Fast Marching-Based Path Generating Algorithm in Anisotropic Environment With Perturbations

JINSHAN XU, YONGRENG CHEN, XI YANG, AND MEIYU ZHANG
College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China
Corresponding author: Meiyu Zhang (zmy@zjut.edu.cn)

ABSTRACT Fast Marching is a widely used method in path planning, especially continuity is demanded due to kinodynamic constraints of automatic vehicles. However, its application in real environment with obstacles and perturbations requires simplifications to be made, resulting in non-optimal or even unfeasible path. We introduce a risk function \( \zeta \) to describe the effect of obstacles to the safety of a generated path. Combined with the physical law based description of external flow perturbation, we formulate the path planning problem in realistic environment as a wave front propagation in terms of an extended Eikonal equation. Inspired by the underlying mechanism of the numerical recipe of Fast Marching algorithm, we transform the equation into its canonical form based on the physical rules behind. This enables solutions of the equation without any simplification nor introduction of additional complexity. Comparing to the classical fast marching based method in anisotropic environment, numerical simulations show that the proposed method is more efficient and robust to external flow perturbations, especially in the case of weak or strong external flow environment.

INDEX TERMS Anisotropic environment, fast marching, path planning.

I. INTRODUCTION

With the recent development of technologies, applications of unmanned robotics are increasing. For example, autonomous underwater vehicles are widely used in oil and gas exploration, ocean floor mapping, search and rescue, etc. To accomplish these tasks, path planning plays an important role [1]. It figures out a feasible path that can be followed by the autonomous vehicle from one position to another without the assistance of human operation. However, due to the limited power supply and the requirement of long endurance as well as the uncertainties of its surrounding environment, path planning is not an easy task [2], which aims at finding a best path in terms of minimum cost and collision free. Although it has been studied for decades, it is still one of the hot research topics in mobile robotics [3]–[5].

In the past, the path planning problem was typically associated with the shortest length. As a result, methods in graph theory were widely used [6]–[8]. These methods share common contributes: the terrain where robotic vehicles plan to traverse is explored by a set of randomly sampled nodes and the cost for traveling between them described as a function of the distance between nodes [9]. As lack of global information of the terrain, path feasibility can not be guaranteed. Further more, the generated paths are typically noncontinuous and might not be feasible for robotics with kinematic and dynamic constraints [10]. In contrast, deterministic path planning approaches divide the whole terrain with grids. Each grid is associated with a cost which describes the distance/risks of traversing it. After computing cost over the entire terrain, applying gradient descent on the cost map to trace back from the goal point leads to a good approximation of the optimal path to the start point.

The artificial potential field path planning method is of this type [11]. Its main idea is to introduce an artificial field that attracts the robot to the goal point and repels it from obstacles [12]. However, it suffers from local minimum and does not take into account the influences from environment (like flows and wind) [13]. The fast marching algorithm introduced by Sethian is another alternative deterministic path
planning method [14]. Initially, this method uses a symbolic wavefront expansion to calculate the shortest time path and also determines the departure time of the robot from a starting point. Thereafter, modifications were introduced to cope with influences from environments: in 2007 Pétrès and co-authors introduced an effective cost function to describe the effect of external flows and gave a feasible path that takes advantages of perturbation flows in anisotropic environment [15]. Although this method shares same order of complexity of the original fast marching method, it generates infeasible path under strong environmental perturbations.

This paper targets the difficulties in applying fast marching to path planning problems in anisotropic environment with strong perturbations. The main contributions of the paper are: 1) we detail a mathematical description of a path planning problem in complex environment with both obstacles and external field perturbations based on the general fast marching principle, and derive an easy-to-implement numerical receipt to find optimal path; 2) we explore parameters that govern the geometrical properties of the generated path, which can be incorporated in the path planning process to overcome kinematic constraints of robots and other unmanned automatic vehicles. To serve the purpose of better description of the proposed method, the rest of paper is organized as following: in Section II, we give a definition of path planning problem which we are going to tackle. It follows an introduction of the general fast marching method in Section III. Section IV presents the way how we cope with obstacles and external field perturbation according to the first-arrival principle. We then give numerical experiments to show the effectiveness of the propose method in Section V and conclude the paper in Section VI.

II. PROBLEM STATEMENTS
Given a configuration space $\Omega$, the path planning problem is to find a curve (defined in Eq. (1)) that minimizes a set of internal and external constraints (time, fuel consumption or danger etc).

$$ C : [0, 1] \rightarrow C_{\text{free}}, $$

$$ s \mapsto C(s) = (x(s), y(s)) $$

where $s$ is a parameterization variable describing the path in the free space. Introducing the cost function $\tau$ to represent all the constraints in the configuration space, the index of feasibility $\rho$ of a path can be described as:

$$ \rho = \int_0^1 \tau(C(x_0,x'))ds $$

where $C(x_0,x')$ is the path in the configuration space that connects the starting point $x_0$ and the target point $x'$. $\tau$ is a non-negative function, representing the difficulties of reaching $x'$. For example, an obstacle in the configuration space is a region with $\tau \sim \infty$; an automatic vehicle moving upstream to $x'$ will have a larger $\tau$ comparing to that moving downstream. With these definitions, the path planning can be further formulated into a minimal value problem as:

$$ u(x_0, x) = \inf_{C(x_0,x')} \rho(x_0, x) $$

$$ u(x_0, x_0) = 0 $$

(3)

where $u(x_0, x)$ stands for the minimal cost along a path $C$ starting from point $x_0$ to the destination $x$ in configuration space $\Omega$. As $\tau$ is a non-negative function, there will be only one local minimum which is the starting point $x_0$. However, due to the fact that $\tau$ is position dependent, solving Eq. (3) is not an easy task.

III. FAST MARCHING PRINCIPLE
It has been proven that the minimization problem defined in Eq. (3) is equivalent to solution of the Eikonal equation [15]:

$$ \| \nabla u \| \cdot F(x, y, n) = 1 $$

(4)

where $F(x, y, n)$ is the propagating velocity of the level sets of $u$, and $n = \nabla u / \| \nabla u \|$ the outwards unit normal vector. For the purpose of easy understanding while without losing generality, we view the cost function $u$ as the first arrival time of an automatic vehicle or robot (other merits can be constructed based on robot’s dynamical and kinetic properties, see below for explanation). The solution of Eq.(4) gives the smallest time required to reach any point $x$ from the starting point $x_0$. The optimal path can then be found by a simple backtracking on the cost map starting from the goal point.

Exploiting the entire configuration space to build a cost map is not an easy task. One way is through the Huygens principle construction. The solution to Eq. (4) is developed by imagining wavefront propagating from the boundary with a speed $F$. Since each point on the wavefront can be seen as a wave source, which initiates outward information propagation, the point used to update a nearby neighbor is determined according to the notion of an entropy condition, i.e., a point is updated, it keeps unchanged no matter what is the information coming later [14].

Based on these observations, Sethian proposed to use the upwind approximation for the gradient [14]. At each point in the configuration space $(i, j)$, the unknown $u$ satisfies:

$$ \| \nabla u \| = \frac{1}{F_{i,j}} = (\max(D^{-x}u, D^{+x}u, 0)^2 + \max(D^{-y}u, D^{+y}u, 0)^2)^{1/2} $$

(5)

where $D^{-}$ and $D^{+}$ are standard finite difference notation:

$$ D^{-x}u = \frac{u_{i,j} - u_{i-1,j}}{h} \quad D^{+y}u = \frac{u_{i+1,j} - u_{i,j}}{h} $$

$$ D^{-y}u = \frac{u_{i,j} - u_{i,j-1}}{h} \quad D^{+x}u = \frac{u_{i,j+1} - u_{i,j}}{h} $$

(6)

where $h$ is the grid size. Considering the fact that the first arrival time $u$ can only grow, an order in selecting the grid points to update is introduced. Algorithm 1 details the implementation of upwind scheme. This Fast-Marching algorithm is an efficient way to approximate the viscosity solution of the Eikonal equation for every position [14], without introducing additional complexity [16]. To proceed, three different queues
we introduce the risk map that gives maximum possible speed of the propagating front. As shown in Fig. 2, according to Huygens principle, any points \((O', O', \ldots)\) on the wavefront \(\mathcal{C}(A, B)\) can be the potential point source that updates \(x\). Taking \(O'\) for example, due to the influence of the external current \(v_e\), the vehicle has to move with speed \(v_e\) in order to give a resultant velocity \(v' = (x, n)\) toward \(X\), which is required to update \(u(x)\). However, the entropy requirement fixes the source to be \(O\), at which point maximum \(F(x, n)\) is expected. As a consequence, the difference with respect to the isotropic case is to find \(F(x, n)\) as [16]:

\[
F(x, n) = \max_{v_e \in \mathcal{C}(n \cdot f(v_e, v_e))} 
\]

Substituting Eq. (7) and (8) into Eq. (4) gives us the equation governing the update of \(u\) in complex environment as:

\[
\| \nabla u \| = \frac{\zeta(x)}{\max_{v_e \in \mathcal{C}(n \cdot f(v_e, v_e))}} 
\]

### B. NUMERICAL SCHEMES

Fast marching algorithm was firstly proposed by Sethian to propagate a wave front to an isotropic configuration space. Petres and co-authors [15] extended this method to...
anisotropic case. They introduced a scaling parameter to approximate the influence of external current flow as an additional cost. However, this simplification might result in a non-optimal path or even non-feasible path, depending on the strength of external disturbance (See Fig. 5).

Noticing the difference between Eq. (9) and (4), the real challenge to solve Eq. (9) is to extract the effective nominal velocity Eq. (8). To simplify the derivation, we view an automatic vehicle as a point on the wavefront. It moves forward in any direction with speed $|v_{e}|$. As a consequence, the maximum nominal velocity can be easily determined as:

$$F(x, n) = v_{e} + v_{c} \cdot n$$ (10)

To numerical solve Eq. (9), we partition the configuration space into Cartesian grid and adopt the 4-connexity discrete model. Notice that only dead points are used to estimate $u$ at point $x$. To update point a grid point $(i, j)$, we denote its two dead neighbors in opposite axis as $D_{x}$ and $D_{y}$, with their values as $u_{x}$ and $u_{y}$ respectively. This allows us to define the normal direction of the wavefront $n$ as:

$$n = \frac{\xi_{x}(u_{x} - u_{i})x_{i} + \xi_{y}(u_{y} - u_{j})y_{y}}{\sqrt{(u_{x} - u_{i})^{2} + (u_{y} - u_{j})^{2}}}$$ (11)

where $\xi_{x}$ is introduced to denote the relative position of its dead neighbors with respective $(i, j)$. $\xi_{x} = 1$ when $(i, j)$ is on the left side of its dead neighbor $u_{x}$, while $\xi_{x} = -1$ when $(i, j)$ is on the right side of $u_{x}$ and $\xi_{x} = 0$ when there is no dead neighbors along $x$-axis. The same goes to $\xi_{y}$. Substituting it into Eq. (10) gives

$$F(x, y, n) = v_{o} + v_{e} \cdot n$$

$$= |v_{o}| + \frac{\xi_{x}(u_{x} - u_{i})v_{x}^{i} + \xi_{y}(u_{y} - u_{j})v_{y}^{j}}{\sqrt{(u_{x} - u_{i})^{2} + (u_{y} - u_{j})^{2}}}$$ (12)

where $v_{x}^{i}$ and $v_{y}^{j}$ are the components of $v_{e}$ along $x$- and $y$-axis respectively. Discretizing Eq. (9) gives us the quadrature equation that updates the point $(i, j)$ as

$$\frac{(u - u_{x})^{2}\xi_{x}^{2}}{h^{2}} + \frac{(u - u_{y})^{2}\xi_{y}^{2}}{h^{2}} = \frac{\xi_{x}^{2}}{F_{i,j}}$$ (13)

where $h$ is the grid size. After some simple algebra calculation, we end up with a quadratic equation:

$$Au^{2} + Bu + C = 0$$ (14)

with

$$A = 2v_{0} - D^{2}$$

$$B = -2v_{0}^{2}(u_{x} + u_{y}) + 2D(E + \zeta h)$$

$$C = v_{0}(u_{x} + u_{y}) - \tau^{2}h^{2} - E(2\zeta h)$$

$$D = (\xi_{x}v_{x}^{i} + \xi_{y}v_{y}^{j})^{2}$$

$$E = \xi_{x}^{2}u_{x} + \xi_{y}^{2}u_{y}$$ (15)

Solving Eq. (14, 15) allows us to build an entire cost map in the configuration space. Although these equations are derived from the case of two dead point, it is applicable to the 1-dead neighbor case, simply by setting $\xi_{x} = 0$ or $\xi_{y} = 0$.

C. IMPLEMENTATION DETAILS

The key idea to Fast Marching Methods lies on the observation that the previous iteration contains a very specific order. In the first arrival time problem presented here, a far point should be updated to the possible smallest $u$ from its dead neighbors. As a far point might be surrounded by multiple dead neighbors, to guarantee the entropy condition, a set of trial estimations of the far point using different combination of dead points should be done [15]:

- **One Dead neighbor**

  Under this case, the update of the far point $u(x)$ is directly given by Eq. (14, 15) with $\xi_{x} = 0$ when the dead neighbor is along $x$-axis or $\xi_{y} = 0$ when the dead neighbor is along $y$-axis (see Fig.3(a) for details).

- **Two Dead neighbors**

  If the Dead neighbors are on opposite sides of the far point (Fig.3 (b) case (1)), estimate the far point $u$ twice with each of the Dead neighbors separately. The smaller is then selected as the value of $u$ at this point. If the Dead neighbors are on different directions, three estimations have to be done: estimate the open point $u$ twice each time using one of these two Dead neighbors; do the estimation using both Dead neighbors. After finishing these estimations, the smallest is chosen (Fig.3 (b) case (2)).

- **Three or Four Dead neighbors**

  The method for updating a far point surrounded by three or four Dead is same as for the two Dead points, i.e., estimate $u$ with each Dead neighbors separately, then do the estimation with each couple of two consecutive Dead neighbors. After these trial estimations, choose the smallest.

D. COMPLEXITY ANALYSIS

The proposed method for solving path planning problem in anisotropic environment is derived from the classical Fast Marching method, which is of complexity order of $N \cdot \log N$, with $N$ representing the number of grid points we discretize the configuration space. Comparing to the original fast
marching, the main differences are 1) calculating the risk map of each grid and 2) deriving the maximum nominal velocity $F(x, n)$ at each point. These two additional operations are all of complexity order of $N$. Thus proposed method still keeps the same complexity order of $N \cdot \log N$ as the original fast marching method.

V. SIMULATION RESULTS

Based on the analysis given above, the proposed method copes with the collision risks with obstacles and external flow perturbation in an union form. We would expect that the method generates path that take good advantage of external force while maintains low risk of collision with obstacles. This section provides detailed simulation results to verify these merits.

A. EFFECTIVENESS OF PATH GENERATION IN COMPLEX ENVIRONMENT

To verify the effectiveness of the proposed method, we carried out a set of numerical experiments, in which the configuration space was divided into a $200 \times 200$ lattice; An automatic vehicle initially locating at $S(20, 20)$ plans to move to a target point $T(117, 156)$. Two circles with radius 10 centered at $(50, 120)$ and $(120, 50)$ were added as obstacles in the configuration space (See Fig. 4). Apparently, without the influence of external flow perturbations, one can imagine the optimal path should pass through the region between these two obstacles; if the consideration of least risk of collision is taken into account, the optimal path should not go through any region between these two obstacles. As shown in the figure, the optimal paths generated by applying the proposed method (solid black curves in panel (a) and (b)) exactly match the intuitive analysis results: when a vehicle is not sensitive to risk (panel (a) with $\lambda = 1$), the optimal path goes between the obstacles and when $\lambda = 3$ (panel (b)), the shortest path is no longer the optimal one.

Beside the ability of ensuring safety, the proposed method is also robust in coping with external flow perturbations. As illustrated in panel (c), under the influence of external flow current shown with little arrows, the generated path can take full use of the propelling force to reach its destination in the case of low safety requirement ($\lambda = 1$), although it is not the shortest. When the benefit from taking use of external flow currents is big enough comparing to the risk of collision with obstacles, the generated path goes through the region between the obstacles (see the solid black curve in panel (d)), even though the vehicle is more demanding on safety ($\lambda = 3$).

B. ROBUSTNESS UNDER PERTURBATIONS

Comparing to the method proposed by Petres et al. [15], the main difference comes from an exact description of the influence of external perturbation in propagating the wavefront without introducing any simplification. This implies that the proposed method is applicable to all possible external interfering force, no matter how strong it is. To verify this, we focus on a path planning problem that an automatic vehicle initially locating at $S(100, 120)$ plans to move to a target point $T(90, 180)$. A rectangular obstacle with length $l = 120$ and width $w = 40$ locates in the center of the configuration space (see Fig. 5). The external current flows with strength $v_c$, are constraint in the region labeled by arrows, indicating their directions. To avoid collision with the obstacle, a safe distance $d_0 = 20$ was introduced in Eq. (7). With out losing generality, the vehicle velocity $v_c$ is normalized into 1.

Fig. 5(a) shows the cost map and the resulted optimal path generated using the proposed method in the case of $v_c = 0.6$. Clearly shown in the figure, the cost optimal path (solid curve) takes full use of the external flows by following the current. However, the shortest length path (dashed
curved in Fig. 5(a)) moves counter-flow and passes over the obstacle.

To qualitatively demonstrate the profits brought by the cost optimal path algorithm, we calculated the fuel consumption $E$ along paths under the assumption that a ship consumes same fuel to keep velocity $|v_e|$ as:

$$E \propto \text{Time} = \int_{s} \frac{ds}{v_e(s) + |c|}$$  \hspace{1cm} (16)

where the integration takes along the generated path. Apparently, minimizing fuel consumption can be achieved by reducing path length or taking advantages of external current flow. In the configuration presented in Fig. 5(a), the shortest length path should be a curve above the obstacles (dashed curve in Fig. 5(a)) while the cost optimal path goes through the region below the obstacle (solid curve in Fig. 5(a)). As the cost optimal path generated using the proposed method takes full use of external flow currents ($|c| = 0.6$), although it has longer length, following it to reach the destination requires about 40% less fuel as that required by following the shortest length path. However, when the profits from using external current flows can not compensate the cost of taking longer path (in the case of weak flow currents), the algorithm should take the shortest length as the criteria to plan a feasible path. Consistent with our intuitive requirement, the proposed method plans a path (red solid curve) that passes above the obstacle when the external flow is weak ($|v_e| = 0.2$). These two curves overlap in the case $|v_e| > 0.35$, while big gaps are seen under weak $v_e$. The reason is that method in [15] is insensitive to the strength of external flow currents during path planning, while the proposed method assesses the real benefits that it would bring by using external flows.

The robustness of the proposed method is not limited to the case of $|v_e| \gg |c|$. As mentioned by many of other authors [17], the method in [15] might be incapable of generating feasible path under strong external flow perturbations, we further test the proposed method in these cases. For this purpose, we consider a configuration space given in Fig. 7: a flow of strength $|v_e| = 1$ coming from right to left. Due to the existence of obstacles, the current splits into two parts: one moves directly toward the left, the other first moves upward and comes down after passing the obstacle with a

---

**FIGURE 5.** Countour plot of the cost function $\psi$ in the configuration space and the generated path. An obstacle (the rectangle area) is treated as a region with risk factor $\zeta \sim \infty$, the regions with external flow perturbations are marked with allows, which also indicate the directions of the current flows. (a) Cost optimal path generated using the proposed method with $|c| = 0.6$ is shown as the solid curve. The length optimal path generated by fast marching method without taking the external current into consideration is shown as dashed curved. (b) The cost optimal path generated using the proposed method (red solid curve) and that obtained using the method given in [15] (blue solid curve) under weak external perturbation flow $|c| = 0.2$.

**FIGURE 6.** Comparison between consumed energy $E$ of by following paths generated using the proposed method (red solid curve) and the method in [15] under environmental configuration same as in Fig. 5. Under strong current perturbations, both method give paths of same quality, while under weak current, the proposed method bring more benefits. Inner panel shows the extra energy required for taking the path generated from the method in [15] in comparing to our method.

**FIGURE 7.** Comparison between the proposed method to that in [15] under environment with obstacles and strong maximum external flow current $v_e = 1.0$. Flow currents are labeled by the arrows with their length indicating the strength relative to the max. The cost maps and resulted paths generated from the method in [15] (a) and our method (b).
diminished flow velocity \(|v_c| = 0.5\). An automatic vehicle plans to move from the starting point \(S\) to its destination \(T\). Applying Pétres’ method to the problem gives a path as shown the solid red curve in the panel (a). Although it uses the part of the current flow, to reach the destination it has to pass through an upstream region of \(|v_c| = 1.0\). Clearly this path is infeasible, as the upstream current prevents the vehicle to move. Different from [15], our method calculates the cost map in the configuration space strictly according to the underlying physical rules. As a result, cost values in the region with \(|v_c| > |v_v|\) can not be updated using downstream points. This is the reason why a non-continuous change of cost value seen in the bottom of Fig. 7(b). Backtracking on this cost map gives a path all in the region with \(|v_c| < |v_v|\), where the vehicle can always overcome these obstructive force and reach its destination.

VI. CONCLUSION AND FUTURE WORKS
Fast marching is a widely used method in path planning which discretizes the isometric configure space and builds a cost map upon it. The global optimal path is then generated by a descent back-tracking on the cost map from the goal point to the starting point. However, when extending this method to anisotropic case, planning accuracy as well as feasibility is challenged due to required simplifications in order to solve the Eikonal equation.

To solve path planning problems in realistic environment, where there exists obstacles and external flow perturbation, we introduce the risk function \(\zeta\) to describe the effect of obstacles to the safety of a generated path. We then formulate the influence of external flow current to the propagation of a wavefront in the form of an Eikonal equation. This allows the consideration of safety and efficiency of automatic vehicle during path planning procedure at the same time. As a result, solution to the equation could give a cost map and an optimal feasible path at end. Inspired by the underlying mechanism of numerical recipe of fast marching algorithm, we transform the extended Eikonal equation into the canonical form based on the physical rules behind. It enables us solve the global optimal path generation problem without any simplification nor introduction of additional complexity. Comparing to the classical fast marching based method in anisotropic environment, numerical simulations show that the proposed method is more efficient and robust to external flow perturbations, especially in the case of weak or strong external flow environment.

Comparing to the sampling based path planning method, like RRT [18] or its variations [19], the unique merit of fast marching based algorithm is the continuity of the generated path, which is a direct consequence of the continuity of the cost map in the configuration space. Here we introduce two additional parameters, i.e., safe distance \(d_0\) and scaling parameter \(\lambda\) in the process of the cost map build-up. An open question on how these two parameters effect the maximum curvature of the generated path is untouched. In particular, the feasibility of the generated path to an automatic vehicle with not only kinodynamic constraints but also their own dynamical property, like unmanned cargos, should be considered [20].

REFERENCES
[1] Y. Liu and R. Bucknall, “Path planning algorithm for unmanned surface vehicles in a practical maritime environment,” Ocean Eng., vol. 97, pp. 126–144, Mar. 2015.
[2] N. E. D. Toit and J. W. Burdick, “Robot motion planning in dynamic, uncertain environments,” IEEE Trans. Robot., vol. 28, no. 1, pp. 101–115, Feb. 2012.
[3] M. Liu, “Robotic online path planning on point cloud,” IEEE Trans. Cybern., vol. 46, no. 5, pp. 1217–1228, May 2016.
[4] S. Mahmoudzadeh, D. M. W. Powers, K. Sammut, and A. M. Yazdani, “A novel versatile architecture for autonomous underwater vehicle’s motion planning and task assignment,” Soft Comput., vol. 22, no. 5, pp. 1687–1710, Mar. 2018.
[5] S. Mahmoudzadeh, A. Yazdani, K. Sammut, and D. Powers, “Online path planning for AUV rendezvous in dynamic cluttered undersea environment using evolutionary algorithms,” Appl. Soft Comput., vol. 70, pp. 929–945, Sep. 2018.
[6] P. Bhattacharya and M. Gavrilova, “Roadmap–based path planning—Using the Voronoi diagram for a clearance–based shortest path,” IEEE Robot. Autom. Mag., vol. 15, no. 2, pp. 58–66, Jun. 2008.
[7] J. Nasir, F. Islam, U. Malik, Y. Ayaz, and O. Hasan, “RRT-smart: A rapid convergence implementation of RRT,” Int. J. Adv. Robotic. Syst., vol. 10, no. 29, pp. 1–12, 2013.
[8] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Sep. 2014, pp. 2997–3004.
[9] M. Elbanhawi and M. Simic, “Sampling–based robot motion planning: A review,” IEEE Access, vol. 2, pp. 56–77, 2014.
[10] K. Jolly, R. S. Kumar, and R. Vijayakumar. “A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits,” Robot. Auton. Syst., vol. 57, no. 1, pp. 23–33, Jan. 2009.
[11] N. Leonard and E. Fiorelli, “Virtual leaders, artificial potentials and coordinated control of groups,” in Proc. 40th IEEE Conf. Decis. Control, Jul. 2003, pp. 2968–2973.
[12] S. M. LaValle, Planning Algorithms. Cambridge, U.K.: Cambridge Univ. Press, 2006. [Online]. Available: http://planning.cs.uiuc.edu/
YONGRENG CHEN received the B.Sc. degree in electronic and information engineering from the Communication University of Zhejiang, Hangzhou, China, in 2016. He is currently pursuing the master’s degree in software engineering with the College of Software, Zhejiang University of Technology.

XI YANG received the B.S. degree from Xi’an Jiaotong University, Xi’an, China, in 2004, the M.S. degree from Zhejiang University, Hangzhou, China, in 2007, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2011. Since 2012, he has been with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou. His research interests include nonlinear control and optimization theory with applications in output regulation, multiagent systems, intelligent transportation systems, and image processing.

MEIYU ZHANG received the B.S. and M.S. degrees from Zhejiang University, Hangzhou, China, in 1988 and 1997, respectively. She is currently a Full Professor with the Zhejiang University of Technology. Her research interests include image processing and automations.