Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Obstacle Avoidance of Manipulators Based on Improved Artificial Potential Field Method

Hao Li[1,2,3,4], Zhiyang Wang[1,3,4], and Yongsheng Ou[1,3,4,⋆]

***Abstract*— Obstacle avoidance is a key problem of robot motion planning. This paper introduces an improved method to realize obstacle avoidance of manipulators based on artificial potential field method which offers a relatively fast and effective way to solve the problem. We establish the attractive potential field function of the goal and the repulsive potential filed functions of obstacles in the workspace instead of the configuration space. Then a series of control points on the manipulator are chosen to be influenced by the total potential. A policy is proposed that only the end point is attracted to the goal while all selected points are repulsed by obstacles. Afterwards, we transfer the potential force of each point in workspace into the moment in configuration space to make the manipulator move by iteration of joint angles. The effectiveness of the method is confirmed by simulation experiments.**

***Index Terms*— Obstacle avoidance Artificial potential field Motion planning Robot manipulation**

I. Introduction

Nowadays manipulators are mainly used to accomplish pipelined tasks in traditional industrial productions. Generally, they are used to implement human pre-programmed tasks in a structured environment. In fact, this approach is inefficient and has little generalization ability. With the promotion of Industry 4.0, the traditional mode of production can no longer meet the demand of development, and manipulators need to be more adaptable. When the working environment changes, how to solve the adaptability of the robot becomes a serious problem. In order to improve the autonomous movement ability and reduce the application cost of the robot, motion planning technology is essential. It can be described as follows: given a robot and a description of its working environment, plan a collision free path between two specified locations that satisfies some certain optimization criteria [1]. Usually we are interested in finding a path that minimizes some cost or distance metric. In this paper, we discuss the problem of obstacle avoidance, which requires manipulators effectively to avoid obstacles in the environment in order to ensure the safety of work.

Obstacle avoidance is the task of satisfying some control objective subject to non-intersection or non-collision position constraints. It is a very important issue in motion planning which is one of the most active subfields of robotics. When it comes to manipulators, the problem becomes challenging because with the increase of the degrees of freedom, the calculation of motion planning increases exponentially. Nonetheless, researchers have proposed many popular approaches to solve this problem [2]. Graph-based plan methods are fundamental. This category includes a few algorithms that can be used to plan paths between a start node and a goal node including the breadth first search or grassfire algorithm, Dijkstras algorithm and the A^* procedure. An improved A^* algorithm is proposed for collision detection on a 6-DoF manipulator in the workspace by Wang [3]. In addition, planning via cell decomposition is also feasible in any dimension theoretically. It is probability complete and optimal but can cause a dimension explosion. Sampling-based planning methods are widely applied. The principle is to randomly select sampling points in the configuration space, and then forge collision free edges between neighboring sample points to form a graph that captures the structure of the configuration space. The two most classical algorithms are Probabilistic Road Maps (PRM) and Randomly Exploring Rapid Trees (RRT). One advantage of this kind of approaches is that they can be used in cluttered environment and high-dimensional space, ensuring global path planning with global convergence properties, albeit at the expenses of being computationally expensive. However, these methods are usually evaluated offline and remain limited to quasi static environments [4]. Park et al. propose efficient real-time path planning by combining PRM and reinforcement learning to deal with uncertain dynamic environments [5]. Actually, with the progress of artificial intelligence, reinforcement learning has become a very popular area of research at present. Researchers model motion planning as a Markov Decision Process and then solve it through classical methods of reinforcement learning like dynamic programming or monte carlo [6][7]. Also there are optimization-based approaches. Genetic algorithms and ant colony algorithms

[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, P.R.China. [2]University of Chinese Academy of Sciences, Beijing 100049, P.R.China. [3]Guangdong Provincial Key Laboratory of Robotics and Intelligent System, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. [4]CAS Key Laboratory of Human-Machine Intelligence-Synergic Systems, Shenzhen Institutes of Advanced Technology.

⋆Yongsheng Ou is the corresponding author. (email: ys.ou@siat.ac.cn)

This research is funded by National Natural Science Foundation of China (Grant No. U1613210), Guangdong Special Support Program (2017TX04X265), Science and Technology Planning Project of Guangdong Province (2019B090915002) and Shenzhen Fundamental Research Program (JCYJ20170413165528221).

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

are typical examples. Khoogar introduced a novel method for obstacle avoidance with redundant robot manipulators based on genetic algorithms [8]. Zhou proposed an improved ant colony algorithm to improve planning of space manipulator end-effector [9]. In this paper, this problem is effective and simple. Basically, we consider motion planning as a nonlinear optimization problem. The goal is to design a control law, for example a trajectory $T(t)$, minimizing some cost functional J under some constraints, such as the total energy consumed or the duration of the motion, such that the dynamic equations are satisfied at all times, the controls are feasible, the motion is collision free, and the trajectory takes the start state to the goal state. A model predictive control has been used in dynamic path planning and obstacle avoidance [10].

Another approach to motion planning involves constructing artificial potential fields (APF) which are designed to attract the robot to the desired goal configuration and repel it from configuration space obstacles. The APF method was first proposed by Khatib in 1986 [11]. The robots motion can then be guided by considering the gradient of this potential function. Kim et al. use harmonic functions which completely eliminate local minima even for a cluttered environment [12]. Based on this idea, Ecole Polytechnique Federale de Lausanne (EPFL) has done abstruse studies. Zadeh proposed a dynamic system approach based on closed-form equation of harmonic potential flow around simple convex obstacles to realize realtime obstacle avoidance [13]. Lukas Huber then offers an approach to avoid multiple concave obstacles, that preserves the asymptotic stability and inherits the convergence properties of harmonic potential fields [14].

In this paper, we use an improved artificial potential field method for the multi-DOF manipulators to achieve obstacle avoidance of the whole arm, and solve the local minimum problem caused by the method.

II. Problem formulation

A. Configuration space

In the real world, most of the robots we are going to build can move continuously through space. Configuration space (C-space) is a handy mathematical and conceptual tool, which was developed to help us think about these kinds of problems in a unified framework. Basically, the configuration space of a robot is the set of all configurations or positions that the robot can attain. Now we introduce the notion of configuration space obstacles which are regions in configuration space that the robot cannot inhabit because of obstacles or other impediments. Conversely, the region of configuration space that the robot can attain is referred to as the free space of the robot. Actually the dimensions and shape of the configuration space obstacle are obtained by considering both the obstacle and the shape of the robot. The configuration of the robot corresponds to a point in the configuration space. And the dark areas correspond to configurations that the robot cannot attain shown in Fig.1. In this setting, the task of planning a path for our robot

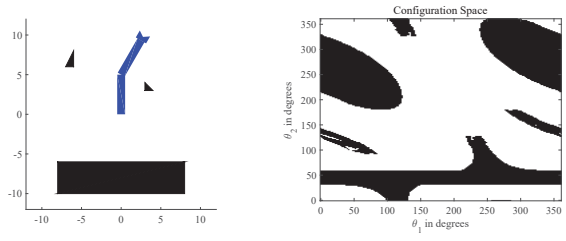


Fig. 1. The two link manipulator in an environment with obstacles and the corresponding configuration space.

corresponds to planning a trajectory through configuration space from the starting configuration to the ending configuration. So we generally apply artificial potential field method to solve problems in configuration space. For a manipulator, it can be described by a point using a generalized coordinate (usually the joint angles). For example, a six-degree-of-freedom manipulator can be described by a six-dimensional vector space $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$. However, the description of the corresponding obstacles is quite cumbersome. Because of the problems of multiple solutions and singularities in the inverse kinematics of the manipulators, the mapping from the workspace to the configuration space is nonlinear. It is difficult to give a distinct description of the obstacle space in C-space, i.e., the "curse of dimensionality" makes that difficult.

To solve it, we consider to establish potential field functions in the workspace instead of the C-space. Since most of the structures (continuously rotation joints, translational joints, etc.) form a differentiable manifold, which is homomorphic to the Euclidean space at any point, the above method is also applicable in the workspace. When defined in the C-space, since the robot can be regarded as a particle, the definition process is relatively simple. However, in the workspace, the robot is a rigid body with a certain volume and cannot be represented by a particle. Therefore, the definition process is more complicated. A feasible method is to select several discrete points along length of each link. For each of these control points, we can construct a potential field which guides it away from the obstacles in the workspace and towards its final goal. Here we adopt a policy that only the end point is attracted to goal but all points are repulsed by obstacles. We can do this by considering the distance between each point and the workspace obstacles. Afterwards, an aggregate potential function can be constructed by computing the sum of these individual potential functions, then we obtain net force acting on each point by computing the gradient of this function with respect to the configuration space coordinates. This gradient then provides information about how to move the robot towards the goal and away

from obstacles. Next, we define the artificial potential field in the workspace. The artificial potential field needs to be able to ensure that it has a minimum value at the focus, and the artificial potential field increases as the distance from the obstacle is shortened. Then, we map the artificial potential field obtained in the workspace to the configuration space, and finally apply the gradient descent method to solve the robot trajectory in the configuration space.

B. The attractive potential field

We use $P = \{p_1, p_2, ..., p_N\}$ to indicate the evaluation points on the manipulator, then we define an artificial potential field at each point. We always choose the end effector as the Nth evaluation point. The simplest method of attractive potential field selection is to make it increase linearly with distance, i.e., the conical potential. However, in this case, the gradients are constant everywhere, and the gradient at the end point is 0. Therefore, the gradient is not continuous, which may cause system instability. We also hope that the attractive potential field will be smooth everywhere. So we construct quadratic potential function:

$$U_{att,i}(q) = \frac{1}{2} K_{att} d^2(q) \tag{1}$$

Where K_{att} is a positive weight coefficient, $d = ||p_i(q) - p_i(q_{goal})||$, which represents the distance between the ith evaluation point in configuration q and the goal configuration q_{goal}. It should be noted that if the start point is very far from the end point, the attractive force could be very large. In this case, it may be desirable to have distance functions that grow more slowly to avoid huge velocities far from the goal. Therefore, we usually combine the conical field with quadratic field structure. We use the conical one when the distance is far, and quadratic one when it is closer. Due to the limited scope of the workspace, we do not consider this situation in this paper. The negative gradient of the attractive field is the attractive force.

$$F_{att,i}(q) = -\nabla U_{att,i}(q) \tag{2}$$

$$= -\frac{1}{2} K_{att} \nabla d^2(q) \tag{3}$$

$$= -K_{att}(p_i(q) - p_i(q_{goal})) \tag{4}$$

C. The repulsive potential field

When constructing the repulsive potential field, it is necessary to ensure that the robot and the obstacle repel each other so that the two never touch; meanwhile, when the robot is away from the obstacle, the repulsive force should be gradually reduced. Thus, one of the simplest methods of constructing a repulsive field is to make the repulsive field tend to infinity at the obstacle boundary and to zero at a distance from the obstacle. We define this distance as Q^*. It is the radius within which the repulsive potential is "activated" and can influence the robot.

$$U_{rep,i}(q) = \begin{cases} \frac{1}{2} K_{rep} \left(\frac{1}{D(q)} - \frac{1}{Q^*} \right)^2 & D(q) \le Q^* \\ 0 & D(q) > Q^* \end{cases} \tag{5}$$

Where K_{rep} is a positive weight coefficient, $D = ||p_i(q) - p_i(obs)||$, which represents the nearest distance between the ith evaluation point and the obstacle. Suppose the obstacle is convex so that $D(q)$ is smooth everywhere, The repulsive force to the robot can be represented by the negative gradient of the repulsive field as follows:

$$F_{rep,i}(q) = \begin{cases} -K_{rep} \left(\frac{1}{D(q)} - \frac{1}{Q^*} \right) \frac{1}{D^2(q)} \nabla D(q) & D(q) \le Q^* \\ 0 & D(q) > Q^* \end{cases} \tag{6}$$

Where $\nabla D(q)$ represents the gradient of the shortest distance in the workspace. If p is the point on obstacle which is closest to the ith evaluation point, then we have

$$\nabla D(q) = \frac{p_i(q) - p}{||p_i(q) - p||} \tag{7}$$

Which is the unit vector from p to $p_i(q)$.

It should be noted that if the shape of the obstacle is not convex, the distance function $D(q)$ is not necessarily differentiable everywhere, which means the force vector is discontinuous. So the selection of the evaluation points does not guarantee that the robot and the obstacle do not collide. When the evaluation point and the obstacle position are shown as Fig.2, collision occurs probably. We can select more floating evaluation points to ensure that the nearest distance exists. Considering that the length of the manipulator link is short, generally we only select the end point and center point of the link.

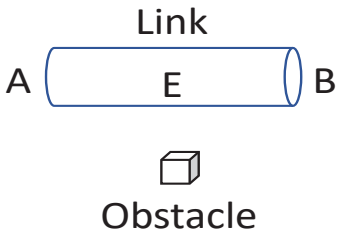


Fig. 2. Position relation between the obstacle and the evaluation points. A and B are end points of the link, E is the middle point. In this case, point E is the closest point to the link rather than A or B or other points.

Through the above two sections, we finally get the total potential energy of the ith evaluation point under the configuration q :

$$U_{net,i}(q) = U_{att,i}(q) + U_{rep,i}(q) \tag{8}$$

and the net force is

$$F_{net,i}(q) = -\nabla U_{att,i}(q) - \nabla U_{rep,i}(q) \tag{9}$$

$$= F_{att,i}(q) + F_{rep,i}(q) \tag{10}$$

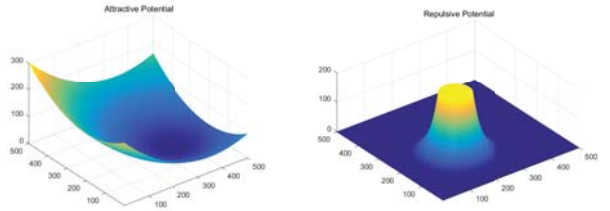


Fig. 3. Display of the attractive and repulsive potential with one goal at (350,200) and only one obstacle at (200,200).

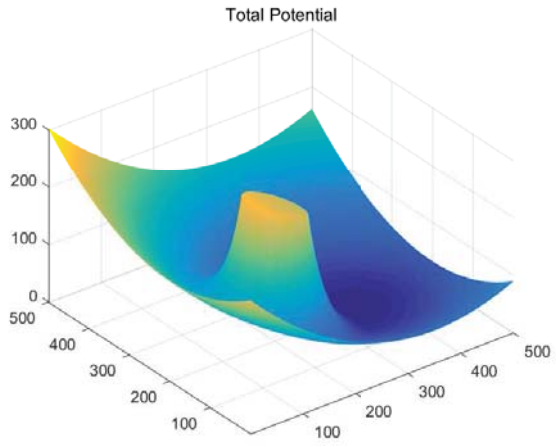


Fig. 4. Display of the total potential energy.

D. Relate linear forces to joint angles

Next we discuss how to map the force in the workspace to the force or moment in the configuration space. Suppose we use M to represent the moment (rotational pair) and force (moving pair) at the joint of the robot. Then we can use the virtual displacement method to obtain the relationship between the force and the moment in the C-space and in the workspace. Assume potential forces on robot body are static. $(\delta x, \delta y, \delta z)$ describes the virtual displacement in the workspace, δq describes the virtual displacement in the C-space, according to reciprocal virtual work theorem, we have

$$F \cdot (\delta x, \delta y, \delta z)^T = M \cdot \delta q \tag{11}$$

by definition of the jacobian matrix,

$$(\delta x, \delta y, \delta z)^T = J \cdot \delta q \tag{12}$$

so we can get

$$M = J^T F \tag{13}$$

through which we can map the forces in workspace to C-space. The joint moment is the potential field force in C-space. Then we use the resulting moment in C-space to move the manipulator. Now the angles can be updated in discrete simulation in similar fashion:

$$\theta(t+1) = \theta(t) + \alpha M \tag{14}$$

where α is an iterative weight constant defined by user. For multi-link manipulators, the net moment vector

$$M_{net} = \sum_{i=1}^{N} J_i^T F_i \tag{15}$$

where

$$M_{net} = [M_1 \quad M_2 \quad \ldots \quad M_m] \tag{16}$$

and M_i corresponds to the moment that will change θ_i, N represents total number of selected points on manipulator, m represents total number of joints (DoFs). Consider the strategy in this article, we regulate that only end point is attracted to goal but all points are repelled from obstacles.

$$F_{att}(q) = F_{att,N}(q) = -\nabla U_{att,N}(q) \tag{17}$$

$$= K_{att}(p_N(q_{goal}) - p_N(q)) \tag{18}$$

$$M_{att} = J_N^T F_{att} \tag{19}$$

$$M_{net} = \sum_{i=1}^{N} M_{rep,i} + M_{att,N} \tag{20}$$

$$= \sum_{i=1}^{N} J_i^T F_{rep,i} + J_N^T F_{att,N} \tag{21}$$

Because of the existence of obstacles, potential field is non-linear and discontinuous. Hence, we must compute equilibrium angles iteratively. Here we use a batch update rule:

$$\Theta \leftarrow \Theta + \alpha M_{net} \tag{22}$$

where

$$\Theta = [\theta_1 \quad \theta_2 \quad \ldots \quad \theta_m]^T \tag{23}$$

The pseudo-code for the proposed iterative algorithm is shown in the following table.

TABLE I
ITERATIVELY UPDATING THE JOINT ANGLE TO MOVE.

Step 1: Initialize Θ, α and $U_{net,i}$
Repeat:
Step 2: Calculate J_i and p_i for $\forall\ i$
Step 3: Obtain $F_{net,i}$ and $M_{net,i}$ through the previous step
Step 4: Update joint angles $\Theta \leftarrow \Theta + \alpha M_{net,i}$

Here are stopping criteria for equilibrium angle calculation. *a*. Goal reached. We think that end effector is close enough to the goal if $||p_i(q) - p_i(q_{goal})|| < threshold$.

b. Minimum threshold for joint moment met. Running algorithm for more iterations will not change angles significantly. *c*. Maximum number of iterations reached. The value is chosen by user to prevent algorithm from running forever.

III. Local minimum problem

Ideally, we want to construct a total potential field with only a single global minimum located at the desired configuration and no local minimum. However, when closely spaced obstacles exist or there is a dead end, the attractive and repulsive forces conspire to produce local minimum which means $M_{net} = 0$ at locations other than the desired location, so the manipulator makes no progress. We call the angle of the joint in this state the equilibrium angle θ^*. This is equivalent to either the manipulator has reached the goal (global minimum of U_{net}) or it cannot get any closer to the goal (local minimum of U_{net}). The former corresponds to criterion *a* while the latter corresponds to criterion *b*. Local minima are where robot is in equilibrium but has not reached goal. They correspond to local valleys other than the goal, where the robot can get stuck. What this means in practice is that the simple gradient based control strategy may or may not converge the goal depending upon where the robot starts. It turns out that these kinds of local minima are very hard to eliminate, some configurations of obstacles and parameter settings lead to local minima while others do not. The more obstacles there are in space, the more complex the total potential energy of the manipulator is, and local minima are more likely to appear. Actually in practice it is hard to know before hand when this problem will occur. Once that happens, usually we use a back tracking procedure to detect these situations and to switch to a different planning strategy.

Fig.5 shows normal experimental case and local minimum case in C-space. Fig.5 (a) and (c) show that the robot can reach the target successfully avoiding the obstacle while (b) and (d) tell that the robot falls into a local minimum in gradient descent. At the end of the red line, the net force of the robot is 0, it is unable to move continuously. So we see that the robot can not converge to the target point. Fig.5 (c) and (d) correspond to the situation of local minimization when there are two obstacles in the workspace. To solve this problem, we think about changing the balance of forces to make the robot get out of the local minimal situation. Under the circumstances, virtual target points or virtual obstacles are added to change the environment typically. The selection principle is that the virtual point can generate enough attractive or repulsive force to overcome the gradient around the local minimum and escape it as quickly as possible. In other words, the descending gradient of the potential energy of the virtual point is greater than the ascending gradient around the minimal point, which makes successful obstacle avoidance possible.

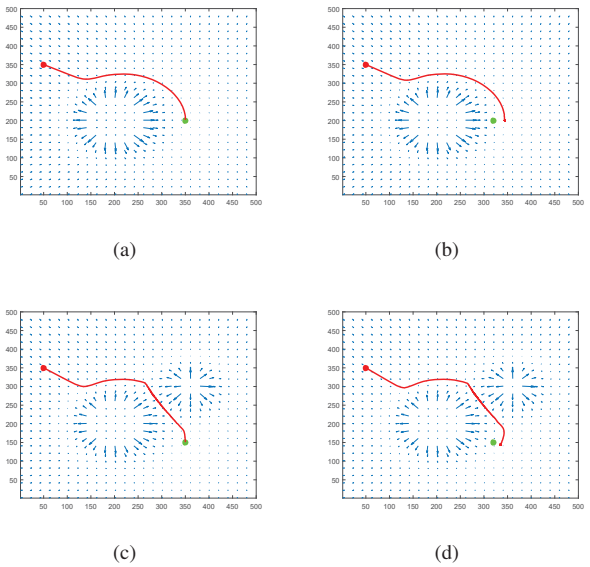
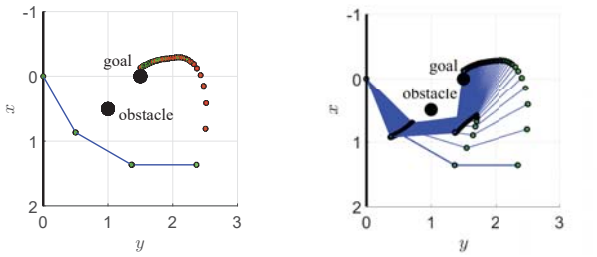


Fig. 5. (a) and (c) show normal experimental case while (b) and (d) show local minimum case in the presence of one obstacle and two obstacles.

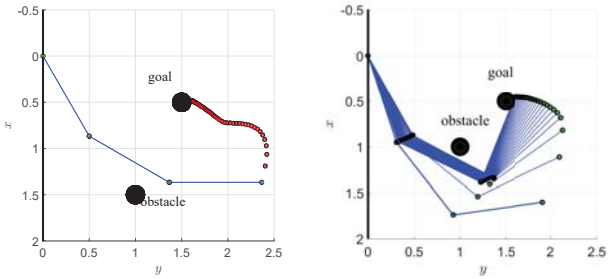
IV. Simulation and experiment

We test the artificial potential with a three-DoF planar manipulator. We choose three joint points and the middle points of every links as control points which influenced by the potential field. Here we assume that the obstacle is a circle shape with a certain radium. Different sets of the positions of obstacle center and goal are selected to verify the method. The simulation results show the feasibility of the method we proposed. In the first experiment, we set the coordinates of the target point to be (0,1.5), and the coordinates of the obstacle to be (0.5,1). Fig.6 (a) shows the trajectory of end effector, that is the last control point in the manipulator. Because of the attractive force on it, we can see that it moves incrementally towards the target point. Fig.6 (b) shows the path diagram of the whole arm during iteration. Due to the repulsive force of every control points, during the movement of the end effector towards the target point, all the other control points are guaranteed not to collide with the obstacle which means the manipulator can avoid it from beginning to end successfully. Fig.7 (a) shows the situation that the goal changes into (0.5,1.5) and the obstacle located at (1.5,1). By contrast, (b) only changes the position of the obstacle. The result tells that the manipulator can also avoid the obstacle without collision. Fig.8 describes the simulation when there are two obstacles in the scene. It is based on the last experiment as we can see that it only adds an obstacle at (1.5,0.5) compared with Fig.7 (b). But we can see visually the difference between the two trajectories. This illustrates the extra obstacle has a visual impact on the manipulator. All of the experiments complete the task as we expect.



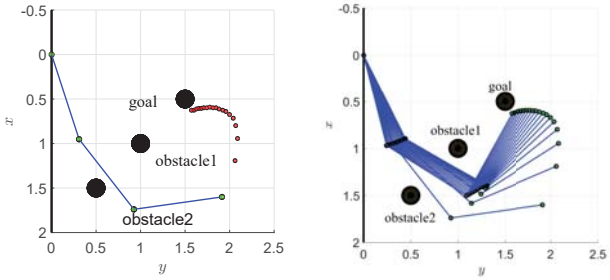
(a) (b)

Fig. 6. The trajectory of end effector when the position of the goal is (0,1.5) and that of the obstacle is (0.5,1). The trajectory of the whole arm when the position of the goal is (0,1.5) and that of the obstacle is (0.5,1).



(a) (b)

Fig. 7. (a) The trajectories of end effector when the position of the obstacle is (1.5,1). (b) The trajectories of arm when the obstacle is at (1,1). The goal position of the above two is the same at (0.5,1.5).



(a) (b)

Fig. 8. The trajectories of end effector and the whole arm when the position of the goal is (0.5,1). There are two obstacles in this experiment, the first one is located at (1,1), and the other one is (1.5,0.5).

V. Conclusion and Future Work

The artificial potential field method has good real-time performance on the multi-DoF manipulator and can be combined with control algorithms. It offers a relatively fast and effective way to solve for safe paths around obstacles which can potentially be implemented in real time with proximity sensors. The correctness and superiority of the algorithm are verified by simulation results. Once we get the information of the obstacle surface point, we could ignore the specific shape of the obstacle to achieve real-time obstacle avoidance. The method is less complex and multiple obstacles are allowed. And we just need to update the potential field rapidly so that it can be applied to dynamic obstacles. However, it is still neither complete nor optimal, so it is very likely that the problem will encounter a situation that cannot be solved.

In the future work, we can add dynamic obstacles in workspace and rapidly update the potential functions in real-time. Multiple obstacles can exist and each one generates a repulsive potential, jointly affect the points on manipulator. In the engineering, we consider to combine obstacle avoidance algorithms with vision to make it more widely used.

References

[1] J. T. Schwartz and C. K. Yap, *Advances in Robotics: Algorithmic and Geometric Apsects of Robotics*, 1986.

[2] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, 2005.

[3] S. Wang and L. Zhu, "Motion planning method for obstacle avoidance of 6-dof manipulator based on improved a* algorithm," *Journal of Donghua University(English Edition)*, vol. 32, no. 1, pp. 79–85, 2015.

[4] S. M. LaValle, J. J. Kuffner, and Jr., "Rapidly-exploring random trees: Progress and prospects," 2000.

[5] J.-J. Park, J.-H. Kim, and J.-B. Song, "Path planning for a robot manipulator based on probabilistic roadmap and reinforcement learning," *International Journal of Control Automation and Systems*, vol. 5, 12 2007.

[6] A. G. Barto and R. S. Sutton, "Reinforcement learning: An introduction," 1998.

[7] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[8] A. R. Khoogar and J. K. Parker, "Obstacle avoidance of redundant manipulators using genetic algorithms," in *Southeastcon 91, IEEE*, 1991.

[9] D. Zhou, L. Wang, and Q. Zhang, "Obstacle avoidance planning of space manipulator end-effector based on improved ant colony algorithm," *Springerplus*, vol. 5, no. 1, p. 509, 2016.

[10] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952–964, 2017.

[11] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Rob. Res.*, vol. 5, no. 1, pp. 90–98, Apr. 1986.

[12] J. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, June 1992.

[13] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, no. 4, pp. 433–454, 2012, the final publication is available at www.springerlink.com.

[14] L. Huber, A. Billard, and J. Slotine, "Avoidance of convex and concave obstacles with convergence ensured through contraction," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1462–1469, 2019.