Robot Path Planning Method Based on Level Set

Ting Wang¹ and Jianghua Sui²

¹Applied Technology College, Dalian Ocean University, Dalian, China
²School of Navigation and Naval Architecture, Dalian Ocean University, Dalian, China
Email: wangwt_1978@163.com; sjh@dlou.edu.cn

Abstract. For the problem of robot path planning under the action of external force field, the directionality of outward force can promote and hinder the movement of the robot. The continuous path planning method can break through the traditional robot path planning algorithm based on grid search Limit, effectively utilize the external force field existing in the space, and improve the numerical implementation method to ensure the feasibility of the planned path in the strong external force field. Based on the Level Set method, the path planning problem is defined. The particle tracking of the robot is transformed into the numerical evolution of the curve. The optimal time path is obtained by solving the Hamilton-Jacques equation. The numerical calculation accuracy of the improved backward path tracking is guaranteed. The feasibility of planning the path. The simulation results show that the algorithm can effectively deal with a variety of complex external force fields and maintain the path in a strong external force field.

Keywords. Directional external force field; speed field; path planning; time optimal; level set; numerical propagation; robotic; numerical computation.

1. Introduction

The path planning problem of robots has always been a research hotspot. Traditional ground robot path planning focuses on the analysis of obstacles or hazardous areas [1-3] to ensure the safety of the robot while minimizing the path. In some new applications, such as lightweight rotorcraft in the wind field, or underwater robots in the flow field, the behavior of the robot is obviously affected by the outward force in the space. This outward force can promote or hinder the behavior of the robot. Therefore, in the path planning, the robot should also consider the use of external forces to optimize an indicator of the robot’s completion of tasks, such as the shortest time and the lowest energy consumption. Alvarez et al. [4] used dynamic programming method and used energy consumption as the optimization criterion to obtain the optimal path of AUV in different spatial scale flow fields. Garau et al. [5] used the A* algorithm to obtain the most energy-efficient AUV path. Kruger et al. [6] used task completion time as another optimization indicator and designed a path planning method to enable robots to make full use of external forces. Rhoads et al. [7] indirectly solved the extreme value curve field by indirectly solving the Hamilton-Jacobi-Bellman equation and using the feedback control rate of the Euler-Lagrange equation (with boundary points). Finally, the time optimal path of the robot is obtained. Tsitsiklis et al. [8] used the first order fast marching algorithm to solve the trajectory optimization problem by solving the discrete Hamilton-Jacobi partial differential equation in the stable field.
Petres et al. [9] proposed a continuous path planning method based on fast marching algorithm. The method defines an anisotropic cost function, in which there is an outward force field in the environment as a constraint for the optimization problem. This method is a breakthrough of the robot based grid point-based path planning method under the action of the external force field. However, this method still has a big problem, mainly because it only applies to the linear cost function, which leads to the planning of infeasible paths in the excessive external force field. Refs. [10-11] and so on did some improvement work on this problem. In Ref. [12], the problem of underwater robot path planning under the influence of flow field in marine environment is proposed, which breaks through the limitation of path planning problem based on grid search, and theoretically guarantees the feasibility of path in strong current. However, due to the discretization of numerical calculations in computers, it is subject to numerical methods. In practice, the feasibility of the path in the strong current is not guaranteed.

This article has the following two contributions. First, by redefining the problem, the situation of the current is extended to an external force, which makes the method broader. The second is to improve the numerical method to make the path feasible more highly under the action of strong external force. The first part first describes the problem to be solved and introduces the level set method. The second part will explain the path planning problem based on the level set method under the influence of external forces in combination with the problems we describe. The third part puts forward the implementation of our algorithm and algorithm. The fourth part gives the simulation results. In the fifth part, we will summarize the work of this paper.

2. Robot Path Planning under Directional Force

2.1. Path Planning Problem Description

Given a configuration space \( \Omega = \{ \Omega_x, \Omega_y \} \), the point in space is defined as \( X = \{x, y\} \). There is a velocity field \( V_f = \{V_{fx}, V_{fy}\} \) caused by the outward force in the space. The path planning method should find a path between the starting point \( X_s \) and the target point \( X_f \). It is defined that the path \( \gamma \) satisfies the differential equation \( \frac{dT}{d\tau} \), where \( u \) determines how the shape of the path \( \gamma \) will change. In the path planning problem of the robot, \( u \) can be regarded as the control rate of the robot. The purpose of the method is to find the optimal control rate \( u^* \) that can form the optimal path \( \gamma^* \).

Defining the single-step movement cost function of the robot under the action of outward force is \( L(X,u) \geq 0 \), then the total cost of moving along path \( \gamma \) is \( J(\gamma) = \int_0^T L(X(\tau), u(\tau))d\tau \). \( J(\gamma) \) is a functional function, because \( J(\gamma) \) is a function of \( \gamma \), and \( \gamma \) is a function of \( t \). The optimal path can be defined as \( \gamma^* = \arg \min_{\gamma} J(\gamma) \), and the optimal total cost function can be written as

\[
J(\gamma^*) = \min_{\gamma} \left\{ \int_0^T L(X(\tau), u(\tau))d\tau \right\} = \int_0^T L(X^*(\tau), u^*(\tau))d\tau
\]

In robotics, a basic path planning problem, also known as the “Zermelo’s navigation problem”, can be described as finding a path from the starting point to the target point in the shortest time, that is, the time optimal path. Assuming \( L(X,u)=1 \), then \( J(\gamma) = \int_0^T d\tau = t \), that is the time the robot is driving.

The motion of the robot in the directional velocity field can be described as \( \frac{dX}{dt} = V_n u + V_f(X,t) \), where \( V_n \) is the speed of the robot, where \( u \) is a unit vector, i.e., \( \| u \| = 1 \), \( V_f(X,t) \) is the velocity field at point \( X \) at time \( t \).
The model does not consider the specific dynamics of the robot. In the application scenario where the robot’s motion space is much larger than the robot’s size, this protocol is completely reasonable. Planning on the higher level of the robot body has very important practical significance. This article will also expound the problem at this level.

2.2. Level Set Method

The idea of the level set method originated from the curve evolution theory, but it overcomes the shortcomings of the curve evolution theory and has a wider scope of application. The curve evolution problem can be described as a process in which a smooth closed curve in a two-dimensional European space $\mathbb{R}^2$ moves at a certain speed along its normal direction to form a cluster of curves with time as a variable. Suppose $C = C(p)$ is a smooth closed curve, $p$ is an arbitrary parameterized variable, let $k$ denote the curvature, $T$ denote a tangent, and $N$ denotes a normal, then the following relationship exists:

$$\frac{\partial C}{\partial t} = \alpha T + \beta N.$$  

Where $\alpha$ and $\beta$ are the components of the velocity function in the tangential direction and the normal direction, respectively. Since the motion of the curve in the tangential direction only affects the parameterization of the curve, it does not change the shape and geometric properties of the curve. According to the curve evolution theorem mechanism, there is always a corresponding $(0, \beta)$ for any velocity function $(\alpha, \beta)$ to ensure that the final evolution result on the curve is the same, so in the specific implementation, only the motion of the curve evolution on the normal can be considered. As shown in figure 1. Let $N$ be the unit normal vector, then the curve evolution equation can be rewritten as:

$$\frac{\partial C}{\partial t} = HN$$  

(1)

Figure 1. The evolution diagram of the curve which propagates along the normal direction.

$H$ in equation (1) is a speed function. After a period of evolution, the curve may undergo topology changes, such as breaks or merges. Curve evolution generally involves parameterization of curves, and the biggest disadvantage of parameterization of curves is that it is not easy to calculate the geometric parameters of curves such as curvature and normal vector, and it is difficult to deal with the topological changes of curves. The level set method is a numerical method for interface tracking and shape modeling. The biggest advantage is that the evolution of the curve (surface) can be numerically calculated in a Cartesian grid without having to calculate the curve (surface). An important theoretical premise of the level set method is the concept of implicit function. The purpose of introducing the level set in the curve evolution theory is to provide an implicit expression for the curve, thus avoiding a series of problems caused by the explicit expression of parameterization. The main idea is to embed the surface of the moving deformation into a high-dimensional function, that is, the change of the topological structure of the curve is expressed as the change of the intersection of a continuously changing surface and a fixed plane (such as a plane with a height of zero). The surface itself can be changed without topology, which makes the complex curve motion process become the evolution
process of high-dimensional functions. For a higher dimensional surface $\phi$, the selection should be as simple as possible. If there is a one-to-one correspondence between the surface and a simple function, we will of course choose this simple function to replace this “high dimensional surface”. In the level set problem, $\phi$ is usually chosen as the “distance symbol function”, assuming that the curve to be tracked (i.e., the zero level set) is $\partial \omega$, and the distance function $d(x)$ is the shortest distance from the periphery of the curve or a point to the curve inside, in the computer calculation process. In the middle, $\partial \omega$ itself is composed of a series of discrete points, so the mathematical expression of $d(x)$ is as follows: $d(x)=\min_{\xi} |x-\xi|$, where $\xi$ is all the points that make up $\partial \omega$. The symbol distance function is defined as follows:

$$\phi(x)=\begin{cases} d(x), & \text{if } x \text{ is in } \partial \omega \\ -d(x), & \text{if } x \text{ is outside } \partial \omega \end{cases}$$ (2)

For the $\xi$ on the $\partial \omega$, there is a $\phi(\xi)=0$. Obviously, there are many advantages to choosing this “symbolic distance function” as the “high-dimensional surface”. First of all, it is very smooth, and it can avoid the steep gradient on the contour, which will make the calculation later easier. The basic idea of the level set method is to continuously evolve $\phi(x,t)$ over time. Its zero level set $\phi=0$ corresponds to the curve to be tracked. $\phi(x)$ is time-dependent, so it is also written as $\phi(x,t)$. Because it contains the closed curve $C$, it can also be written as $\phi(C(t),t)$, abbreviated as $\phi$.

For the $t$ moment, the evolved curve $C(t)$ and the zero level set $\phi(C(t),t)=0$ are equal, and they are derived from time:

$$\frac{\partial \phi}{\partial t} + \nabla \phi \cdot \frac{\partial C}{\partial t} = 0$$ (3)

In differential geometry, the inner unit normal vector and gradient on a curve or surface have the following relationship:

$$N = -\frac{\nabla \phi}{|\nabla \phi|}$$ (4)

It can be seen that the gradient is in the same direction as the normal, and equations (3) and (4) are connected together to obtain:

$$\frac{\partial \phi}{\partial t} = -\nabla \phi \cdot N = -H|\nabla \phi|$$ (5)

3. Path Set Method Based on Level Set

After Considering the path $\gamma$ and $\gamma(t)=X$, the change of $\phi$ along $\gamma$ satisfies.

$$\frac{d}{dt} \phi(x,t) = \frac{\partial \phi}{\partial t} + \nabla \phi \cdot X$$, and for the zero level set, i.e., $\frac{d}{dt} \phi(X,t) = 0$, there is $\frac{\partial \phi}{\partial t} + \nabla \phi \cdot X = 0$. Among them, $X$ is the motion speed of the robot itself, which is consistent with the direction of the curve expansion normal. Combining equation (4), there is $X = v_n \cdot N + V_f(X,t)$, and the upper formula is $\phi$, then the following Hamilton-Jacques equation is satisfied.

$$\frac{\partial \phi(x,t)}{\partial t} + V_n |\nabla \phi(x,t)| + V_f \cdot \nabla \phi(x,t) = 0$$ (6)
Assume that the zero level set is a very small closed curve at the initial moment, that is, the area enclosed by the curve approaches zero, but the normal presence at each point on the curve can be guaranteed. In combination with definition (4), equation (6) has the following initial conditions:

\[
\phi(x, t=0) = \|x\|_2
\]

Here \(\|\cdot\|_2\) represents the second type of norm, the Euclidean distance.

At the \(t = 0\) moment, the zero level set \(\phi(x, t)=0\) is the starting point of the robot, and then the curve evolves with time from the starting point according to equation (6). For any point \(y\), the time when the zero-level set curve first meets \(T(y)\) is \(\phi(y, T(y))=0\). When \(\phi(d, T(d))=0\) or \(T(y)>t\) (evolution timeout), the evolution process ends.

Since the zero level set at the \(T(y)\) moment is derived from the \(t = 0\) moment, if the position of the glider at the \(T(y)\) moment is regarded as coincident with the \(y\), the position of the glider at the previous moment can be obtained. According to this idea, the position of the glider at the time of \(t = 0\) can be reversed. The reverse path reversal process satisfies:

\[
\frac{dx}{dt} = \nabla \phi(x(t), t) \left. \frac{\phi(x, t)}{\|\phi(x, t)\|} \right|_{t=t_0}
\]

By connecting all of these reversed points, you can get the path \(\gamma\) and you can see that \(T(y)\) is just the time from the start to the end.

4. Numerical Implementation of the Algorithm
According to the previous section, the algorithm is divided into two parts, namely the forward zero level set evolution and the backward path tracking.

4.1. Forward Zero Level Set Evolution
First, four differential operators are defined, which are forward and backward differences in the x and y directions, respectively, which correspond to partial derivatives in the continuous x and y directions.

\[
\phi^+_x = \frac{1}{\Delta h}(\phi_{i+1,j} - \phi_{i,j}), \quad \phi^+_y = \frac{1}{\Delta h}(\phi_{i,j+1} - \phi_{i,j})
\]

\[
\phi^-_x = \frac{1}{\Delta h}(\phi_{i,j} - \phi_{i-1,j}), \quad \phi^-_y = \frac{1}{\Delta h}(\phi_{i,j} - \phi_{i,j-1})
\]

According to equation (9), the following two gradients are defined:

\[
\nabla^+ = [\max(\phi^+_x, 0) + \min(\phi^+_x, 0) + \max(\phi^+_y, 0) + \min(\phi^+_y, 0)]^{1/2}
\]

\[
\nabla^- = [\max(\phi^-_x, 0) + \min(\phi^-_x, 0) + \max(\phi^-_y, 0) + \min(\phi^-_y, 0)]^{1/2}
\]

Substituting equations (9) and (10) into equation (8) yields:

\[
\frac{\phi^{+\Delta t}_{i,j} - \phi^{-\Delta t}_{i,j}}{\Delta t} = -(\max(V_{x_{i,j}}^{+\Delta t}, 0)\nabla^+ + \min(V_{x_{i,j}}^{-\Delta t}, 0)\nabla^-)
\]

\[
- (\max(u^{+\Delta t}_{y_{i,j}}, 0)\phi^+_x + \min(u^{+\Delta t}_{y_{i,j}}, 0)\phi^+_y)
\]

\[
- (\max(v^{+\Delta t}_{y_{i,j}}, 0)\phi^-_x + \min(v^{+\Delta t}_{y_{i,j}}, 0)\phi^-_y)
\]
Through the evolution mechanism of equation (11), the level set function can evolve continuously. Since the finite difference method is used in the numerical calculation, it is necessary to pay attention to the selection of the time step $\Delta t$. Under the condition of the space grid spacing $\Delta h$, the CFL condition must be satisfied:

$$H:\Delta t \leq \Delta h$$

(12)

Set the target point $x_T$. When the $\phi(x_T)$ value at the target point is not greater than zero, it can be judged that the zero level set has reached the target point. At this time, the time required for path planning can be obtained as $T_{s_f} = \sum_{i=1}^{N} \Delta t_i$.

4.2. Backward Path Tracking

In the forward evolution of the zero level set, we have proposed and saved the zero level set curve at each moment. At $T_{s_f}$ time, the robot should be below the target point $x_f$, and for discrete reasons, this lower part is not necessarily directly below. The target point $x_f$ corresponds to the projection $\hat{x}_f$ of the zero level set curve at time $T_{s_f}$. From this point on, we start to inversely calculate the path and discretize (8) to obtain:

$$\frac{x(t-\Delta t)-x(t)}{\Delta t} = -V(x,t) - V_n \hat{\eta}$$

(13)

5. Simulation Results and Analysis

In the simulation experiment of the robot path planning problem under the action of external force, the space is configured on the two-dimensional plane, and the two coordinate axes $x$ and $y$ of the space are between $[-1, 1]$. For numerical calculation, the $x$ and $y$ directions are equally divided into $m \times n$ grids, indexed by $i$ and $j$ respectively. The external force velocity field $V_i$ in the space takes values on these grids, and the external force velocity on the $(i, j)$ grid point is $V_{ij}(i, j)$. In the following simulation experiment, the speed $V_e$ of the robot is kept constant and set to $V_e = 0.28m/s$ in the simulation.

In order to verify the basic functions of the algorithm and the accuracy of the numerical method, there is no external force in the space in figure 2, and the robot only evolves at its own speed. The black closed curve is the zero level set in each step of evolution. The simulation results show that the basic functions of the algorithm have been implemented, and the accuracy of the numerical algorithm can guarantee the optimal path generation.

In figure 3, there is a constant external force velocity field $V_e = 0.2m/s$ in the space. It can be found that under the influence of external force, the robot moves to the left to slow down and the right to accelerate. The curve on the curve shows that the curve to the left of the origin of the unit time is squeezed, and the curve on the right side is stretched.

In figure 4, there are two opposite velocity fields $V_e = 0.2m/s$ in the space. The starting point and the target point of the robot are distributed on both sides of the velocity field, and the starting point and the target point are consistent in the $Sy$ direction. The simulation results show that the path planning algorithm can effectively use external forces to accelerate its own motion, and the algorithm can still maintain the original accuracy in the complex flow field.

The path planning results of the bending external force field are shown in figure 5, a common bending external force field, such as the eddy existing in the atmosphere and the ocean. The simulation results show that under the bending external force field, the path can effectively utilize the external
force field, and the generated path breaks through the traditional way of node picking and forms a continuous smooth curve path.

In figure 6, there is a band-shaped external force field in the space. The curves of different colors in the figure represent the path of the robot under different external force fields. The color from light to dark indicates that the velocity field is small to large. It can be seen that in the smaller velocity field, the angle between the path passing through the velocity field and the x-axis is larger, indicating that the robot can more easily cross the velocity field. In the larger velocity field, the angle becomes smaller, indicating that the range in which the robot can move is smaller in a larger velocity field. At the same time, the algorithm can accurately determine the position of the entering velocity field to determine that the planned path is still feasible under the action of strong external force.

**Figure 2.** Path planning for robots in a free-forced space.

**Figure 3.** Path planning for robots in a constant forced space.

**Figure 4.** Path planning for robots in a space wherein exists double force stripes with inversive directions with each other.

**Figure 5.** Path planning for robots in a bend-like force space.
6. Conclusion
This paper studies the path planning problem of robots under the action of external force field. Firstly, the mathematical model of path planning is given. Based on the level set method, the Hamilton-Jacques equation of continuous path planning is derived. By solving the equation, the time optimal path is obtained. By improving the second-order upwind numerical calculation method, the numerical accuracy of the algorithm is improved. This ensures the continuity of the planning path and breaks through the grid-based search method of the traditional path planning method, while ensuring the feasibility of the path in the strong external force field.

References
[1] Carroll K P, et al. 1996 AUV path planning: An A* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology (IEEE) pp 79-84.
[2] Vasudevan C and Ganesan K 1996 Case-based path planning for autonomous underwater vehicles Autonomous Robots 3 79-89.
[3] Warren C W 1990 A technique for autonomous underwater vehicle route planning IEEE Journal of Oceanic Engineering 15 201-205.
[4] Alvarez A, Caiti A and Onken R 2004 Evolutionary path planning for autonomous underwater vehicles in a variable ocean IEEE Journal of Oceanic Engineering 29 418-429.
[5] Garau B, Alvarez A and Oliver G 2005 Path planning of autonomous underwater vehicles in current fields with complex spatial variability: An A* approach Proceedings of the 2005 IEEE International Conference on Robotics and Automation pp 194-198.
[6] Kruger D, Stolkin R, Blum A and Briganti J 2007 Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments Proceedings of the IEEE Int. Conf. on Robotics and Automation pp 4265-4270.
[7] Rhoads B, I Mezic and Poje A 2010 Minimum time feedback control of autonomous underwater vehicles Proceedings of the 49th IEEE Conference on Decision and Control, CDC 2010 pp 15-17.
[8] Tsitsiklis J N 1995 Efficient algorithms for globally optimal trajectories IEEE Trans. Autom. Control 40 1528-1538.
[9] Petres C, Pailhas Y, Patron P, Petillot Y, Evans J and Lane D 2007 Path planning for autonomous underwater vehicles IEEE Trans. Robot 331-341.
[10] Soulignac M, Taillibert P and Rueher M 2009 Time-minimal path planning in dynamic current fields IEEE International Conference on Robotics and Automation 2473-2479.
[11] Soulignac M 2011 Feasible and optimal path planning in strong current fields *IEEE Trans. Robot* 27 89-98.

[12] Lolla T, Ueckermann M P, Yi˘git K, Haley Jr. P J and Lermusiaux P F J 2012 Path planning in time dependent flow fields using level set methods *2012 IEEE International Conference on Robotics and Automation*. 