Robotica (2022), **40**, pp. 599–610
doi:10.1017/S0263574721000709

CAMBRIDGE
UNIVERSITY PRESS

REPLY

Biharmonic navigation using radial basis functions

Xu-Qian Fan and Wenyong Gong*

Department of Mathematics, Jinan University, Guangzhou, China
*Corresponding author. Email: gongwenyong@jnu.edu.cn

Received: 20 January 2021; **Revised:** 9 May 2021; **Accepted:** 10 May 2021; **First published online:** 17 June 2021

Keywords: Path planning, Radial basis function, Biharmonic function

Abstract

Path planning has been widely investigated by many researchers and engineers for its extensive applications in the real world. In this paper, a biharmonic radial basis potential function (BRBPF) representation is proposed to construct navigation fields in 2D maps with obstacles, and it therefore can guide and design a path joining given start and goal positions with obstacle avoidance. We construct BRBPF by solving a biharmonic equation associated with distance-related boundary conditions using radial basis functions (RBFs). In this way, invalid gradients calculated by finite difference methods in large size grids can be preventable. Furthermore, paths constructed by BRBPF are smoother than paths constructed by harmonic potential functions and other methods, and plenty of experimental results demonstrate that the proposed method is valid and effective.

1. Introduction

With the development of smart robots, collision-free navigation to destinations using path planning methods in complex environments has been widely paid attention by both research and industrial communities. This benefits from the fact that optimal paths designed by path planning algorithms can satisfy various requirements in different scenarios. Therefore, path planning is one of the central components for many real-world applications including autonomous driving [1], smart service robots [2, 3, 4, 5], and unmanned aerial vehicles (UAV) [6, 7].

Path quality criterion determines the final path shapes and cost functions. Path length and path duration are two common quality criteria, for example, the well-known A^* [8] and Dijkstra's algorithm [9] are two classical shortest path solutions. Smoothness [10] is another requirement for path planning since robots steering in a smooth path becomes easy. Moreover, designers can construct more sophisticated cost function to accomplish their tasks. In computer-aided geometric design (CAGD) [11], bounded curvature is a basic constraint in curve and surface reconstruction. However, designing a path with the overall bounded curvature along the path is still an open issue [12]. In this paper, we focus on designing paths with sufficient smoothness, which are more accordant with practical robot steering.

Many literatures have reported smooth path planning methods [8, 10, 13, 14, 15, 16, 17], and harmonic potential field [13, 18] is a representative approach among them. Harmonic functions are very useful for path planning since they possess two important properties: uniqueness of solutions and maximum/minimum principle. The maximum/minimum principle guarantees that a harmonic function achieves its maximal and minimal solutions on the boundaries, and the unique solution can ensure the destination arrival for robots. Furthermore, paths constructed by harmonic potential methods are also C^2 smooth at least. However, the large curvature may occur in paths constructed from harmonic functions. For example, the maximal curvature (see the definition in Experimental Results Section) of the path shown in Fig. 1(a) is $\kappa_m = 7.2789$. Biharmonic functions, on the other hand, satisfy a fourth-order partial differential equation, and they are C^4 at least. Existing path planning methods [12, 19] based on

© The Author(s), 2021. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574721000709 Published online by Cambridge University Press

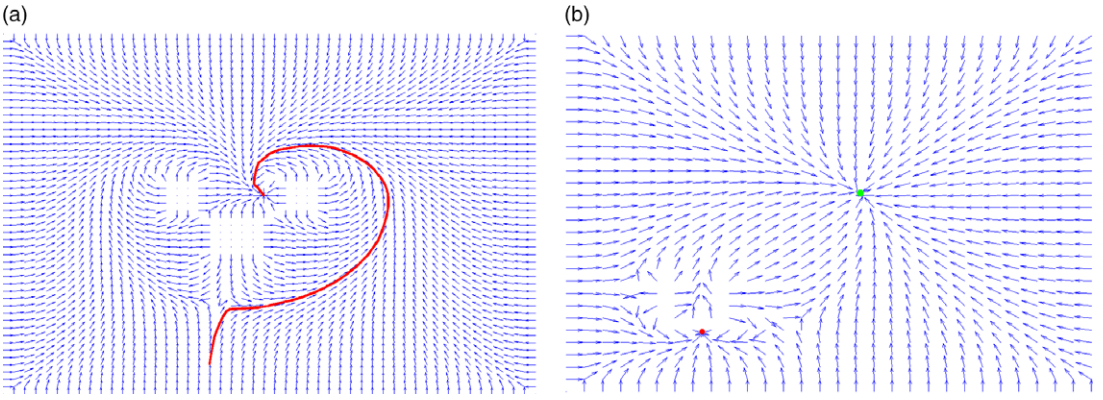


Figure 1. *(a) Path computed using harmonic function with Dirichlet boundary conditions. The path is smooth, but the maximal curvature is* $\kappa_m = 7.2789$. *(b) An invalid gradient field is constructed in a* 300×300 *grid by solving the harmonic equation with central difference method. The red point is the unique real destination or sink, while the green point is a pseudo-sink.*

biharmonic functions have reported satisfactory results. In this paper, we present a biharmonic radial basis potential function to design paths with enough smooth path planning requirement.

Numerical algorithms (e.g., Jacobi iteration, Gauss–Seidel, and SOR) are feasible for solving harmonic and biharmonic equations, but numerical solution depends heavily on the resolution of grids, and the most drawback is that iterative algorithms may fail to find a valid gradient for large grid size due to numerical precision. See Fig. 1(b), for example, the gradient field is constructed in a 300×300 grid by solving the harmonic equation $\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$ using the central difference method, and the iteration number is set to 10,000. The red point is the unique real destination or sink position, but in the example, a green pseudo-sink point is appeared. Radial basis functions (RBFs) [20] are a high-accuracy meshless method to solve partial differential equations, thus RBFs are irrelevant to grid size and can also overcome invalid gradients caused by numerical precision. On the other hand, proper boundary conditions are crucial for concrete PDEs. Therefore, we design distance-related boundary conditions to construct our biharmonic field. In this paper, we propose and use the radial basis functions (RBFs) to approximate biharmonic equations with our proposed boundary conditions.

The main contribution of this paper is threefold. (1) A biharmonic potential field is proposed to design collision-free smooth paths. (2) A biharmonic equation with distance-related boundary condition is presented, which can be used to construct the biharmonic potential field. (3) The RBF approach is used to solve the biharmonic equation, and it can approximate biharmonic functions as well as on large size grids comparing with finite difference methods. The rest of this paper is organized as follows. In the next section we first review some related works, and then preliminary knowledge is revisited in Section 3. The biharmonic equation system with distance-related boundary conditions is presented in Section 4, and the relevant approximate computation and path planning procedure is also introduced in this section. In Section 5, some experimental results are shown, and at last we conclude the whole paper.

2. Related work

Numerous path planning methods have been proposed for given start and goal positions in the past decades, and many of them have been used in real-world products. In this section, we mainly review representative path planning algorithms in the two-dimensional maps, and more references can be found in the survey [21]. Generally, most path planning algorithms fall into one of the following categories.

https://doi.org/10.1017/S0263574721000709 Published online by Cambridge University Press

- **Geometric method** Geometry-based algorithms, also known as the road map algorithms, are classic path planning methods. There are four major algorithms in the group: visibility graph, cell decomposition, Voronoi diagram, and triangulation-based algorithms. Geometric methods work with geometric representations of environments and obstacles, and generate paths by connecting nodes and edges of geometric graphs. In visibility graph methods [22, 23], the start point, the goal point, and the vertices of polygons (obstacles) are connected to form a graph, and edges of the graph has no intersections with obstacles. Therefore, any two points in the graph can be reachable in the graph. Then, graph search methods (e.g., Dijkstra's algorithm) are used to find a path. Comparing to visibility graph methods, edges in Voronoi diagram methods [24] are not necessary to be straight lines. The curved edges in Voronoi diagrams can improve the safety for moving robots. The cell decomposition methods [25] divide the free configuration space into many small regions called cells, and then these cells are viewed as nodes to construct a graph according to neighboring relationships. Triangulation-based pathfinding methods[26, 27, 28] represent a planar environment using Constrained Delaunay Triangulations, and then path is found in the triangulated environment. Geometric path planning methods run very fast in general, but smoothness of paths constructed by the methods is unsatisfactory.
- **Sampling-based method** For a path planner, constructing a graph from sensory data are always a time-consuming task. Instead of a geometric representation of maps, sampling-based planners, such as RRT [29, 30] and its variant RRT*[31] and PRM [32, 33], randomly choose enough points in the free configuration space, and then the steering and collision checking routines are applied to explore reachable paths. In general, this kind of algorithms are available for high-dimensional and complex constraints path planning problems. RRT [29, 30] and PRM [32, 33] are probabilistically complete, but the path generated by them is not optimal (e.g., not to be shortest) and not always unique. In general, approaches fallen into the category are efficient, but they (e.g., RRT) may also become time consuming when maps contain narrow passages.
- **Potential field method** The artificial potential field (APF) [34] is a classical potential field design method, which was first proposed by Khatib [35] in 1986. Robots are pushed away from obstacles in APF by the repulsive field, and they are also attracted to the goal position by the attractive field. APF can generate smooth paths, but robots may not achieve the goal due to local minimizers. Harmonic navigation fields [6, 13, 18] are extensively used in robotic path planning areas. The harmonic function attains its global minimal/maximal value at boundaries, which can guarantee and always find a path to the goal from any locations. Moreover, paths generated by harmonic fields are generally smooth, which tend to be easily steering for robots. However, harmonic function methods have several drawbacks. The computational cost is very extensive comparing to sampling-based path planning methods (e.g., RRT and PRM). Large curvatures may occur in paths generated by following a harmonic navigation field. On the other hand, biharmonic fields (BHF) satisfy a fourth-order PDE, and the paths generated by following BHF shows no large curvature results [12, 19]. In fact, the algorithm proposed in this paper falls into the category.
- **Graph-based search method** In essence, graph-based planning methods (e.g., Dijkstra's and A^* algorithms) are a distance propagation process according to computed distances. Dijkstra's algorithm [9] adopts a brute-force way to calculate the shortest path. On the other hand, A^* algorithm [8] and its variants (e.g., any-angle A^* [36], D^* [37], and D^* lite [38]) introduce a heuristic function to reduce the search scope and improve the efficiency comparing to Dijkstra's algorithm. Graph-based methods can always find a path joining given start and target positions if there at least exists a path. However, the real-time performance of A^* algorithm and its variants on large-scale graphs does not reveal the superiority. However, graph-based search method needs to first construct a graph, and then search algorithms are used to find paths, and it does not demonstrate superiority comparing to other path planning methods.

https://doi.org/10.1017/S0263574721000709 Published online by Cambridge University Press

- **Intelligence planning method** When robots locate in a completely unknown and dynamic environment, learning-based methods [39, 40, 41] can plan a reasonable path via local image and depth information captured by equipped sensors. However, for unseen information that was not covered in sample data, learning-based methods cannot be well dealt with. Various smart optimization methods, such as simulated annealing algorithm [42], genetic algorithm [43], and ant colony algorithm [44], are also used to explore feasible path planning strategies.

In fact, there are some path planning methods that are not covered by our category scheme. For example, combinations of several methods [45, 46] stated above are also widely used in vehicle navigation. In the next section, we give preliminary knowledge of biharmonic functions and radial basis functions.

3. Preliminaries

3.1. Biharmonic functions

In a 2D planar region $\Omega \subset \mathbb{R}^2$, a function $F(x, y)$ is biharmonic if it satisfies

$$\Delta^2 F(x, y) = 0, \tag{1}$$

where Δ is the Laplacian operator, and Eq. (1) is also the well-known biharmonic equation. Biharmonic functions have many graceful properties that can be used for path planning. Rather than a second-order elliptic function as the Laplace equation, the biharmonic equation is a fourth-order elliptic PDE, thus two boundary conditions are required to solve it and attain a unique solution [47]. If every function $f_i(\cdot)$ $(i = 1, 2, \cdots, K)$ is biharmonic, then the sum $\sum_{i=1}^{K} f_i(\cdot)$ of K biharmonic functions is also biharmonic. That is the superposition property.

The first property states that any biharmonic equation has a unique solution up to proper boundary conditions, and the superposition property illustrates that we can compose multiple biharmonic functions to accomplish complex tasks. Unfortunately, the maximum/minimum principle is not satisfied by biharmonic equations. Interestingly, let $g(x, y) = \Delta F(x, y)$, then solving Eq. (1) with proper boundary conditions is equivalent to solve system of equations with corresponding boundary conditions:

$$\begin{cases} \Delta F(x, y) = g(x, y) & in \quad \Omega \\ \Delta g(x, y) = 0 & in \quad \Omega \end{cases} \tag{2}$$

The function $g(x, y)$ is obviously a harmonic function, which can be used to construct a navigation field.

3.2. Radial Basis Functions (RBFs)

RBFs are a powerful approximation (interpolation) tool, which can approximate any multivariable functions by linear combinations of translates of a single radial basis function. One of the greatest advantages of RBFs is that they can be applied to almost an arbitrary dimensional point set due to no distribution requirements for data. Approximation using RBFs does not need to triangulate a given point set, thus RBFs are a meshless method and an arbitrary approximate accuracy can also be gained in principle. For these reasons, RBFs have been widely used to solve PDEs [20] in scientific computing and engineering.

Given m center points $c_i \in \mathbb{R}^2$ $(i = 1, \cdots, m)$, the RBFs approximation scheme for $x \in \mathbb{R}^n$ can be generally expressed as

$$f(x) = \sum_{i=1}^{m} \omega_i \phi(\|x - c_i\|), \tag{3}$$

where $\|\cdot\|$ is the Euclidean distance, ω_i are unknown coefficients that must be determined, and $\phi(\cdot): \mathbb{R}^+ \to \mathbb{R}$ is the RBF kernel. At some cases, polynomial terms are added at the end of Eq. (3) to guarantee the existence of RBF interpolation and approximation.

https://doi.org/10.1017/S0263574721000709 Published online by Cambridge University Press

Clearly, RBF kernel functions are heavily relevant to the approximation quality. Commonly used kernel functions include Gaussian kernel $\phi(r) = \exp(-s^2 r^2)$, linear kernel $\phi(r) = r$, multiquadric kernel $\phi(r) = \sqrt{1 + s^2 r^2}$, and thin plate spline $\phi(r) = r^2 \ln r$, where $r > 0$ and s is the shape parameter.

Since the approximant $f(\cdot)$ is differentiable with respect to parameters ω_i, then these parameters can be estimated by iterative methods or matrix methods of linear least squares.

4. Biharmonic navigation field

In this section, we mainly give the construction method of the biharmonic navigation field using biharmonic radial basis representation. Therefore, the RBF representation and relevant computational procedures are presented here.

4.1. Biharmonic equation with distance-related boundary conditions

Let Ω be an open bounded domain whose boundary is $\partial\Omega$, and $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_K\}$ are K topological disks in Ω whose boundary are $\partial\mathcal{O} = \{\partial\mathcal{O}_1, \partial\mathcal{O}_2, \cdots, \partial\mathcal{O}_K\}$, respectively. We also require $\partial\mathcal{O}_i \neq \emptyset$ and $\partial\mathcal{O}_i \cap \partial\mathcal{O}_j = \emptyset$. In the path planning background, $\mathcal{O}_i$ in fact represents an obstacle region or a goal region, and $\partial\mathcal{O}_i$ is the boundary of $\mathcal{O}_i$.

We consider a function $u(x, y)$ defined in the domain Ω that is governed by a biharmonic equation:

$$\begin{cases} \Delta^2 u(x,y) = 0, & (x,y) \in \Omega \backslash \mathcal{O} \\ \Delta u(x,y) = S, \quad u(x,y) = T & (x,y) \in \partial\mathcal{O}_{V+1} \cup \cdots \cup \partial\mathcal{O}_K \\ \Delta u(x,y) = 0, \quad u(x,y) = \frac{1}{\gamma \cdot d(x,y)} & (x,y) \in \partial\Omega \cup \partial\mathcal{O}_1 \cup \cdots \cup \partial\mathcal{O}_V \end{cases} \tag{4}$$

that is equivalent to a system of equations if we define $g(x, y) = \Delta u(x, y)$:

$$\begin{cases} \Delta g(x,y) = 0, & (x,y) \in \Omega \backslash \mathcal{O} \\ g(x,y) = S, & (x,y) \in \partial\mathcal{O}_{V+1} \cup \cdots \cup \partial\mathcal{O}_K \\ g(x,y) = 0, & (x,y) \in \partial\Omega \cup \partial\mathcal{O}_1 \cup \cdots \cup \partial\mathcal{O}_V \end{cases} \tag{5}$$

and

$$\begin{cases} \Delta u(x,y) = g(x,y), & (x,y) \in \Omega \backslash \mathcal{O} \\ u(x,y) = T, & (x,y) \in \partial\mathcal{O}_{V+1} \cup \cdots \cup \partial\mathcal{O}_K \\ u(x,y) = \frac{1}{\gamma \cdot d(x,y)}, & (x,y) \in \partial\Omega \cup \partial\mathcal{O}_1 \cup \cdots \cup \partial\mathcal{O}_V \end{cases} \tag{6}$$

where S, T, and $\gamma > 0$ are real numbers, and $d(x, y)$ is the Euclidean distance between a point (x, y) and the goal position. Now, we recall a theorem that characterizes the uniqueness and smoothness of biharmonic equations in Eq. (4) under Navier boundary condition.

Theorem 1. *The equation Eq. (4) has a unique* $C^4(\Omega \backslash \mathcal{O}) \cap C(\overline{\Omega \backslash \mathcal{O}})$ *solution.*

Proof. By the theory of the elliptic equation under the Dirichlet boundary condition [48], one can obtain a unique classical solution f to the Laplace equation Eq. (5). Similarly, one can also obtain a unique classical solution u to the Poisson equation Eq. (6). Hence the solution u to (6) is the unique classical solution to Eq. (4), and u is also $C(\overline{\Omega \backslash \mathcal{O}})$ according to ref. [48]. □

4.2. Solving biharmonic equations using radial basis functions

In the following, we concentrate on the following biharmonic equation:

$$\begin{cases} \Delta^2 u(x,y) = 0, & (x,y) \in \Omega \backslash \mathcal{O} \\ \Delta u(x,y) = G_1, \quad u(x,y) = G_2 & (x,y) \in \partial\mathcal{O}_K \\ \Delta u(x,y) = 0, \quad u(x,y) = \frac{1}{\gamma \cdot d(x,y)} & (x,y) \in \partial\Omega \cup \partial\mathcal{O}_1 \cup \cdots \cup \partial\mathcal{O}_{K-1} \end{cases} \tag{7}$$

https://doi.org/10.1017/S0263574721000709 Published online by Cambridge University Press

where G_1 and G_2 are real numbers. From Eq. (7), it is obvious we mainly focus on a single goal $\mathcal{O}_K$ and multiple obstacles $\mathcal{O}_i$ $(i = 1, 2, \cdots, K-1)$. Here, we have to point out $\mathcal{O}_K$ in fact is a single point. According to Theorem 1, however, $\mathcal{O}_K$ can not be a point. In practice, we set $\mathcal{O}_K = \{(x, y) : \|(x, y) - (x_g, y_g)\|_2 \leq \mathcal{R}\}$, where $\mathcal{R} > 0$ represents a radius and (x_g, y_g) is the goal position.

Assuming $\{(x_i, y_i)\}_{i=1}^m$ are m points locating in the boundaries $\partial\mathcal{O}_1 \cup \cdots \cup \partial\mathcal{O}_K$. Due to the general solution of biharmonic equations can be formulated as [49] $f_1(x, y) + r^2 f_2(x, y)$ with suitable harmonic functions f_1 and f_2, and $r^2 = x^2 + y^2$, and the biharmonic equation Eq. (7) is homogeneous, then its approximation solution can be expressed as [50]

$$u_h(x, y) = \sum_{i=1}^{m} \alpha_i \phi(r_i) + \sum_{i=1}^{m} \beta_i \psi(r_i) \tag{8}$$

where $\phi(r_i) = ln(r_i)$ and $\psi(r_i) = r_i^2 ln(r_i)$ with $r_i = \|(x, y) - (x_i, y_i)\|_2$, and $\alpha_i \in \mathbb{R}$ and $\beta_i \in \mathbb{R}$ are scalar parameters. Also (x_i, y_i) is called the centers of RBF. Obviously, $u_h(x, y)$ satisfies the biharmonic equations. In this paper, we call u_h the biharmonic radial basis representation function (BRBRF). Similarly, u_h also has to satisfy the boundary condition:

$$\begin{cases} \Delta u_h(x, y) = G_1, \quad u_h(x, y) = G_2 & (x, y) \in \partial\mathcal{O}_K \\ \Delta u_h(x, y) = 0, \quad u_h(x, y) = \frac{1}{\gamma \cdot d(x,y)} & (x, y) \in \partial\Omega \cup \partial\mathcal{O}_1 \cup \cdots \cup \partial\mathcal{O}_{K-1} \end{cases} \tag{9}$$

where $\mathcal{O}_K$ is the goal region, and $\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_{K-1}$ is the union of the obstacle regions.

Due to u_h is biharmonic, we just need to determine $2m$ parameters α_i and β_i such that the boundary conditions (9) are satisfied. In essence, RBF approximation solution u_h interpolates given nodes. Therefore we require $u_h(x_i, y_i)$ $(i = 1, 2, \cdots, m)$ must satisfy the boundary constraints (9). On the other hand, in order to approximate boundary constraints on non-given boundary points and avoid the vanishing distance $r_i = 0$, we replace the centers of BRBRF in Eq. (8) as

$$(\hat{x}_i, \hat{y}_i) = \begin{cases} (x_i, y_i) + \lambda \cdot n_i, & (x_i, y_i) \in \partial\Omega \\ (x_i, y_i) - \lambda \cdot n_i, & (x_i, y_i) \in \partial\mathcal{O}_1 \cup \cdots \cup \partial\mathcal{O}_K \end{cases} \tag{10}$$

where $\lambda \geq 1$ and n_i is the unit outer normal at point (x_i, y_i). Then the distance r_i in Eq. (8) is changed to $r_i = \|(x, y) - (\hat{x}_i, \hat{y}_i)\|_2$.

Note that there are $2m$ unknown parameters $\{\alpha_i\}_{i=1}^m$ and $\{\beta_i\}_{i=1}^m$ in u_h, and we also have $2m$ constraints according to Eq. (9). Therefore, we can give the closed-form solution of $\{\alpha_i\}_{i=1}^m$ and $\{\beta_i\}_{i=1}^m$. Let $\omega = (\alpha_1, \alpha_2, \cdots, \alpha_m, \beta_1, \beta_2, \cdots, \beta_m)^T$, m_1 and m_2 be the numbers of points locating in $\partial\mathcal{O}_K$ and $\partial\Omega \cup \partial\mathcal{O}_1 \cup \cdots \cup \partial\mathcal{O}_{K-1}$, respectively. It is obvious that $m_1 + m_2 = m$. Then the four constraints in Eq. (9) can be represented as a matrix form $A \cdot \omega = B$. Let $\phi_{i,j} = ln(r_{i,j})$ and $\psi_{i,j} = r_{i,j}^2 ln(r_{i,j})$ with $r_{i,j} = \sqrt{(x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2}$, then we set

$$A_1 = \left(\begin{array}{cccc|cccc} 0 & 0 & \cdots & 0 & \Delta\psi_{1,1} & \Delta\psi_{1,2} & \cdots & \Delta\psi_{1,m} \\ 0 & 0 & \cdots & 0 & \Delta\psi_{2,1} & \Delta\psi_{2,2} & \cdots & \Delta\psi_{2,m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \Delta\psi_{m_1,1} & \Delta\psi_{m_1,2} & \cdots & \Delta\psi_{m_1,m} \\ \hline \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,m} & \psi_{1,1} & \psi_{1,2} & \cdots & \psi_{1,m} \\ \phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,m} & \psi_{2,1} & \psi_{2,2} & \cdots & \psi_{2,m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{m_1,1} & \phi_{m_1,2} & \cdots & \phi_{m_1,m} & \psi_{m_1,1} & \psi_{m_1,2} & \cdots & \psi_{m_1,m} \end{array}\right)$$

https://doi.org/10.1017/S0263574721000709 Published online by Cambridge University Press

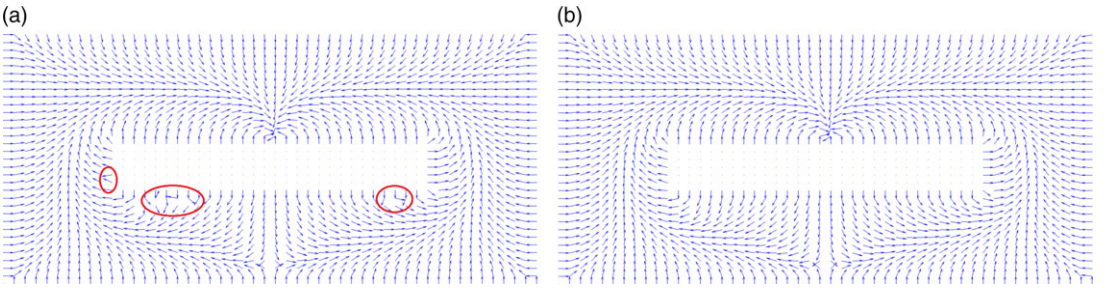


Figure 2. *(a) In three red circle regions, some incorrect directions may lead to unsatisfactory path planning results. (b) The misleading direction field in (a) is rectified by the harmonic interpolation.*

$$A_2 = \left(\begin{array}{cccc|cccc} 0 & 0 & \cdots & 0 & \Delta\psi_{l_1,1} & \Delta\psi_{l_1,2} & \cdots & \Delta\psi_{l_1,m} \\ 0 & 0 & \cdots & 0 & \Delta\psi_{l_2,1} & \Delta\psi_{l_2,2} & \cdots & \Delta\psi_{l_2,m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \Delta\psi_{l_{m_2},1} & \Delta\psi_{l_{m_2},2} & \cdots & \Delta\psi_{l_{m_2},m} \\ \hline \phi_{l_1,1} & \phi_{l_1,2} & \cdots & \phi_{l_1,m} & \psi_{l_1,1} & \psi_{l_1,2} & \cdots & \psi_{l_1,m} \\ \phi_{l_2,1} & \phi_{l_2,2} & \cdots & \phi_{l_2,m} & \psi_{l_2,1} & \psi_{l_2,2} & \cdots & \psi_{l_2,m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{l_{m_2},1} & \phi_{l_{m_2},2} & \cdots & \phi_{l_{m_2},m} & \psi_{l_{m_2},1} & \psi_{l_{m_2},2} & \cdots & \psi_{l_{m_2},m} \end{array}\right)$$

where $l_i = m_1 + i$. Immediately, we get $A = \left[\frac{A_1}{A_2}\right]$, and B is a $2m \times 1$ column vector with $B = (M_{G_1}^T, M_{G_2}^T, M_{S_1}^T, M_{S_2}^T)^T$, where $M_{G_1} = (G_1, G_1, \cdots, G_1) \in R^{1\times m_1}$, $M_{G_2} = (G_2, G_2, \cdots, G_2) \in R^{1\times m_1}$, $M_{S_1} = (0, 0, \cdots, 0) \in R^{1\times m_2}$ and $M_{S_2} = (\frac{1}{r\cdot d(x_1,y_1)}, \frac{1}{r\cdot d(x_2,y_2)}, \cdots, \frac{1}{r\cdot d(x_{m_2},y_{m_2})}) \in R^{1\times m_2}$. Then the unknown ω can be solved by $\omega = A^{-1}B$. Due to the Gaussian elimination is used to compute A^{-1}, then the time complexity is $O(m^3)$. Here we do not give the proof that A is invertible, but in practice the matrix A is rarely not invertible.

4.3. Path planning using biharmonic navigation field

Once all parameters α_i and β_i have been estimated, then we can reconstruct the biharmonic function $u_h(x, y)$ for $(x, y) \in \Omega$. At last the biharmonic navigation field is represented as the negative gradient field $-\nabla u_h(x, y)$, where ∇ is the gradient operator. Robots can arrive at the goal position from any point in Ω by following the navigation field.

We have to point out that $-\nabla u_h(x, y)$ may occur misleading directions at some points near boundaries. For example, there are some incorrect directions at some locations in three red circle regions in Fig. 2(a). If robots follow these directions, paths will be fairly unexpected. Therefore post-processing is needed to rectify the flaws. We observed that most directions are correct, and we also note that directions in local regions are gradually changed. Hence we adapt harmonic interpolation to correct wrong directions. In fact, we first detect positions where wrong directions are occurred, and then we update these directions using the harmonic equation, and directions at other positions are served as the Dirichlet conditions. Note that although the grid size maybe large, the number of wrong directions is small, and thus iterative methods for solving harmonic equations are valid. For example, there are some misleading directions in the red regions in Fig. 2(a). Using the harmonic interpolation for these vectors, we can get the final biharmonic navigation field as shown in Fig. 2(b), which can be used to complete the path planning tasks.

https://doi.org/10.1017/S0263574721000709 Published online by Cambridge University Press

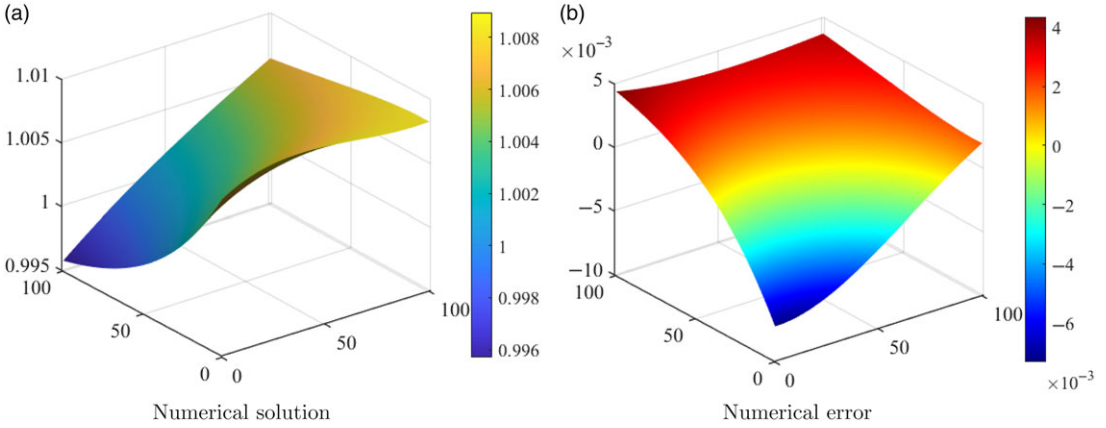


Figure 3. *(a) Numerical solution of a biharmonic equation* $\Delta^2 f(x, y) = 0$ *with* $f(x, y) = x^2 + y^2 + e^x \cos(y)$ *using the approximation method presented in this paper. (b) Numerical error.*

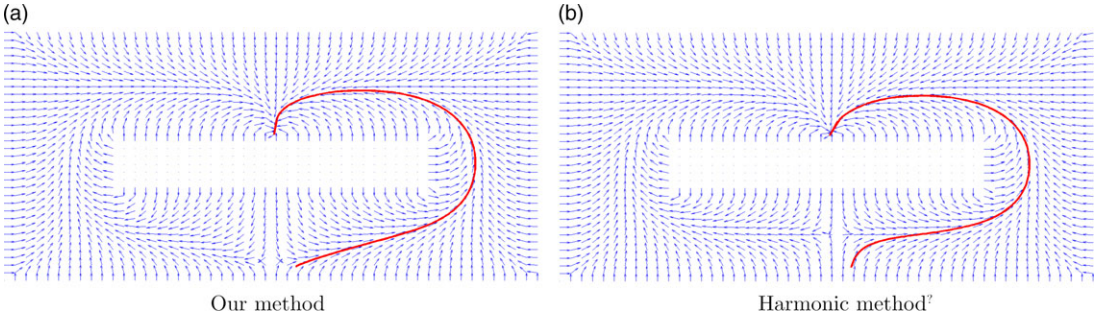


Figure 4. *Path planning results constructed by using our method and the harmonic function method [13] with Dirichlet boundary conditions in (a) and (b), respectively.*

5. Experimental results

In this section, we give implementation details of our algorithm and show several examples. All examples, including Figs. 1(a), (b), 2(a), (b), are implemented using MATLAB on a 3.40 GHz laptop computer with 4 GB of memory. In the next, several examples are shown, and relevant parameters are also given.

We first give a numerical example that can verify the effectiveness of our RBF approximation scheme for the biharmonic equation with boundary conditions. Let $f(x, y) = x^2 + y^2 + e^x \cos(y)$, then the function $f(x, y)$ is a biharmonic function since it satisfies $\Delta^2 f(x, y) = 0$. The boundary conditions are determined by the exact solution $f(x, y)$. In the example, we assume the computation domain is $\Omega = [0, 0] \times [1, 1]$, which is discretized into 100×100 grids, and also let the goal position is (0.5, 0.5). The results are shown in Fig. 3(a) and (b), and the maximal relative error is about 0.73%. The example shows that the presented approximation strategy for biharmonic equation is useful and valid.

By setting $G_1 = 0.1$, $G_2 = -20$, and $\gamma = 10$ in Eq. (9(a)), our biharmonic navigation field method (see Fig. 4(a)) is compared to the harmonic field method (see Fig. 4(b)). Both figures have the same size 165×247 and obstacle configurations. In the two figures, both algorithms have the same start and goal positions, and the navigation field information is also shown in both figures. The results demonstrate that the path constructed by our method is more attractive and steerable than the path constructed by the harmonic function method. It is obvious that the path in Fig. 4(a) that tends to keep far distances to obstacles, and meanwhile maintains local flatness as much as possible.

https://doi.org/10.1017/S0263574721000709 Published online by Cambridge University Press

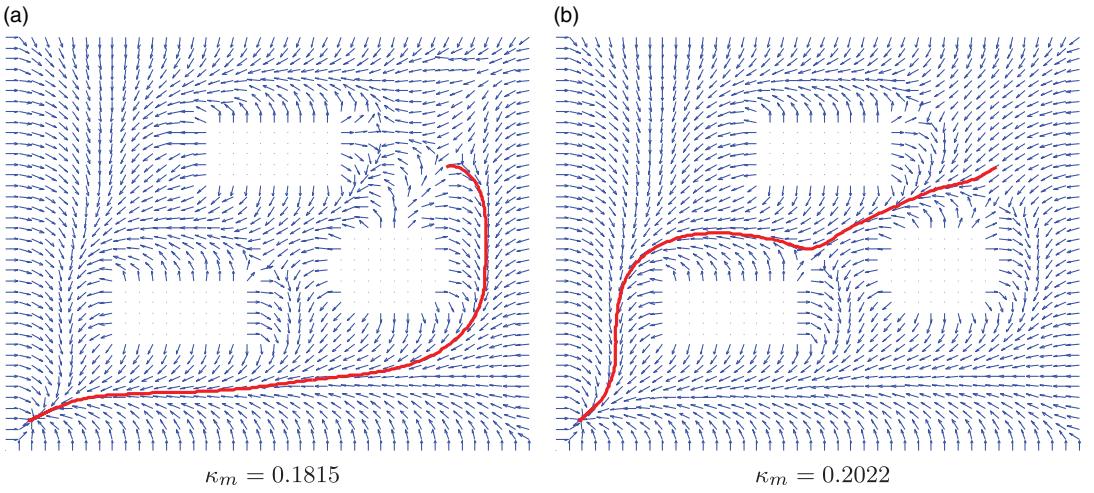


Figure 5. *Path planning results constructed by using our method and the harmonic function method [13] with Dirichlet boundary conditions in (a) and (b), respectively.*

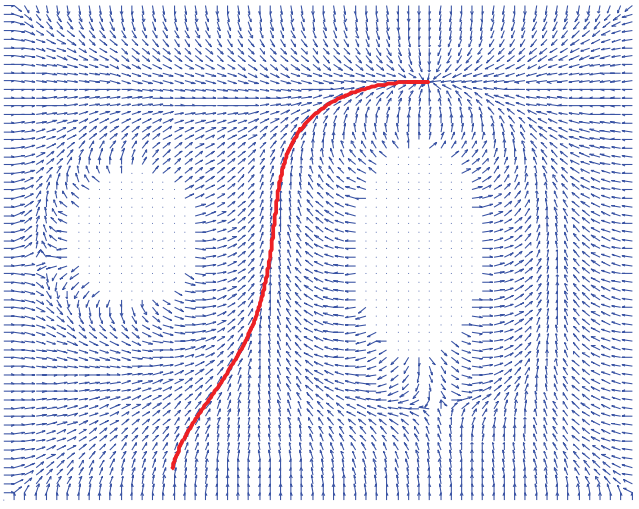


Figure 6. *Our path planning result in a large grid with size* 300×300. *The maximal curvature of the path is* $\kappa_m = 0.1054$.

In Fig. 5, the path shown in (a) is constructed by our method with setting $G_1 = 0.1$, $G_2 = -25$, and $\gamma = 40$, and the result in (b) is generated by following the harmonic field constructed from the harmonic equation with Dirichlet boundary conditions. Both subfigures have the same size 200×200, obstacle configurations, and start and goal positions. In Fig. 5, the path in (a) tends to arrive at the goal position with large distances to obstacles compared to distances between the path and obstacles in (b). Furthermore, it is obvious the maximal curvature of the path in (a) is less than that of the path in (b). The results verify that our algorithm can generate more smoother paths than the harmonic method. Supposing p_0, p_1, and p_2 are three successive points in a polyline, and then the *discrete curvature* at p_1 is computed as $\kappa = \frac{2\theta}{d}$, where d is the sum of two line segments p_0p_1 and p_1p_2, and θ is the angle between $\overrightarrow{p_0p_1}$ and $\overrightarrow{p_1p_2}$.

In Fig. 6, we test our algorithm in a large grid with size 300×300, and the result demonstrates that our algorithm can also run very well in large size grids. Taking Fig. 1(b) as an example, however, solving

https://doi.org/10.1017/S0263574721000709 Published online by Cambridge University Press

Table 1. *Statistical data of paths shown in Fig. 7.*

–	APF [35]	RRT* [31]	Our method
time(s)	26.69	762.67	28.14
κ_m	1.305	2.57	0.054

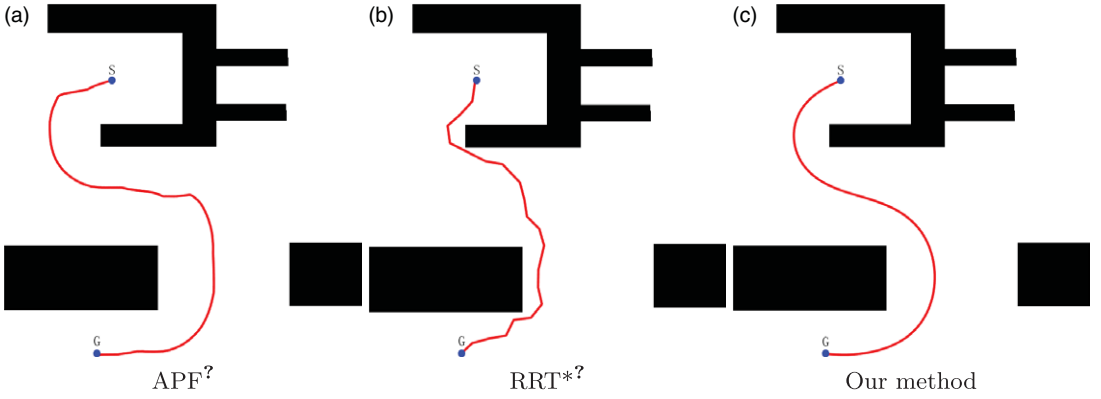


Figure 7. *Given a start position S and a goal position G, path planning results shown in (a), (b), and (c) are obtained by performing the APF [35], RRT* [31], and our proposed method, respectively.*

harmonic equations using numerical iterative methods (e.g., finite difference method) in large size grids cannot arrive at desirable solutions.

We also estimate the performance of our proposed method and other two classical path planning algorithms, Artificial Potential Function (APF) [35] and RRT* [31], in an indoor map shown in Fig. 7. Given a start position S and a goal position G, paths generated by three different path planning algorithms are shown in Fig. 7 with map size 500×500. The maximum curvature and the run time are also computed, and the related data are reported in Table 1. Note that due to a random algorithm, RRT* is performed five times in the experiment, κ_m and the run time are corresponding mean values as well. The time efficiency of APF and our method is similar, but both are significantly better than that of RRT*. The maximum curvature κ_m of the path generated by the proposed biharmonic method, on the other hand, outperforms the other two methods' results.

Examples shown in this section demonstrate that the biharmonic path planning algorithm proposed in this paper is practical and feasible.

6. Discussion and conclusion

Biharmonic functions are an powerful tool for many real applications, and they also show great potential in path planning. The proposed biharmonic navigation method belongs to the potential field function category, and an obvious advantage is that it can generate a more smooth path compared to APF, harmonic methods, and other path planning algorithms. Similar to all potential field methods, the proposed algorithm also suffers the computational inefficiency.

In this paper, we present BRBPF to design a path joining given start and goal positions. The navigation field is the negative gradient of the solution of a biharmonic equation with distance-related boundary conditions. In order to deal with invalid gradient problems for large size grids by using finite difference methods (such as Jacobi iteration and Gauss–Seidel), the RBF approach is adopted to solve our biharmonic equation system. An obvious benefit is that paths constructed by biharmonic navigation fields are smoother than paths constructed by harmonic fields. We have to point out that the RBF method for solving the biharmonic equation presented in this paper is not suitable for single point obstacles due

https://doi.org/10.1017/S0263574721000709 Published online by Cambridge University Press

to outer normals (see Eq. (10)) are need for boundary points of the obstacles. In the future, we would like to develop our method to deal with path planning tasks in real sensor data scenes, even on some curved surfaces, because it is possible to solve the Poisson equation on some Riemannian manifolds [51, 52, 53, 54, 55, 56, 57].

Acknowledgments. This work was supported by the Natural Science Foundation (NSF) of China (61802147), the Natural Science Foundation of Guangdong Province (2018A030310634), and Fundamental Research Funds for the Central Universities (21617348).

References

[1] B. Paden, M. Cáp, S. Z. Yong, D. S. Yershov and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.* **1**(1), 33–55 (2016).
[2] Z. N. Bien, H. Lee, J. Do, Y. Kim, K. Park and S. Yang, "Intelligent interaction for human-friendly service robot in smart house environment," *Int. J. Comput. Intell. Syst.* **1**(1), 77–93 (2008).
[3] W. Gong, X. Xie and Y. Liu, "Human experience–inspired path planning for robots," *Int. J. Adv. Rob. Syst.* **15**(1), 172988141875704 (2018).
[4] V. Malviya, A. K. Reddy and R. Kala, "Autonomous social robot navigation using a behavioral finite state social machine," *Robotica* **38**(12), 2266–2289 (2020).
[5] Z. Yao, W. Zhang, Y. Shi, M. Li, Z. Liang, F. Li and Q. Huang, "Rimjump: Edge-based shortest path planning for a 2D map," *Robotica* **37**(4), 641–655 (2019).
[6] V. Dobrokhodov, "Cooperative path planning of unmanned aerial vehicles," *J. Guidance Control Dyn.* **34**(5), 1601–1602 (2011).
[7] Y. Hu, D. Li, Y. He and J. Han, "Path planning of UGV based on bézier curves," *Robotica* **37**(6), 969–997 (2019).
[8] P. E. Hart, N. J. Nilsson and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.* **4**(2), 100–107 (1968).
[9] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik* **1**(1), 269–271 (1959).
[10] M. Rufli, D. Ferguson and R. Siegwart, "Smooth Path Planning in Constrained Environments," *IEEE International Conference on Robotics and Automation* (2009) pp. 3780–3785.
[11] R. Schneider and L. Kobbelt, "Geometric fairing of irregular meshes for free-form surface design," *Comput. Aided Geom. Des.* **18**(4), 359–379 (2001).
[12] L. Guys, S. Puechmorel and L. Lapasset, "Automatic conflict solving using biharmonic navigation functions," *Proc. Soc. Behav. Sci.* **54**(4), 1378–1387 (2012).
[13] C. I. Connolly, J. B. Burns and R. Weiss, "Path Planning Using Laplace's Equation," *International Conference on Robotics and Automation* (1990) pp. 2102–2106.
[14] S. S. Ge and Y. Cui, "New potential functions for mobile robot path planning," *IEEE Trans. Rob. Autom.* **16**(5), 615–620 (2000).
[15] W. Gong, "Probabilistic model based path planning," *Phys. A Stat. Mech. Appl.* **568**, 125718 (2021).
[16] L. E. Kavraki, P. Svestka, J. Latombe and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Rob. Autom.* **12**(4), 566–580 (1996).
[17] J. J. Kuffner and S. M. Lavalle, "RRT-Connect: An Efficient Approach to Single-Query Path Planning," *International Conference on Robotics and Automation*, vol. 2 (2000) pp. 995–1001.
[18] C. I. Connolly, "Applications of Harmonic Functions to Robotics," *International symposium on Intelligent Control* (1992) pp. 498–502.
[19] A. A. Masoud, S. A. Masoud and M. M. Bayoumi, "Robot Navigation Using a Pressure Generated Mechanical Stress Field: 'The Biharmonic Potential Approach'," *IEEE International Conference on Robotics and Automation* (1994) pp. 124–129.
[20] B. Fornberg and N. Flyer, "Solving PDES with radial basis functions," *Acta Numerica* **24**, 215–258 (2015).
[21] B. Paden, M. Čáp, S. Z. Yong, D. Yershov and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.* **1**(1), 33–55 (2016).
[22] H.-P. Huang and S.-Y. Chung, "Dynamic Visibility Graph for Path Planning," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566)*, vol. 3 (IEEE, 2004) pp. 2813–2818.
[23] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Commun. ACM* **22**(10), 560–570 (1979).
[24] P. Bhattacharya and M. L. Gavrilova, "Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path," *IEEE Rob. Autom. Mag.* **15**(2), 58–66 (2008).
[25] M. Šeda, "Roadmap Methods vs. Cell Decomposition in Robot Motion Planning," *Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation* (World Scientific and Engineering Academy and Society (WSEAS), 2007) pp. 127–132.
[26] D. Demyen and M. Buro, "Efficient Triangulation-based Pathfinding," *AAAI*, vol. 6 (2006) pp. 942–947.

https://doi.org/10.1017/S0263574721000709 Published online by Cambridge University Press

[27] M. Kallmann, "Path Planning in Triangulations," *Proceedings of the IJCAI Workshop on Reasoning, Representation, and Learning in Computer Games* (2005) pp. 49–54.
[28] H. Yan, H. Wang, Y. Chen and G. Dai, "Path Planning Based on Constrained Delaunay Triangulation," *2008 7th World Congress on Intelligent Control and Automation* (IEEE, 2008) pp. 5168–5173.
[29] L. Jaillet, J. Cortés and T. Siméon, "Transition-based RRT for Path Planning in Continuous Cost Spaces," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2008) pp. 2145–2150.
[30] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Technical report, Computer Science Department, Iowa State University (1998).
[31] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Rob. Res.* **30**(7), 846–894 (2011).
[32] D. Hsu, J.-C. Latombe and H. Kurniawati, "On the probabilistic foundations of probabilistic roadmap planning," *Int. J. Rob. Res.* **25**(7), 627–643 (2006).
[33] L. Kavraki, P. Svestka and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Rob. Autom.* **12**(4), 566–580 (1996).
[34] Y. Huang, H. Ding, Y. Zhang, H. Wang, D. Cao, N. Xu and C. Hu, "A motion planning and tracking framework for autonomous vehicles based on artificial potential field-elaborated resistance network (apfe-rn) approach," *IEEE Trans. Ind. Electron.* **67**(2), 1376–1386 (2020).
[35] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Rob. Res.* **5**(1), 90–98 (1986).
[36] P. Yap, N. Burch, R. C. Holte and J. Schaeffer, "Any-Angle Path Planning for Computer Games," *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, AIIDE'11 (AAAI Press, 2011) pp. 201–207.
[37] A. Stentz, "Optimal and Efficient Path Planning for Partially Known Environments," **In:** *Intelligent Unmanned Ground Vehicles* (Springer, 1997) pp. 203–220.
[38] S. Koenig and M. Likhachev, "D*lite," *Eighteenth National Conference on Artificial Intelligence*, Menlo Park, CA, USA (American Association for Artificial Intelligence, 2002) pp. 476–483.
[39] M. Otte, "A survey of machine learning approaches to robotic path-planning," *Int. J. Rob. Res.* **5**(1), 90–98 (2008).
[40] N. Pérez-Higueras, F. Caballero and L. Merino, "Learning Human-Aware Path Planning with Fully Convolutional Networks," *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018) pp. 1–5.
[41] M. Wulfmeier, D. Rao, D. Z. Wang, P. Ondruska and I. Posner, "Large-scale cost function learning for path planning using deep inverse reinforcement learning," *Int. J. Rob. Res.* **36**(10), 1073–1087 (2017).
[42] H. Miao and Y.-C. Tian, "Robot Path Planning in Dynamic Environments Using a Simulated Annealing Based Approach," *2008 10th International Conference on Control, Automation, Robotics and Vision* (IEEE, 2008) pp. 1253–1258.
[43] H. Burchardt and R. Salomon, "Implementation of Path Planning Using Genetic Algorithms on Mobile Robots," *2006 IEEE International Conference on Evolutionary Computation* (IEEE, 2006) pp. 1831–1836.
[44] F.-Z. Yue, P.-Y. Cui and H.-T. Cui, "Planetary rover path-planning based on ant colony optimization algorithm," *Control and Decision* **21**(12), 1437 (2006).
[45] X. Du, H.-H. Chen and W.-K. Gu, "Neural network and genetic algorithm based global path planning in a static environment," *J. Zhejiang Univ. Sci. A* **6**(6), 549–554 (2005).
[46] E. Masehian and M. R. Amin-Naseri, "A voronoi diagram-visibility graph-potential field compound algorithm for robot path planning," *J. Rob. Syst.* **21**(6), 275–300 (2004).
[47] R. J. Duffin, "The maximum principle and biharmonic functions," *J. Math. Anal. Appl.* **3**(3), 399–405 (1961).
[48] D. Gilbarg and N. S. Trudinger, *Elliptic Partial Differential Equations of Second Order* (Springer, Berlin, 1983).
[49] C. Loewner, "On generation of solutions of the biharmonic equation in the plane by conformal mappings," *Pac. J. Math.* **3**(2), 417–436 (1953).
[50] G. D. Manolis and D. Polyzos, *Recent Advances in Boundary Element Methods. A Volume to Honor Professor Dimitri Beskos* (Springer, New York, 2009).
[51] X.-Q. Fan, "On the existence of solutions of poisson equation and poincaré - lelong equation," *Manuscripta Mathematica* **120**(4), 435–467 (2006).
[52] X.-Q. Fan, "Poisson equation on some complete noncompact manifolds," *Acta Mathematica Sinica-English Ser.* **23**(4), 623–638 (2007).
[53] P. Li and L.-F. Tam, "Green's functions, harmonic functions, and volume comparison," *J. Differ. Geom.* **41**(2), 277–318 (1995).
[54] P. Li and S.-T. Yau, "On the parabolic kernel of the schrödinger operator," *Acta Mathematica* **156**(3–4), 153–201 (1986).
[55] O. Munteanu, C.-J. Sung and J. Wang, "Poisson equation on complete manifolds," *Adv. Math.* **348**, 81–145 (2019).
[56] L. Ni, "The poisson equation and Hermitian–Einstein metrics on holomorphic vector bundles over complete noncompact kähler manifolds," *Indiana Univ. Math. J.* **51**(3), 679–704 (2002).
[57] L. Ni, Y. Shi and L.-F. Tam, "Poisson equation, poincaré–lelong equation and curvature decay on complete kähler manifolds," *J. Differ. Geom.* **57**(2), 339–388 (2001).

Cite this article: X.-Q. Fan and W. Gong (2022). "Biharmonic navigation using radial basis functions", Robotica **40**, 599–610. https://doi.org/10.1017/S0263574721000709

https://doi.org/10.1017/S0263574721000709 Published online by Cambridge University Press