applied sciences

MDPI

Article

Optimizing Kinematic Modeling and Self-Collision Detection of a Mobile Manipulator Robot by Considering the Actual Physical Structure

Lijun Qiao [1], Xiao Luo [2,*], Qingsheng Luo [1], Minghao Li [3] and Jianfeng Jiang [1]

[1] School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China; 3120170132@bit.edu.cn (L.Q.); luoqsh@bit.edu.cn (Q.L.); 3120170122@bit.edu.cn (J.J.)
[2] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[3] School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China; 3120170189@bit.edu.cn
* Correspondence: luox@bit.edu.cn

Abstract: In this paper, an optimized kinematic modeling method to accurately describe the actual structure of a mobile manipulator robot with a manipulator similar to the universal robot (UR5) is developed, and an improved self-collision detection technology realized for improving the description accuracy of each component and reducing the time required for approximating the whole robot is introduced. As the primary foundation for trajectory tracking and automatic navigation, the kinematic modeling technology of the mobile manipulator has been the subject of much interest and research for many years. However, the kinematic model established by various methods is different from the actual physical model due to the fact that researchers have mainly focused on the relationship between driving joints and the end positions while ignoring the physical structure. To improve the accuracy of the kinematic model, we present a kinematic modeling method with the addition of key points and coordinate systems to some components that failed to model the physical structure based on the classical method. Moreover, self-collision detection is also a primary problem for successfully completing the specified task of the mobile manipulator. In traditional self-collision detection technology, the description of each approximation is determined by the spatial transformation of each corresponding component in the mobile manipulator robot. Unlike the traditional technology, each approximation in the paper is directly established by the physical structure used in the kinematic modeling method, which significantly reduces the complicated analysis and shortens the required time. The numerical simulations prove that the kinematic model with the addition of key point technology is similar to the actual structure of mobile manipulator robots, and the self-collision detection technology proposed in the article effectively improves the performance of self-collision detection. Additionally, the experimental results prove that the kinematic modeling method and self-collision detection technology outlined in this paper can optimize the inverse kinematics solution.

Keywords: mobile manipulator robot; kinematic modeling; self-collision detection; actual physical structure

Citation: Qiao, L.; Xiao, L.; Luo, Q.; Li, M.; Jiang, J. Optimizing Kinematic Modeling and Self-Collision Detection of a Mobile Manipulator Robot by Considering the Actual Physical Structure. *Appl. Sci.* **2021**, *11*, 10591. https://doi.org/10.3390/app112210591

Academic Editor: Manuel Armada

Received: 18 October 2021
Accepted: 5 November 2021
Published: 10 November 2021

1. Introduction

The mobile manipulator robot fixes the traditional manipulator on the mobile platform, thereby extending the substantial potential applications of the fixed manipulator [1,2]. Due to the combination of the high mobility of the mobile platform and the dexterous maneuverability of the manipulator, it can be widely used in current life and work, such as in planetary exploration, nuclear reactor maintenance, landmine detection, agriculture missions, and fire rescue operations [3–5]. The mobile manipulator will be able to smoothly complete several complex tasks if the degree of freedom of the manipulator system is increased and a more reasonable structure for the entire system is designed. [6–8]. However, improving the accuracy of the kinematic model and avoiding self-collision remain challenging problems in robot operations [9,10]. The resolution of these challenges is extremely promising for the development of mobile manipulator systems.

Kinematic modeling is a fundamental problem and a commonly discussed issue in regard to the high-precision control of the mobile manipulator robot [11–13]. The current methods of kinematic modeling include the standard Denavit–Hartenberg (SDH) method, the modified Denavit–Hartenberg (MDH) method, the screw theory method, and the dual-quaternions method. As the most typical methods, SDH and MDH are based on the homogeneous transformation matrix, but the description of the two matrices are very different [14–17]. The difference between the two methods is mainly caused by the differences in selecting the origin of the joint-fixed frames and the direction of three axes. Meanwhile, due to the difference in the number of input and output variables, the mobile platform includes holonomic and non-holonomic constraint equations. The non-holonomic constraints of the mobile platform increase the complexity of system control [18–20]. Researchers combined screw theory and Lie algebra to simplify the kinematic modeling of the manipulator robot in order to solve the kinematic modeling of the manipulator robot. However, the computational complexity of screw theories is not significantly reduced since the analysis process is still carried out via matrix operations [21–24]. With the rapid development of mathematical theory, researchers have successfully established the kinematic modeling of the manipulator robot based on dual-quaternions to reduce computational complexity. However, the dual-quaternions method is still in the research stage due to incompleteness of the theory [25–27]. Therefore, the SDH and MDH methods are the main methods in the kinematic modeling process.

The kinematic modeling of the mobile manipulator robot consists of two parts: mobile platform modeling and manipulator modeling. For the mobile platform subsystem, researchers regarded the mobile platform as two translation joints and one rotation joint. Then, they added the nonholonomic constraints in the kinematic modeling to establish the kinematic model of the mobile platform. For the manipulator subsystem, researchers have successfully developed and designed a variety of mature arms, including UR series robots, KUKA LBR series robots, and the Jaco arm robot. According to the difficulty in completing tasks and the requirements of work performance, UR series manipulators have become more commonly used manipulator systems. For the UR-serial robot, researchers have established the corresponding coordinate system directly at the end position of the link according to the SDH method. Then, they solved the position and posture of the tool end of the robotic arm via the homogeneous transformation matrix.

Nevertheless, the kinematic modeling result is notably different from the actual model of the mobile manipulator. For example, the structural parameters of the manipulator subsystem in Figure 1 are similar to that of the typical UR5 robot shown in Figure 2a, we mark these motors from the base to the end-effector as the joint i $(i = 1, 2, \ldots\ldots, 6)$, and define link 1 (link 2) as the connecting part between motor 2 and motor 3 (motor 3 and motor 4). The positions of joints 2 and 3 cannot be solved in the mathematical model established by the traditional method; the position of connecting link 1 cannot be accurately expressed in the mathematical model. Therefore, according to the mathematical model, the control of the mobile manipulator robot may not be able to meet the actual motion requirements. For example, the end tool of the robot cannot reach the specified position due to the limitation of the physical structure of the connecting link 2; the above-mentioned joints and connecting link may collide with the external environment. By comparing the kinematic model and the actual physical model, we found that researchers are mainly focused on the relationship between the rotation angle of the driving joints and the position and posture of the robot. In order to ensure that the mathematical model is more consistent with the physical model of the mobile manipulator with the UR-like arm, we should consider the actual physical structure of the manipulator system in the kinematic modeling process.

Self-collision detection is another major problem in controlling the mobile manipulator, and it has been concerned and studied by various researchers for many years [28–30]. According to the literature on self-collision detection without external sensors, the self-collision detection of mobile manipulators mainly includes three steps: approximating each component, describing the position of each approximate object, and determining self-collision judgment. The first step aims to simplify each component through standard geometry to facilitate the mathematical analysis of self-collision detection. Owing to the complex structure of the mobile manipulator, the standard geometry, also called an approximate element, mainly includes the ellipsoidal object [31], the cylindrical object [32] and the spherical object [33]. Furthermore, multiple approximate elements are utilized to simplify each component for improving the approximating performance [34]. The second step describes the actual position of each approximation in the three-dimensional space during the movement of the mobile manipulator robot. Currently, the basic principle of the second step is the spatial transformation method, which needs to consider the positional relationship of each component at different times [35]. After completing the above steps, various self-collision detection technologies are proposed according to the actual needs of a given task. The simplest collision detection method directly uses the shortest distance between the surface points of different components as the primary judgment [36]. However, this method takes a long time and cannot be used for real-time collision detection. To reduce the required time for collision detection, the researchers proposed to use the relationship between spatial geometric objects as the principle of determination [37]. Additionally, the researcher reduced self-collision detection of components that never collide for optimizing self-collision detection technology [38].

In this paper, we developed a kinematic modeling method to accurately describe the actual structure of the mobile manipulator robot with a UR-like arm. Meanwhile, we specifically improved the performance of the self-collision detection process by reasonably approximating the structure of the mobile manipulator robot and reducing the time required for position description. The main contributions of this study are as follows:

(1) This paper proposes a new type of kinematic modeling method for mobile manipulator robots. We optimized the difference between the kinematic model and the actual physical model by considering the structure of the manipulator subsystem. Therefore, the new modeling method can make the kinematic model similar to the actual model. Moreover, the modeling method of the manipulator subsystem is also applicable to other UR series robots.

(2) We improve self-collision detection technology by considering the structural characteristics of the mobile manipulator robot. To improve the accuracy of self-collision detection, we approximate each component by using different spherical objects instead of a uniform one. Meanwhile, the positioning of each approximate object is described by the structural parameters instead of the previous spatial transformation, which can effectively reduce the required time for preparing self-collision detection.

(3) The simulations prove the correctness of the kinematic modeling method for the mobile manipulator robot with a UR-like arm. By comparing with other studies, the improvement proposed in the approximation and the positioning of each approximate object can enhance the performance of self-collision detection.

The rest of the article is structured as follows. In Section 2, the physical model of the mobile manipulator with a UR-like arm is developed. Section 3 proposes a new kinematic modeling method to compensate for the defects in classical kinematic modeling. Section 4 describes the optimization of the self-collision detection technology in the approximation and positioning of each component based on the actual physical structure. In Section 5, the basic parameters of the mobile manipulator robot with a UR-like arm are identified for the design of simulations, and the results demonstrate the validity of the kinematic modeling method and the improvement in self-collision detection proposed in this paper. Section 6 summarizes the full text and describes the future work.

2. Physical Structure of Mobile Manipulator Robot

The object studied in this paper is shown in Figure 1. The physical structure of the mobile manipulator robot is composed of three parts, namely the mobile platform subsystem, the mechanical manipulator subsystem, and the carrying load. The mobile platform subsystem is controlled by the differential rotational speed of two driving wheels, it is mainly used to transport the robot to the location near the task position. The manipulator subsystem is composed of motors and links, and it serves as the end actuator to detect the location environment and grab the target object. Finally, the carrying load can handle maintenance items and place various tools.

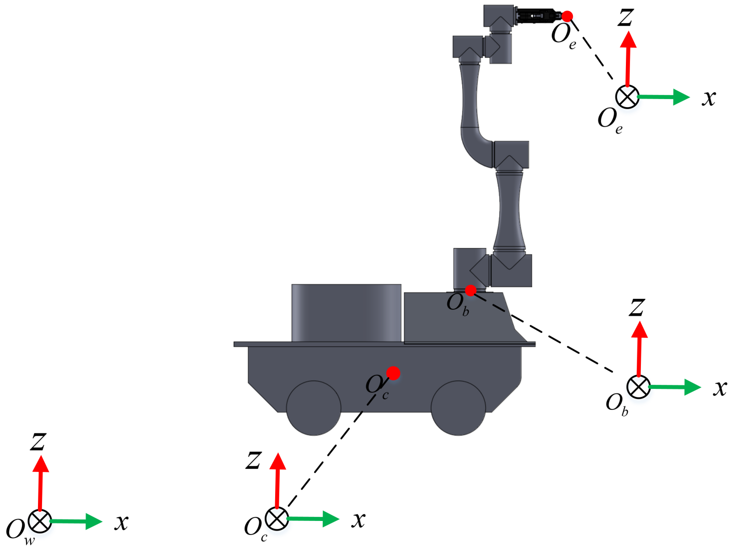


Figure 1. The physical structure of the mobile manipulator robot.

The various coordinate frames should be primarily established to determine the position and posture of the mobile manipulator robot. Generally, the position and posture of the manipulator subsystem are determined by the homogeneous matrix after establishing the coordinate frame based on the standard Denavit–Hartenberg (SDH) method and the modified Denavit-Hartenberg (MDH) method. The SDH and MDH methods have significant differences in choosing the origin and the z-axis of each coordinate system. In the SDH method [39], the next motor rotation direction is regarded as the z-axis of each joint-fixed frame. In contrast, the current motor rotation direction is regarded as the z-axis of each joint-fixed frame in the MDH method [40]. To determine the position and posture of the manipulator subsystem in the world frame, we need to acquire the fixed position and posture of the manipulator's base in the mobile coordination frame. Moreover, the position and posture of the mobile platform in the world frame must be determined during the movement.

To accomplish the specified tasks, we established the following different types of coordinate systems in the mobile manipulator robot:

(1) The world frame O_W. The origin of the world frame is fixed at the specified position and remains unchanged. It is also defined as the global coordinate system in the space in which the robot is located.
(2) The body-fixed frame O_c. The origin of the body-fixed frame is attached to the center of the mobile platform, and the direction of each axis is defined according to the motion requirements. It is used to describe the position and posture of the mobile platform.
(3) The base-fixed frame O_b. The origin of the base-fixed frame is fixed at the connection position of the manipulator subsystem on the mobile platform, and the direction of each axis is established according to the manipulator coordinate system. It is used to describe the position and posture of the manipulator's base.
(4) The end-fixed frame O_e. The origin of the end-fixed frame is attached to the end of the manipulator subsystem, and the direction of each axis is established based on

the principle of robotic arm modeling. It is mainly used to describe the position and posture of the end of the manipulator robot.

3. The Optimized Kinematic Modeling Method

This section mainly examines the forward kinematic modeling method of mobile manipulator robots with UR-like arms. According to the defects in classical modeling methods, this paper proposes an optimized kinematic modeling method to ensure that the mathematical model is similar to the actual physical model.

3.1. Traditional Kinematic Modeling of the Mobile Manipulator Robot

For the mobile manipulator robot in Figure 1, the manipulator subsystem is composed of two links and six motors. To establish a mathematical model, the subsystem can be simplified to six joints and seven links according to the standard Denavit–Hartenberg (SDH) method. Then, the position and posture of the end tool of the manipulator subsystem can be solved by the homogeneous matrix transformation between adjacent coordinate frames. Assuming the mobile platform moves on the horizontal plane, since the mobile platform is driving by the rear wheels, the position and posture of the midpoint of the driving wheels can be solved by the movement along the x-axis and y-axis and the rotation around the z-axis of the world frame. Therefore, the control variables of the entire mobile robotic arm system can be specifically expressed as

$$q = [q_m, q_a]^T = [d_x, d_y, q_z, q_1, q_2, \ldots\ldots, q_n]^T \tag{1}$$

where q, q_m, and q_a indicate the control variables of mobile manipulator subsystem, the mobile platform, and manipulator subsystem, respectively; d_x and d_y indicate the distance moved along the x-axis and y-axis; q_z indicates the rotation angle around the z-axis of the world frame; and q_i indicates the rotation angle of the i joint in the manipulator subsystem.

In practical tasks, the matrix transformation from the midpoint coordinate frame of the driving wheels to the center of gravity of the mobile platform is fixed. Moreover, the transformation matrix from the mobile-fixed frame to the base-fixed frame does not change. According to the SDH method, we can obtain the position and posture of the end-effector of the mobile manipulator in the world frame, which is expressed as

$${}^{o}T_e = {}^{o}T_c \cdot {}^{c}T_b \cdot {}^{b}T_1 \cdot \ldots\ldots \cdot {}^{n-1}T_n \cdot {}^{n}T_e \tag{2}$$

where ${}^{o}T_c$ indicates the position and posture of the center of gravity of the mobile platform in the world frame, ${}^{c}T_b$ indicates the position and posture of the manipulator's base in the body-fixed frame, ${}^{b}T_1$ indicates the homogeneous transformation matrix from the 1-th joint to the base-fixed frame, ${}^{n}T_e$ indicates the homogeneous transformation matrix from the end-fixed frame to the n-th joint-fixed frame, and ${}^{i-1}T_i$ indicates the homogeneous transformation matrix from the i-th joint-fixed frame to the $(i-1)$-th joint-fixed frame.

3.2. Limitations of Kinematic Modeling

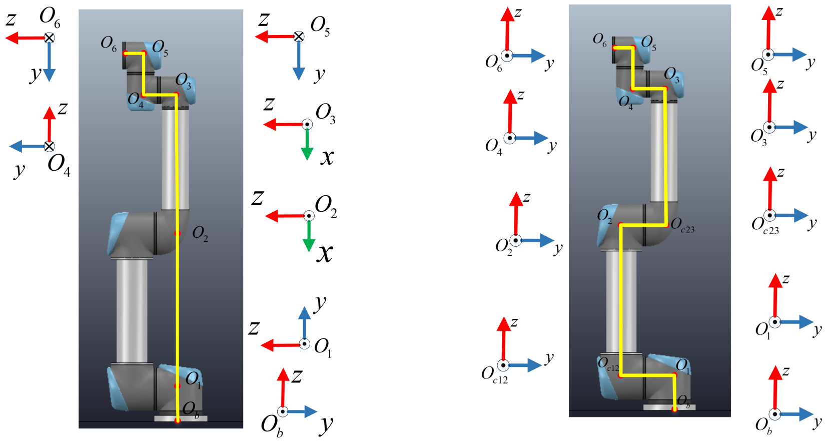
The forward kinematic of the mobile manipulator robot is mainly used to solve the position and posture of the mobile manipulator. In this article, the mobile manipulator is composed of two parts: the mobile platform and the manipulator system. Therefore, kinematic modeling is established based on the basic parameters of each link and the essential parameters of the mobile platform. However, many shortcomings have been observed in the kinematic modeling results of the mobile manipulator robot based on other studies [41,42].

3.2.1. Limitations of Kinematic Modeling of the Manipulator

In this article, the initial parameters and installation positions of these motors are similar to the typical UR5 robot. To simplify the control of the UR arm, researchers have developed a classical kinematic modeling result based on the Standard Denavit–Hartenberg

(SDH) method [41]. Therefore, the kinematic modeling of the manipulator subsystem based on the classical method is shown in Figure 2a. In the classical method, motor 2, link 2 and motor 3 are regarded as a complete connecting link. With the comparison between the actual physical model and the mathematical model, we found that (1) the huge difference between the position of the second joint and the location of the actual motor cannot be ignored, and it can also be seen in the third joint; (2) the position of the connecting link between the second joint and the third joint acquired by the classical method is unable to match the actual position.

Comparing kinematic modeling results, we can find that the mathematical model based on the classical kinematic modeling method by only establishing the local coordinate frame at each joint cannot accurately describe the actual physical model. Due to the great difference between the mathematical model and the physical model, the collision between one component and another component or the external environment cannot be accurately judged. Furthermore, the establishment of each coordinate frame needs to be complicated according to the specified requirement, which is not suitable for quickly establishing a kinematic model for the UR5 robot. With in-depth research on the SDH method, we can establish seven local coordinate frames for the UR series robot based on the driving joints, but more coordinate frames were required to describe the basic structure of the manipulator. Therefore, the inability to establish some coordinate frames on the structure of some components is the primary factor that leads to the inaccuracy of kinematic modeling.



(**a**) (**b**)

Figure 2. Some coordinate frames established for the kinematic modeling of the manipulator subsystem. (**a**) Kinematic modeling based on the classical method; (**b**) Kinematic modeling based on the optimized method.

3.2.2. Limitations of Kinematic Modeling of the Mobile Platform

To determine the position and posture of the mobile platform during the movement, we need to simplify the mobile platform into two translation joints and one rotation joint. After simplifying the mobile platform, we still need to establish the corresponding coordinate system on the center of gravity of the mobile platform, which can be seen in Figure 3a. However, three different coordinate systems were established at the same location according to the SDH method, which inevitably increased the difficulty of system analysis and reduced the speed of the solution.

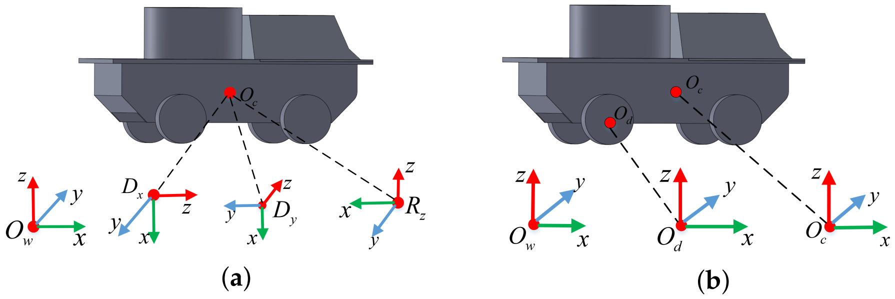


Figure 3. The kinematic modeling of the mobile platform. (**a**) Kinematic modeling based on the DH method; (**b**) Kinematic modeling based on the optimized method.

3.3. Optimized Kinematic Modeling of the Mobile Manipulator Robot

In order to improve the accuracy of the kinematic model for a UR-like manipulator subsystem and reduce the complexity of the coordinate frames for the mobile subsystem, this paper optimized the kinematic modeling technology of each subsystem of the mobile manipulator robot.

3.3.1. Optimized Kinematic Modeling of the Manipulator

When establishing kinematic modeling using the classical method, the physical structure of the actual links and the actual position of the joint are ignored to emphasize the relationship between the location of the end tool and the driving joints. At the same time, the establishment of the joint-fixed coordinate frame is very complicated. To meet the structural requirements of the mathematical model and reduce the complexity of kinematic modeling, we optimized the kinematic modeling method by re-establishing the coordinate frames on the joint and adding some coordinate frames on the physical structure of the connecting links. Therefore, we selected the key points both on each joint and on the conversion positions in the actual connecting link, which is shown in Figure 2b. Some key points are mainly used to express the location of joints, but two adjacent points can be used to describe the structure of each link. The specific steps of the optimized kinematic modeling method are as follows:

Step 1: Number links and joints.

For the manipulator robot with n-degree-of-freedom, the links are marked from the base (0) to the final end-effector (n). Furthermore, the joints are identified from 1 to n, the joint i is established between the link $(i-1)$ and the link i.

Step 2: Judge whether to add the key point on the structure.

After connecting the adjacent joints through a straight line, we need to observe whether the straight line between the corresponding joints is similar to the structure of the connecting link. If they are the same, we do not need to add other key points. Otherwise, the key points must be added to the location where the structure changes. After establishing the key points, we connect the joint points and the key point again to determine whether the structure of the connecting link can be expressed by these lines. If it cannot be expressed, we add the other key points again until the structure of connecting link can be expressed by these straight lines.

Step 3: Establish all coordinate frames.

(1) Establish the joint-fixed coordinate frame. The origin of the joint-fixed coordinate frame was established at the location of the corresponding joint. The direction of one axis needs to be the same as the rotation direction of the joint, and the other axis directions need to meet the left-hand rule.

(2) Establish the key points-fixed coordinate frames. We establish the coordinate frame at the positions of key points, and the three-axis directions of those coordinate frames need to meet the left-hand rule.

(3) Establish the other coordinate frames. We establish the base-fixed coordinate frame at the base position according to the actual requirement. The origin of the base-fixed

coordinate frame is fixed in the convenient position of the robot's base, and three-axis directions need to meet the left-hand rule.

Step 4: Identify transformation parameters.

(1) Obtain three distances (d_x^i, d_y^i, d_z^i). The origin of the coordinate frame (O_i) moves to the origin of the next coordinate frame (O_{i+1}) along the three coordinate axes with the order of x_i-axis, y_i-axis, and z_i-axis.

(2) Obtain three angles (q_x^i, q_y^i, q_z^i). The three-axis directions of the i-th coordinate frame are rotated to the three-axis directions of the $(i+1)$-th coordinate frame around the three coordinate axes with the order of x_i-axis, y_i-axis, and z_i-axis.

(3) Determine the angle variable which is controlled by the driving joint.

Step 5: Determine the homogeneous transformation matrix.

The current coordinate frame can reach the adjacent coordinate frame through rotation and translation relative to the current coordinate frame. Therefore, the corresponding homogeneous transformation matrix is as follows:

$$ {}^{i-1}T_i = Tr_x(d_x^i)Tr_y(d_y^i)Tr_z(d_z^i)R_x(q_x^i)R_y(q_y^i)R_z(q_z^i) \tag{3} $$

where $Tr_i()$ represents the translation transformation matrix along the corresponding coordinate axis, and $R_i()$ represents the rotation transformation matrix around the corresponding coordinate axis. The specific representation of various transformation matrices is expressed as:

$$ \begin{gathered} R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_{q_x} & -S_{q_x} & 0 \\ 0 & S_{q_x} & C_{q_x} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; R_y = \begin{bmatrix} C_{q_y} & 0 & S_{q_y} & 0 \\ 0 & 1 & 0 & 0 \\ S_{q_y} & 0 & C_{q_y} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; R_z = \begin{bmatrix} C_{q_z} & -S_{q_z} & 0 & 0 \\ S_{q_z} & C_{q_z} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \\ Tr_x = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; Tr_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & 0 \\ 0 & & 0 & 1 \end{bmatrix}; Tr_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}; \end{gathered} \tag{4} $$

where C_{q_i} indicates $cos(q_i)$ and S_{q_i} indicates $sin(q_i)$.

Step 6: Determine the position and posture of the coordinate frames in the base-fixed frames.

The position and posture of each coordinate frame in the base-fixed coordinate system can be described by the homogeneous transformation matrix from the base-fixed coordinate frame to the corresponding coordinate frame, which can be expressed as:

$$ \begin{cases} {}^bT_i{=}{}^bT_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot \ldots\ldots \cdot {}^{i-1}T_i, (i = 1, 2, ..., n); \\ {}^bP_i = [{}^bT_i(1,4), {}^bT_i(2,4), {}^bT_i(3,4)]^T; \end{cases} \tag{5} $$

where ${}^bT_i$ indicates the position and posture of each coordinate frame in the base-fixed coordinate frame, and ${}^{i-1}T_i$ indicates the homogeneous transformation matrix from the $(i-1)$-th coordinate system to the i-th coordinate frame.

By adding the point on the structure of the connecting link, the kinematic modeling result is similar to the actual physical model rather than the kinematic model based on the classical method, as shown in Figure 2. The position of the manipulator's joint met the actual position requirements, and the structure of the kinematic model has a small difference from the actual physical model. Meanwhile, the establishment of each local coordinate frame and the acquisition of key parameters are simpler than the classical methods.

Many different coordinate frames are attached to each joint, some structures of the connecting links, and the end-effector. The relationship between each coordinate frame can be determined by the homogeneous transformation matrix. Therefore, the position and posture of the end-effector of the manipulator in the base-fixed frame can be directly solved by the homogeneous transformation matrix multiplication from the base-fixed frame to the end-fixed frame, which can be described as

$$ {}^{b}T_e = {}^{b}T_1 \cdot {}^{1}T_2 \cdot \ldots \cdot {}^{n-1}T_n \cdot {}^{n}T_e \tag{6}$$

where ${}^{b}T_e$ indicates the homogeneous transformation matrix from the end-fixed coordinate frame to the base-fixed frame, ${}^{n}T_e$ indicates the homogeneous transformation matrix from the end-fixed frame to the n-th joint-fixed frame, and ${}^{i-1}T_i$ indicates the homogeneous transformation matrix from the i-th joint-fixed frame to the $(i-1)$-th joint-fixed frame.

3.3.2. Optimized Kinematic Modeling of the Mobile Platform

We assume that the mobile manipulator robot is mainly composed of rigid components to guarantee the stability and precision of the manipulator robot's movement. Meanwhile, the mobile platform mainly moves on the road with high friction and no slide or slip during movement. Therefore, the mobile platform is directly driven by the angular velocity of the motors mounted on the left and right wheels. With in-depth research on mobile manipulator robots, the mobile platform mainly realizes the movement of the mobile manipulator, which can be regarded as the transformation of the based-fixed frame. Therefore, we can determine the position and the posture of the mobile platform by the homogeneous transformation matrix from the world frame to the midpoint of the driving wheels.

We directly established a coordinate system at the midpoint of the two driving wheels to replace the three coordinate systems previously established at the center of gravity of the mobile platform, which can be seen in Figure 3b. This method can reduce the difficulty of establishing the coordinate system of the mobile platform and increase the speed of the kinematic modeling. The homogeneous transformation matrix from the world frame to the midpoint of the two driving wheels can denote the position and posture of the mobile platform, which can be expressed as follows:

$$ {}^{O}T_d = \begin{bmatrix} C(q_{zcs}) & -S(q_{zcs}) & 0 & p_{xd} \\ S(q_{zcs}) & C(q_{zcs}) & 0 & p_{yd} \\ 0 & 0 & 1 & p_{zd} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where ${}^{O}T_d$ indicates the position and posture of the midpoint of the driving axis in the world frame; $C()$ and $S()$ indicate $cos()$ and $sin()$, p_{xd}, p_{yd}, p_{zd} indicate the position of the midpoint of the drive axis in the world frame; and q_{zcs} indicates the rotation angle around the z-axis of the body-fixed frame.

3.3.3. Optimized Kinematic Modeling of the Entire System

Since the manipulator's base is fixed on the mobile platform subsystem, the position and posture of the manipulator's end-effector in the frame will inevitably be affected by the fixed position and posture. To determine the position of the manipulator's end-effector, we need to convert the reference coordinate system from the base-fixed frame to the world frame by considering the homogeneous transformation matrix expressing the different fixing positions. Therefore, the position and posture of the base-fixed frame can be denoted as:

$$\begin{aligned} {}^{d}T_b &= {}^{d}T_c \cdot {}^{c}T_b = {}^{d}T_c \cdot Tr_x(x_{cb})Tr_y(y_{cb})Tr_z(z_{cb})R_x(q_{xcb})R_y(q_{ycb})R_z(q_{zcb}) \\ &= \begin{bmatrix} 1 & 0 & 0 & x_{dc}+x_{cb} \\ 0 & 1 & 0 & y_{dc}+y_{cb} \\ 0 & 0 & 1 & z_{dc}+z_{cb} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{8}$$

where $Tr_i()$ indicates the translation transformation matrix; $R_x(), R_y(), R_z()$ represent the rotation transformation matrix; x_{cb}, y_{cb}, and z_{cb} indicate the translational distance of the base-fixed frame along the three axes of the body-fixed frame; q_{xcb}, q_{ycb}, and q_{zcb} represent the rotation angle of the base-fixed frame around the three axes of the body-fixed frame; x_{dc}, y_{dc}, and z_{dc} indicate the translational distance from the midpoint of the driving axis to the body-fixed frame along the three axes.

According to the above Equations (6)–(8), the position and posture of the end of the mobile manipulator can be specifically expressed with the homogeneous transformation matrix in Equation (2). For the designated task, the position and posture of the robot's end are mainly described as follows:

$$ {}^{o}T_e = {}^{o}T_d \cdot {}^{d}T_b \cdot {}^{b}T_e \tag{9}$$

where ${}^{o}T_e$ indicates the position and posture of the manipulator's end-effector in the world frame, ${}^{o}T_d$ indicates the position and posture of the midpoint of two driving wheels in the world frame, ${}^{d}T_b$ indicates the position and posture of the manipulator's base in the midpoint-fixed frame, and ${}^{b}T_e$ indicates the position and posture of the manipulator's end in the base-fixed frame.

The position and posture of the manipulator's end are mainly described as six variables in the Cartesian coordinate system. To improve the efficiency of performing tasks, we generally only set the position requirements to complete the given task. Therefore, it is necessary to convert the homogeneous transformation matrix into the position and angle in the Cartesian space:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ O_x \\ O_y \\ O_z \end{bmatrix} = \begin{bmatrix} {}^{o}T_e(1,4) \\ {}^{o}T_e(2,4) \\ {}^{o}T_(3,4) \\ atan\left(\frac{{}^{o}T_e(3,2)}{{}^{o}T_e(3,3)}\right) \\ atan\left(-\frac{{}^{o}T_e(3,1)}{\sqrt{{}^{o}T_e(3,2)^2+{}^{o}T_e(3,3)^2}}\right) \\ atan\left(\frac{{}^{o}T_e(2,1)}{{}^{o}T_e(1,1)}\right) \end{bmatrix} \tag{10}$$

where P_x, P_y, P_z indicate the position of the end of the manipulator subsystem in the world frame, and O_x, O_y, O_z indicate the Euler angle of the end of the manipulator subsystem around the world frame.

4. The Improved Self-Collision Detection Technology

This section mainly focuses on the improvement of the self-collision detection technology of the mobile manipulator robot. Self-collision detection needs to be processed through multiple key steps, which affect the result. Research on the optimization of each key step can not only improve the accuracy of self-collision detection but also reduce the time required for it. The remainder of this section specifically shows the optimization steps.

4.1. Approximate Improvement Via Multiple Sphere Elements

Approximating the components is the first key step in self-collision detection, which mainly simplifies the description of the robot to make the component more suitable for self-collision detection than the actual physical structure. However, the objects used by researchers to approximate the structure shapes of various components mainly include the spherical element, the cylindrical element, and the elliptical element. When approximating the components of the robot, the standard approximate element needs to meet the following basic conditions: for the spherical element, the sphere's radius is regarded as the largest value to ensure that it surrounds the outer boundary of each component; for the cylindrical element, the length of the cylinder is equal to the maximum distance in the three directions, and the radius of the cylinder can envelop the lateral boundary of each component; finally, for the elliptical element, the longest axis of the ellipse needs to envelop the longest endpoint of the component, in addition to the shortest axis also being required to envelop the shortest cross-sectional length of the component, and the length of the other axis is selected as the minimum distance. The approximate results of the same components utilizing different approximate elements are shown in Figure 4. The basic conditions can completely distinguish the possible collisions between the surface points of different components.

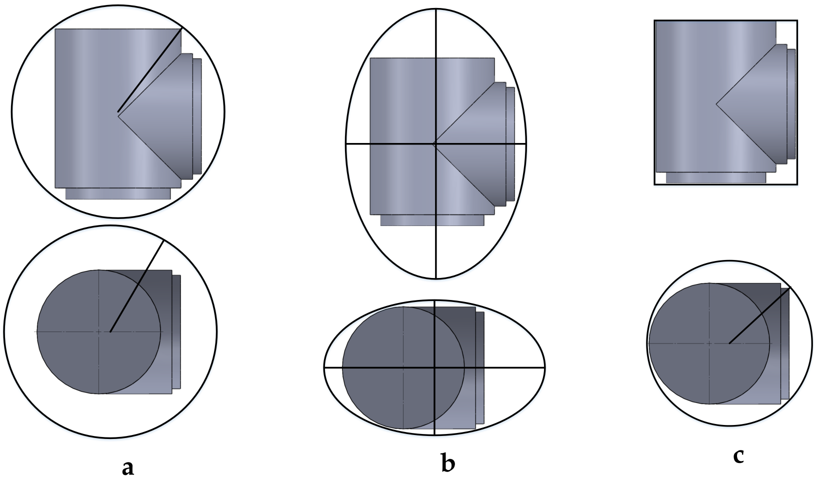


Figure 4. The basic approximate elements. (**a**) Rough spherical approximation of a motor; (**b**) Rough ellipsoidal approximation of a motor; (**c**) Rough cylindrical approximation of a motor.

Currently, spherical elements are commonly selected to approximate each component. This is because other elements have many defects in approximating some components of the mobile manipulator robot. For example: first, collision detection is impossible to be determined by using the cylindrical element to approximate each component of the entire system, which is caused by different types of collisions in different positions of two cylindrical objects, including the arc surfaces of two cylindrical objects, the horizontal surface of two cylindrical objects, and the horizontal surface of one cylindrical object and the arc surface of another cylindrical object; second, collision deletion cannot be quickly determined using elliptical elements to describe each component, as the collisions of two elliptical objects can only be determined by the shortest distance between two surface points; third, collision detection of two spherical objects can be determined by the shortest distance between two surface points as well as by the shortest distance from the center of the radius of one sphere to that of another. Therefore, to accurately describe the various components and quickly determine the real-time self-collision detection, spherical elements were used to pre-process the self-collision detection of the mobile manipulator robot. As such, various suitable spherical objects were selected for the approximation of each component.

One spherical element cannot simplify all components of the mobile manipulator robot, and we should simplify the special parts by using multiple sphere elements. In the mobile manipulator robot, self-collision detection requires different approximations of various components, including the mobile platform subsystem, the manipulator subsystem, and the carrying load. The mobile platform has a complete and closed structural feature, and the motors of the manipulator subsystem were also designed with a complete and closed structural feature. However, the connecting link between the adjacent joints of the manipulator subsystem are designed according to the actual requirements and appearance beautification. To quickly establish the approximation of the entire system for collision detection, each component needs to be rapidly approximated by selecting a simplified approximate object. For these motors in the manipulator subsystem, we can directly select a single sphere element to approximate, which can be seen in Figure 4a. However, some components cannot be directly regarded as a single spherical object, such as the connecting link and the mobile platform. In this case, we should select multiple sphere elements to describe the component in order to achieve a specific structure shape, which is shown in Figure 5.

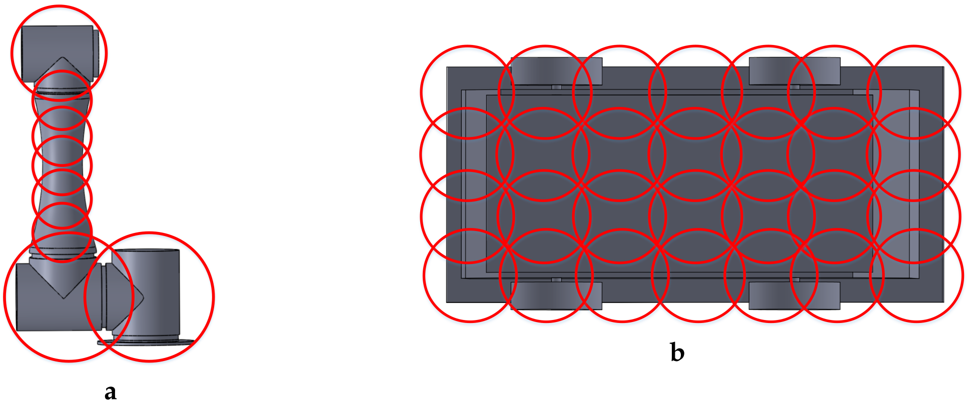


Figure 5. Multi-element approximation of the component. (**a**) The approximation of the component in the manipulator subsystem; (**b**) The approximation of the mobile platform.

4.2. Determine the Position of Each Approximation Object

During the movement of the robot, the description of the position and posture of the approximation of each component in the three-dimensional space are carried out during the second key step. The specific structural shape of each component can be described by the spherical element, but the actual position and posture of each component in three-dimensional space cannot be represented by the approximation obtained by the above method, which is only described in the local self-coordinate system. To acquire the actual position of each compartment in three-dimensional space, the traditional method needs to transform the approximation from the local self-coordinate system into the world frame through the homogeneous transformation matrix. The traditional methods and the transformation results can be found in [30,36].

However, the traditional method has many defects in the positioning of approximation, and these defects will seriously affect the efficiency of robot expansion detection. For example, (1) to acquire the surface point and the center point of each spherical object, the most time required for the self-collision detection is spent on determining the position of the surface point and the center of the spherical objects by corresponding homogeneous transformation; (2) in the movement of the robot, the positioning of different approximations requires the spatial transformation matrix corresponding to the component at different times to be solved, which increases the complexity of the positioning of approximation; (3) during the positioning of each component, the surface points of each approximation require spatial transformation by the homogeneous matrix of the corresponding component, which will inevitably increase the required time for positioning the entire approximation of the mobile manipulator robot.

To reduce the complexity of the solution and shorten the required time for the positioning of approximation, we improved the positioning of approximation based on the structure of the mobile manipulator robot. In Section 3, the mathematical model is similar to the actual physical model of the mobile manipulator robot. Therefore, the position of two adjacent key coordinate systems can directly describe the position and shape of the corresponding component in the world frame. The surface point and the center of the spherical objects can be obtained by effectively dispersing the component, which reduces the required time for approximating each component and improves the efficiency of collision detection. The specific steps are described as follows:

4.2.1. Determine the Position of the Motor and the Carrying Load

For the motor and the carrying load of the manipulator subsystem, we can directly describe the positioning of each motor by the corresponding joint-fixed coordinate frame, and we can also confirm the radius of the spherical approximation based on the basic parameters of the motor. Therefore, the corresponding spherical myopic object space is defined as

$$\begin{cases} p_{xi}^{J} = C_x^J(i) + R_s^J(i) \cdot cos(m^J); \\ p_{yi}^{J} = C_y^J(i) + R_s^J(i) \cdot sin(m^J) \cdot cos(n^J); \\ p_{zi}^{J} = C_z^J(i) + R_s^J(i) \cdot sin(m^J) \cdot sin(n^J); \end{cases} \tag{11}$$

where $p_{xi}^J, p_{yi}^J, p_{zi}^J$ represent the positioning result of the corresponding approximation of the i-th motor; $R_s^J(i)$ represents the radius of the corresponding approximation of the i-th motor; $C_x^J(i), C_y^J(i), C_z^J(i)$ represent the position of the center of the corresponding approximation; and m^J, n^J represent the minimum divided distance. The value range of m and n is $[0, 2\pi]$, $[0, \pi]$.

4.2.2. Determine the Position of the Link and the Tool

For the link or the tool of the manipulator subsystem, we can directly describe their positioning by two adjacent joint coordinate systems, such as the second link in Figure 5a. The lower-key point is $p_1 = [x_1, y_1, z_1]^T$, the upper-key point is $p_2 = [x_2, y_2, z_2]^T$, the radius of the corresponding link is R_{L2}, and the height is H_{L_2}, so the unit vector of the straight line regarded as the corresponding link at any time can be expressed as

$$e_{p_1 p_2} = \frac{[x_2, y_2, z_2]^T - [x_1, y_1, z_1]^T}{\parallel [x_2, y_2, z_2]^T - [x_1, y_1, z_1]^T \parallel} \tag{12}$$

where $e_{p_1 p_2}$ represents the unit vector of the straight line for the corresponding link.

We determined the center of each spherical object. Due to analysis of the basic structural parameters of the connecting link, the radius of the corresponding multiple spheres is determined as $^L_s(i)$. For the surrounding outer boundary of the connecting link and the motors with multiple spheres, the small spherical objects envelop the connecting link from the center to the two ends, and the large sphere envelops the motors at both ends of the link. Thus, the center position of multiple spherical objects can be expressed as

$$\begin{cases} C_i^L = \dfrac{[x_2 + x_1, y_2 + y_1, z_2 + z_1]^T}{2} - k \cdot e_{p_1 p_2} \\ k = -\dfrac{i}{2} : 1 : \dfrac{i}{2}; \\ i = \dfrac{H_{L_i}}{\sqrt{2} R_s^L(i)}; \end{cases} \tag{13}$$

where C_i^L represents the position of the center of the corresponding approximation for the link, and H_{L_i} represents the length of the corresponding link.

We determined the location of the surface point. After determining the center and the radius of each sphere based on the basic structure of the corresponding component and Formula (13), the location of the surface point can be acquired via the surface expression of each sphere, which can be denoted as

$$\begin{cases} p_{xi}^{L} = C_x^L(i) + R_s^L(i) \cdot cos(m^L); \\ p_{yi}^{L} = C_y^L(i) + R_s^L(i) \cdot sin(m^L) \cdot cos(n^L); \\ p_{zi}^{L} = C_z^L(i) + R_s^L(i) \cdot sin(m^L) \cdot sin(n^L); \end{cases} \tag{14}$$

where $p_{xi}^J, p_{yi}^J, p_{zi}^J$ represent the positioning result of the corresponding approximation of the corresponding link; $C_x^L(i), C_y^L(i), ^L C_z(i)$ represent the center of the spherical object approximated the corresponding link; and m^L, n^L represent the minimum divided distance of the corresponding spherical object. The value range is $[0, 2\pi]$, $[0, \pi]$.

4.2.3. Determine the Position of the Mobile Platform and the Support Frame

For the mobile platform or the support frame of the manipulator subsystem, we can directly describe their positioning by the mobile-fixed coordinate frame and the basic structural parameters, and we can also confirm the radius of the spherical approximation based on the basic parameters of the mobile platform or the support frame. For example, for the mobile platform, the positioning of the midpoint of the driving wheels of the mobile platform is expressed as $[p_{xd}, p_{yd}, p_{zd}]$; the positioning of the center of gravity of the driving wheel of the mobile platform is expressed as $[p_{xc}, p_{yc}, p_{zc}]$. Therefore, the unit vectors along the x- and y-axes of the body-fixed frame can be expressed as

$$\begin{cases} e_{xdc} = \dfrac{[p_{xd}, p_{yd}, 0]^T - [p_{xc}, p_{yc}, 0]^T}{\parallel [p_{xd}, p_{yd}, 0]^T - [p_{xc}, p_{yc}, 0]^T \parallel}; \\ e_{ydc} = e_{xdc} \times e_z; \end{cases} \tag{15}$$

where e_{xdc} represents the unit vector along the x-axis of the body-fixed frame; $e_z = [0, 0, 1]$ represents the unit vector along the z-axis of the body-fixed frame; and e_{ydc} represents the unit vector along the y-axis of the body-fixed frame.

We determined the center of each spherical object. The length, width, and height of the mobile platform are three fixed known quantities, and the radius of the corresponding myopic sphere is expressed as R_s^M. Thus, the center position of multiple spherical objects can be expressed as

$$\begin{cases} C_{ij}^M = C^M - p \cdot e_{ydc}, (p = -\dfrac{j}{2} : 1 : \dfrac{j}{2};) \\ C^M = [P_{xc}, P_{yc}, P_{zc}]^T - k \cdot e_{xdc}, (k = -\dfrac{i}{2} : 1 : \dfrac{i}{2}); \\ i = \dfrac{L_{M_i}}{\sqrt{2}R_s^M}; j = \dfrac{B_{M_i}}{\sqrt{2}R_s^M}; \end{cases} \tag{16}$$

where C_{ij}^M represents the position of the center of the corresponding approximation for the mobile platform, L_{M_i} represents the length of the mobile platform, and B_{M_i} indicates the width of the mobile platform.

We determined the location of the surface point. After determining the center and the radius of each sphere based on the basic structure of the corresponding component and Formula (16), the location of the surface point can be acquired via the surface expression of each sphere, which can be denoted as

$$\begin{cases} p_{xij}^M = C_x^M(i) + R_s^M(i) \cdot cos(m^M); \\ p_{yij}^M = C_y^M(i) + R_s^M(i) \cdot sin(m^M) \cdot cos(n^M); \\ p_{zij}^M = C_z^M(i) + R_s^M(i) \cdot sin(m^M) \cdot sin(n^M); \end{cases} \tag{17}$$

where $p_{xij}^J, p_{yij}^J, p_{zij}^J$ represent the positioning result of the corresponding approximation of the mobile platform; $C_x^M(i), C_y^M(i), ^M C_z(i)$ represent the center of the spherical object approximated the corresponding link; and m^M, n^M represent the minimum divided distance of the corresponding spherical object. The value range is $[0, 2\pi]$, $[0, \pi]$.

After completing the positioning of each component in the three-dimensional space, the pre-processing of self-collision detection of the manipulator robot is complete. Next, we need to determine whether there is a collision between the different components.

4.3. Improve the Judgment of Self-Collision Detection

At present, the main judgment principles are mainly determined directly by the surface points of different parts or by the relationship between the centers and the radius of different spherical approximations. In the following, we describe in detail the principles of various types of collision determination.

After completing the positioning of each component in the three-dimensional space, the surface of approximation can be used to describe the structure shape of each component in the mobile manipulator robot. Therefore, self-collision detection between different components can be directly judged by the nearest point on the surface of different spheres. Particular attention should be paid to determining the surface points of each spherical object and solving the minimum distance between the multiple points on various surfaces. Following the transformation of the approximate object to the position of the corroding component, the shortest distance between non-adjacent spherical objects can be obtained. After acquiring all distances of the surfaces' points, the minimum distance can be regarded as the basic judgment of collision detection. In theory, the distance between the surfaces' points maybe not be smaller than zero in all states. If the minimum distance between the surface points of two objects is equal to zero, the corresponding components are colliding. Otherwise, it is assumed that no collision occurs. This can be described as follows:

$$\begin{cases} collision(i,j) = 1, \qquad min(dis(i,j)) = 0; \\ collision(i,j) = 0, \qquad min(dis(i,j)) > 0; \\ dis(i,j) = \sqrt{(p_{xi} - p_{xj})^2 + (p_{yi} - p_{yj})^2 + (p_{zi} - p_{zj})^2}, \qquad (1 \le i \ne j \le 11); \end{cases} \tag{18}$$

where $collision()$ expresses the collision detection; $min()$ is the shortest distance of the surface points between part i and part j; $collision = 1$ means occurrence of collision between two objects; $collision = 0$ means that the objects do not collide; $dis()$ indicates the distance between each surface point of different spherical objects; $[p_{xi}, p_{yi}, p_{zi}]$ indicates the position of the discredited surface point of one approximation; and $[p_{xj}, p_{yj}, p_{zj}]$ indicates the position of the discredited surface point of another approximation.

When these components of the mobile manipulator are converted into multiple spherical objects, the shortest distance between the spheres' center and the sum of the corresponding radius of different spherical objects can also be used as the criterion of self-collision detection. When the shortest distance between the two approximations is greater than the sum of the radius of the sphere, the two components of the mobile manipulator robot will not collide. Otherwise, there will be a collision between the two components. The corresponding collision determination formula can be expressed as follows:

$$\begin{cases} collision(i,j) = 1, \qquad min(dis(S_i, B_j)) \le R_i + R_j; \\ collision(i,j) = 0, \qquad min(dis(S_i, B_j)) > R_i + R_j; \\ dis(S_i, B_j) = \| S_i - B_j \|, \qquad (1 \le i \ne j \le 11); \end{cases} \tag{19}$$

where $min()$ represents the shortest distance between two components, R_i represents the radius of one spherical object, R_j represents the radius of another spherical object that is not adjacent to another spherical object, $dis(S_i, B_j)$ indicates the distance between the center of different spherical approximations, S_i indicates the position of the center of the sphere objects for the approximation of one component, and B_j indicates the position of the center of the sphere objects for the approximation of another component.

However, the shortest distance of the surfaces' points of the two spherical objects may not equal zero in the actual self-collision detection. Moreover, not all components in the mobile manipulator robot will collide with other components. Therefore, we can improve the existing collision detection principles to improve the basic performance of self-collision detection.

4.3.1. Improve the Accuracy of Self-Collision Detection

There must be a certain distance between the surface points in one spherical element, which is caused by the different sizes of the spherical objects, the various surfaces of the components, and the incompleteness of the point cloud. Therefore, Formulas (18) and (19) cannot be effectively detected by self-collision even if a collision occurs. To solve this problem, self-collision detection is judged by the sum distance of the minimum separation for the acquisition of the surface points of the two spherical objects. When the sum distance is greater than the shortest distance of the surfaces' points, the two spherical objects will not collide; otherwise, a collision is deemed to occur. Moreover, we should consider the response distance of the mobile robot arm to address the collision phenomenon during the movement, which can be described as follows:

$$collision(i,j) = \begin{cases} 1, & min(dis(i,j)) \leq d_i + d_j + d_{sf}; \\ 0, & min(dis(i,j)) > d_i + d_j + d_{sf}; \end{cases} \tag{20}$$

where d_i and d_j represent the smallest segmentation distance between the points of the two objects, respectively; d_{sf} represents the response distance to address the collision.

We used the shortest distance to judge self-collision detection. In order to successfully implement the self-collision detection of the mobile manipulator robot, we need to determine the shortest distance between the components and the radius distance of the corresponding two spheres. When the shortest distance between the two approximations is greater than the sum of the radius of the sphere, the two components of the mobile manipulator robot will not collide. Otherwise, there will be a collision between the two components. The corresponding collision determination formula can be expressed as follows:

$$\begin{cases} collision(i,j) = 1, & min(dis(S_i, B_j)) \leq R_i + R_j + R_{sf}; \\ collision(i,j) = 0, & min(dis(S_i, B_j)) > R_i + R_j + R_{sf}; \\ dis(S_i, B_j) = \| S_i - B_j \|, & (1 \leq i \neq j \leq 11); \end{cases} \tag{21}$$

where $min()$ represents the shortest distance between two components, R_i represents the radius of the spherical object, R_j represents the radius of another spherical object that is not adjacent to the first one, and R_{sf} represents the safe distance to avoid the collision.

4.3.2. Reduce the Number of Self-Collision Detection

In the self-collision detection of the mobile manipulator robot, the traditional method requires collision detection for all non-adjacent components. However, in the actual movement, many components will not collide with other components of the mobile manipulator robot. In order to shorten the required time for completing self-collision detection, we only need to detect the components that may collide with others. According to the basic structure of the mobile manipulator robot, the components we need to perform collision detection on are mainly expressed as follows:

$$\begin{cases} Task1 = [(M, J_3), (M, J_4), (M, J_5), (M, J_6), (M, L_1), (M, L_2), (M, L_T)]; \\ Task2 = [(G, J_3), (G, J_4), (G, J_5), (G, J_6), (M, L_1), (M, L_2), (M, L_T)]; \\ Task3 = [(J_1, J_4), (J_1, J_5), (J_1, J_6), (J_1, L_2), (J_1, L_T)]; \\ Task4 = [(J_2, J_4), (J_2, J_5), (J_2, J_6), (J_2, L_2), (J_2, L_T)]; \\ Task5 = [(L_1, J_4), (L_1, J_5), (L_1, J_6), (L_1, L_T)]; \\ Task6 = [(J_3, J_5), (J_3, J_6), (J_3, L_T)]; \\ Task7 = [(L_2, J_5), (L_2, J_6), (L_2, L_T)]; \\ Task8 = [(J_4, J_6), (J_4, L_T)]; \end{cases} \tag{22}$$

where M represents the approximation object for the mobile platform, J_i represents the approximation object for the i-th motor in the manipulator subsystem, L_i represents the approximation object for the i-th link in the manipulator subsystem, and G represents the approximation object for the carry load.

The above self-collision detection method in Formulas (20) and (21) can accurately determine the apparent collision, and the detection of self-collision components in Formula (22) can effectively reduce the collision time of the system.

5. Simulation Results

To validate the performance of kinematic modeling and self-collision detection, which was optimized by adding some key points to some components that failed to model the physical structure, we conducted some typical simulations in the mobile manipulator robot with a UR5-like arm, including the kinematic model, approximation results, the time required to approximate the robot, and the effectiveness of self-collision detection. Since this research aims to develop a method suitable for the usage of a conventional industrial environment, we choose a typical personal computer to test the performance of kinematic modeling and self-collision detection. The computer is equipped with Intel® Core™ i5-7500, CPU @3.4 HZ, RAM @8 G, whose configurations are mid-sized computers in terms of computational speed in 2021. The flowing subsection summarizes the processes and the results of these typical simulations.

5.1. Equipment Parameters

This simulation experiment mainly takes the mobile manipulator robot illustrated in Figure 1 as the research object. To establish the kinematic modeling of the robot, we need to determine the following related parameters:

(1) Parameters of the mobile platform. The length, width, and height of the mobile platform are $L_c * B_c * H_c = 1104$ mm $*$ 518 mm $*$ 258 mm, respectively; the radius of the wheels is $R_c = 100$ mm; the midpoint of the driving axis in the body-fixed frame is $D_c = [340, 0, 119]$ mm. At the initial moment, the position of the midpoint of the driving axis in the world frame is $P_d = [0, 0, 100]$ mm; the rotation angle of the robot around the Z-axis of the midpoint-fixed frame is 0rad.

(2) Parameters of the manipulator subsystem. The manipulator subsystem is redesigned based on the typical UR manipulator in Figure 2, and the initial rotation angle of each joint is set $(0, 0, 0, 0, 0, 0)$. Therefore, the coordinate system of each joint is established based on the optimized method, and the specific parameters of the manipulator robot were shown in Table 1. In this table, d_i^c, a_i^c, α_i^c indicate the basic parameters obtained of the manipulator subsystem, which are determined by the classical kinematic modeling method, d_i^t, a_i^t, α_i^t are obtained by the traditional kinematic modeling method.

(3) Parameters of the other objects. The height of the carrying load object is $H_{Ld} = 200$ mm, the radius is $R_{Ld} = 200$ mm, and the distance between the fixed position of the carrying load and the gravity of the mobile platform in the x-axis direction is $L_{cx} = 140$ mm. The height of the supporting platform is $H_{sp} = 137$ mm, the length and width of the lower surface are 450 mm $*$ 358 mm, respectively; and the horizontal distance from the center of the lower support surface to the gravity of the mobile platform is 297 mm. Meanwhile, the length and width of the upper surface are 360 mm $*$ 190 mm, respectively, and the horizontal distance from the center of the upper support surface to the gravity of the mobile platform is 252 mm. In addition, the maximum length, width, and height of the tool at the end of the manipulator subsystem are 199 mm $*$ 89 mm $*$ 89 mm, respectively.

5.2. Comparison of Kinematic Modeling Results

In this section, we evaluate the performance of our proposed kinematic modeling in a mobile manipulator with the UR-like arm in Figure 1. Namely, we mainly focus on the difference between the mathematical model and the actual physical model of the mobile manipulator robot. As a mature industrial robot, the UR manipulator has been extensively researched and developed. At present, the industrial application of the UR arm mainly includes UR3, UR5, and UR10. The basic shapes of these robots are similar, but there are great differences in structural parameters and working performance. Therefore, we selected the manipulator similar to the UR5 arm as the object of this experiment for comparing the kinematic modeling methods. The basic structure of the manipulator subsystem is shown in Figure 1, and the coordinate frames established based on different kinematic methods are the same as Figure 2.

To verify the modeling performance in this study, we established the kinematic modeling of the UR robot based on the classical kinematic modeling method and the optimized kinematic modeling method mentioned in this study. The corresponding basic parameters of the UR5 arm with different kinematic modeling methods are shown in Table 1.

According to these basic parameters of the manipulator subsystem, we can build different kinematic modeling of the mobile manipulator robot with the UR5-like arm, which are shown in Figure 6a,b.

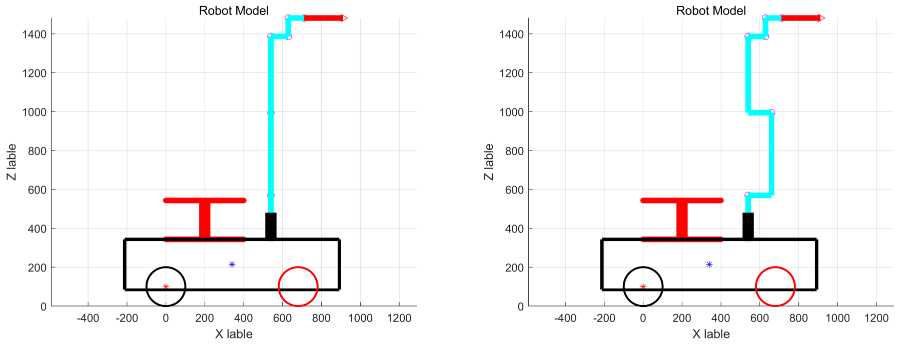
Table 1. The key parameters obtained for different kinematic modeling methods.

(a) Classical Kinematic Modeling Method

$Joint$	d_i^c (mm)	a_i^c (mm)	α_i^c (rad)	Off (rad)
J_1	89.2	0	$\pi/2$	0
J_2	0	-425	0	$-\pi/2$
J_3	0	-392.25	0	0
J_4	109.15	0	$\pi/2$	$-\pi/2$
J_5	94.65	0	$-\pi/2$	0
J_6	82.3	0	0	0

(b) Optimized Kinematic Modeling Method

$Start$	End	d_x^i (mm)	d_y^i (mm)	d_z^i (mm)	q_x^i (rad)	q_y^i (rad)	q_z^i (rad)
O_b	O_1	0	0	89.2	0	0	q_1
O_1	O_{c12}	0	-120	0	0	q_2	0
O_{c12}	O_2	0	0	425	0	0	0
O_2	O_{c23}	0	120	0	0	q_3	0
O_{c23}	O_3	0	0	392.25	0	0	0
O_3	O_4	0	-89.15	0	0	q_4	0
O_4	O_5	0	0	94.65	0	0	q_5
O_5	O_6	0	-82.3	0	0	q_6	0



(**a**) (**b**)

Figure 6. Kinematic modeling of the mobile manipulator robot with different methods (the '*' is the midpoint of these driving wheels, and the '*' is the gravity center of the mobile platform). (**a**) The classical kinematic modeling of the mobile manipulator robot; (**b**) The optimized kinematic modeling of the mobile manipulator robot.

In the classical kinematic modeling method, the initial angle of each joint in the manipulator robot is setting as $(90^{\circ},0^{\circ},0^{\circ},0^{\circ},180^{\circ},0^{\circ})$, the location of each component in the manipulator is the same as the result in Figure 1. The position and posture of all coordinate frames in the based-fixed frame were solved by these two methods. Due to the difference in the establishment of these coordinate frames, we solve the position of some joint-fixed coordinate frames of the manipulator in the world frame, which can be expressed as:

$$\begin{aligned}
{}^{0}T_{1} &= {}^{1}T_{Cla} = {}^{1}T_{Opt} = \begin{bmatrix} 540.0, & 0.0, & 569.2 \end{bmatrix}^{T}; \\
{}^{0}T_{2} &= {}^{2}T_{Cla} = {}^{O_{c23}}T_{Opt} = \begin{bmatrix} 540.0, & 0.0, & 994.2 \end{bmatrix}^{T}; \\
{}^{0}T_{3} &= {}^{3}T_{Cla} = {}^{3}T_{Opt} = \begin{bmatrix} 540.0, & 0.0, & 1386.4 \end{bmatrix}^{T}; \\
{}^{0}T_{4} &= {}^{4}T_{Cla} = {}^{4}T_{Opt} = \begin{bmatrix} 629.1, & 0.0, & 1386.4 \end{bmatrix}^{T}; \\
{}^{0}T_{5} &= {}^{5}T_{Cla} = {}^{5}T_{Opt} = \begin{bmatrix} 629.1, & 0.0, & 1481.1 \end{bmatrix}^{T}; \\
{}^{0}T_{6} &= {}^{6}T_{Cla} = {}^{6}T_{Opt} = \begin{bmatrix} 711.5, & 0.0, & 1488.1 \end{bmatrix}^{T};
\end{aligned} \tag{23}$$

where, T_{Cla} and T_{Opt}, respectively, represent the solution results of the classical method and the optimized method, the classic method is the most conventional kinematic modeling method for UR5 manipulator, and the optimized method refers to the kinematic modeling method proposed in this study.

By comparing the kinematic modeling results and the key parameters of the mobile manipulator robot, we can obtain the following:

(1) Comparing the results of these coordinate frames in Figure 6a,b, the optimized kinematic modeling method can establish a suitable coordinate frame at any location according to actual requirement instead of the designated location based on the classical method. Furthermore, the relationship between adjacent coordinate frames is more accessible to solve than the classical method.

(2) Comparing Figure 6a,b, the classic method cannot describe the position of the driving joint and the structure of the connecting links, but the optimized kinematic modeling method in this paper can describe the position of each joint and the structure of the connecting link.

(3) Comparing Formula (23) and the modeling results in Figure 2a,b, the positions of each driving joint in the mobile manipulator are the same. Therefore, the kinematic modeling method mentioned in this article can accurately determine the position and posture of each component.

Therefore, the optimized kinematic modeling method can improve the similarity between the mathematical and physical models. At the same time, it reduces the time required to establish these coordinate frames and the complexity of solving key parameters.

5.3. Comparison of the Performance of Self-Collision Detection

To verify the superior performance of the self-collision detection technology proposed in this article, we designed a variety of different test experiments based on these improvements of self-collision detection technology in this study. The main performance parameters include the detection accuracy and the required time for solving self-collision detection. For comparing the detection accuracy, the best approximating method is regarded as the approximate object which is the most suitable for the actual structure of mobile manipulator robot by observing different approximate results. Furthermore, we choose the required time for preprocessing process of self-collision detection and that for judging self-collision detection as another performance to prove the effectiveness of the improved self-collision detection. The specific experiments are described below.

5.3.1. Comparison of the Approximation Results

This simulation compares the efficiency of the approximation used for simplifying the robot before self-collision detection processes. If a single spherical object is used for the approximation of each component of the mobile manipulator robot and we use the methods of article [36] and this article to approximate each component of the mobile manipulator system, the results are as shown in Figure 7.

To compare the accuracy between the approximation and the actual structure of the mobile manipulator during the self-collision detection process, we performed a quantitative calculation by comparing the volume of the approximation and the volume enveloped by the actual part surface. Assuming: the volume enveloped by an actual component surface of the mobile manipulator robot is V_1, the volume of the corresponding approximation object is V_2, and the approximation degree of the corresponding component using the corresponding approximation object can be expressed as $n = \frac{V_1}{V_2} * 100\%$. When the value of the approximation degree is larger, the approximation object of the corresponding component is better, and the value range of the approximation degree is between $[0, 1]$. Therefore, we use the approximation degree to quantitatively compare the performance of approximating each component of the mobile manipulator robot based on different methods, which can be summarized in Table 2.

Table 2. Approximation degree of each component.

Name	**Physical Model**	**Typical Method**		**Optimized Method**	
Number	**Original Volume** V_1 **(mm^3)**	**Typical Volume** V_2^a **(mm^3)**	**Degree** n_1	**Optimized Volume** V_2^b **(mm^3)**	**Degree** n_2
1–Platform	1.6993×10^8	3.8956×10^8	43.62%	3.8956×10^8	43.62%
2–Load	4.9637×10^6	4.7080×10^7	10.54%	4.7080×10^7	10.54%
3–Motor1	2.0390×10^6	4.1888×10^6	48.68%	4.1888×10^6	48.68%
4–Motor2	2.0390×10^6	4.1888×10^6	48.68%	4.1888×10^6	48.68%
5–Link1	1.7107×10^6	1.2566×10^7	**13.61%**	3.6191×10^6	**47.27%**
6–Motor3	1.1786×10^6	4.1888×10^6	**28.14%**	2.1447×10^6	**54.95%**
7–Link2	1.3585×10^6	8.5786×10^6	**15.84%**	4.0532×10^6	**33.52%**
8–Motor4	7.0638×10^5	1.4368×10^6	49.17%	1.4368×10^6	49.17%
9–Motor5	7.0638×10^5	1.4368×10^6	49.17%	1.4368×10^6	49.17%
10–Motor6	7.0638×10^5	1.4368×10^6	49.17%	1.4368×10^6	49.17%
11–Tool	9.0120×10^5	4.3103×10^6	**2.09%**	1.5268×10^6	**5.90%**

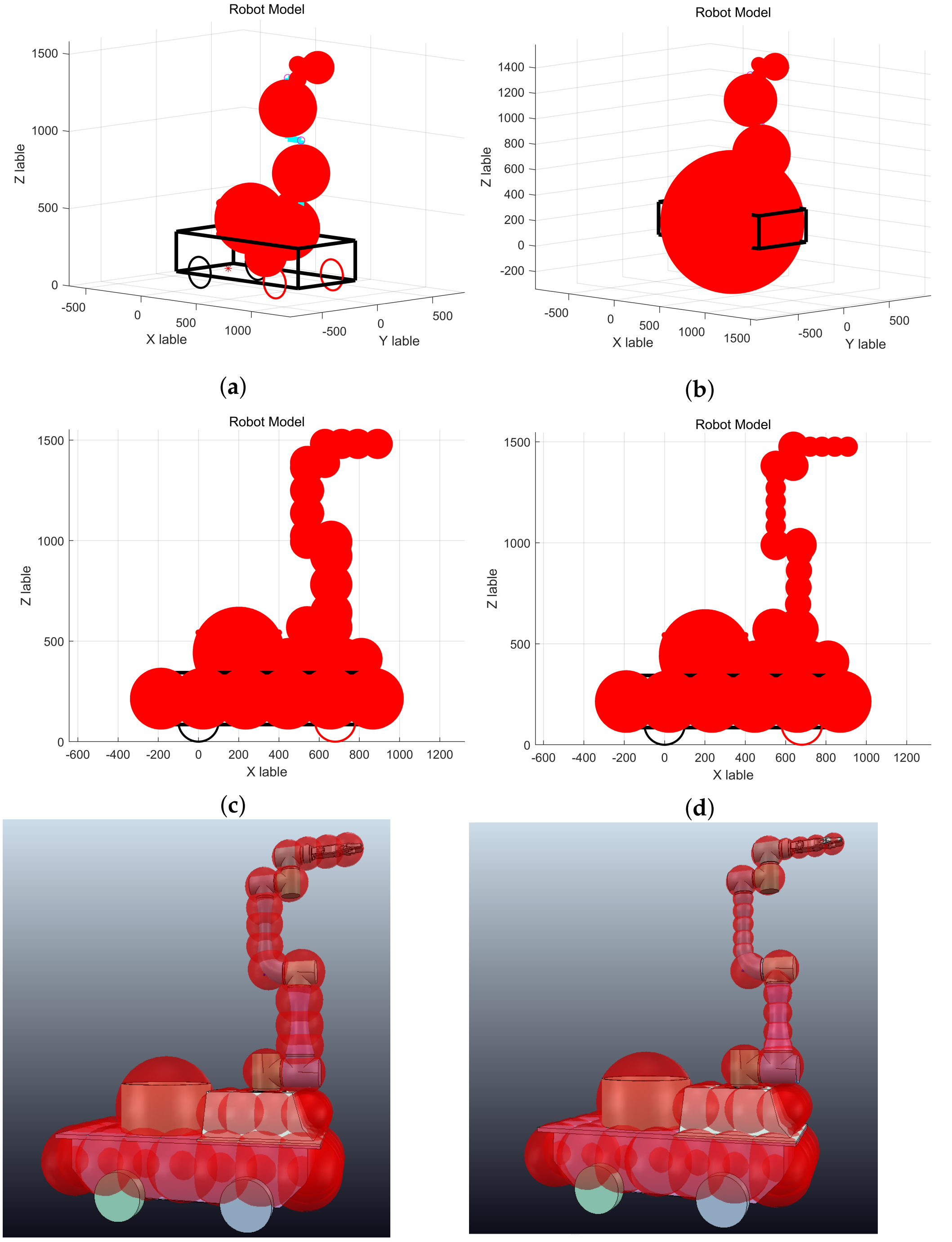
By comparing different approximation results of the mobile manipulator robot, we can obtain the following:

(1) Comparing Figure 7a,b, the second link, the third link, and the mobile platform are described by a large or small spherical object, and the two components will inevitably collide with each other at the initial time. This is caused by the significant difference in length and width of the manipulator and the mobile platform.

(2) Comparing Figure 7c–f, the basic structure of the manipulator subsystem can be accurately described by adding key point technology, which proves the correctness of the optimized kinematic modeling method and the effectiveness of the approximating process based on the physical structure model.

(3) Comparing Figure 7c–f, the connecting links and mobile platform of the robot can be described by multiple spherical objects. Moreover, the misjudgment of self-collision detection can be effectively eliminated at the initial moment. Therefore, the multiple spherical objects can improve the performance of self-collision detection.

(4) Comparing Table 2 and Figure 7e,f, the approximation degrees of these links and the end-effector obtained by the optimized method are more excellent than these degrees obtained by the typical method. Furthermore, the appearance of connecting links in Figure 7f was more accurately described by the multiple spherical objects than by the same spherical objects in Figure 7e. Therefore, multiple spherical objects can be effectively used to approximate the complex components in the mobile manipulator.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 7. Approximating the mobile manipulator robot by a single spherical object. (**a**) The approximation described by the single small spherical object; (**b**) The approximation described by the single large spherical object; (**c**) The approximation described by the method proposed in article [36]; (**d**) The approximation described by the method proposed in this study; (**e**) The approximation result of the mobile manipulator robot using the method proposed in article [36]; (**f**) The approximation result of the mobile manipulator robot using the method proposed in this study.

5.3.2. Comparison of the Time Required for Approximating Robot

The time required for the approximation of the mobile manipulator is another key parameter that can be used to improve the performance of self-collision detection. Therefore, to compare the real-time performance, we calculated the required time for the approximation of all components of the mobile manipulator at different moments in Figure 7e,f. Assumed: the initial position and posture of the mobile platform are $q_m^{int} = (0, 0, 0^{\circ})$, and the initial angle of each joint of the manipulator robot is $q_a^{int} = (99^{\circ}, 9^{\circ}, -9^{\circ}, 0^{\circ}, 81^{\circ}, 0^{\circ})$, the final position of the mobile platform is the same as the initial position, $q_m^{end} = (0, 0, 0^{\circ})^T$, but the end angle of each joint of the manipulator robot is $q_a^{end} = (180^{\circ}, 90^{\circ}, -90^{\circ}, 0^{\circ}, 90^{\circ}, 0^{\circ})$. The total time required for the task is $t = 10$ s. Then, we compared the performance of approximating processes based on the spatial transformation method and the point method used in the approximating process. The time used in solving the center position of all spherical objects is summarized in Table 3. In this table, the approximation object 1 and the approximation object 2 obtained by the VREP software are shown in Figure 7e,f, respectively; T_a^i and T_b^i indicate the required time for approximating the result i based on the spatial transformation method in article [36] and the method proposed in this study, respectively; T_c^i and T_d^i, respectively, indicate the required time for solving the spherical objects′ center in the result i based on the spatial transformation method and method proposed in this study, respectively.

Table 3. The time required for different approximating methods.

Name	**Approximation Object 1**				**Approximation Object 2**			
Time	T_a^1 **(s)**	T_b^1 **(s)**	T_c^1 **(s)**	T_d^1 **(s)**	T_a^2 (s)	T_b^2 **(s)**	T_c^2 **(s)**	T_d^2 **(s)**
$T1$	0.6820	0.4494	0.000714	0.000721	0.6216	0.2392	0.000783	0.000642
$T2$	0.6319	0.2317	0.000607	0.000436	0.6180	0.2321	0.000755	0.000404
$T3$	0.6220	0.2305	0.000675	0.000378	0.6191	0.2346	0.000604	0.000398
$T4$	0.6248	0.2365	0.000611	0.000426	0.6247	0.2323	0.000614	0.000425
$T5$	0.6258	0.2322	0.000596	0.000381	0.6267	0.2321	0.000723	0.000401
$T6$	0.6266	0.2306	0.000586	0.000438	0.6190	0.2356	0.000613	0.000412
$T7$	0.6298	0.2291	0.000607	0.000404	0.6200	0.2329	0.000610	0.000551
$T8$	0.6177	0.2368	0.000913	0.000397	0.6211	0.2303	0.000614	0.000401
$T9$	0.6297	0.2353	0.000597	0.000382	0.6238	0.2424	0.000616	0.000394
$T10$	0.6202	0.2364	0.000612	0.000378	0.6271	0.2345	0.000607	0.000411
AVE	0.6311	0.2548	0.000652	0.000434	0.6221	0.2346	0.000654	0.000444

By comparing the spatial transformation method with the point method proposed in this article, the following can be observed:

(1) Comparing T_a^1, T_b^1, T_a^2 and T_b^2 in Table 3, the required time based on the critical point method is shorter than the time based on the transformation method in the same position of each component. For example, through the average sampling of 10 points, the key point modeling method is about 47% of the time used based on the spatial transformation method for different approximation robots. Therefore, the key point method can effectively improve the real-time performance of the approximating process and reduce the overall time for self-collision detection.

(2) Comparing T_c^1, T_d^1, T_c^2 and T_d^2 in Table 3, the required time to solve the centers of all spherical objects based on the key point method takes less time-consuming than the required time based on the spatial transformation. For example, the required time is shorter by 61% of the time based on the spatial transformation when comparing the average time required for solving the centers of all spherical objects at different moments.

(3) Comparing T_a^1, T_c^1, T_b^2, and T_d^2 in Table 3, the improved self-collision detection proposed in this study can effectively reduce the time required for approximating the mobile manipulator robot.

5.3.3. Comparison of the Judgment of Self-Collision Detection

We compared the effectiveness of self-collision detection for the mobile manipulator robot after completing the approximation of each component. In the movement, most collisions may occur at other driving joint angles besides the obvious collision angles.

Therefore, we used the joint angles generated by solving the inverse kinematic to study the self-collision detection of the mobile manipulator. Considering the self-collisions phenomenon, the driving joints of the mobile manipulator cannot reach the solved joint angle in the actual movement. Therefore, the kinematic modeling method and self-collision detection technology mentioned in this paper are added to the differential evolution method to optimize the inverse kinematics solution of the mobile manipulator. We can directly choose the traditional differential evolution method proposed in [43] to solve the inverse kinematics problem. Then, the solution result is directly verified by self-collision detection. When it cannot satisfy the collision-free constraint, the inverse kinematics problem is resolved by the differential evolution method and judges the self-collision detection again. When no self-collision occurs, the current solution is saved as the actual control angle. Based on the optimized differential evolution method, the robot can smoothly reach the designated position.

To verify the performance of the improved self-collision detection, we compared the method mentioned in this study with the traditional differential evolution method in [43]. Assuming that the corresponding initial joint angle of the mobile manipulator is set to $(0.0, 0.5, \pi/4, 0, \pi/2, -\pi/4, \pi/4, 0.0, 0.0, 0.0)$, the first desired position is set to $(800, 500, 1000)$, and the second desired position is set to $(540, -131, 974)$. Then, we obtained the collision results by the improved method and the typical self-collision methods. Furthermore, we compared the success rates and the required time for solving the self-collision detection with different methods, which is shown in Table 4. In this table, SN_i represents the number of inverse kinematics solutions, CN_i represents the number of collisions obtained by the self-collision detection method, SR_i represents the success rate of inverse kinematics results without the self-collision, and AT_i represents the time required for self-collision detection in solving the inverse kinematics of the mobile manipulator.

Table 4. Comparison performance of self-collision detection in practical application.

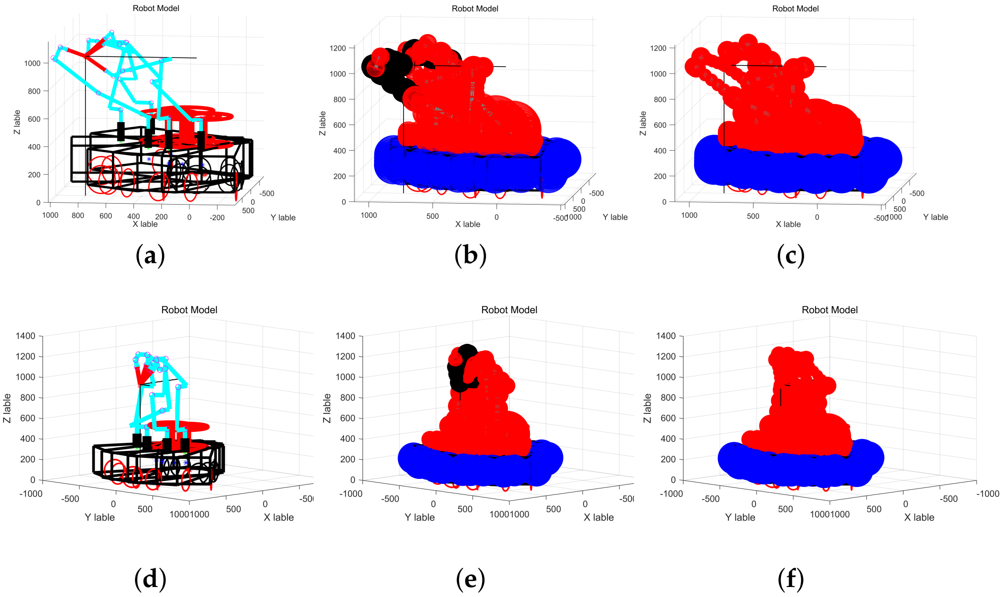
Method	**Typical Self-Collision Detection**				**Improved Self-Collision Detection**			
Name	SN_1	CN_1	SR_1	AT_1 **(s)**	SN_2	CN_2	SR_2	AT_2 **(s)**
Task1	100	73	27.00%	1.9053	100	63	37.00%	1.6700
Task2	100	86	14.00%	1.9353	100	65	35.00%	1.7081

Some typical solution results based on the differential evolution method are shown in Table 5, and draw upon some of the solution results, as shown in Figure 8. In Table 5, Q_{si} represents the solution result without collision based on both methods, Q_{fi} indicates the solution with self-collision based on the typical method, and ME is the maximum error between the result position and the actual position. In order to more intuitively identify the location of the collision, the approximate object of the manipulator subsystem uses a red spherical object. However, the approximate object of the mobile platform uses the blue spherical object. Furthermore, we use the approximation with black color to describe the corresponding components when the collision occurs. By comparing the solution results in the Tables 4 and 5, the following can be observed:

(1) Comparing Figure 8a–f and Table 5, the inverse kinematics of the mobile manipulator have different results for the designated reachable position. In theory, these solutions have the same validity for inverse kinematics, but some driving joints cannot reach the solved angle due to self-collision constraints. Therefore, the optimized differential evolution algorithm can reach the designated position without considering other collision avoidance technology.

(2) Comparing Figure 8c,f, The optimized differential evolution method can guarantee collision-free results. Due to the high-precision kinematic modeling and effective self-collision detection, the solutions which included the self-collision phenomenon can be entirely discarded, and the actual results fully meet the actual requirement.

(3) Comparing the average time (AT_1 and AT_2) and the success rate of self-collision (SR_1, SR_2) in Table 4, the improved self-collision detection can reduce the required time for solving collision detection than the typical technology. Furthermore, the performance of improved self-collision detection can be effectively improved by increasing the modeling accuracy of the mobile manipulator robot.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 8. Inverse kinematics analysis of the mobile manipulator robot. (**a**) Solution results for desired position 1; (**b**) Collision detection results based on the typical method for position 1; (**c**) Collision detection results based on the improved method for position 1; (**d**) Solution results for desired position 2; (**e**) Collision detection results based on the typical method for position 2; (**f**) Collision detection results based on the improved method for position 2.

Table 5. Solutions of inverse kinematics.

Name	d_x (mm)	d_y (mm)	q_z (rad)	q_1 (rad)	q_2 (rad)	q_3 (rad)	q_4 (rad)	q_5 (rad)	q_6 (rad)	ME (mm)
Q_{s1}	−0.0008	0.8659	1.6368	1.0156	0.9326	−0.0388	0.4905	2.8275	0.6857	0.0359
Q_{s2}	−0.7738	−0.0247	−0.6463	2.0489	−1.0538	0.3505	0.0355	−2.0784	1.1763	0.0509
Q_{s3}	−1.2539	−0.0196	1.1871	1.4097	−0.0675	1.9153	0.1931	−2.4686	1.1753	0.0366
Q_{f1}	11.6327	33.8740	0.8827	2.5442	1.1058	−0.2049	−1.3091	−0.9356	−1.4209	0.0129
Q_{f2}	60.2222	24.4381	0.1224	0.7637	0.3481	−1.6334	−1.7833	0.8309	0.2869	0.0240
Q_{s1}	−0.1388	0.0208	0.3756	0.4912	0.8297	−1.1704	−1.7821	−2.1422	−0.8794	0.0148
Q_{s2}	−2.2584	1.4400	−0.4380	2.1788	−1.1323	1.2635	0.7801	1.4986	0.3924	0.0403
Q_{s3}	−13.9452	−37.8080	0.7383	0.8655	−0.0554	1.0174	0.3473	1.7498	−0.0186	0.0118
Q_{f1}	−2.2584	1.4400	−0.4380	2.1788	−1.1323	1.2635	0.7801	1.4986	0.3924	0.0403
Q_{f2}	1.2887	1.8100	−0.0748	0.5451	0.5286	−1.0827	−1.4539	−1.5473	−1.1939	0.0058

5.4. Experiments of Self-Collision Detection

In this section, we design experiments to prove the application of the improved self-collision detection technology in the actual movement. Due to the different position and posture required for various tasks, the collision will occur among different components of the mobile manipulator robot, including the collision between the components of the manipulator subsystem, the collision between the mobile platform and the manipulator subsystem, and the collision between the carrying load and the manipulator subsystem. Therefore, many simulations were designed to test the performance of the self-collision detection technologies based on the improved method proposed in this study and based on the typical method in article [36]. Assuming the manipulator subsystem stops moving when a collision occurs, these self-collision detection technologies were used to judge the collision status, which can also identify the joint angle of the manipulator subsystem when a collision occurs. These simulations are illustrated below.

Situation 1: To study the collision between the components of the manipulator subsystem, we set the initial angle of the manipulator subsystem as $(90^{\circ}, 0^{\circ}, 18^{\circ}, 18^{\circ}, 0^{\circ}, 0^{\circ})$, and the final desired joint angle is $(90^{\circ}, 0, 180^{\circ}, 180^{\circ}, 0^{\circ}, 0^{\circ})$, the motion time of each joint is $t_1 = 10$ s, and the overall joints use uniform motion. To obtain the actual movement and determine the collision results, we first carry out a simulation experiment in the VREP-EDU software, and solve the shortest distance of two components which will occur collision during the movement, which is shown in Figure 9a,b. Then, we use the typical method and the improved method proposed in this study to determine the detection of a collision between different components of the manipulator. The specific results are shown in Figure 9(c1,c2). To accurately recognize the collision, we select the driving joint angle at the time of the collision and the joint angle at the previous time to control the mobile manipulator robot, and the overall time is $t_2 = 20$ s.The specific results are shown in Figure 9(c3,c4).

Situation 2: To study the collision between the carrying load and the manipulator subsystem, we set the initial angle of the manipulator as $(90^{\circ}, 90^{\circ}, 0^{\circ}, -90^{\circ}, 0^{\circ}, 0^{\circ})$, the final desired joint angle is $(0^{\circ}, 90^{\circ}, 0^{\circ}, -90^{\circ}, 0^{\circ}, 0^{\circ})$, and the motion time of each joint is $t_1 = 10$ s, and the overall joints use uniform motion. We use the typical method and the improved method proposed in this study to determine the detection of a collision between the carrying load and the manipulator subsystem. The specific results are shown in Figure 10(c1,c2). To accurately recognize the collision, we also select the driving joint angle at the time of the collision and the joint angle at the previous time to control the mobile manipulator robot, and the overall time is $t_2 = 20$ s. The specific results are shown in Figure 10(c3,c4).

Situation 3: To study the collision between the mobile platform and the manipulator subsystem, we set the initial angle of the manipulator as $(90^{\circ}, 0^{\circ}, 0^{\circ}, 0^{\circ}, 0^{\circ}, 0^{\circ})$, the expected final angle is $(90^{\circ}, 180^{\circ}, 0^{\circ}, 0^{\circ}, 0^{\circ}, 0^{\circ})$, the motion time of each joint is $t_1 = 10$ s, and the overall joints move at a uniform speed. We select the typical method and the improved method proposed in this study to determine the detection of a collision between the mobile platform and the manipulator subsystem. The specific results are shown in Figure 11(c1,c2). To accurately recognize the collision, we also select the driving joint angle at the time of the collision and the joint angle at the previous time to control the mobile manipulator robot, and the overall time is $t_2 = 20$ s.The specific results are shown in Figure 11(c3,c4).

According to the basic structure of the mobile manipulator robot, the manipulator subsystem will inevitably collide with the other components of the mobile manipulator robot. The possible collisions of the mobile manipulator robot are described in Figures 9–11.

Figure 9 shows the result of the movement of the mobile manipulator and the change of the shortest distance of the collision components in situation 1. According to the movement result of the mobile manipulator and the shortest distance change with time, the sixth motor and the first link were occurring collision between Figure 9(a9,a10) in VREP software. When two components collide, the shortest distance between the surfaces of the two parts is zero. From Figure 9b, we can get that the time of occurring self-collision is $t = 8.65$ s, and the corresponding joint angle is $(90^{\circ}, 0^{\circ}, 155.7^{\circ}, 155.7^{\circ}, 0^{\circ}, 0^{\circ})$. When using the basic method and the improved method to detect the self-collision of the mobile manipulator, the larger step size can be used to detect the collision of two components at $t_1^a = 8$ s and $t_1^b = 9$ s. When a smaller step size is used, the typical method will detect the collision at $t_2^a = 18$ s, and the joint angle at this moment is $(90^{\circ}, 0^{\circ}, 142.2^{\circ}, 142.2^{\circ}, 0^{\circ}, 0^{\circ})$; the improved method in this paper will detect the collision at $t_2^b = 4$ s, and the corresponding joint angle is $(90^{\circ}, 0^{\circ}, 147.6^{\circ}, 147.6^{\circ}, 0^{\circ}, 0^{\circ})$. The collision joint angle based on the improved method is closer to the actual joint angle when the collision occurs.

Figure 10 shows the result of the movement of the mobile manipulator and the change of the shortest distance of the collision components in situation 2. In the simulation of

VREP software, the manipulator subsystem reaches the designated position smoothly according to Figure 10a. However, the shortest distance of the manipulator subsystem and the carrying load is becoming to zero between $t = 7.7$ s, the corresponding joint angle is $(26^{\circ}, 90^{\circ}, 0^{\circ}, -90^{\circ}, 0^{\circ}, 0^{\circ})$, and the robot's motion results are shown in Figure 10a,b. When using the basic method and the improved method to detect the self-collision of the mobile manipulator, the larger step size can detect the collision of two components at $t_1^a = 6$ s and $t_1^b = 6$ s. When a smaller step size is used, the typical method will detect the collision at $t_2^a = 11$ s, and the joint angle at this moment is $(40.05^{\circ}, 90^{\circ}, 0^{\circ}, -90^{\circ}, 0^{\circ}, 0^{\circ})$; the method in this paper will detect the collision at $t_2^b = 15$ s, and the corresponding joint angle is $(38.25^{\circ}, 90^{\circ}, 0^{\circ}, -90^{\circ}, 0^{\circ}, 0^{\circ})$. In the second simulation with the VREP software, the manipulator subsystem can move close to the edge of the carrying load without collision. However, we use the spherical element to approximate the manipulator subsystem and the carrying load for practical applications, the typical method and the improved method will be judged as a collision occurs before the manipulator subsystem touches the edge of the carrying load. Therefore, the results of the self-collision detection based on the two methods are quite different from the actual collision joint angle.

Figure 11 shows the result of the movement of the mobile manipulator and the change of the shortest distance of the collision components in situation 3. According to the shortest distance change with time in VREP software, the mobile platform and the second link occur collision at $t = 6.2$ s, and the corresponding joint angle is $(90^{\circ}, 110.7^{\circ}, 0^{\circ}, 0^{\circ}, 0^{\circ}, 0^{\circ})$. When using the typical method and the improved method to detect the self-collision, the larger step size can be used to detect the collision of two components at $t_1^a = 5$ s and $t_1^b = 6$ s. When a smaller step size is used, the typical method will detect the collision at $t_2^a = 6$ s, and the joint angle at this moment is $(90^{\circ}, 77.4^{\circ}, 0^{\circ}, 0^{\circ}, 0^{\circ}, 0^{\circ})$; the improved method in this paper will detect the collision at $t_2^b = 10$ s, and the corresponding joint angle is $(90^{\circ}, 99^{\circ}, 0^{\circ}, 0^{\circ}, 0^{\circ}, 0^{\circ})$. Comparing the typical method, the collision joint angle based on the improved method in this article is closer to the actual joint angle which occur collision. This is due to the more detailed description of the connecting link.

By comparing these self-collision detection results with different methods, we can obtain the following:

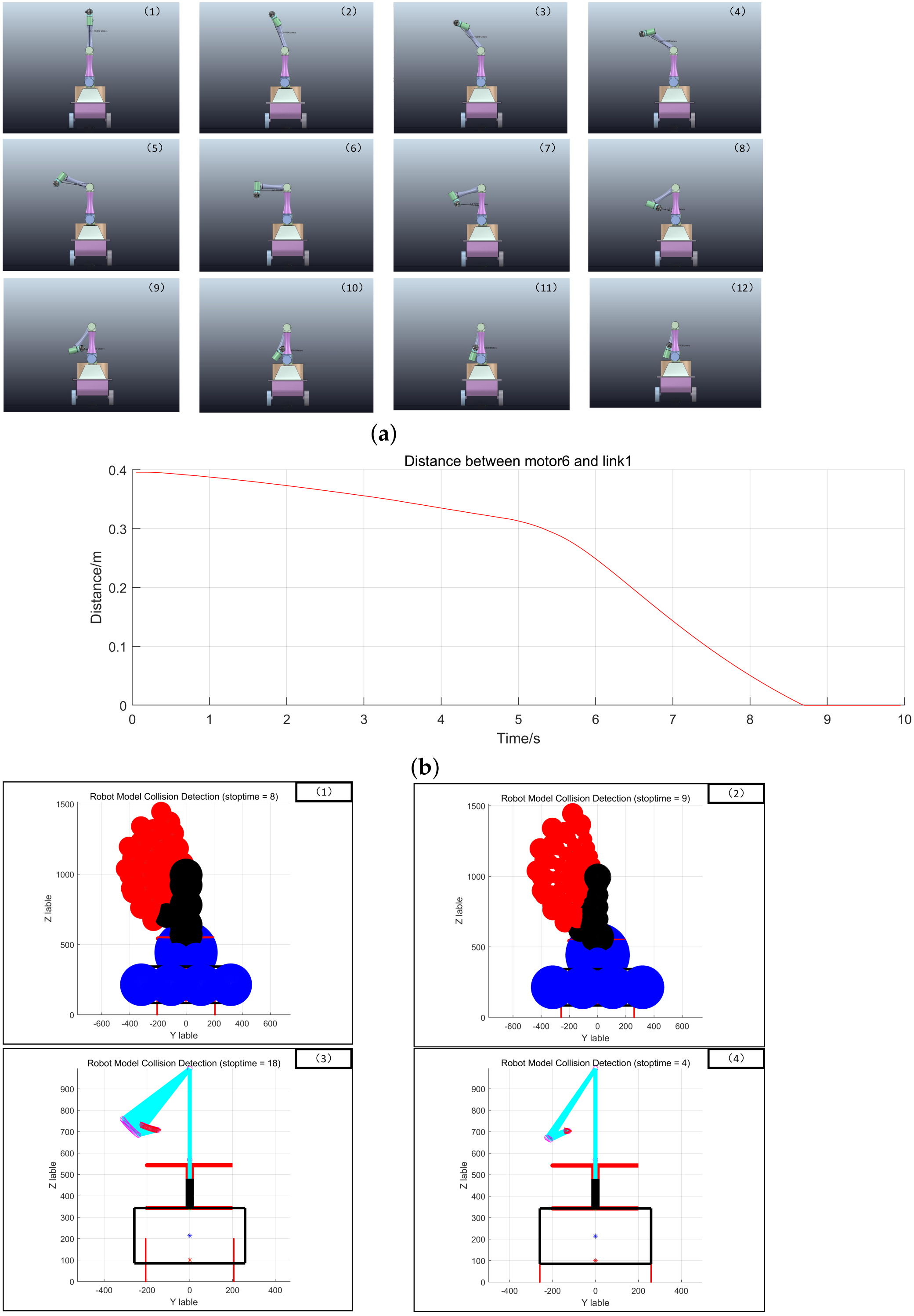
(1) Comparing Figures 9c, 10c and 11c, both the improved method and the typical method can effectively achieve self-collision detection, determine the collision location and identify the joint angle of the mobile manipulator robot.

(2) Comparing Figures 9a,c, 10a,c and 11a,c, both the improved methods and the typical method can obtain the detection of self-collision before occurring the actual collision between different components, which can prevent the damage caused by the collision phenomenon.

(3) Before reaching the inappropriate angle in Figures 9c, 10c and 11c, the manipulator subsystem will inevitably not collide with other components. If shorting the time interval of system movement, we can redefine the angular where the system collides due to the influence of the motion accuracy. However, this result only improves the angle accuracy before self-collision detection.

(4) Comparing Figure 10a,c, the mobile manipulator can move to the specified position according to the given joint angles in the Vrep software shown in Figure 10a, but these self-collision detection methods will detect the collision phenomenon in Figure 10c. Therefore, the above methods may incorrectly determine that the collision has occurred, but the actual parts will not collide. This case is caused by the collision of the near-sighted object boundary, and these methods cannot determine that the different parts are in contact without collision.

(5) Comparing Figures 9a,c, 10a,c and 11a,c, the improved self-collision detection technology can describe the approaching angle of collision more accurately than the typical method used in article [36]. This is due to the consideration of the deformation of the connecting link.

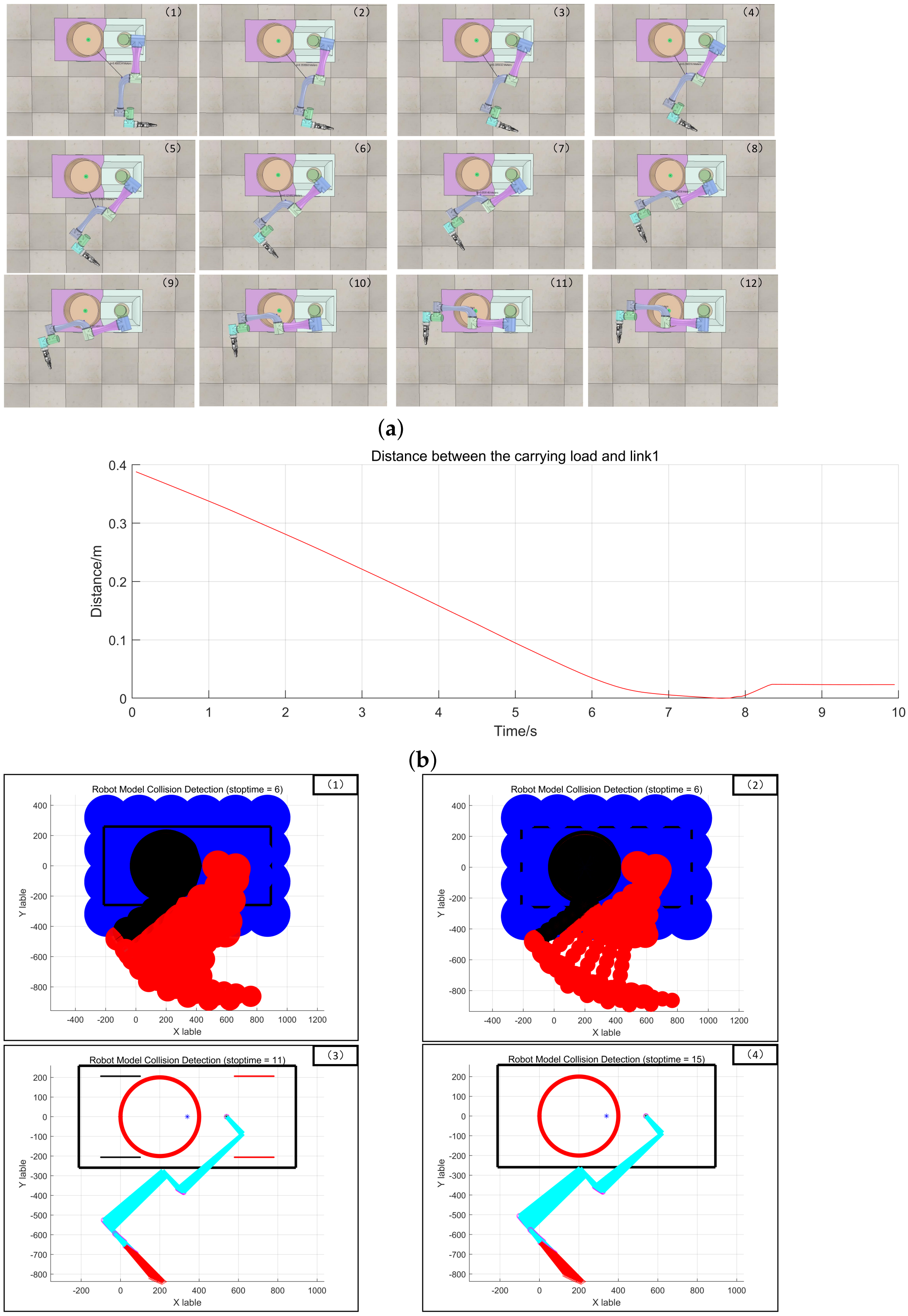


(**a**)

(**b**)

(**c**)

Figure 9. Detection of self-collision between different components of the manipulator subsystem. (**a**) The movement of the mobile manipulator in situation 1 based on the VREP software; (**a1**–**a9**) The movement of the mobile manipulator before reaching collision location in situation 1 based on the VREP software; (**a10**–**a12**) The movement of the mobile manipulator after reaching collision location in situation 1 based on the VREP software; (**b**) Distance changes between two components of the manipulator subsystem; (**c1**) Detection of self-collision using the typical method; (**c2**) Detection of self-collision using the improved method; (**c3**) Simplify the detection of self-collision using the typical method; (**c4**) Simplify the detection of self-collision using the improved method.

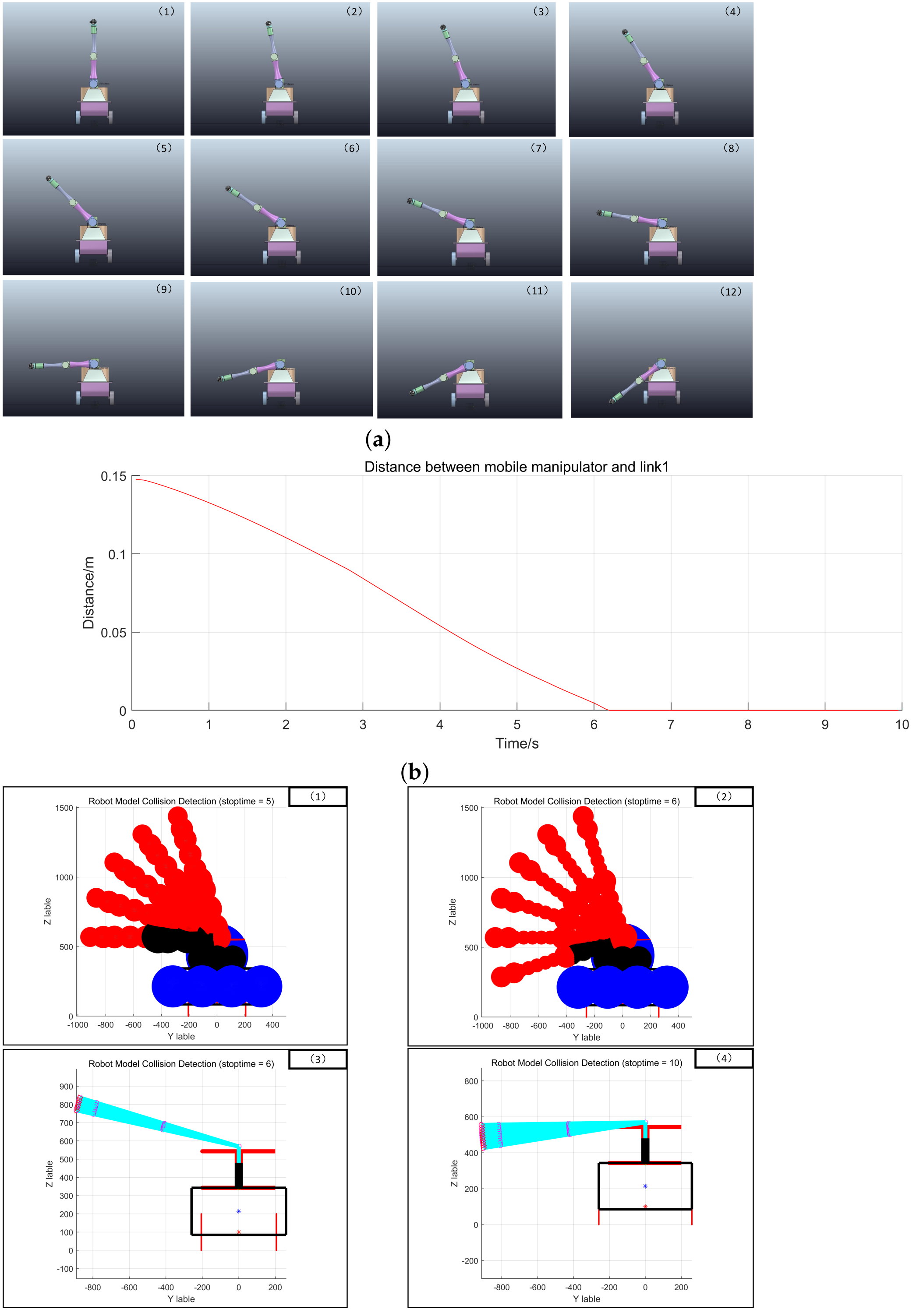


(a)

(b)

(c)

Figure 10. Detection of self-collision between different components of the manipulator subsystem. (**a**) The movement of the mobile manipulator in situation 2 based on the VREP software; (**a1**–**a9**) The movement of the mobile manipulator before reaching collision location in situation 1 based on the VREP software; (**a10**–**a12**) The movement of the mobile manipulator after reaching collision location in situation 1 based on the VREP software; (**b**) Distance changes between the carrying load and the manipulator subsystem; (**c1**) Detection of self-collision using the typical method; (**c2**) Detection of self-collision using the improved method; (**c3**) Simplify the detection of self-collision using the typical method; (**c4**) Simplify the detection of self-collision using the improved method.



(**a**)

(**b**)

(**c**)

Figure 11. Detection of self-collision between the mobile platform and the manipulator subsystem. (**a**) The movement of the mobile manipulator in situation 3 based on the VREP software; (**a1**–**a9**) The movement of the mobile manipulator before reaching collision location in situation 1 based on the VREP software; (**a10**–**a12**) The movement of the mobile manipulator after reaching collision location in situation 1 based on the VREP software; (**b**) Distance changes between two components of the manipulator subsystem; (**c1**) Detection of self-collision using the typical method; (**c2**) Detection of self-collision using the improved method; (**c3**) Simplify the detection of self-collision using the typical method; (**c4**) Simplify the detection of self-collision using the improved method.

6. Conclusions

The significant contribution of this paper is the proposal of a new type of kinematic modeling method and efficient self-collision detection technology. In order to reduce the difference between the mathematical model and the actual physical model of the mobile manipulator robot, we optimized the kinematic modeling by adding some key points to compensate for the defects of the traditional modeling method. Meanwhile, by replacing the complex calculation of the traditional spatial transformation method, the key points were directly used to determine the position of each approximation. The result proves that the key point method reduces the required time for approximating the entire system before self-collision detection. Therefore, the modeling technology and self-collision detection methods in this study can effectively improve the accuracy of system modeling and enhance the real-time performance of self-collision detection.

However, this article does not describe a method for detecting collisions between the robot and the surrounding environment. In this case, obstacles may be subdivided into static state and motion state. Identifying a method of judging the collision state and effectively avoiding possible collisions will be the next step in future work.

Author Contributions: Conceptualization, L.Q. and Q.L.; methodology, L.Q.; software, L.X. and Q.L.; validation, L.Q., L.X., J.J. and Q.L.; formal analysis, L.Q.; investigation, L.X. and Q.L.; resources, L.Q.; data curation, L.Q., J.J. and M.L.; writing—original draft preparation, L.Q.; writing—review and editing, L.X. and Q.L.; visualization, L.Q., J.J. and M.L.; supervision, L.X. and Q.L.; project administration, L.X. and Q.L.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (2016YFC0803000, 2016YFC0803005).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Engemann, H.; Du, S.; Kallweit, S.; Cönen, P.; Dawar, H. OMNIVIL-An Autonomous Mobile Manipulator for Flexible Production. *Sensors* **2020**, *20*, 7249. [CrossRef]
2. Park, D.; Hoshi, Y.; Mahajan, H.P.; Kim, H.K.; Erickson, Z.; Rogers, W.A.; Kemp, C.C. Active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned. *Rob. Auton. Syst.* **2020**, *124*, 103344. [CrossRef]
3. Wu, R.; Zhou, C.; Chao, F.; Zhu, Z.; Lin, C.M.; Yang, L. A developmental learning approach of mobile manipulator via playing. *Front. Neurorobot.* **2017**, *11*, 53. [CrossRef] [PubMed]
4. Seo, I.S.; Han, S.I. Dual closed-loop sliding mode control for a decoupled three-link wheeled mobile manipulator. *ISA Trans.* **2018**, *80*, 322–335. [CrossRef] [PubMed]
5. Gong, Z.; Xie, F.; Liu, X.J.; Shentu, S. Obstacle-crossing Strategy and Formation Parameters Optimization of a Multi-tracked-mobile-robot System with a Parallel Manipulator. *Mech. Mach. Theory* **2020**, *152*, 103919. [CrossRef]
6. Jin, L.; Li, S.; Yu, J.; He, J. Robot manipulator control using neural networks: A survey. *Neurocomputing* **2018**, *85*, 23–34. [CrossRef]
7. Safeea, M.; Neto, P.; Bearee, R. On-line collision avoidance for collaborative robot manipulators by adjusting off-line generated paths: An industrial use case. *Rob. Auton. Syst.* **2019**, *119*, 278–288. [CrossRef]
8. Park, J.; Baek, H. Stereo vision based obstacle collision avoidance for a quadrotor using ellipsoidal bounding box and hierarchical clustering. *Aerosp. Sci. Technol.* **2020**, *103*, 105882. [CrossRef]
9. Bostelman, R.; Foufou, S.; Hong, T.; Shah, M. Model of Mobile Manipulator Performance Measurement using SysML. *J. Intell. Robot. Syst. Theory Appl.* **2018**, *92*, 65–83. [CrossRef] [PubMed]
10. Farelo, F.; Alqasemi, R.; Dubey, R. Optimized dual-trajectory tracking control of a 9-DoF WMRA system for ADL tasks. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 1786–1791. [CrossRef]
11. Du, B.; Zhao, J.; Song, C. Dexterity analysis for omni-directional wheeled mobile manipulator based on double quaternion. *Chin. J. Mech. Eng.* **2013**, *26*, 585–593. (In English) [CrossRef]
12. Mashali, M.; Alqasemi, R.; Dubey, R. Mobile manipulator dual-trajectory tracking using control variables introduced to end-effector task vector. *World Autom. Congr. Proc.* **2016**, *2016*, 1–6. [CrossRef]

13. Jin, L.; Li, S.; La, H.M.; Luo, X. Manipulability Optimization of Redundant Manipulators Using Dynamic Neural Networks. *IEEE Trans. Ind. Electron.* **2017**, *64*, 4710–4720. [CrossRef]
14. Chen, F.; Selvaggio, M.; Caldwell, D.G. Dexterous Grasping by Manipulability Selection for Mobile Manipulator with Visual Guidance. *IEEE Trans. Ind. Inform.* **2019**, 15, 1202–1210. [CrossRef]
15. Mishra, S.; Mohan, S.; Vishvakarma, S.K. Simplified motion control of a vehicle-manipulator for the coordinated mobile manipulation. *Def. Sci. J.* **2020**, *70*, 72–81. [CrossRef]
16. Zhong, G.; Kobayashi, Y.; Hoshino, Y.; Emaru, T. System modeling and tracking control of mobile manipulator subjected to dynamic interaction and uncertainty. *Nonlinear Dyn.* **2013**, *73*, 167–182. [CrossRef]
17. Li, W.; Xiong, R. Dynamical Obstacle Avoidance of Task- Constrained Mobile Manipulation Using Model Predictive Control. *IEEE Access* **2019**, *7*, 88301–88311. [CrossRef]
18. Wang, H.; Li, R.; Gao, Y.; Cao, C.; Ge, L. Comparative study on the redundancy of mobile single- and dual-arm robots. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 1–19. [CrossRef]
19. Tawfik, M.A. Motion Control of Non-Holonomic Wheeled Mobile Robot Based on Particle Swarm Optimization Method (PSO). *Assoc. Arab Univ. J. Eng. Sci.* **2019**, *26*, 55–66. [CrossRef]
20. My, C.A.; Makhanov, S.S.; Van, N.A.; Duc, V.M. Modeling and computation of real-time applied torques and non-holonomic constraint forces/moment, and optimal design of wheels for an autonomous security robot tracking a moving target. *Math. Comput. Simul.* **2020**, *170*, 300–315. [CrossRef]
21. Wang, Y.; Liang, X.; Gong, K.; Liao, Y. Kinematical research of free-floating space-robot system at position level based on screw theory. *Int. J. Aerosp. Eng.* **2019**, 2019, 6857106. [CrossRef]
22. Tao, B.; Zhao, X.; Yan, S.; Ding, H. Kinematic modeling and control of mobile robot for large-scale workpiece machining. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2020,** *10*, 1–10. [CrossRef]
23. Tang, C.P.; Bhatt, R.M.; Abou-Samah, M.; Krovi, V. Screw-theoretic analysis framework for cooperative payload transport by mobile manipulator collectives. *IEEE/ASME Trans. Mechatron.* **2006**, *11*, 169–178. [CrossRef]
24. Qiu, C.; Cao, Q. Modeling and analysis of the dynamics of an omni-directional mobile manipulators system. *J. Intell. Robot. Syst. Theory Appl.* **2008**, *52*, 101–120. [CrossRef]
25. Savino, H.J.; Pimenta, L.C.A.; Shah, J.A.; Adorno, B.V. Pose consensus based on dual quaternion algebra with application to decentralized formation control of mobile manipulators. *J. Franklin Inst.* **2020**, *357*, 142–178. [CrossRef]
26. Quiroz-Omaña, J.J.; Adorno, B.V. Whole-Body Kinematic Control of Nonholonomic Mobile Manipulators Using Linear Programming. *J. Intell. Robot. Syst. Theory Appl.* **2018**, *91*, 263–278. [CrossRef]
27. Silva, F.F.A.; Adorno, B.V. Whole-body Control of a Mobile Manipulator Using Feedback Linearization and Dual Quaternion Algebra. *J. Intell. Robot. Syst. Theory Appl.* **2018**, *91*, 249–262. [CrossRef]
28. Chang, J.W.; Wang, W.; Kim, M.S. Efficient collision detection using a dual OBB-sphere bounding volume hierarchy. *CAD Comput. Aided Des.* **2010**, *42*, 50–57. [CrossRef]
29. Xiaodong, Z.; Zixin, T.; Xin, L. Real-time Detection of Space Manipulator. *World Acad. Sci. Eng. Technol.* **2015**, *9*, 849–853.
30. Han, D.; Nie, H.; Chen, J.; Chen, M. Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection. *Robot. Comput. Integr. Manuf.* **2018**, *49*, 98–104. [CrossRef]
31. Ju, M.Y.; Liu, J.S.; Shiang, S.P.; Chien, Y.R.; Hwang, K.S.; Lee, W.C. Fast and accurate collision detection based on enclosed ellipsoid. *Robotica* **2001**, *19*, 381–394. [CrossRef]
32. Patel, R.V.; Shadpey, F.; Ranjbaran, F.; Angeles, J. A collision-avoidance scheme for redundant manipulators: Theory and experiments. *J. Robot. Syst.* **2005**, *22*, 737–757. [CrossRef]
33. Figueredo, L.F.C.; Adorno, B.V.; Ishihara, J.; Borges, G. Robust kinematic control of manipulator robots using dual quaternion representation. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 1949–1955. [CrossRef]
34. Henten, E.J.V.; Schenk, E.J.; van Willigenburg, L.G.; Meuleman, J.; Barreiro, P. Collision-free inverse kinematics of the redundant seven-link manipulator used in a cucumber picking robot. *Biosyst. Eng.* **2010**, *106*, 112–124. [CrossRef]
35. Duguleana, M.; Barbuceanu, F.G.; Teirelbar, A.; Mogan, G. Obstacle avoidance of redundant manipulators using neural networks based reinforcement learning. *Robot. Comput. Integr. Manuf.* **2012**, *28*, 132–146. [CrossRef]
36. Liao, J.; Huang, F.; Chen, Z.; Yao, B. Optimization-based motion planning of mobile manipulator with high degree of kinematic redundancy. *Int. J. Intell. Robot. Appl.* **2019**, *3*, 115–130. [CrossRef]
37. Lei, M.; Wang, T.; Yao, C.; Liu, H.; Wang, Z.; Deng, Y. Real-Time Kinematics-Based Self-Collision Avoidance Algorithm for Dual-Arm Robots. *Appl. Sci.* **2020**, *10*, 5893. [CrossRef]
38. Jang, K.; Kim, S.; Park, J. Reactive Self-Collision Avoidance for a Differentially Driven Mobile Manipulator. *Sensors* **2021**, *21*, 890. [CrossRef]
39. Spong, M.W. Seth Hutchinson, and Mathukumalli Vidyasagar. In *Robot Modeling and Control*, 2nd ed.; Wiley: New York, NY, USA, 2004; pp. 61–81.
40. Craig, J.J. *Introduction to Robotics: Mechanics and Control*, 3rd ed.; Pearson Education: Karnataka, India, 2009; pp. 62–100.
41. Pollák, M.; Kočiško, M.; Paulišin, D.; Baron, P. Measurement of unidirectional pose accuracy and repeatability of the collaborative robot UR5. *Adv. Mech. Eng.* **2020**, *12*, 1–21. [CrossRef]

42. Zhang, H.; Sheng, Q.; Sun, Y.; Sheng, X.; Xiong, Z.; Zhu, X. A novel coordinated motion planner based on capability map for autonomous mobile manipulator. *Robot. Auton. Syst.* **2020**, *129*, 103554. [CrossRef]
43. López-Franco, C.; Hernández-Barragán, J.; Alanis, A.Y.; Arana-Daniel, N.; López-Franco, M. Inverse kinematics of mobile manipulators based on differential evolution. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1–22. [CrossRef]