sensors

MDPI

Article

Path Planning for Obstacle Avoidance of Robot Arm Based on Improved Potential Field Method

Xinkai Xia [1,2], Tao Li [3], Shengbo Sang [1,4], Yongqiang Cheng [1,4], Huanzhou Ma [1,2], Qiang Zhang [1,4] and Kun Yang [1,4,*]

[1] Shanxi Key Laboratory of Micro Nano Sensor & Artificial Intelligence Perception, College of Information and Computer, Taiyuan University of Technology, Taiyuan 030024, China
[2] Shanxi Institute of 6D Artificial Intelligence Biomedical Science, Taiyuan 030031, China
[3] Medical Big Data Research Center, Department of Medical Innovation Research, Chinese PLA General Hospital, Beijing 100853, China
[4] Key Lab of Advanced Transducers and Intelligent Control System of the Ministry of Education, Taiyuan University of Technology, Taiyuan 030024, China
* Correspondence: yangkun01@tyut.edu.cn

Abstract: In medical and surgical scenarios, the trajectory planning of a collaborative robot arm is a difficult problem. The artificial potential field (APF) algorithm is a classic method for robot trajectory planning, which has the characteristics of good real-time performance and low computing consumption. There are many variants of the APF algorithm, among which the most widely used variants is the velocity potential field (VPF) algorithm. However, the traditional VPF algorithm has inherent defects and problems, such as easily falling into local minimum, being unable to reach the target, poor dynamic obstacle avoidance ability, and safety and efficiency problems. Therefore, this work presents the improved velocity potential field (IVPF) algorithm, which considers direction factors, obstacle velocity factor, and tangential velocity. When encountering dynamic obstacles, the IVPF algorithm can avoid obstacles better to ensure the safety of both the human and robot arm. The IVPF algorithm also does not easily fall into a local problem when encountering different obstacles. The experiments informed the RRT* algorithm, VPF algorithm, and IVPF algorithm for comparison. Compared with the informed RRT* and VPF algorithm, the result of experiments indicate that the performances of the IVPF algorithm have significant improvements when dealing with different obstacles. The main aim of this paper is to provide a safe and efficient path planning algorithm for the robot arm in the medical field. The proposed algorithm can ensure the safety of both the human and the robot arm when the medical and surgical robot arm is working, and enables the robot arm to cope with emergencies and perform tasks better. The application of the proposed algorithm could make the collaborative robots work in a flexible and safe condition, which could open up new opportunities for the future development of medical and surgical scenarios.

Keywords: surgical robot; medical robot; human–robot interaction; trajectory planning

check for updates

Citation: Xia, X.; Li, T.; Sang, S.; Cheng, Y.; Ma, H.; Zhang, Q.; Yang, K. Path Planning for Obstacle Avoidance of Robot Arm Based on Improved Potential Field Method. *Sensors* **2023**, *23*, 3754. https://doi.org/10.3390/s23073754

Academic Editor: Andrey V. Savkin

Received: 9 March 2023
Revised: 21 March 2023
Accepted: 25 March 2023
Published: 5 April 2023

1. Introduction

With the rapid development of the robot industry, the robot arm is no longer operated by an operator or a fixed program, but it rather behaves more like an human staff in the foreseeable future. In the medical field, surgical robots have begun to be used and are developed rapidly. The Da Vinci System is already in service and can operate on patients in the medical field [1]. Although, at present, robotic surgical arms are mainly operated by doctors to perform operations indirectly, autonomous robots have been developed. The autonomous robots can operate autonomously on patients without having to be operated by a doctor or operator [2]. This can be seen as a developing trend for human doctors and robot doctors to cooperate in surgery. For future surgical robots, the preoperative and intraoperative path planing of a surgical robot arm will become an important research and application content [3].

At present, a lot of the global planning algorithms of the robot arm focus on planning the optimal trajectory, such as A-star algorithm (A*) [4,5], probabilistic roadmap (PRM) [6–8], rapidly-exploring random tree (RRT) [9–12], and so on. The RRT series algorithms are one the most widely used methods. It samples the detection state of robot arm in the Cartesian space directly, which greatly simplifies the collision detection of the robot arm. However, the random sampling characteristic of RRT makes the planned paths not smooth enough. RRT* [13] is an asymptotic optimization algorithm that adds the reselection of the parent node on the basis of the RRT. The smoothness of the paths planned by RRT* and its modified algorithms are greatly improved, and can meets the demands of the practical application [14–16]. Whereas, RRT* has the disadvantages of high computational cost and long computation time. In view of the shortcomings of RRT*, Gammell et al. [17] proposed informed RRT*, which greatly reduces the computation time by optimizing the node sampling area. Therefore, informed RRT* has become one of the most widely used planning algorithms for robotic arms. However, these algorithms have obvious disadvantages in safety-critical applications such as surgical and industrial collaborative robotics, such as poor real-time performance and excessive computational costs [18].

In these human–robot collaboration scenarios, the robot arm needs to ensure the safety of the human, which means that the algorithm needs good real-time performance and dynamic obstacle avoidance ability. Compared with the global planning algorithm, the local planning algorithm is more suitable for these collaborative tasks that require continuous trajectory modification based on sensory feedback [19,20]. Khatib et al. [21] firstly proposed the APF algorithm in 1986, which has good real-time performance to solve the obstacle avoidance problem. The core idea of the APF algorithm is to construct virtual force fields in the Cartesian space; the attraction of the target point and the repulsion of obstacles to the robot arm are also introduced in the algorithm. APF is the local planning algorithm and calculates two virtual forces at each time step to drive the movement of the robot arm. Therefore, this algorithm can adjust the planned trajectory in real time according to the feedback of the sensor. To enhance the safety of the robotic operations, Ye et al. [22] combined the quadratic programming based control scheme equipped with the APF method to achieve joint-limit and obstacle avoidance. Christoff et al. [23] used the artificial velocity-potential field method to implement the motions and trajectory planning of two six degrees of freedom (6-DoF) surgery robot arms. Tian ey al. [24] proposed an overall configuration planning method, which is based on the improved APF method to tackle the problem of trajectory planning of medical continuum hyper-redundant manipulators. Aiming at the two aspects of reaching the surgical tool pose and performing the planned path in an active robotic surgery, Hao et al. [24] proposed the improved APF and the PDNN algorithm to increase surgical robots with high safety and accuracy.

As a branch method of APF, VPF was soon applied to the robot obstacle avoidance scene [25]. In robotic applications, VPF is used more often because the velocity fields are simpler, more efficient, and easier to calculate than force fields in most other scenarios. However, there are still some problems in the VPF algorithm, such as easily falling into local minimum, the target being unreachable, oscillations in the presence of obstacles, poor dynamic obstacle avoidance ability, and so on [26]. Therefore, many researchers have made improvements on the VPF algorithm. Volpe et al. [27] presented a potential function based on superquadrics. The potential function also prevented the creation of local minimum when it was added to spherically symmetric attractive wells. In order to optimize the attractive field function, Wang et al. [20] introduced boundaries into Cartesian space components. The trajectory obtained by this method can be smooth and safe. Zhao et al. [28] established dynamic virtual target points to enhance the predictive ability of the manipulator to the road ahead. Xu et al. [29] proposed a saturated function in the attractive velocity potential field algorithm, which slows down to the goal progressively. The algorithm could improve the safety of the robot arms. Park et al. [30] proposed a dynamic potential filed method that incorporated the position and velocity of the actuator, which provided a better obstacle avoidance ability than the traditional VPF algorithm. Zhang et al. [26] proposed an obstacle avoidance strategy for a dual-arm robot based on speed field with improved VPF. They introduced a new repulsion function and added virtual constraint points to eliminate the existing limitations and improve the VPF.

However, all these methods cannot handle neither dynamic or static obstacles problems well in surgery tasks. Therefore, these methods are not suitable for human–robot interaction scenarios with high real-time and safety requirements, as in this paper. In this work, an improved VPF method is proposed to solve the above mentioned problems, which can make the robot arm cope better with both dynamic and static obstacles. This work is structured as follows. Section 1 introduces the relevant information, advantages, and disadvantages of several trajectory generation methods. The related works of the APF and VPF algorithm are summarized. Section 2 describes the improvement strategies of the VPF algorithm in detail in this work. In Section 3, simulation and actual experiments are conducted to verify the proposed algorithm. In Section 4, the work of the whole article is summarized.

2. Improved Velocity Potential Field

2.1. Traditional Potential Field

The VPF algorithm is a type of virtual velocity method that builds the environment into potential fields of attraction and repulsion. The target is built as a attractive potential field and generates attractive velocity. The obstacle is built as a repulsive potential field and generates repulsive velocity.

The attractive potential function is defined as

$$U_{att}(q) = \frac{1}{2}\zeta{\rho_g}^2 \tag{1}$$

where q is the current joint positions of the robot arm, ζ is the attractive potential gain, and ρ_g is the relative Cartesian distance between the end effector of robot arm and the target.

The construction of the repulsive potential field is similar to that of the attractive potential field, which can be defined as

$$U_{rep}(q) = \begin{cases} \frac{1}{2}k(\frac{1}{\rho_b} - \frac{1}{\rho_0})^2, & \rho_b \leq \rho_0 \\ 0, & \rho_b > \rho_0 \end{cases} \tag{2}$$

where k is the repulsive potential gain, ρ_b is the minimum distance between the robot's body and the obstacle, and ρ_0 is the max range of the repulsive potential field. When the minimum distance between the robot's body and the obstacle is beyond ρ_0, then the magnitude of the potential field is 0, and the robot will not be affected by the obstacle.

The velocity is the gradient of the potential field. The functions of velocities can be expressed as

$$v_{att}(q) = -\nabla U_{att}(q) = \zeta\rho_g \tag{3}$$

$$v_{rep}(q) = -\nabla U_{rep}(q) = \begin{cases} k(\frac{1}{\rho_b} - \frac{1}{\rho_0})\frac{1}{\rho_b^2}, & \rho_b \leq \rho_0 \\ 0, & \rho_b > \rho_0 \end{cases} \tag{4}$$

where v_{att} is the attraction velocity caused by the target, which is acting on the end effector. v_{rep} is the repulsive velocity caused by the obstacle, which is acting on the robot links or the end effector.

The velocity in Cartesian space cannot recognized by the robot arm directly, and needs to be converted into joint space. The 6 × 1 velocity vector can be calculated as

$$V_{att}(q) = \left[\omega_{att}\ v_{att}\right]^T \tag{5}$$

$$V_{rep}(q) = \left[0\ 0\ 0\ v_{rep}\right]^T \tag{6}$$

where ω_{att} is the attractive angular velocity.

The traditional VPF algorithm does not calculate the part of the angular velocity of attraction, so ω_{att} is a 3 × 1 zero vector. According to Equations (5) and (6), the Cartesian velocity can be converted to the joint speed as follows:

$$\dot{q}_{att} = J^{-1}\, V_{att}(q) \tag{7}$$

$$\dot{q}_{rep} = J^{-1}\, V_{rep}(q) \tag{8}$$

where J is the Jacobian matrix of the robot. $\dot{q}_{att}$ and $\dot{q}_{rep}$ are the attractive joint speed and repulsive joint speed, respectively, which can drive the robot arm to move toward the target or move away from obstacles.

This work makes improvements based on the traditional VPF algorithm, and changes are inflicted on the attractive field, repulsive field, and singularity avoidance strategy.

2.2. Improved Attractive Field

The tradition VPF produces first-order linear attraction velocity, which is not flexible enough. When the effector is far away from the target, the velocity becomes high, which can make the robot arm move at a high joint speed. Therefore, the robot arm may not have enough time to decelerate, which will greatly increase the possibility of collisions with obstacles. This work adopts the segmented attractive potential field method to build an attractive potential field. The attractive potential field is defined as

$$U_{att}(q) = \begin{cases} \frac{1}{2}\zeta {\rho_g}^2, & \rho_g < \rho_{g0} \\ s\zeta\rho_g, & \rho_g \geq \rho_{g0} \end{cases} \tag{9}$$

where, ζ is the attraction potential field coefficient, ρ_g is the distance between the end effector and target, ρ_{g0} is the distance constant decided by the environment, and s is the constant coefficient.

The attraction linear velocity can be calculated by

$$v_{att}(q) = \left[v_x\ v_y\ v_z\right]^T = \begin{cases} \zeta\rho_g, & \rho_g < \rho_{g0} \\ \zeta s, & \rho_g \geq \rho_{g0} \end{cases} \tag{10}$$

When ρ_g exceeds ρ_{g0}, the attractive velocity is a constant value. It ensures that the attractive joint speed will not be too high. The traditional VPF algorithm does not consider the calculation of the angular velocity, but this work defines how to calculate the angular velocity. Firstly, the direction vector of the end effector A and the direction vector of the actuator pointing to the target B should be determined. The direction vector B is considered to be obtained by rotating the direction vector A around a space vector k by angle θ. The quaternion rotation matrix can be obtained from k and θ. Then, the Euler angle $[\ \alpha\ \beta\ \gamma\ ]^T$ can be calculated by the quaternion rotation matrix. The angular velocity is processed in the same way as the linear velocity, it can be calculated by

$$\omega_{att}(q) = \left[\omega_x\ \omega_y\ \omega_z\right]^T = \zeta_a\left[\alpha\ \beta\ \gamma\right]^T \tag{11}$$

where ζ_a is the coefficient of the attraction angular velocity.

2.3. Improved Repulsive Field

The construction of the improved repulsive potential field is more complicated than that of the attractive potential field. As the Figure 1 shows, O_j and P_j are the two closest points on the robot arm and the obstacle. T is the Target point. The vector O_jT is the direction vector from O_j to the target point. The vector O_jP_j is the direction vector from O_j to P_j. V_{obs} is obstacle's velocity. θ_{rob_tar} is the angle between the vector O_jP_j and the vector O_jT, and θ_{rob_Vobs} is the angle between P_jO_j and the direction of V_{obs}.

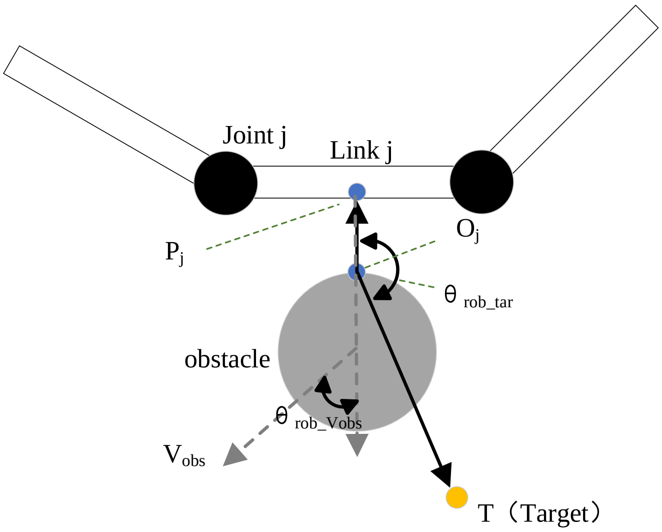


Figure 1. Obstacle avoidance velocity generation in workspace.

The function of the improved repulsive field is defined in Equation (12).

$$U_{rep}(q) = \begin{cases} \frac{1}{2}k(\frac{1}{\rho_b} - \frac{1}{\rho_0})^2 e^{m(\sigma_s+\sigma_d)}, & \rho_b < \rho_0 \\ 0, & \rho_b \geq \rho_0 \end{cases} \tag{12}$$

$$\sigma_s = \frac{a\theta_{rob\_tar} + (rV_{obs})^n}{(1 + rV_{obs})^n} \tag{13}$$

$$\sigma_d = bV_{obs}\theta_{rob\_V_{obs}} \tag{14}$$

where σ_s and σ_d are the static and dynamic obstacle velocity-direction factor, respectively. k is the repulsive potential gain, ρ_b is the minimum distance between the robot arm and the obstacle, and ρ_0 is the maximum range of repulsive potential filed. a and b are constants. m is the order. r is the reciprocal of V_{ref}, which is the reference velocity of obstacles and can be regarded as a constant. V_{ref} is set to be very small and used to determine whether the obstacle is a dynamic obstacle or a static obstacle. When V_{obs} is much greater than V_{ref}, the obstacle can be considered a dynamic obstacle. When the V_{obs} is much less than V_{ref} and close to 0, the obstacle will be considered as a static obstacle.

The traditional potential field method does not considered the positional relationship among obstacles, robotic arms, and target points. When the target point and obstacles are very close, the attractive velocity and the repulsive velocity may balance, and the robot arm will fall into local minimum problem and cannot reach the target. θ_{rob_tar} is introduced to improve this situation. The effect of θ_{rob_tar} is to deform the repulsive potential field. When the robot arm is on the side that is closed to the target point around the obstacle, θ_{rob_tar} is smaller. When other parameters remain unchanged, the repulsive potential field U_{rep} will be small. The robot arm can reach the target point more easily. When the robot arm is on the side that is far away from the target point around the obstacle, θ_{rob_tar} increases. The repulsive potential field U_{rep} will be larger, which makes the robot arm stay away from obstacles.

The traditional potential field method does not take measures against dynamic obstacles. When the obstacle moves fast, collisions may occur. In order to enhance the potential field's ability to react to dynamic obstacles, the velocity of the obstacle V_{obs} is introduced. When the V_{obs} is much less than V_{ref} and close to 0, σ_d is approximately equal to 0 and σ_s is approximately equal to $a\theta_{rob\_tar}$. This means that the repulsive velocity potential field U_{rep} is mainly affected by σ_s. When V_{obs} is much greater than V_{ref}, the obstacle can be considered a dynamic obstacle, and σ_s is approximately equal to 1. At this time, the repulsive velocity potential field U_{rep} is mainly affected by σ_d. When an obstacle moves towards the robot arm, $\theta_{rob\_V_{obs}}$ increase and U_{rep} increase accordingly. The larger repulsive velocity reduces the possibility of the robot arm colliding with obstacles greatly. When the obstacle moves away from the robot arm, $\theta_{rob\_V_{obs}}$ decreases and U_{rep} decreases, reducing the obstacle's impact on the robot arm. The effect of θ_{rob_tar} is to deform the repulsive potential field. When the robot arm is on the side which is closed to the target point around the obstacle, θ_{rob_tar} is smaller. When other parameters remain unchanged, the repulsive potential field U_{rep} will be smaller. When the robot arm is on the side that is far away from the target point around the obstacle, θ_{rob_tar} increases, and the repulsive potential field U_{rep} will be larger.

The velocity-direction factors σ_s and σ_d explain the effect of different obstacles on the potential field well. In addition, the introduction of θ_{rob_tar} and V_{obs} can make the U_{rep} deal with dynamic and static obstacles better at the same time.

The repulsive velocity V_{rep} can be expressed as follows:

$$V_{rep} = \left[\omega_{rep}\ v_{rep}\right]^T \tag{15}$$

where the repulsive angle velocity ω_{rep} is $[0\ 0\ 0]^T$. The repulsive linear velocity is the gradient of the repulsive potential field and it can be obtained from Equation (12)

$$v_{rep} = -\nabla U_{rep}(q) = w_1 \Delta\rho_b + w_2 \Delta\theta_{rob\_tar} + w_3 \Delta\theta_{rob\_Vobs} \tag{16}$$

$$\omega_{rep} = [0\,0\,0]^T \tag{17}$$

The w_1, w_2, w_3 are expressed as

$$w_1 = s_1 e^{m((\frac{a\theta_{rob\_tar}+(rV_{obs})^n}{(1+rV_{obs})^n})+bV_{obs}\theta_{rob\_Vobs})} \tag{18}$$

$$w_2 = \frac{s_2 am}{(1+rV_{obs})^n} e^{m((\frac{a\theta_{rob_tar}+(rV_{obs})^n}{(1+rV_{obs})^n})+bV_{obs}\theta_{rob\_Vobs})} \tag{19}$$

$$w_3 = s_2 mbV_{obs} e^{m((\frac{a\theta_{rob\_tar}+(rV_{obs})^n}{(1+rV_{obs})^n})+bV_{obs}\theta_{rob\_Vobs})} \tag{20}$$

where, $s_1 = k\left(\frac{1}{\rho_b} - \frac{1}{\rho_0}\right)\left(\frac{1}{\rho_b}\right)^2$, the $s_2 = \frac{1}{2}k\left(\frac{1}{\rho_b} - \frac{1}{\rho_0}\right)^2$.

In order to deal with fast and dynamic obstacles, the strategy of adjusting the coefficient of the obstacle's range of action has been added. The function of ρ_0 is expressed as follows:

$$\rho_0 = \begin{cases} \rho_{02}, V_{obs} > \ V_{obs0} \\ \rho_{01} + \frac{(\rho_{02}-\rho_{01})}{V_{obs0}} V_{obs}, 0 < V_{obs} \leq V_{obs0} \\ \rho_{01}, V_{obs} = 0 \end{cases} \tag{21}$$

where ρ_{02} and ρ_{01} are the maximum and the minimum of obstacle's range of action, respectively, and V_{obs0} is the obstacle velocity that maximizes the range of action. ρ_0, which changes according to the transformation of V_{obs}, is more flexible and enables the robot arm to better respond to static obstacles and dynamic obstacles at the same time. According to the above, safety and efficiency are all taken into account in this work.

2.4. Tangential Velocity

The traditional potential field does not have a good processing method for large cubic obstacles such as walls. Both the mobile robot and robot arm are prone to to fall into local minimum in front of large cube obstacles. To solve this problem, this work uses the virtual tangential velocity to make the robot arm get rid of the trap. When the boundary of the obstacle is determined, the side boundary of the obstacle is expanded to prevent the robot arm from getting too close to obstacles. On the side closer to the robot arm, the point which moves a certain distance from the apex of the expanded obstacle boundary to the back side of the obstacle is the tangential point.

As Figure 2 shows, EE is the end effector, P_{tan} is the tangent point and the direction of tangential linear velocity is from EE to P_{tan}. The function of the linear velocity v_{tan} is

$$v_{tan} = \mu\rho_t + \delta \tag{22}$$

where ρ_t is the distance between P_{tan} and the end effector, μ is the coefficient factor and δ is the constant velocity. At the same time, the use of tangential velocity may conflict with the original attraction velocity. When using tangential velocity, the weight of attraction velocity should be greatly reduced. When the end effector moves to the vicinity of the tangential point, the use of tangential velocity ends. The calculation of tangential angle velocity is done in the same way as that of the attractive angle velocity.

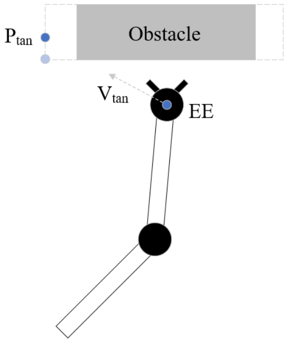


Figure 2. Schematic diagram of tangential velocity.

2.5. Singularity Avoidance Strategy Based on Velocity Space

Although the VPF algorithm calculates the path in Cartesian space, it also needs to map velocity to joint space. The mapping relationship between Cartesian space velocity and joint speed of the VPF can be expressed as

$$\dot{q} = J^{-1}\dot{x} \tag{23}$$

where $\dot{q}$ is the joints speed and $\dot{x}$ is the Cartesian velocity.

An important problem encountered in planning path in Cartesian space is singularity. When the robot arm encounters singularity, it will lose one or more degrees of freedom and the joint speeds will increase a lot. For the singularity problem, this work adopts the damped least squares solution with dynamic damping factor [31]. This method sacrifices the accuracy of the end trajectory, but can avoid singularities effectively. The function of the damped least square solution is defined as

$$\dot{q} = (J^T\ J\ + \lambda^2\ I)^{-1} J^T \dot{x} \tag{24}$$

where I is the identity matrix and λ is dynamic damping factor. The function of λ can be expressed as

$$\lambda = \frac{k_{const}}{tan(d_{thershold} + c) + k_{const}} \tag{25}$$

where k_{const} is a constant and $d_{thershold}$ is the bias of the tangent trigonometric function. The main role of $d_{thershold}$ is to amplify the small changes in c. c is the number of conditions to judge whether the robot arm has fallen into singularity and c can be described as

$$c = \sqrt{det(J\ J^T)} \tag{26}$$

When the robot arm approaches the singularity, the determinant of the Jacobian matrix approaches 0, which leads c close to 0, too. According to Equation (25), the value of λ will increase. Although the accuracy is sacrificed slightly, the singularity can be avoided. When the robot arm is far away from the singular point, c is much greater than 0, which leads λ close to 0. Then, Equation (24) is equal to Equation (23).

The repulsive joint speed cannot be calculated directly, similar to the attractive joint speed, because obstacles do not act on the end effector necessarily. In most cases, they act on other links. When an obstacle acts on the link i (usually not an end effector), this link is regarded as an end effector, and the Jacobian matrix with link i as the end of the robot arm is used to solve the joint speed, that is, only the repulsive speed of joint i and all previous joints are calculated. Therefore, J is replaced by J_i in Equations (24) and (26), and J^{-1} is replaced by ${J_i}^+$ (the pseudo-inverse of J_i) in Equation (23).

3. Experiments

In this section, a series of simulations are designed to compare the informed RRT*, VPF, and IVPF. The experiments are carried out on the KINOVA GEN2 collaborative robot arm with 6 DOF (J2N6S200). This lightweight robot arm has a unique structure without a bulky control cabinet and teaching pendant. It is small in size and weight, and the maximum working radius is 0.985 m. The simulation experiments of the informed RRT*, VPF, and IVPF are compared in MATLAB2021a. Besides, the performances of IVPF are also proved on the Kinova robot.

In the experiment results, the purple and orange circles are the waypoints of the end effector and dynamic obstacle, respectively. The green and magenta cross markers are the final position of the end effector and the target point, respectively.

The parameters of the experiments are defined as follows:

d_0 is the distance between the end effector and the target point.

d_1 is the distance between the robot arm and obstacle A.

d_2 is the distance between the robot arm and obstacle B.

The key parameters of the IVPF algorithm are defined in Table 1.

Table 1. Key parameters of the IVPF algorithm.

Parameter	ζ	k	m	n	a	b	r
No. of Equation	(1)	(4)	(4)	(5)	(5)	(6)	(5)
value	0.1	0.01	5	10	2	5	200

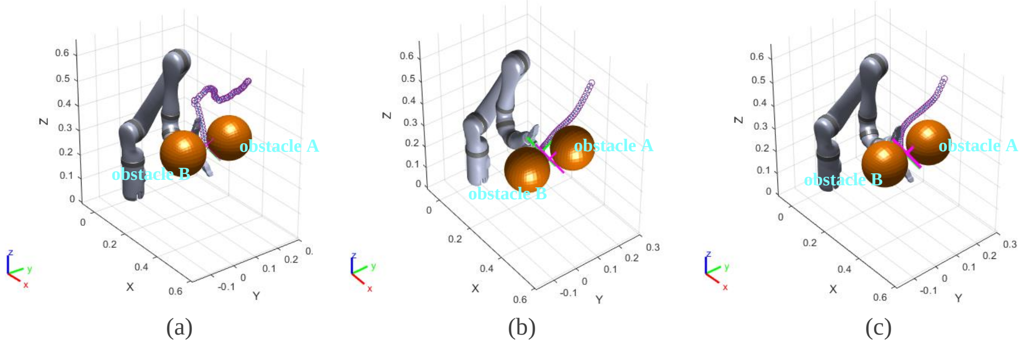
3.1. Static Obstacles Avoidance Experiment I

In this group of comparative experiments, the robot arm base is located at the origin of the coordinate system. The unit of the coordinate axis is m. Two static spherical obstacles (obstacle A and obstacle B) are set at (0.45, 0.1, 0.4) and (0.45, −0.1, 0.4), respectively, and the radius of the two obstacles are both 8 cm. The target point is set between two obstacles and the coordinate is (0.45, 0, 0.4). The purpose of the experiments is to plan the path so that the end effector can move to the target point. The experimental process and results are shown in Figures 3 and 4 and Table 2.

Figures 3a and 4a show the motion trajectory and state of the robot arm when it is using the informed RRT* algorithm. It can be seen that the robot arm can reach the target point and does not collide with the obstacles.

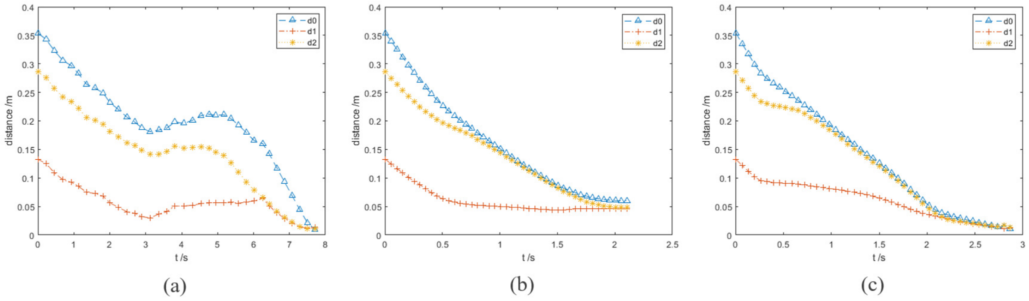
Due to the short distance between the target point and the obstacle, the robot arm falls into the local minimum problem and cannot reach the target point when using the VPF algorithm (Figure 3b), and the distance between the target point and stop point is 7.4 cm (d_0 in Figure 4b).

The motion trajectory and state of the robot arm when it is using IVPF are shown in Figures 3c and 4c. The smoothness of the trajectory is much higher than that of informed RRT*, and the movement running time of informed IVPF (2.9 s) is much shorter than that of informed RRT* (7.78 s). Compared with the VPF algorithm, IVPF also does not easily fall into the local minimum problem. In this case, the robot arm can reach the target point quickly and smoothly when using the IVPF algorithm, and the robot arm always maintains a safe distance from the two obstacles in the movement.



(a) (b) (c)

Figure 3. Experiment A: (**a**–**c**) the running process of the robot arm when using three different algorithms. (**a**) informed RRT*; (**b**) VPF; (**c**) IVPF.



(a) (b) (c)

Figure 4. Experiments A: the distance between the robot arm and the target point and obstacles. (**a**–**c**) The results of the three algorithms. (**a**) informed RRT*; (**b**) VPF; (**c**) IVPF.

Table 2. The result of static obstacles avoidance experiment I.

Experiments A	Informed RRT*	VPF	IVPF
arrived goal	yes	no	yes
collision	no	no	no
time (s)	7.78	-	2.9
minimum d_1 (cm)	1	5.1	1

3.2. Static Obstacles Avoidance Experiment II

In this group of experiments, a wall obstacle is set to verify the ability of three different algorithms to deal with wall obstacles. The static wall obstacle is located at (0.36, 0, 0.4). Its length, width, and height are 36 cm, 1 cm, and 36 cm. The coordinates of target point are located at (0.4, −0.15, 0.4). A static spherical obstacle A is set near the target point and its coordinates are (0.4, −0.3, 0.4). The static spherical obstacle A is set near the target point and its coordinates are (0.4, −0.3, 0.4). The running processes and distance curves of the different algorithms are shown in Figures 5 and 6. The result of different algorithms are compared in Table 3.

Figure 5a exhibits the motion trajectory of the robot arm when using informed RRT* algorithm. The motion trajectory indicates that the robot arm can cross the wall obstacle smoothly, and the whole process only took 6.22 s.

However, the VPF algorithm cannot deal with this wall obstacle well. The motion trajectory in Figure 5b shows that the robot arm falls into the local minimum problem and cannot reach the target. The robot arm cannot cross the wall obstacle when it is using the VPF algorithm. In addition, the robot arm cannot keep a safe distance from the obstacle.

Figure 5c shows the motion trajectory of IVPF and that the robot arm finally reached the target point. The whole process took 6.38 s and the running time is almost the same as that of the informed RRT*. Compared with the VPF algorithm, the IVPF algorithm falls less easily into a local minimum in front of wall obstacles.

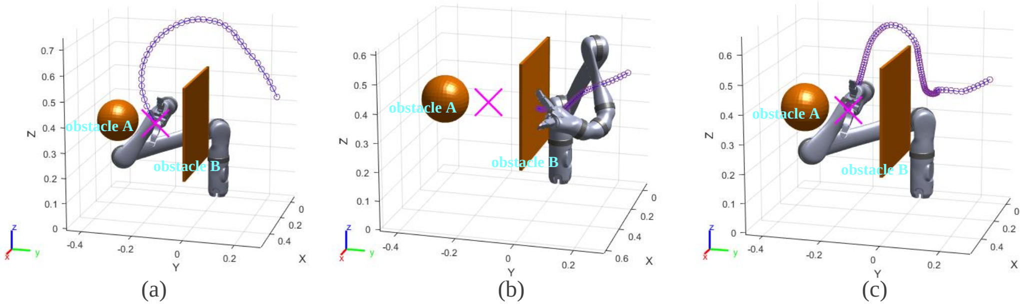


Figure 5. Experiments B: the distance between the robot arm and the target point and obstacles. (**a**–**c**) The results of the three algorithms. (**a**) informed RRT*; (**b**) VPF; (**c**) IVPF.

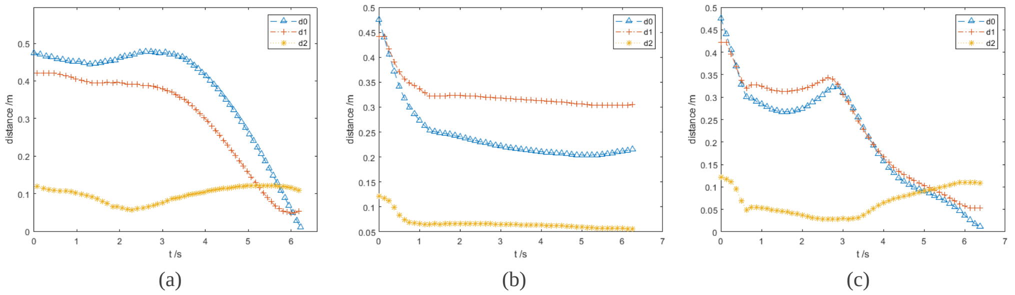


Figure 6. Experiments B: the distance between the robot arm and the target point and obstacles. (**a**–**c**) The results of the three algorithms. (**a**) informed RRT*; (**b**) VPF; (**c**) IVPF.

Table 3. The result of static obstacles avoidance experiment II.

Experiments A	Informed RRT*	VPF	IVPF
arrived goal	yes	no	yes
collision	no	no	no
time (s)	6.22	-	6.38
minimum d_1 (cm)	5.9	0.05	2.8

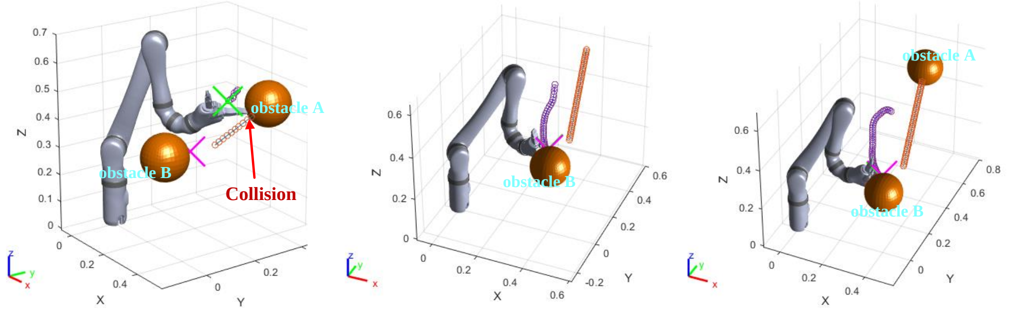
3.3. Dynamic Obstacles Avoidance Experiment I

In this group of experiments, the obstacle A becomes a dynamic one. The velocity vector of obstacle A is $(-0.05, 0.2, 0.05)$ and its unit is m/s. The running processes and distance curves of the different algorithms are shown in Figures 7 and 8. The result of different algorithms are compared in Table 4.

Figure 7a shows the motion trajectory of the robot arm when using the informed RRT* algorithm. In Figure 8a, the robot arm collides with the obstacle A in 1.5 s. The informed RRT* algorithm does not have real-time obstacle avoidance capability in this experiment.

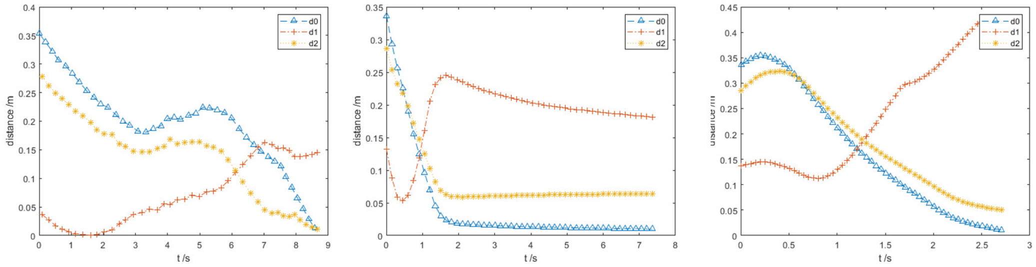
The VPF algorithm is used in the same experimental conditions. Figure 7b exhibits the whole motion trajectory of the end effector of the robot arm. From Figure 8b, it can be seen that the minimum value of d_1 is 5.4 cm (0.48 s). However, if the obstacle suddenly accelerates, this distance may not guarantee that the robot arm has enough reaction time to avoid obstacles. It can be seen from the trajectory in Figure 7b that the end effector of the robot arm made a small obstacle avoidance action. The VPF algorithm takes 7.5 s for the robot arm to reach the target point.Two seconds after starting to move, the robot arm moves very slowly. This is because the target point and obstacle B are very close, and the robot arm is almost at a standstill state. This experiment explains how the VPF algorithm cannot deal with both dynamic and static obstacles well at the same time. Although the robot arm reaches the target point in final, it takes long and does not achieve satisfactory results for dynamic obstacles.

Finally, the performances of the IVPF are verified. Figure 7c shows the whole motion trajectory of the end effector of the robot arm. The motion trajectory of the big obstacle avoidance action of the end effector can be seen clearly from this figure. This evasive action can make the robot arm stay away from obstacle and is verified by the curve of d_1 in Figure 8c. The curve of d_1 shows the distance between the robot arm and dynamic obstacle A, and the shortest distance is 11.8 cm. This distance increased 118.5% by that of VPF. Furthermore, it only took 2.75 s for the robot arm to reach the target point, which is much faster than that of VPF. Although the target point is very close to obstacle B, it has little effect on the robot arm and the robot arm can still reach the target point quickly.



(a) (b) (c)

Figure 7. Experiments B: the running process of the robot arm when it is using three different algorithms. (**a**) informed RRT*; (**b**) VPF; (**c**) IVPF.



(a) (b) (c)

Figure 8. Experiments B: the distance between the robot arm and the target point and obstacles in the simulation experiments. (**a**–**c**) The results of the three algorithms. (**a**) informed RRT*; (**b**) VPF; (**c**) IVPF.

Table 4. The result of static obstacles avoidance experiment I.

Experiments B	Informed RRT*	VPF	IVPF
arrived goal	yes	yes	yes
collision	yes	no	no
time (s)	8.9	7.5	2.75
minimum d_1 (cm)	0	5.4	11.8

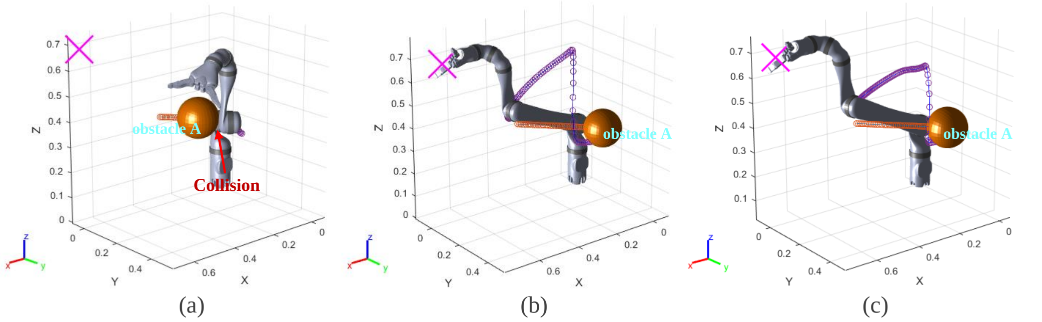
3.4. Dynamic Obstacles Avoidance Experiment II

In this group of comparative experiments, the target point is located at (0.7, 0, 0.7). The initial location of a dynamic spherical obstacle is set at (0.4, 0.1, 0.4), and the radius of the obstacle is 8 cm. The obstacle will move towards joint 3 of the robot arm with the velocity vector of $(-0.1, 0.1, 0)$ (m/s). The result of different algorithms are compared in Table 5.

The motion process of the informed RRT* algorithm is exhibited in Figure 9a. In Figure 10a, the robot arm collides with the obstacle A in 1.02 s. The informed RRT* algorithm does not have real-time obstacle avoidance capability in this experiment.

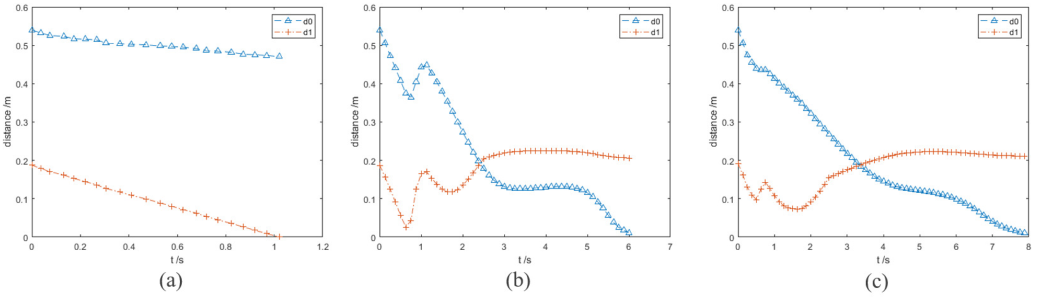
The VPF algorithm is used in the same experimental conditions. The motion process of the VPF algorithm is exhibited in Figure 9b. The trajectory in the figure shows that the robot arm made an evasive action when the obstacle moved towards joint 3. However, it can be seen that the robot arm is very close to the obstacle at 0.68 s, which can be verified by curve d_1 in Figure 10b. The minimum value of d_1 is 2 cm. Although the robot arm can react to the dynamic obstacle when it is using VPF, it is difficult for the robot arm to maintain a sufficient safe distance from the dynamic obstacle. Therefore, the ability handling of the dynamic obstacle using the VPF algorithm is not ideal.

The moving performances of the IVPF algorithm are exhibited in Figures 9c and 10c. The curve d_1 shows that the robot arm reaches the closest position to the obstacle twice. The two minimum distances are 9.5 cm and 6.9 cm. These distances can ensure that the robot arm and obstacles will not collide temporarily. However, these distances are the minimum between the robot arm and the obstacle at a whole motion. The robot arm has avoided the front of the obstacle's movement at this time. The minimum distance in this processing is still longer than that of the VPF algorithm, and increases by 245%.



(a) (b) (c)

Figure 9. Experiments C: the running process of the robot arm when it is using three algorithms. (**a**) informed RRT*; (**b**) VPF; (**c**) IVPF.



(a) (b) (c)

Figure 10. Experiments C: the distance between the robot arm and the target point and obstacle in the simulation experiments. (**a**–**c**) The results of the three algorithms. (**a**) informed RRT*; (**b**) VPF; (**c**) IVPF.

Table 5. The result of static obstacles avoidance experiment II.

Expriments C	Informed RRT*	VPF	IVPF
arrived goal	no	yes	yes
collision	yes	no	no
time (s)	-	6.1	8
minimum d_1 (cm)	0	2	6.9

3.5. The Actual Experiments

In the actual experiment, a simple robot obstacle avoidance scenario is designed to verify the feasibility of the IVPF algorithm. The initial state of the Kinova robot arm is shown in Figure 11a. There is a wall obstacle (iron plate) set near the robot arm. The obstacle is 54 cm high and 42 cm wide. The coordinates of the obstacle's center is (0.2, 0, 0.27). The purpose of this experiment is to verify that the IVPF algorithm can make the robot arm safely cross the obstacle and reach the other side of the obstacle. In order to ensure the safety during the experiment, the maximum joint speed of the robot arm is set as 15°/s.

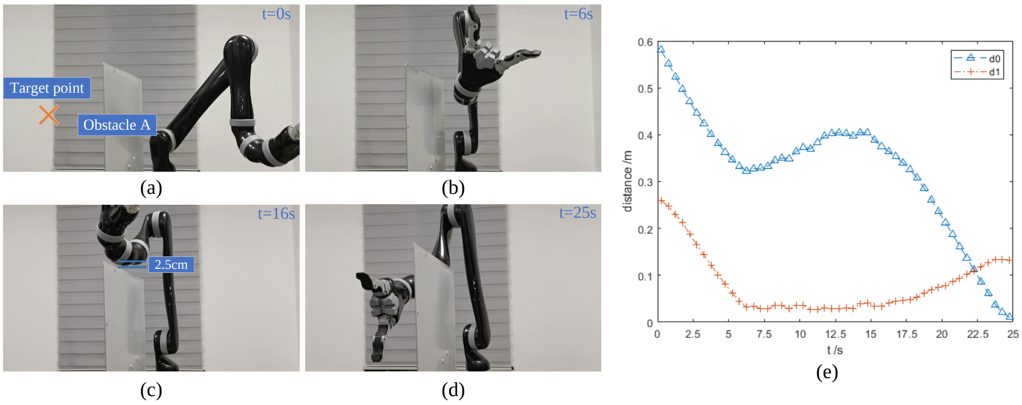


Figure 11. Experiments E: actual experiments. (**a**–**d**) The running process of the robot arm when it is using the IVPF algorithm. (**e**) The distance between the robot arm and the target point and the obstacle in the experiments.

Figure 11 exhibits the process of the movement of the robot arm. Figure 11a shows the initial state of the robot arm, and (b) to (d) show the state of the robot arm in the experiment. It can be seen from the figures that the robot arm can cross the obstacle and maintain a distance with the obstacle. The minimum distance between the robot arm and the obstacle is 2.5 cm, which can be seen in Figure 8c. The curves in Figure 11e can also confirm this situation.

3.6. Discussion

It can be seen from several comparative experiments that compared with VPF and the informed-RRT* algorithms, the IVPF algorithm has better dynamic obstacle avoidance ability and a fast response speed. When the medical and surgical robot arm uses the IVPF algorithm, the robot arm can always keep a safe distance from the dynamic obstacles, which can ensure the safety of both the human and the robot arm. The fast response to obstacles can help the robot arm address unexpected situations better. In addition, the IVPF algorithm reduces the influence of static obstacles, which can make the robot reach the target point to perform a task easily. However, IVPF also has shortcomings. Although the tangential velocity is used to get rid of the local minimum problem, the path quality planned by theIVPF algorithm is not good enough compared with the informed-RRT* algorithm. Therefore, the IVPF algorithm needs to be improved in how to execute tasks more efficiently in some scenarios.

4. Conclusions

This work proposed an improved VPF algorithm to deal with the inherent disadvantages of the traditional algorithm, such as easily falling into local minimum, being unable to reach the target and also showcasing poor dynamic obstacle avoidance ability. In the proposed IVPF algorithm, dynamic and static obstacle velocity-direction factors are introduced into the attractive and repulsive field, respectively, and the singularity problem is also discussed. Experiments including static and dynamic obstacles avoidance showed that the proposed algorithm has good performances in the obstacle avoidance effect and trajectory smoothness. The actual experiment also elucidates the algorithm's manifestation on the wall obstacle. The results of the study could be applied to relevant human–robot interaction conditions, which could greatly promote the scientific development of robot safety control. Compared with other algorithms, the proposed IVPF algorithm can make the medical and surgical robot cooperate with humans without collision, while ensuring the safety of the humans at the same time to complete the task. With its ability to provide a faster response time, enhanced safety, and improved task execution, the IVPF algorithm is likely to have a positive impact on medical and surgical scenarios. The IVPF algorithm can ensure the safety of humans when the robot arm is performing a task, and it enables the robot arm to address unexpected situations and perform tasks better. The IVPF algorithm has the potential to significantly improve the performance of medical and surgical robots, making them safer and more effective tools for doctors and surgeons. The application of this algorithm could make the collaborative robots work in a flexible and safe condition, which could open up new opportunities for the future development of medical and surgical scenarios.

Author Contributions: Methodology, X.X., K.Y., Y.C. and T.L.; validation, X.X., H.M. and S.S.; writing, X.X., K.Y. and Q.Z.; funding acquisition, T.L. and K.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China (No. 2021ZD0140409), National Natural Science Foundation of China (No. 62003232), STIP of Higher Education Institutions in Shanxi (No. 2020L0072).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Moran, M.E. Evolution of robotic arms. *J. Robot. Surg.* **2007**, *1*, 103–111. [CrossRef] [PubMed]
2. Saeidi, H.; Opfermann, J.D.; Kam, M.; Wei, S.; Léonard, S.; Hsieh, M.H.; Kang, J.U.; Krieger, A. Autonomous robotic laparoscopic surgery for intestinal anastomosis. *Sci. Robot.* **2022**, *7*, eabj2908. [CrossRef] [PubMed]
3. Zhang, W.; Li, H.; Cui, L.; Li, H.; Zhang, X.; Fang, S.; Zhang, Q. Research progress and development trend of surgical robot and surgical instrument arm. *Int. J. Med. Robot. Comput. Assist. Surg.* **2021**, *17*, e2309. [CrossRef] [PubMed]
4. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
5. Li, F.; Huang, Z.; Xu, L. Path planning of 6-DOF venipuncture robot arm based on improved a-star and collision detection algorithms. In Proceedings of the 2019 IEEE international conference on robotics and biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 2971–2976.
6. Kavraki, L.E.; Svestka, P.; Latombe, J.C.; Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [CrossRef]
7. Kunz, T.; Reiser, U.; Stilman, M.; Verl, A. Real-time path planning for a robot arm in changing environments. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 5906–5911. [CrossRef]
8. Siméon, T.; Laumond, J.P.; Nissoux, C. Visibility-based probabilistic roadmaps for motion planning. *Adv. Robot.* **2000**, *14*, 477–493. [CrossRef]
9. LaValle, S.M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*; Iowa State University: Ames, IA, USA, 1998.
10. Zhaochu, H.; Yuanlie, H.; Bi, Z. Obstacle avoidence path planning for robot arm based on mixed algorithm of artificial potential field method and RRT. *Ind. Eng. J.* **2017**, *20*, 56.
11. Wang, X.; Li, X.; Guan, Y.; Song, J.; Wang, R. Bidirectional potential guided rrt* for motion planning. *IEEE Access* **2019**, *7*, 95046–95057. [CrossRef]
12. Li, S.; Zhao, D.; Sun, Y.; Yang, J.; Wang, S. Path Planning Algorithm Based on the Improved RRT-Connect for Home Service Robot Arms. In Proceedings of the 2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR), Tokoname, Japan, 4–6 March 2021; pp. 403–407.
13. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [CrossRef]
14. Islam, F.; Nasir, J.; Malik, U.; Ayaz, Y.; Hasan, O. Rrt-smart: Rapid convergence implementation of rrt towards optimal solution. In Proceedings of the 2012 IEEE International Conference on Mechatronics and Automation, Chengdu, China, 5–8 August 2012; pp. 1651–1656.
15. Qureshi, A.H.; Iqbal, K.F.; Qamar, S.M.; Islam, F.; Ayaz, Y.; Muhammad, N. Potential guided directional-RRT* for accelerated motion planning in cluttered environments. In Proceedings of the 2013 IEEE International Conference on Mechatronics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 519–524.

16. Jeong, I.B.; Lee, S.J.; Kim, J.H. RRT*-quick: A motion planning algorithm with faster convergence rate. In *Robot Intelligence Technology and Applications 3*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 67–76.
17. Gammell, J.D.; Srinivasa, S.S.; Barfoot, T.D. Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 2997–3004.
18. Sozzi, A.; Bonfè, M.; Farsoni, S.; De Rossi, G.; Muradore, R. Dynamic motion planning for autonomous assistive surgical robots. *Electronics* **2019**, *8*, 957. [CrossRef]
19. Atawnih, A.; Papageorgiou, D.; Doulgeri, Z. Kinematic control of redundant robots with guaranteed joint limit avoidance. *Robot. Auton. Syst.* **2016**, *79*, 122–131. [CrossRef]
20. Wang, W.; Zhu, M.; Wang, X.; He, S.; He, J.; Xu, Z. An improved artificial potential field method of trajectory planning and obstacle avoidance for redundant manipulators. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1729881418799562. [CrossRef]
21. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous Robot Vehicles*; Springer: Berlin/Heidelberg, Germany, 1986; pp. 396–404.
22. Ye, M.; Li, W.; Chan, D.T.M.; Chiu, P.W.Y.; Li, Z. A Semi-Autonomous Stereotactic Brain Biopsy Robot With Enhanced Safety. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1405–1412. [CrossRef]
23. Heunis, C.M.; Barata, B.F.; Phillips Furtado, G.; Misra, S. Collaborative Surgical Robots: Optical Tracking During Endovascular Operations. *IEEE Robot. Autom. Mag.* **2020**, *27*, 29–44. [CrossRef]
24. Tian, Y.; Zhu, X.; Meng, D.; Wang, X.; Liang, B. An Overall Configuration Planning Method of Continuum Hyper-Redundant Manipulators Based on Improved Artificial Potential Field Method. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4867–4874. [CrossRef]
25. Cho, W.J.; Kwon, D.S. A sensor-based obstacle avoidance for a redundant manipulator using a velocity potential function. In Proceedings of the 5th IEEE International Workshop on Robot and Human Communication, RO-MAN'96 TSUKUBA, Tsukuba, Japan, 11–14 November 1996; pp. 306–310. [CrossRef]
26. Zhang, H.; Zhu, Y.; Liu, X.; Xu, X. Analysis of Obstacle Avoidance Strategy for Dual-Arm Robot Based on Speed Field with Improved Artificial Potential Field Algorithm. *Electronics* **2021**, *10*, 1850. [CrossRef]
27. Volpe, R.; Khosla, P. Manipulator control with superquadric artificial potential functions: Theory and experiments. *IEEE Trans. Syst. Man Cybern.* **1990**, *20*, 1423–1436. [CrossRef]
28. Zhao, M.; Lv, X. Improved manipulator obstacle avoidance path planning based on potential field method. *J. Robot.* **2020**, *2020*, 1701943. [CrossRef]
29. Xu, X.; Hu, Y.; Zhai, J.; Li, L.; Guo, P. A novel non-collision trajectory planning algorithm based on velocity potential field for robotic manipulator. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1729881418787075. [CrossRef]
30. Park, D.H.; Hoffmann, H.; Pastor, P.; Schaal, S. Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In Proceedings of the Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots, Daejeon, Republic of Korea, 1–3 December 2008; pp. 91–98.
31. Cao, Q.; Sun, M.; Xue, W.; Xia, S. Research on dynamic obstacle avoidance and singularity avoidance of manipulator under human-machine cooperation. *J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.)* **2020**, *48*, 55–59. (In Chinese)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.