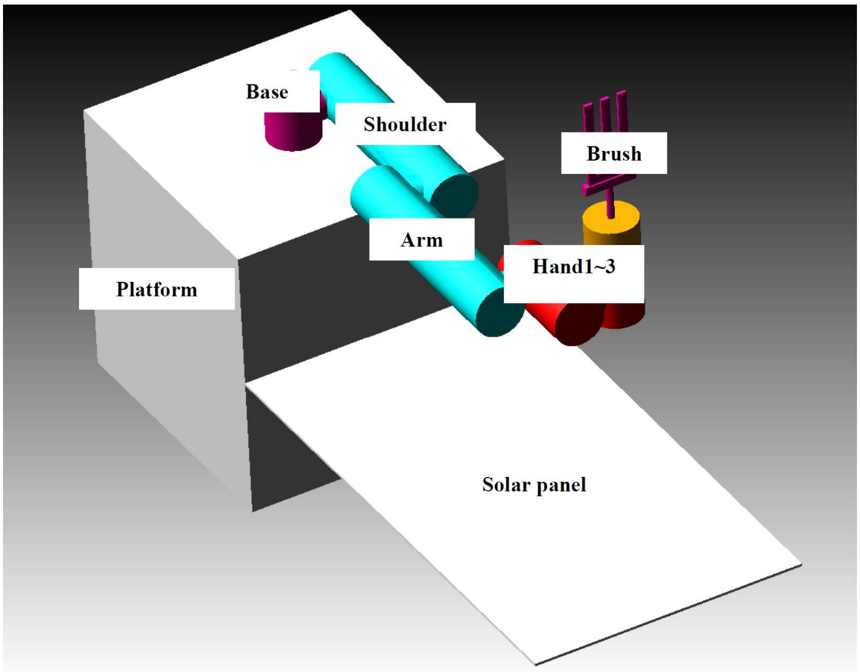


Figure 5. Configuration of the space detumbling robot.

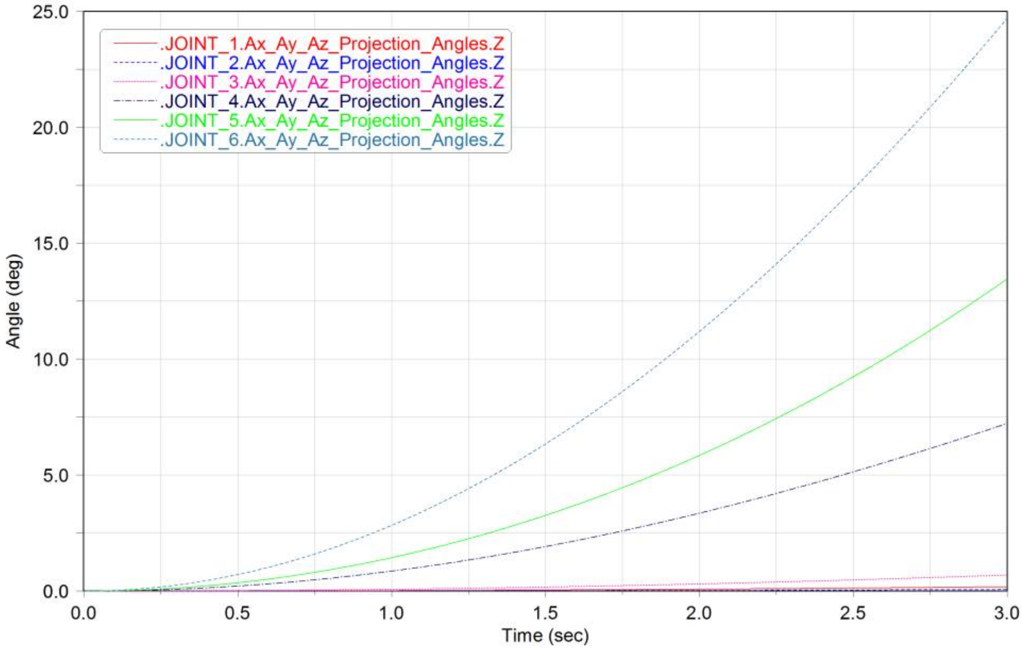


Figure 6. Diagram of the rotation angle of each joint over time (ADAMS).

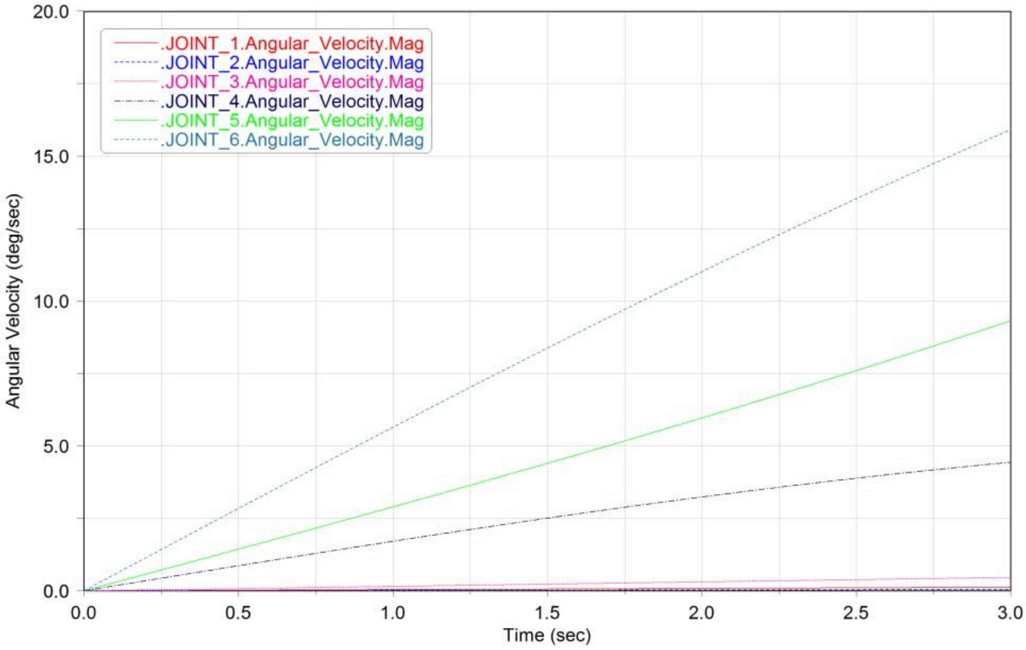
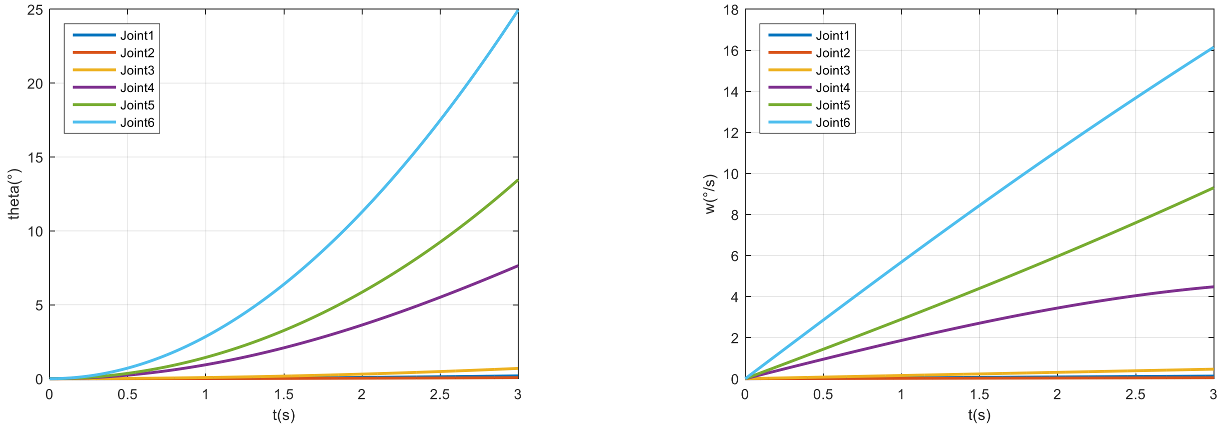
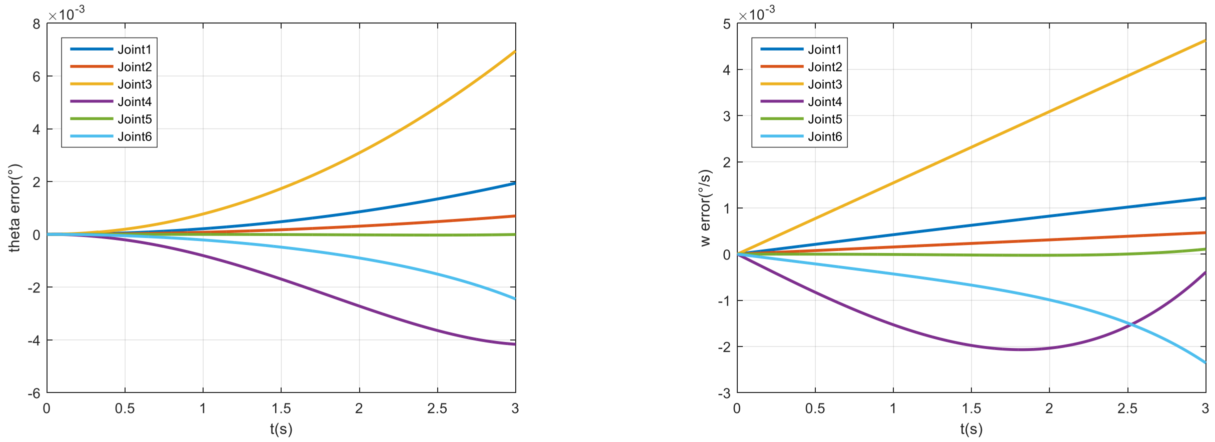


Figure 7. The angular velocity of each joint over time (ADAMS).



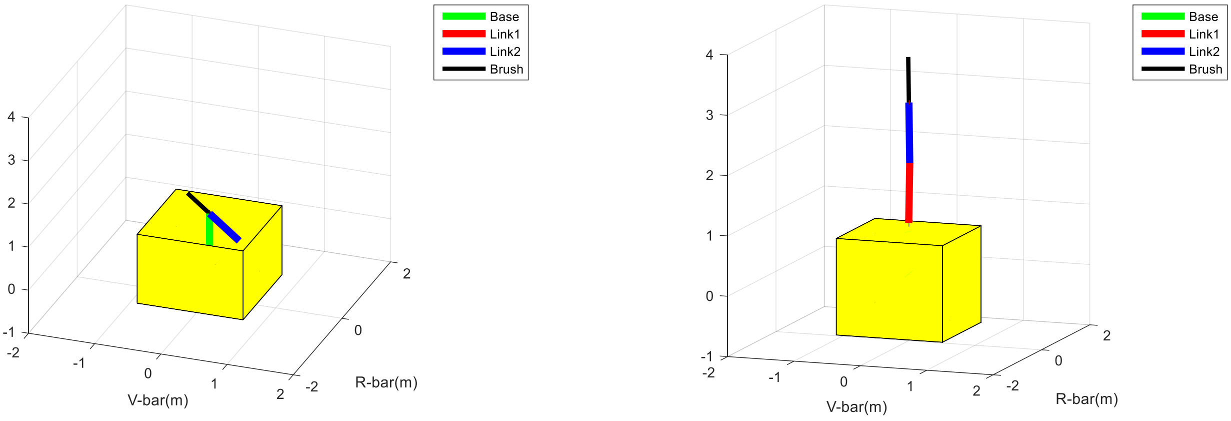
(**a**) Rotation angle (**b**) Angular velocity

Figure 8. MATLAB simulation results.



(**a**) Rotation angle (**b**) Angular velocity

Figure 9. Deviation vs. time graph.



(**a**) Initial state (**b**) Goal state

Figure 10. Schematic diagrams of initial and goal states.

It should be noted that the arm wrist joint can be locked, and thus the sampling state space can be expressed as

$$\Theta = \begin{pmatrix} \theta_1 & \vartheta_2 & \theta_3 & 0 & \frac{\pi}{2} & 0 \end{pmatrix} \tag{26}$$

On the other hand, under this simplified condition, the non-collision condition can be expressed as

$$\begin{aligned} &0 \leq \theta_2 \leq \pi \\ &-2\theta_2 \leq \theta_3 \leq \pi \end{aligned} \tag{27}$$

Considering that the flexible brush has a length, the state of the manipulator must also be satisfied:

$$0 \leq \theta_2 + \theta_3 \leq \pi \tag{28}$$

In summary, the sampling state space is

$$\Theta = \{(\theta_1\ \theta_2\ \theta_3\ 0\ \frac{\pi}{2}\ 0)\ \begin{matrix} |0 \leq \theta_2 \leq \pi \\ 0 \leq \theta_2 + \theta_3 \leq \pi \end{matrix} \tag{29}$$

while

$$\begin{aligned} &T_{plan} = 600\ \text{s} \\ &N = 1000 \\ &\omega_{\max} = 0.5^{\circ}/\text{s} \\ &\dot{\omega}_{\max} = 5^{\circ}/\text{s}^2 \end{aligned}$$

The data structure defining the sampling points in the robot path planning is as follows:

$$\Theta i = \{i\theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ \theta_6\ iftoinitialiftogoaljJijJinitialJgoal\} \tag{30}$$

where i is the sample point number; *iftoinitial* and *iftogoal* are identifiers for determining that the node is connected to the initial state or the goal state; j denotes the node from the node to the initial connection point.

The hardware conditions are Intel(R)Core(TM)i7-4720HQ CPU@2.60GHz, 8.00GB RAM. The simulation platform adopts MATLAB2015b. The simulation results are shown in Figure 11. Under the hardware conditions presented in this paper, the path planning time is 0.567 s when the sampling number is 1000. It can be seen from the figure that the robot arm can reach the goal configuration, which shows the validity of the robot arm path planning algorithm. Considering that the increase in sampling points makes the solution matrix very large, this paper adopted a two-step strategy: first, according to the full sampling space to carry out the path planning, we generated the sequence of path sampling points; then, we selected a small sample to generate the trajectory in the obtained sequence. Taking Figure 11 as an example, the resulting trajectory equation is shown in Figure 12.

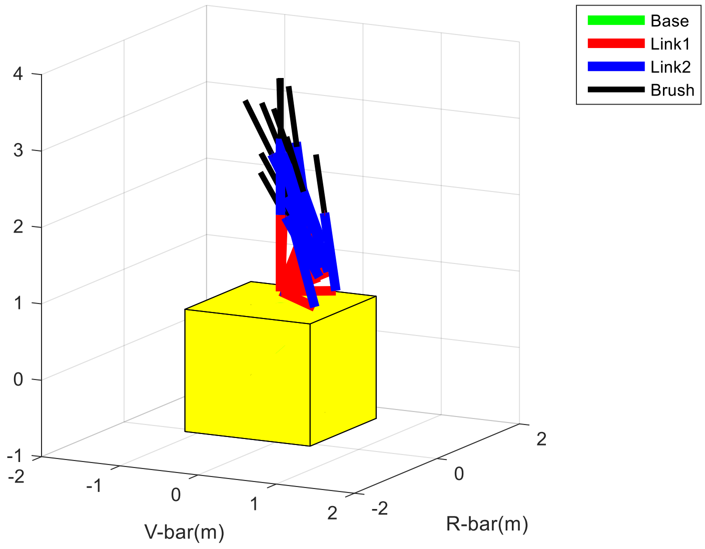


Figure 11. Path planning results.

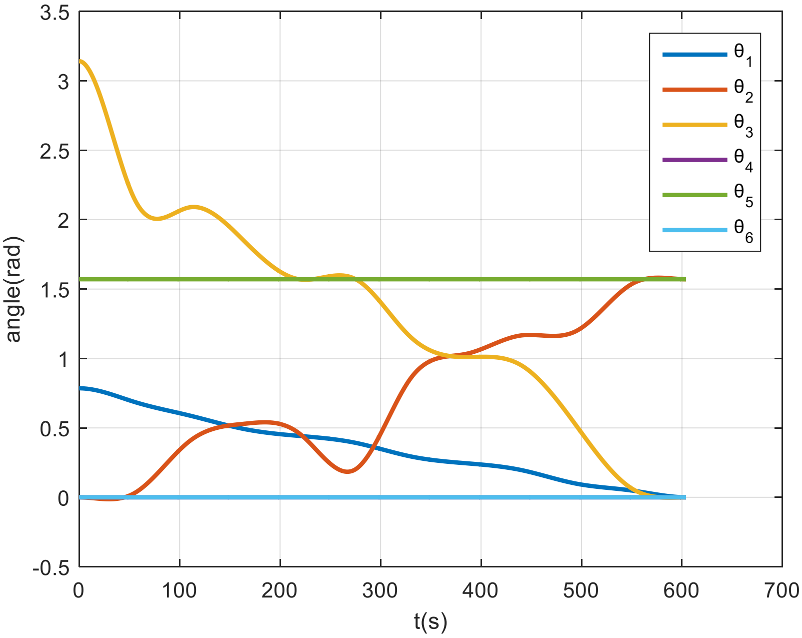


Figure 12. Manipulator trajectory planning results.

5. Conclusions and Future Work

With the development of space exploration technology and space commercial activities, the number of spacecrafts in space is sharply increasing, and space resources and the environment are facing enormous challenges. On-orbit service (OOS, which consists of on-orbit refueling, on-orbit repairing, on-orbit upgrading and space debris removal) is an effective means to achieve successful space exploration missions and keep the space environment safe. Whether on-orbit assembly or space debris removal, the proximity to non-cooperative targets is important. However, these non-cooperative targets usually have complex attitude movements, which greatly affect the proximity operation process. In order to avoid damage to service satellites and targets during operation and improve safety and reliability, it is necessary to study how to eliminate or reduce the rotation of targets. A series of technical verification tests have been conducted by the space powers, proposing numerous detumbling methods, including friction, static, net, auxiliary device and electromagnetic. Considering technical maturity and energy consumption, among these methods, frictional detumbling is the most feasible. This paper focused on a space detumbling robot and studied the related technologies including space detumbling robot dynamics and robot arm path planning. A certain space detumbling robot with a 'platform + manipulator + end effector' configuration was proposed. By considering the end effector as a translational joint, the kinematic and dynamic model of the space detumbling robot was presented. Then, ADAMS and MATLAB were used to simulate and verify the model. After that, the robot arm deployment problem was analyzed in detail, and path planning based on the Bi-FMT* algorithm was also proposed and verified by simulation.

Space detumbling is a multi-disciplinary complex system engineering problem involving basic disciplines such as mathematics, physics and materials and combining technical disciplines such as control, computer and simulation. In contrast, the research work conducted in this article is only a small part of the solution, there is still a big gap to fill before practical engineering applications and theoretical research needs to be improved. On the basis of this article, future work directions include the following:

(1) The platform, manipulator and target are all regarded as rigid bodies; in practice, both the manipulator and solar panels have a certain degree of flexibility, and modeling under the condition of multiple flexible bodies is an important research direction.

(2) Semi-physical design and simulation verification of detumbling platforms and mechanisms.

Author Contributions: Conceptualization, N.C. and Y.Z.; methodology, N.C. and Y.Z.; software, W.C.; validation, N.C., Y.Z. and W.C.; formal analysis, N.C.; investigation, N.C.; resources, Y.Z.; data curation, W.C.; writing—original draft preparation, Y.Z.; writing—review and editing, W.C.; visualization, N.C.; supervision, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

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

References

1. Yingzi, H.; Chunling, W.; Liang, T. Research status and development trend of space control technology. *Space Control. Technol. Appl.* **2014**, *40*, 1–8.
2. Ning, C.; Yasheng, Z.; Wenhua, C. Review and Outlook of the Space Operation and Control Project Development and Technology. In Proceedings of the 3rd International Conference on Unmanned Systems (ICUS), Harbin, China, 27–28 November 2020.
3. Weilin, W.; Yangang, L. Guidance and control for satellite proximity operations. *Aircr. Eng. Aerosp. Technol.* **2014**, *86*, 76–86.
4. Bonnal, C.; Ruault, J.M.; Desjean, M.C. Active debris removal: Current status of activities in CNES. In Proceedings of the 6th European Conference on Space Debris, Darmstadt, Germany, 22–25 April 2013.
5. Nao, T.; Hirohisa, K. Autonomous learning-type impact thrust method for damping angular momentum of malfunctioning satellites. *Proc. Jpn. Aerosp. Soc.* **2007**, *55*, 27–33.
6. Ferrari, F.; Benvenuto, R.; Lavagna, M. Gas plume impingement technique for space debris de-tumbling. In Proceedings of the International ESA Conference on Guidance, Navigation & Control Systems, Oporto, Portugal, 2–6 June 2014.
7. Kumar, R.; Sedwick, R.J. Despinning Orbital Debris Before Docking Using Laser Ablation. *J. Spacecr. Rocket.* **2015**, *52*, 1129–1134. [CrossRef]
8. Vetrisano, M.; Thiry, N.; Vasile, M. Detumbling large space debris via laser ablation. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2015.
9. Caubet, A.; Biggs, J.D. Design of an attitude stabilization electromagnetic module for detumbling uncooperative targets. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 1–8 March 2014.
10. Yudintsev, V.; Aslanov, V. Detumbling Space Debris Using Modified Yo-Yo Mechanism. *J. Guid. Control. Dyn.* **2017**, *40*, 1–8. [CrossRef]
11. Zhang, S.; Luo, Z.; Nie, T.; Gu, Y.; Jiang, Y.; Fan, Y. Fast Rolling Target Derotating Cell Sail for On-Orbit Service and Its Working Method. Invention Patent CN105197261A, 2015.
12. Bennett, T.; Schaub, H. Capitalizing on relative motion in electrostatic de-tumbling of axisymmetric GEO objects. In Proceedings of the 6th International Conference on Astrodynamics Tools and Techniques (ICATT), Darmstadt, Germany, 14–17 March 2016.
13. Bennett, T.; Stevenson, D.; Hogan, E.; Schaub, H. Prospects and challenges of touchless electrostatic detumbling of small bodies. *Adv. Space Res.* **2015**, *56*, 557–568. [CrossRef]
14. Zhao, Y. Research on Non-Contact Attitude Control Based on the Coulomb Force. Ph.D. Thesis, Harbin Institute of Technology, Harbin, China, 2016.
15. Bennett, T.; Schaub, H. Touchless electrostatic de-tumble of a representative box-and-panel spacecraft configuration. In Proceedings of the 7th European Conference on Space Debris, Darmstadt, Germany, 18–21 April 2017.
16. Kadaba, P.K.; Naishadham, K. Feasibility of noncontacting electromagnetic de-spinning of a satellite by inducing eddy currents in its skin. I. Analytical considerations. *IEEE Trans. Magn.* **2002**, *31*, 2471–2477. [CrossRef]
17. Sugai, F.; Abiko, S.; Tsujita, T.; Jiang, X.; Uchiyama, M. De-tumbling an uncontrolled satellite with contactless force by using an eddy current brake. In Proceedings of the International Conference on Intelligent Robots and Systems, Hamburg, Germany, 28 September–2 October 2015.
18. Gómez, N.O.; Walker, S.J.I. Eddy currents applied to de-tumbling of space debris: Analysis and validation of approximate proposed methods. *Acta Astronaut.* **2015**, *114*, 34–53. [CrossRef]
19. Jankovic, M.; Kumar, K.; Gómez, N.O. Spacecraft concept for active de-tumbling and robotic capture of Ariane rocket bodies. In Proceedings of the 13th Symposium on Advanced Space Technologies in Robotics and Automation, Noordwijk, The Netherlands, 10–13 May 2015.
20. Figura, J.S.; James, N. A laboratory demonstration of de-tumbling space debris with magnetically-induced eddy current torques. In Proceedings of the 1st IAA Conference on Space Situational Awareness, Orlando, FL, USA, 13–15 November 2017.
21. Nishida, S.I.; Kawamoto, S. Strategy for capturing of a tumbling space debris. *Acta Astronaut.* **2011**, *68*, 113–120. [CrossRef]
22. Matunaga, S.; Kanzawa, T.; Ohkami, Y. Rotational motion-damper for the capture of an uncontrolled floating satellite. *Control. Eng. Pract.* **2001**, *9*, 199–205. [CrossRef]
23. Kawamoto, S.; Matsumoto, K.; Wakabayashi, S. Ground experiment of mechanical impulse method for uncontrollable satellite capturing. In Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space (i-SAIRAS), St-Hubert, QC, Canada, 18–22 June 2001; pp. 1–8.

24. Huang, P.; Zhang, F.; Meng, Z.; Liu, Z. Adaptive control for space debris removal with uncertain kinematics, dynamics and states. *Acta Astronaut.* **2016**, *128*, 416–430. [CrossRef]
25. O'Connor, W.J.; Hayden, D.J. De-tumbling of space debris by a net and elastic tether. *J. Guid. Control. Dyn.* **2017**, *40*, 1829–1836. [CrossRef]
26. Shan, M.; Guo, J.; Gill, E. Tumbling space debris capturing using a net. In Proceedings of the 7th European Conference on Space Debris, Darmstadt, Germany, 18–21 April 2017.
27. Hovell, K.; Ulrich, S. Attitude stabilization of an uncooperative spacecraft in an orbital environment using visco-elastic tethers. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, USA, 5–9 January 2015.
28. Velez, P.; Atie, T.; Alazard, D.; Cumer, C. Space Debris Removal using a Tether: A Model. *IFAC-PapersOnLine* **2017**, *50*, 7247–7252. [CrossRef]
29. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006.
30. Jiang, J.; Ma, Y. Path planning strategies to optimize accuracy, quality, build time and material use in additive manufacturing: A review. *Micromachines* **2020**, *11*, 633. [CrossRef]
31. Omisore, O.; Han, S.; Al-Handarish, Y.; Du, W.; Duan, W.; Akinyemi, T.; Wang, L. Motion and trajectory constraints control modeling for flexible surgical robotic systems. *Micromachines* **2020**, *11*, 386. [CrossRef]
32. Liu, Y.; Wang, D.; Zhang, Y.; Yuan, Z.; Liu, J.; Yang, S.; Yu, Y. Design and experimental study of space continuous robots applied to space non-cooperative target capture. *Micromachines* **2021**, *12*, 536. [CrossRef]
33. Kumar, P.; Gauthier, M.; Dahmouche, R. Path planning for 3-D in-hand manipulation of micro-objects using rotation decomposition. *Micromachines* **2021**, *12*, 986. [CrossRef]
34. Kavraki, L.E.; Svestka, P.; Latombe, J.C.; Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [CrossRef]
35. LaValle, S.M. Randomized kinodynamic planning. *Int. J. Robot. Res.* **2001**, *20*, 378–400. [CrossRef]
36. Phillips, J.M.; Bedrossian, N.; Kavraki, L.E. Guided expansive spaces trees: A search strategy for motion and cost constrained state spaces. In Proceedings of the IEEE International Conference on Robotics & Automation, IEEE, New Orleans, LA, USA, 26 April–1 May 2004.
37. Karaman, S.; Frazzoli, E. *Sampling-Based Algorithms for Optimal Motion Planning*; Sage Publications Inc.: Thousand Oaks, CA, USA, 2011.
38. Karaman, S.; Frazzoli, E. Optimal kinodynamic motion planning using incremental sampling-based methods. In Proceedings of the 49th IEEE Conference on Decision and Control, Atlanta, GA, USA, 15–17 December 2010.
39. Gammell, J.D.; Srinivasa, S.S.; Barfoot, T.D. Batch Informed Trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June 2014.
40. Arslan, O.; Tsiotras, P. Use of relaxation methods in sampling-based algorithms for optimal motion planning. In Proceedings of the IEEE International Conference on Robotics & Automation, Karlsruhe, Germany, 6–10 May 2013.
41. Janson, L.; Schmerling, E.; Clark, A.; Pavone, M. Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *Int. J. Robot. Res.* **2015**, *34*, 883–921. [CrossRef]
42. Urmson, C.; Anhalt, J.; Bagnell, D.; Baker, C.; Bittner, R.; Clark, M.N.; Dolan, J.; Duggins, D.; Galatali, T.; Geyer, C.; et al. Autonomous driving in urban environments: Boss and the Urban Challenge. *J. Field Robot.* **2008**, *25*, 425–466. [CrossRef]
43. Montemerlo, M.; Becker, J.; Bhat, S.; Dahlkamp, H.; Dolgov, D.; Ettinger, S.; Haehnel, D.; Hilden, T.; Hoffmann, G.; Huhnke, B.; et al. Junior: The Stanford entry in the urban challenge. *J. Field Robot.* **2008**, *25*, 569–597. [CrossRef]
44. Fletcher, L.; Teller, S.; Olson, E.; Moore, D.; Kuwata, Y.; How, J.; Leonard, J.; Miller, I.; Campbell, M.; Dan, H. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*; Springer Publishing Company: Berlin, Germany, 2009.
45. Kuwata, Y.; Teo, J.; Fiore, G.; Karaman, S.; Frazzoli, E.; How, J.P. Real-time motion planning with applications to autonomous urban driving. *IEEE Trans. Control. Syst. Technol.* **2009**, *17*, 1105–1118. [CrossRef]
46. Chen, W.; Xu, J.; Zhao, X.; Liu, Y.; Yang, J. Separated sonar localization system for indoor robot navigation. *IEEE Trans. Ind. Electron.* **2020**, *68*, 6042–6052. [CrossRef]
47. Xie, Y.; Zhou, R.; Yang, Y. Improved distorted configuration space path planning and its application to robot manipulators. *Sensors* **2020**, *20*, 6060. [CrossRef] [PubMed]
48. Premachandra, C.; Murakami, M.; Gohara, R.; Ninomiya, T.; Kato, K. Improving landmark detection accuracy for self-localization through baseboard recognition. *Int. J. Mach. Learn. Cybern.* **2017**, *8*, 1815–1826. [CrossRef]
49. Chu, X.; Hu, Q.; Zhang, J. Path planning and collision avoidance for a multi-arm space maneuverable robot. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *54*, 217–232. [CrossRef]
50. Frazzoli, E. Quasi-random algorithms for real-time spacecraft motion planning and coordination. *Acta Astronaut.* **2003**, *53*, 485–495. [CrossRef]
51. Kobilarov, M.; Pellegrino, S. Trajectory planning for cubesat short-time-scale proximity operations. *J. Guid. Control. Dyn.* **2017**, *37*, 566–579. [CrossRef]
52. Frazzoli, E.; Dahleh, M.A.; Feron, E.; Kornfeld, R. A randomized attitude slew planning algorithm for autonomous spacecraft. In Proceedings of the AIAA Guidance Navigation and Control Conference, Montreal, Canada, 6–9 August 2001.

53. Phillips, J.M.; Bedrosian, N.; Kavraki, L.E. Spacecraft rendezvous and docking with real-time randomized optimization. In Proceedings of the AIAA Guidance Navigation and Control Conference, Boston, MA, USA, 19–12 August 2013.
54. Starek, J.A.; Barbee, B.W.; Pavone, M. A sampling based approach to spacecraft autonomous maneuvering with safety specifications. In Proceedings of the 38th Annual American Astronomical Society Conference on Guidance and Control (AAS GNC Conference), Breckenridge, CO, USA, 30 January–4 February 2015; pp. 15–116.
55. Starek, J.A.; Schmerling, E.; Maher, G.D.; Barbee, B.W.; Pavone, M. Real-time, propellant-optimized spacecraft motion planning under Clohessy-Wiltshire-Hill dynamics. In Proceedings of the Aerospace Conference, IEEE, Big Sky, MT, USA, 5–12 March 2016.
56. Starek, J.A.; Schmerling, E.; Maher, G.D.; Barbee, B.W.; Pavone, M. Fast, safe, propellant-efficient spacecraft motion planning under Clohessy–Wiltshire–Hill dynamics. *J. Guid. Control. Dyn.* **2017**, *40*, 418–438. [CrossRef]
57. Starek, J.A. *Sampling-Based Motion Planning for Safe and Efficient Spacecraft Proximity Operations*; MIT: Cambridge, MA, USA, 2016.
58. Chen, N.; Zhang, Y.; Li, Z.; Cheng, W.; Li, J.; Diao, H.; Wang, W.; Fang, Y. An improved sampling-based approach for spacecraft proximity operation path planning in near-circular orbit. *IEEE Access* **2020**, *8*, 41794–41804. [CrossRef]
59. Spong, M.W.; Hutchinson, S.; Vidyasagar, M. Robot modeling and control. *Ind. Robot. Int. J.* **2006**, *17*, 709–737.
60. Pohl, I. *Bidirectional and Heuristic Search in Path Problems*; Stanford University Press: Stanford, CA, USA, 1969.
61. Luby, M.; Ragde, P. A bidirectional shortest-path algorithm with good average-case behavior. *Algorithmica* **1989**, *4*, 551–567. [CrossRef]
62. Hwang, Y.K.; Ahuja, N. Gross motion planning—A survey. *ACM Comput. Surv.* **1992**, *24*, 219–291. [CrossRef]
63. Wenhua, C. Detumbling and Control of Space Non-Cooperative Targets for Proximity Operation. Ph.D. Thesis, Space Engineering University, Beijing, China, 2019.