ISSN 2007-9737

Homotopy-based Motion Planning for Car-like Robots

Yu'ban Yoshua Martinez Vasquez[1,*], Israel Becerra[1], Gerardo Diaz Arango[2]

[1] Centro de Investigación en Matemáticas, Guanajuato, Mexico

[2] Universidad Veracruzana, Departamento de Instrumentación Electrónica, Veracruz, Mexico

{yuban.martinez, israelb}@cimat.mx, guda.diaz.gd@gmail.com

Abstract. Path planning for car-like robots in environments with obstacles constitutes a complex problem, as it requires simultaneously satisfying geometric and kinematic (non-holonomic) constraints. Although there are planners available in the literature that address this setting, this work proposes an alternative two-stage planning methodology based on algebraic and geometric methods. In the first stage, the homotopy continuation method is used to solve the system of nonlinear equations that incorporates the geometric constraints of the vehicle and the environment, obtaining an initial collision-free trajectory in the configuration space of the rigid chassis. In the second stage, a post-processing step applies Reeds-Shepp curves to connect and smooth the trajectory, followed by a randomized shortcutting pass that reduces total path length while preserving kinematic feasibility. To validate the proposed approach, simulation results are presented comparing the performance of the methodology against geometric sampling-based planners from the OMPL library, all operating in the same Reeds-Shepp configuration space. The evaluation metrics focus on computation time and total path length, demonstrating the feasibility and competitiveness of the homotopic method for autonomous navigation problems.

Keywords. Homotopy continuation method, car-like robot, path planning, mobile robotics, Reeds-Shepp curves.

1 Introduction

Path planning is a fundamental problem in mobile robotics: given a start configuration and a goal configuration, the planner must find a collision-free trajectory through a workspace populated by obstacles. For holonomic platforms such as omnidirectional ground robots, the problem reduces to finding a curve in the planar workspace, a task that some of the first motion planning algorithms [7] solve reliably. Car-like robots, however, impose an additional layer of difficulty: their kinematics are governed by non-holonomic constraints that prohibit lateral motion and bound the path curvature through a minimum turning radius. These constraints render many holonomic planners inapplicable and require dedicated algorithms that plan simultaneously over the position and orientation of the vehicle.

Computing paths for car-like robots falls into the category of what is known as kinodynamic planning [11], which is customarily addressed through sampling-based methods [12] or classical potential-field approaches [13], among others. However, such methods are either computationally slow or prone to falling into local minima. To cope with these limitations, this work proposes an alternative two-stage planning method based on algebraic and geometric principles. In the first stage, the homotopy continuation method is employed to find a solution to a system of nonlinear equations that encodes the geometric constraints of the environment and the body of the vehicle, resulting in an initial collision-free trajectory in the configuration space of the rigid chassis. In the second stage, the trajectory is connected and

smoothed by a post-processing phase that uses Reeds-Shepp curves [15].

This is followed by a random shortcutting pass that shortens the overall path length while maintaining kinematic feasibility.

1.1 Related Work

Sampling-based Methods. Rapidly-exploring Random Trees (RRT) [10] and Probabilistic Roadmaps (PRM) [4] are among the most widely used frameworks for motion planning problems. In their kinodynamic variants, when applied to a Reeds–Shepp state space, these methods operate directly in $SE(2)$ and employ the shortest Reeds–Shepp path [15] as the local steering function [11], thus generating kinematically feasible trajectories without the need for a separate post-processing step. Bidirectional variants such as RRT-Connect [6] improve convergence speed by growing trees simultaneously from both the start and goal. Cell-exploration planners such as KPIECE [16] and LBKPIECE [17] improve coverage in high-dimensional spaces, whereas EST [3] is designed to explore expansive regions of the configuration space. Although all of these methods are probabilistically complete, they are not deterministic: different runs may yield different paths, and repeatability is not guaranteed. In addition, their computation times are often high.

Potential-field Methods. Artificial potential fields [5] model obstacles as repulsive charges and the goal as an attractive one, yielding a fast, deterministic planner. The primary weakness of this approach is its susceptibility to local minima, which can trap the robot in configurations far from the goal. Homotopy-based planners [19, 2] address this shortcoming by embedding obstacle boundaries as algebraic singularities in a system of nonlinear equations, producing a continuous, collision-free curve that is immune to local minima by construction. The Enhanced Homotopy Path Planning Method (EHPPM) [2] demonstrated computation times on the order of milliseconds for maps with up to 200 obstacles and outperformed RRT and PRM in a comparative benchmark conducted with OMPL [18].

Nevertheless, prior homotopy planners were developed for holonomic point robots navigating among circular or ellipsoidal obstacles in 2-D workspaces; their direct extension to car-like vehicles with non-holonomic constraints and finite oriented footprints remained an open problem.

Non-holonomic Planning and Reeds-Shepp Curves. A classical result due to Laumond [9] states that any smooth path in the configuration space of a car-like robot can be approximated, to arbitrary precision, by a sequence of Reeds-Shepp curves [15]–the set of shortest paths for a vehicle that can move both forward and backward with a fixed minimum turning radius. This result is widely used both as a steering function inside sampling-based planners and as a post-processing tool that converts a sequence of configurations into a kinematically feasible trajectory. The second stage of our proposal leverages this result.

1.2 Contributions

This work proposes, as its main contribution, a two-stage planning methodology that extends homotopy-based planning to car-like robots. The main features of the proposal are as follows:

— **3-D Configuration-space Planning.** Rather than planning in the planar workspace and assigning headings in a separate step, the homotopy system is formulated directly in the three-dimensional configuration space $SE(2)$, allowing position and orientation to be solved simultaneously.

— **Multi-point Rigid-body Collision Model.** The finite, oriented footprint of the vehicle is approximated by 7 reference points that cover the chassis corners and key midpoints. Each reference point contributes an orientation-dependent repulsion term, coupling obstacle avoidance and heading planning within the homotopy system.

— **Convex-polygon Obstacle Barrier Functions.** A half-plane activation formulation is introduced to construct exact barrier functions for convex polygonal

obstacles, replacing the circular or ellipsoidal approximations of prior work.

— **Guide-function Repulsion with Automatic Sign Assignment.** The per-obstacle scalar repulsion weights of prior methods are replaced by a single guide function whose zero set coincides with the goal configuration, together with an automatic rule that determines the side on which the vehicle passes each obstacle. This eliminates the need for manual parameter tuning.

— **Reeds-Shepp post-processing and Shortcut Optimization.** The discrete configuration sequence produced by the homotopy stage is connected by shortest Reeds-Shepp paths, enforcing the curvature bound at every point of the final trajectory. A subsequent randomized shortcutting pass further reduces total path length while preserving kinematic feasibility.

The remainder of the paper is organized as follows. Section 2 reviews the homotopy continuation framework and the baseline EHPPM. Section 3 presents the four extensions introduced to handle car-like robots. Section 4 reports simulation results and a comparative benchmark. Section 5 draws conclusions and outlines future work.

2 Preliminaries

2.1 Homotopy Continuation Methods

Homotopy continuation methods (HCMs) are a class of numerical techniques commonly used to find solutions of systems of nonlinear algebraic equations (NAES) of the form:

$$f(\mathrm{x}) = 0, \quad f : \mathbb{R}^n \to \mathbb{R}^n, \tag{1}$$

where $\mathrm{x} \in \mathbb{R}^n$ is the vector of unknowns. The key idea is to embed the target system f in a one-parameter family of systems that continuously deforms a *trivial* system $G(\mathrm{x})$–whose solutions are known–into the original system $f(\mathrm{x})$. This deformation is governed by the *homotopy map:*

$$H(\mathrm{x}, \lambda) = \lambda f(\mathrm{x}) + (1-\lambda)\, G(\mathrm{x}) = 0, \quad H : \mathbb{R}^{n+1} \to \mathbb{R}^n, \tag{2}$$

where $\lambda \in [0, 1]$ is the homotopy parameter. By construction:

— at $\lambda = 0$, the system reduces to $G(\mathrm{x}) = 0$, which has a known (trivial) solution.

— at $\lambda = 1$, the system coincides with the original target $f(\mathrm{x}) = 0$.

The solution set $H^{-1}(0) = \{(\mathrm{x}, \lambda) : H(\mathrm{x}, \lambda) = 0\}$ generically forms a family of smooth curves γ in $\mathbb{R}^{n+1}$ [1].

By tracking one such curve from a known starting point at $\lambda = 0$ to $\lambda = 1$, the HCM finds a solution of the original system without requiring a good initial guess.

Newton homotopy. The specific homotopy used throughout this paper is *Newton's homotopy*, obtained by choosing $G(\mathrm{x}) = f(\mathrm{x}) - f(\mathrm{x}_0)$, where $\mathrm{x}_0 \in \mathbb{R}^n$ is an arbitrary starting point. Substituting into (2) and simplifying yields:

$$H(\mathrm{x}, \lambda) = f(\mathrm{x}) - (1 - \lambda)\, f(\mathrm{x}_0) = 0. \tag{3}$$

This choice guarantees that the curve γ passes exactly through x_0 at $\lambda = 0$, making it natural to use the robot's initial configuration as the starting point.

2.2 Homotopy Path Planning Method

The *Homotopy Path Planning Method* (HPPM), introduced in [19] and extended to the Enhanced HPPM (EHPPM) in [2], applies the framework above to the problem of collision-free path planning for a point robot in a normalized 2-D workspace $[0, 1]^2$.

2.2.1 Goal Geometry

The goal configuration $B = (x_g, y_g)$ is encoded as the unique intersection of two straight lines with distinct slopes m_1 and m_2:

$$L_1(x, y) = -y + m_1 x - (m_1 x_g - y_g) = 0, \quad (4)$$

$$L_2(x, y) = -y + m_2 x - (m_2 x_g - y_g) = 0. \quad (5)$$

Both lines vanish simultaneously only at B, providing a scalar attraction mechanism toward the goal. The baseline two-equation system is:

$$f_1(x, y) = L_1(x, y) = 0, \quad (6)$$

$$f_2(x, y) = L_2(x, y) = 0. \quad (7)$$

2.2.2 Obstacle Representation

Each obstacle i is modeled by a scalar algebraic function $\mathcal{O}_i(x, y)$ whose zero set coincides with the obstacle boundary. For circular obstacles of radius r_i centered at $(c_{x,i}, c_{y,i})$:

$$\mathcal{O}_i(x, y) = (x - c_{x,i})^2 + (y - c_{y,i})^2 - r_i^2 = 0. \quad (8)$$

The repulsion field generated by m obstacles is:

$$W(x, y) = \sum_{i=1}^{m} \frac{p_i}{\mathcal{O}_i(x, y)}, \quad (9)$$

where $p_i \in \mathbb{R} \setminus \{0\}$ is a signed repulsion weight. The sign of p_i controls whether the planned path passes to one side or the other of the i-th obstacle; its magnitude governs the clearance. The singularity $1/\mathcal{O}_i \to \infty$ as the robot approaches the obstacle boundary deflecting the homotopy curve around it, analogously to repulsive potentials in artificial-field methods–but without the risk of local minima, because the deflection is a topological effect of the singularity rather than a gradient step.

2.2.3 Complete System

The obstacle repulsion is added to equation (7), and the goal-normalization constant $Q = W(x_g, y_g)$ is subtracted to ensure that $f_2 = 0$ is satisfied at the goal:

$$f_1(x, y) = L_1(x, y) = 0, \quad (10)$$

$$f_2(x, y) = L_2(x, y) + W(x, y) - Q = 0. \quad (11)$$

Applying Newton's homotopy (3) to this system with starting point (x_0, y_0) gives the two-equation homotopy system:

$$\begin{cases} H_1(x, y, \lambda) = f_1(x, y) - (1 - \lambda)\, f_1(x_0, y_0) = 0, \\ H_2(x, y, \lambda) = f_2(x, y) - (1 - \lambda)\, f_2(x_0, y_0) = 0. \end{cases} \quad (12)$$

Since $H : \mathbb{R}^3 \to \mathbb{R}^2$ is underdetermined, its solution set $H^{-1}(0)$ is generically a family of curves in the three-dimensional (x, y, λ) space. The obstacle singularities topologically force these curves to route around the obstacles, producing collision-free trajectories.

2.3 Spherical Path Tracking Algorithm

Since $H^{-1}(0)$ is a family of curves rather than isolated points, a dedicated *curve-following* technique is required to trace the relevant branch from the starting point $(x_0, y_0, 0)$ to a solution at $\lambda = 1$.

The *Spherical Tracking Algorithm* (STA), introduced in [20, 14], accomplishes this by augmenting the homotopy system with the equation of an $(n + 1)$-dimensional hypersphere centered at the most recently computed point on the curve:

$$S_i(x, y, \lambda) = (x - c_x)^2 + (y - c_y)^2 + (\lambda - c_\lambda)^2 - r^2 = 0, \quad (13)$$

where (c_x, c_y, c_λ) is the sphere center and $r > 0$ is its radius, kept constant throughout the tracking. The augmented system:

$$H_S = \begin{cases} H_1(x, y, \lambda) = 0, \\ H_2(x, y, \lambda) = 0, \\ S_i(x, y, \lambda) = 0 \end{cases} \quad (14)$$

is now square (3 equations, 3 unknowns) and its solutions are the two intersection points of the sphere with the homotopy curve.

By advancing the sphere center step by step along the curve–always placing it at the previously found intersection point–the algorithm traces γ from start to goal.

2.4 Predictor–Corrector Scheme

Solving the augmented system (14) directly with Newton–Raphson (N-R) from an arbitrary initial point may converge to the *backward* intersection of the sphere with the curve rather than the *forward* one, stalling the tracking. A predictor–corrector scheme prevents this by providing a warm start that is geometrically guaranteed to lie close to the forward intersection.

2.4.1 Euler Predictor

At the current curve point (x_i, y_i, λ_i), the tangent direction $\vec{v}_p$ to the homotopy curve is obtained from the null space of the Jacobian $\partial H/\partial(x, y, \lambda)$. Specifically, differentiating (12) with respect to an arc-length parameter ρ and applying the chain rule yields the linear system:

$$\begin{pmatrix} \partial_x H_1 & \partial_y H_1 \\ \partial_x H_2 & \partial_y H_2 \end{pmatrix} \begin{pmatrix} \partial_\rho x \\ \partial_\rho y \end{pmatrix} = -\frac{\partial \lambda}{\partial \rho} \begin{pmatrix} \partial_\lambda H_1 \\ \partial_\lambda H_2 \end{pmatrix}, \quad (15)$$

where the sign of $\partial\lambda/\partial\rho$ is determined by:

$$\frac{\partial \lambda}{\partial \rho} = s_g \, (-1)^{n+1} \det \begin{pmatrix} \partial_x H_1 & \partial_y H_1 \\ \partial_x H_2 & \partial_y H_2 \end{pmatrix}, \quad (16)$$

with $s_g \in \{+1, -1\}$ chosen so that λ increases along the tracked branch. The unit tangent vector $\hat{v}_p = \vec{v}_p / \|\vec{v}_p\|$ is assembled from $(\partial_\rho x, \, \partial_\rho y, \, \partial_\rho \lambda)$, and the *predictor point* is placed on the sphere surface in the tangent direction:

$$(x_p, y_p, \lambda_p) = (c_x, c_y, c_\lambda) + r \, \hat{v}_p. \quad (17)$$

2.4.2 Newton–Raphson Corrector

Starting from the predictor point, the N-R method is applied to the augmented system (14):

$$\begin{pmatrix} x_{j+1} \\ y_{j+1} \\ \lambda_{j+1} \end{pmatrix} = \begin{pmatrix} x_j \\ y_j \\ \lambda_j \end{pmatrix} - \left[J_S(x_j, y_j, \lambda_j) \right]^{-1} H_S(x_j, y_j, \lambda_j), \quad (18)$$

where J_S is the $(n+1) \times (n+1)$ Jacobian of H_S. Iterations continue until the residual $\|H_S(z_{j+1})\|$ falls below a prescribed tolerance ε, or until a maximum number of iterations $j_{\max}$ is reached. Because the predictor places the initial guess on the sphere surface close to the forward intersection of γ, convergence is typically achieved in a small number of iterations when the sphere radius r is chosen appropriately relative to the local curvature of the solution curve. The full tracking loop follows the spherical predictor–corrector scheme detailed in [2].

2.5 Reeds-Shepp Curves

A car-like robot with minimum turning radius $\rho_{\min}$ has kinematics governed by:

$$\dot{x} = v \cos\theta, \quad \dot{y} = v \sin\theta, \quad \dot{\theta} = \frac{v}{\rho_{\min}} \tan\phi, \quad (19)$$

where $v \in \mathbb{R}$ is the signed speed (positive forward, negative backward) and ϕ is the front-wheel steering angle, bounded so that $|\tan\phi| \leq \rho_{\min}^{-1}$. Reeds and Shepp [15] proved that the *shortest* path between any two configurations $q_a, q_b \in SE(2)$ under these constraints consists of at most five segments, each belonging to one of three primitive types:

— C^+: circular arc of radius $\rho_{\min}$, driven *forward* (left or right turn).

— C^-: circular arc of radius $\rho_{\min}$, driven *backward*.

— $S^\pm$: straight segment driven forward or backward.

A path is written as a *word* over this alphabet; a vertical bar | denotes a *cusp* (a reversal of driving direction). For example, C^+SC^- denotes a left arc forward, a straight segment forward, and a right arc backward, with no cusps.

By exhaustive analysis of the optimality conditions, Reeds and Shepp identified 48 distinct word types grouped into canonical families (CSC, CCC, CCSC, etc.), obtained by applying time-reversal and left-right reflection symmetries [15, 11].

2.5.1 Computing the Reeds-Shepp Distance

The Reeds-Shepp problem is reduced to unit turning radius by expressing the relative configuration in normalized coordinates: $\Delta\tilde{q} = (\Delta x/\rho_{\min}, \Delta y/\rho_{\min}, \Delta\theta)$. For each of the 48 Reeds-Shepp word types, segment lengths are obtained in closed form as functions of $\Delta\tilde{q}$. Denoting the normalized length of type k by $\ell_k(\Delta\tilde{q})$, the distance is:

$$d_{\mathrm{RS}}(q_a, q_b) = \rho_{\min} \min_{k=1,\ldots,48} \ell_k(\Delta\tilde{q}). \tag{20}$$

Types with no feasible real solution are assigned $\ell_k = +\infty$ [11]. Since the number of candidates is fixed, this evaluation is $O(1)$.

3 Homotopy Planner for Car-like Robots

The formulation described in Section 2.1 was originally developed for a point robot navigating among circular obstacles in a 2-D workspace. Adapting it to a car-like vehicle requires addressing two additional challenges: (i) the robot occupies a finite, oriented footprint that moves as the vehicle turns, and (ii) the orientation θ must itself be planned, not merely inferred after the fact.

This section starts with the problem definition, followed by the four extensions introduced to handle these requirements: the promotion of the planning problem to a 3-D configuration space, the multi-point rigid-body collision model, the replacement of circular obstacle representations by convex-polygon barrier functions and an adaptive repulsion field that does not require manual tuning.

3.1 Problem Definition

The configuration of the car-like robot is described by the vector:

$$\mathrm{x} = \begin{bmatrix} x & y & \theta \end{bmatrix}^\top \in \mathbb{R}^2 \times \mathbb{S}^1, \tag{21}$$

where (x, y) is the rear-axle midpoint and θ is the heading angle.

Therefore, let $\mathcal{C} = SE(2)$ be the configuration space, $\mathcal{C}_{obs}$ the set of configurations where the robot is in collision with an obstacle, and $\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}$ the collision-free configuration set.

Problem 1 *Given a pair of start and goal configurations,* x_0 *and* x_g*, find a trajectory in* $\mathcal{C}_{free}$ *that connects* x_0 *and* x_g *respecting the car-like robot kinematics modeled by* (19)*.*

3.2 Configuration Space Extension

Planning directly in the three-dimensional configuration space, $\mathcal{C} = SE(2)$, allows the planner to determine a collision-free pose sequence $(x(\lambda), y(\lambda), \theta(\lambda))$ simultaneously, rather than computing a planar path and assigning headings in a separate post-processing step.

To accommodate the extra dimension in the homotopy continuation method, the goal geometry is extended with a third scalar equation:

$$f_3(\theta) = \theta - \theta_g = 0, \tag{22}$$

where θ_g is the desired heading at the goal configuration $\mathrm{x}_g = (x_g, y_g, \theta_g)$. The full target system therefore becomes $f : \mathbb{R}^3 \to \mathbb{R}^3$:

$$f_1(x, y) = L_1(x, y) = 0, \tag{23}$$

$$f_2(x, y, \theta) = L_2(x, y) + R(x, y, \theta) - R_{\mathrm{goal}} = 0, \tag{24}$$

$$f_3(\theta) = \theta - \theta_g = 0, \tag{25}$$

where L_1 and L_2 are the goal-defining lines from Section 2.2, $R(x, y, \theta)$ is the orientation-dependent repulsion field defined in Section 3.5 and then a normalization constant is computed by evaluating the field at the goal $R_{\mathrm{goal}} = R(x_g, y_g, \theta_g)$ so that $f_2(\mathrm{x}_g) = 0$ is guaranteed by construction and the goal configuration is always a root of the complete system.

Applying Newton's homotopy to this system gives:

$$H(\mathrm{x}, \lambda) = f(\mathrm{x}) - (1-\lambda)\, f(\mathrm{x}_0) = 0, \quad (\mathrm{x}, \lambda) \in \mathbb{R}^3 \times [0, 1], \tag{26}$$

where $\mathrm{x}_0 = (x_0, y_0, \theta_0)^\top$ is the initial configuration of the vehicle.

3.3 Multi-Point Rigid-Body Collision Model

Treating the car as a point robot would ignore the finite dimensions of the chassis and lead to paths that are geometrically collision-free for the reference point but not for the vehicle body. To address this, the collision check is distributed over a set of K *reference points* that cover the vehicle footprint.

Given the wheelbase l and half-width w, these points are expressed in closed form as rigid-body offsets from the rear-axle midpoint:

$$C_k(x, y, \theta) = \begin{bmatrix} x \\ y \end{bmatrix} + \mathcal{R}(\theta) \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix}, \quad k = 0, \ldots, K-1, \tag{27}$$

where $\mathcal{R}(\theta)$ is the standard 2-D rotation matrix and $(\Delta x_k, \Delta y_k)$ are the local offsets of each reference point.

The current implementation uses $K = 7$ points: the four chassis corners, the front-axle midpoint, and two lateral midpoints at mid-length of the vehicle.

This choice provides a tighter approximation of the rectangular footprint than the four corners alone, particularly for obstacles that can penetrate the lateral or frontal face of the vehicle.

Since C_k depends on θ through (27), every repulsion term evaluated at a reference point becomes an implicit function of the vehicle orientation, coupling obstacle avoidance and heading planning directly inside the homotopy system.

3.4 Convex-Polygon Obstacle Representation

Prior work in homotopy-based planners has modeled obstacles exclusively as circles or ellipses, since these shapes admit simple closed-form barrier functions (see (8)).

This work proposes the use of *convex polygon* barrier functions as a more faithful representation of real-world obstacles. Given a convex polygon with n ordered vertices $\{V_0, \ldots, V_{n-1}\}$ and interior centroid c, each edge defines the half-plane:

$$\ell_j(x, y) = m_j x - y + b_j = 0, \quad j = 0, \ldots, n-1, \tag{28}$$

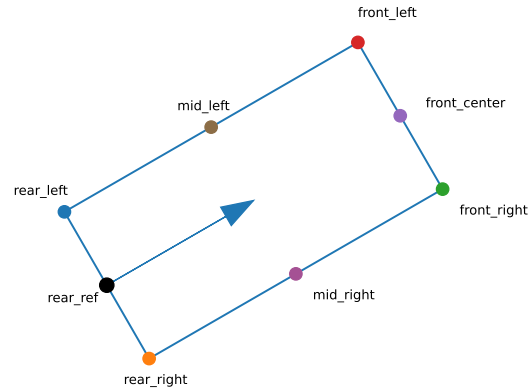

Fig. 1. Reference points used for the rigid-body collision model ($K = 7$): four corners, front-axle midpoint, and two lateral midpoints

where m_j and b_j are the slope and intercept of the j-th edge (with a separate expression for vertical edges). The sign of ℓ_j evaluated at the centroid c indicates which half-plane contains the polygon interior. A one-sided activation term is then constructed per edge:

$$\tau_j(x, y) = \begin{cases} |\ell_j| + \ell_j & \text{if } \ell_j(c) \leq 0, \\ |\ell_j| - \ell_j & \text{if } \ell_j(c) > 0, \end{cases} \tag{29}$$

so that $\tau_j = 0$ when the query point lies on the *interior* side of edge j (same half-plane as the centroid) and strictly positive on the exterior side. The barrier function for the entire polygon is:

$$\mathcal{E}_i(x, y) = \sum_{j=0}^{n-1} \tau_j(x, y). \tag{30}$$

By construction, $\mathcal{E}_i = 0$ for any point *inside* the polygon and $\mathcal{E}_i > 0$ outside it. Consequently, the repulsion term $1/\mathcal{E}_i$ diverges as a vehicle reference point approaches the polygon boundary from the exterior, creating a singularity that deflects the trajectory around the obstacle – analogous to the singularity that circular barriers produce at $\mathcal{O}_i = 0$.

3.5 Adaptive Repulsion Field via Guide Function

In prior homotopy planners, the repulsion weight for each obstacle is a manually tuned scalar

constant p_i (see (9)). This requires per-obstacle parameter selection and provides no systematic way to account for the spatial relationship between an obstacle and the intended route.

This work proposes replacing the scalar weight with a *guide function* $N(x, y)$, evaluated at each reference point of the vehicle, which modulates the repulsion automatically based on proximity to the straight-line path from start to goal.

The guide function is defined as the sum of the two goal lines:

$$N(x, y) = L_1(x, y) + L_2(x, y), \tag{31}$$

whose combined zero set is the goal configuration. This function is large near the start and decreases to zero at the goal, producing a natural scalar field aligned with the direction of motion.

As a result, obstacles that lie close to or on the straight-line route produce a large N-value at the reference points and thus generate a stronger deflection, while obstacles that are laterally distant generate a smaller contribution automatically, without any manual tuning.

3.5.1 Automatic Sign Assignment

A second limitation of the scalar formulation is that the sign of the repulsion, which controls whether the planner routes the vehicle to the left or the right of an obstacle, must be specified manually.

Here, this sign is determined automatically by computing, for each obstacle i, on which side of the straight guide line ℓ_g connecting start to goal the obstacle centroid c_i lies:

$$s_i = \begin{cases} +1 & \text{if } \ell_g(c_i) \leq 0, \\ -1 & \text{otherwise,} \end{cases} \tag{32}$$

where $\ell_g(x, y) = A_g x + B_g y + C_g$ is the signed distance function of the guide line. This assignment ensures that the vehicle is consistently deflected away from the side on which the obstacle centroid lies, without any user intervention.

3.5.2 Complete Repulsion Field

Combining multi-point evaluation, the polygon barrier, and the guide function, the complete orientation-dependent repulsion field is:

$$R(x, y, \theta) = k_{\text{rep}} \sum_{i=1}^{m} s_i \sum_{k=0}^{K-1} \frac{N(C_k(x, y, \theta))}{\mathcal{E}_i(C_k(x, y, \theta))}, \tag{33}$$

where $k_{\text{rep}} > 0$ is a single global scaling constant and all other parameters are determined automatically from the geometry.

3.6 Planning Pipeline

The complete two-stage planning pipeline proceeds as follows:

— **Stage 1: Problem Setup and Homotopy Solution.**

- **Geometry setup.** Given x_0, x_g, vehicle dimensions (l, w), and the obstacle set $\{V_i\}$, the goal lines L_1, L_2, the guide function N, the reference-point offsets $\{(\Delta x_k, \Delta y_k)\}$, the polygon barriers $\{\mathcal{E}_i\}$, and the obstacle signs $\{s_i\}$, are computed.
- **Homotopy.** The system $f(\mathrm{x}) = 0$ in (23)–(25) is assembled and the Newton homotopy (26) is traced from x_0 to a solution x^* with $\lambda = 1$, producing a discrete sequence of collision-free configurations $\{(x_j, y_j, \theta_j)\}$ in $SE(2)$.

— **Stage 2: Path Construction and Optimization.**

- **Reeds-Shepp interpolation.** Inspired by [8], configuration pairs $(\mathrm{x}_j, \mathrm{x}_{j+1})$ are connected by the shortest feasible Reeds-Shepp curve [15], enforcing the curvature bound $\kappa \leq 1/\rho_{\min}$ and producing a smooth, kinematically feasible trajectory. Path length is measured as the sum of the arc lengths of the constituent circular-arc and straight-line segments.

– **Shortcut optimization.** After the full Reeds-Shepp path is assembled, a randomized shortcutting pass is applied: pairs of configurations $(\mathrm{x}_a, \mathrm{x}_b)$ are sampled uniformly at random from the path sequence, and the direct Reeds-Shepp connection $\mathrm{RS}(\mathrm{x}_a, \mathrm{x}_b)$ is tested for collision.

3.7 A Note on Completeness

Tracking succeeds whenever the branch of $H^{-1}(0)$ that contains the start configuration can be followed up to $\lambda = 1$, but there exist topological situations–for example, turning points where $\partial\lambda/\partial\rho$ vanishes and the branch folds back, or branches that escape the working region before reaching the goal–in which, due to numerical errors, the continuation curve does not converge to a solution even though a feasible path exists.

The method should therefore be understood as a fast, deterministic planner whose completeness depends on the resolution used in the underlaying numerical methods. However, a formal analysis on the resolution completeness [1] of the method remains an open problem and a direction for future theoretical work.

4 Simulation Results

4.1 Experimental Setup

All experiments were executed on a machine running Ubuntu 22.04 (Intel® Core™ i7-13620H, 16 GB RAM). The proposed planner was implemented in C++; baselines were evaluated using OMPL [18] with a `ReedsSheppStateSpace`, so every planner operates natively in $SE(2)$ with the same minimum turning radius $\rho_{\min}$ and identical local steering function [11]. Six baselines were evaluated: RRT [10], RRTConnect [6], PRM [4], KPIECE [16], LBKPIECE [17], and EST [3]. Asymptotically optimal algorithms like RRT*, PRM*, and Informed RRT* were excluded as their convergence criterion is path-length optimization rather than a fixed time budget, making them incomparable to single-query planners.

Three benchmark scenarios of increasing difficulty were designed (Table 1); 20 independent runs were executed per planner per scenario. The minimum turning radius $\rho_{\min}$ used for the Reeds-Shepp connection was identical for the proposed method and all OMPL baselines in each scenario, ensuring a fair comparison; the sphere tracking radius r of the continuation algorithm is reported per scenario in Table 1.

Two metrics are reported: *path length* (total Reeds-Shepp arc length, expressed in the workspace's arbitrary distance units, as returned by the OMPL Reeds-Shepp metric) and *computation time* (wall-clock milliseconds). Both the proposed method and the OMPL baselines are displayed as box plots (median, IQR (InterQuartile Range), $1.5\times$ IQR whiskers). The three intermediate outputs of the proposed method (*homotopy*, *interpolation*, *opt*) are reported separately. A *success rate* (fraction of runs within $T_{\max}$) is also reported.

4.2 Parameter Selection

The proposed planner exposes two scalar parameters. The global repulsion scale k_{rep} in (33) is *not* tuned per scenario: it is fixed to $k_{\mathrm{rep}} = 1$ in all experiments, since the spatial weighting of the repulsion is already provided by the guide function N, which scales each obstacle term automatically according to its proximity to the start-to-goal line.

The only parameter adjusted per scenario is the sphere tracking radius r of the predictor–corrector scheme (Section 2.4), set to $r = 30, 15, 15$ for Scenarios 1–3 respectively. This radius controls how far the tracking advances along the homotopy curve at each step: a small r yields finer, higher-quality trajectories at the cost of more Newton–Raphson iterations and therefore computation time, whereas a large r advances faster but can overshoot regions of high curvature and degrade path quality.

The reported values were chosen as a compromise between trajectory quality and computation time, with denser or more tightly constrained environments (Scenarios 2 and 3) using a smaller radius.

4.3 Scenario 1: Obstacle Avoidance and Parking

4.3.1 Description

The first scenario places the vehicle in a 1000×650 workspace with 5 convex polygonal obstacles. The vehicle starts at $(0, 0)$ with $\theta_0 = 0$ and must reach a goal at $(920, 400)$ with $\theta_g = 1.57$, requiring it to navigate around one obstacle and arrive at a precise position and heading–a manoeuvre analogous to parking in a constrained bay. The combination of obstacle avoidance and a heading constraint at the goal makes this a demanding test for planners that plan position and orientation jointly.

4.3.2 Results

Table 2 reports success rates for this scenario. All methods, with the exception of RRT, achieved a 100% success rate. Fig. 2 provides a box plot-based comparison in terms of path length and computation time.

Being deterministic, the homotopy stage always produces the same configuration sequence, and the only source of variance in the box plots comes from the randomized shortcutting pass. The resulting trajectory is notably more direct than those of the sampling-based planners, which explore large portions of free space before converging.

Among the baselines, a clear disproportion emerges: KPIECE and LBKPIECE terminate in under 35 ms but yield paths of 6568 and 9155 units respectively, roughly six to eight times longer than ours, while RRT spends 504 ms and still produces a path 34% longer.

RRTConnect and PRM occupy a middle ground, trading moderate computation time for reasonable path quality, though neither approaches to the length achieved by the homotopic pipeline.

The final trajectory from the last stage (*opt*) was computed in 42 ms with a path length of 1104 units, the shortest among all evaluated methods. Fig. 3 shows example trajectories.

4.4 Scenario 2: Narrow Corridor

4.4.1 Description

The second scenario tests the planner in a 1000×600 workspace dominated by two large rectangular obstacles that form a narrow horizontal corridor (Fig. 6, centre). The vehicle starts at $(120, 100)$ with $\theta_0 = 0$ and must reach $(860, 460)$ with $\theta_g = \pi/2$, requiring it to enter the corridor, traverse it horizontally, and then execute a $90°$ rotation to arrive at the goal heading. This scenario is particularly challenging for sampling-based planners because the narrow passage significantly reduces the probability of a random sample landing in the feasible region [3].

4.4.2 Results

As shown in Table 3, the narrow passage has a measurable impact on success rates: RRT drops to 90%, and RRTConnect, KPIECE, and LBKPIECE each reach 95%, while the proposed method, PRM and EST maintain 100%. The corridor's effect is more apparent in path length.

Inspecting Fig. 4, KPIECE and LBKPIECE remain dramatically longer than every other planner (5884 and 6737 units respectively), and RRTConnect and RRT converge to nearly identical lengths (1144 vs. 1145), suggesting that the bidirectional expansion offers no advantage when the feasible region is constrained.

PRM comes closest among the baselines (1103), at roughly the same computation time as the proposed method (99 ms vs. 31 ms), but still above it. The homotopy formulation is unaffected by passage width: the algebraic singularities route the trajectory through the corridor directly, without sampling, yielding the shortest result (982) at the lowest variance. Fig. 6 shows qualitative results.

4.5 Scenario 3: Dense Obstacle Field

4.5.1 Description

The third scenario uses a 4000×4000 workspace populated with 200 small square obstacles distributed across the environment (Fig. 7). The vehicle starts at the origin with $\theta_0 = 0$ and must reach $(2400, 4000)$ with $\theta_g = \pi/2$.

Table 1. Benchmark scenario parameters. $W_W \times H_W$: workspace dimensions; $l \times w$: wheelbase and half-width; n_{obs}: obstacles; $\rho_{\min}$: minimum turning radius used for the Reeds-Shepp connection (identical for all planners); r: sphere tracking radius of the continuation algorithm; $T_{\max}$: time limit

#	Scenario	$W_W \times H_W$	$l \times w$	n_{obs}	$\rho_{\min}$	r	$T_{\max}$
1	**Avoidance & parking**	1000×650	50×30	5	100	30	1 s
2	**Narrow corridor**	1000×600	40×20	2	60	15	1 s
3	**Dense obstacle field**	4000×4000	40×20	200	60	15	5 s

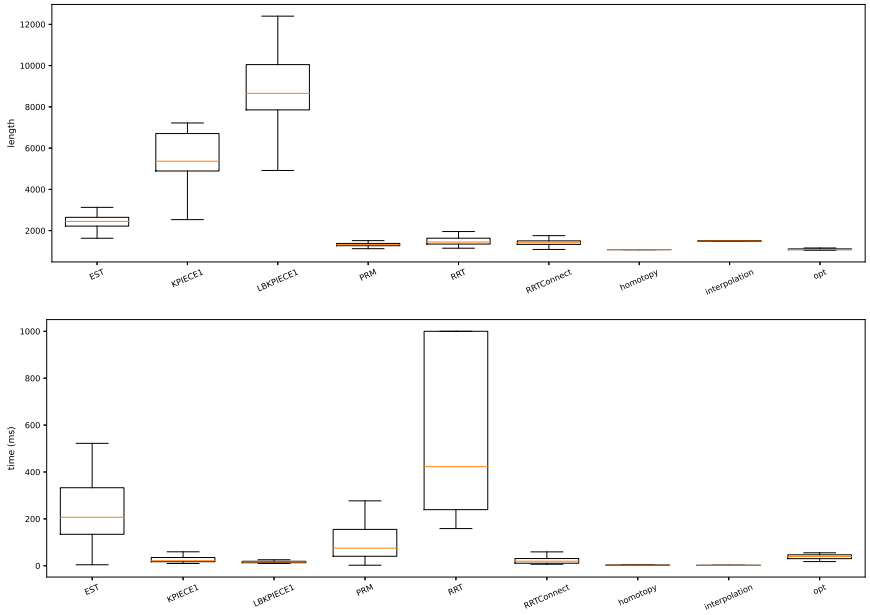

Fig. 2. Scenario 1 benchmark. *Top:* Reeds-Shepp path length. *Bottom:* computation time in milliseconds

4.5.2 Results

Table 4 and Fig. 5 show the quantitative results. With 200 obstacles over a 4000×4000 workspace, the gap between planners widens considerably.

LBKPIECE degrades to a median of 39000 units, nearly eight times longer than the proposed method, and both LBKPIECE and EST fall below 100% success rate.

A counterintuitive result is that RRTConnect, despite the environment's density, terminates in only 70 ms with a path of 5337, its best relative performance across all three scenarios; in a sparsely constrained field without tight passages, the bidirectional strategy can grow and meet quickly without getting trapped.

RRT and KPIECE, by contrast, require the most computation time among the baselines (1300 ms and 1242 ms respectively) while still producing substantially longer paths. The proposed method's aggregate time of 581 ms remains well below both, and its path (4873) is the shortest by a clear margin, with near-zero variance in the length metric given the deterministic nature of the first two pipeline stages. Fig. 7 shows example trajectories.

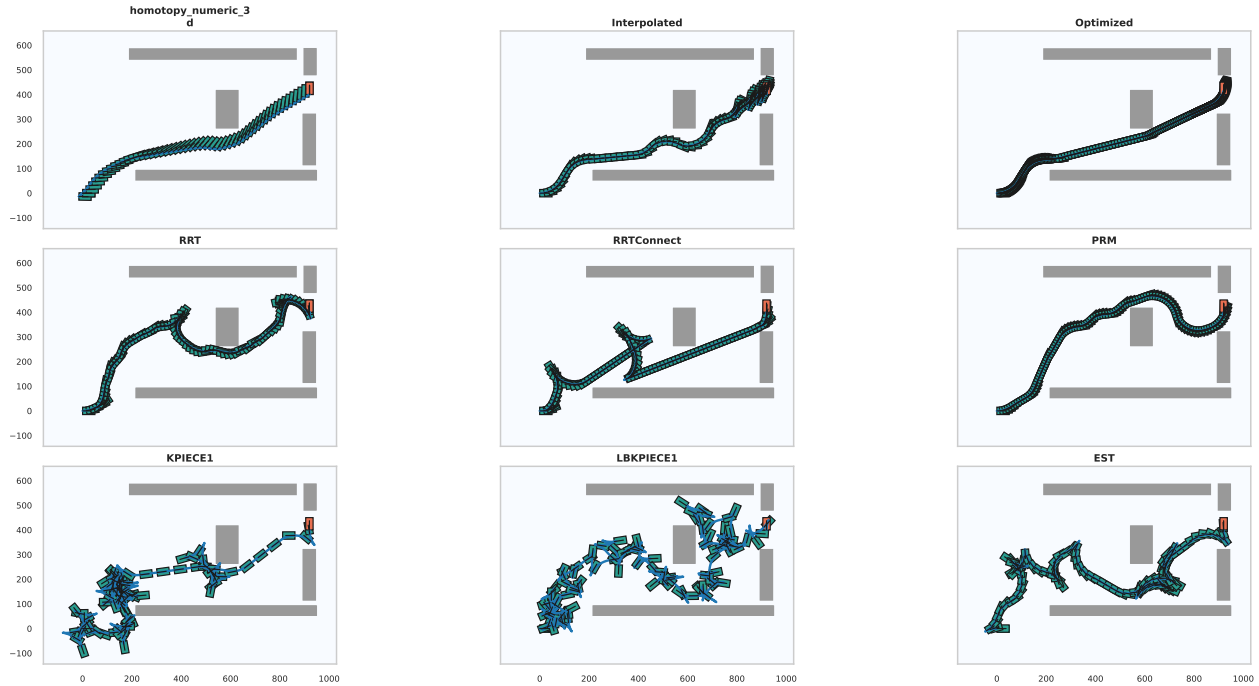

Fig. 3. Scenario 1 example trajectories

Table 2. Scenario 1 success rate (%) over 20 runs

Ours	RRT	RRTConnect	PRM	KPIECE	LBKPIECE	EST
100	75	100	100	100	100	100

Table 3. Scenario 2 success rate (%) over 20 runs

Ours	RRT	RRTConnect	PRM	KPIECE	LBKPIECE	EST
100	90	95	100	95	95	100

Table 4. Scenario 3 success rate (%) over 20 runs

Ours	RRT	RRTConnect	PRM	KPIECE	LBKPIECE	EST
100	100	100	100	100	95	95

Table 5. Per-stage median computation time of the proposed method (in ms). The per-stage medians add up to the **Ours(Opt)** time reported in Table 6

Stage	Sc. 1	Sc. 2	Sc. 3
Homotopy	6.81	1.73	263.71
Interpolation	4.30	1.92	13.36
Optimization	31.30	27.20	303.97
Total	**42.41**	**30.85**	**581.04**

4.6 Cross-Scenario Summary

Table 6 consolidates the median path length and computation time across all three scenarios.

The proposed method yields the shortest paths in every case, a result that holds without run-to-run variance in the homotopy and interpolation stages; the only spread visible in the box plots originates in the randomized shortcutting pass. Among the baselines, path quality and computation time consistently trade off in opposite directions: planners that terminate quickly do so at the cost of substantially longer paths, while those that invest more computation time achieve more reasonable lengths but never match the proposed method.

PRM offers the best balance among the baselines, maintaining competitive path lengths and moderate computation times across all three scenarios. RRTConnect exhibits a stronger scenario dependence: its bidirectional strategy is highly effective in the open dense field of Scenario 3, where it is the fastest planner overall, but offers little advantage in the narrow corridor.

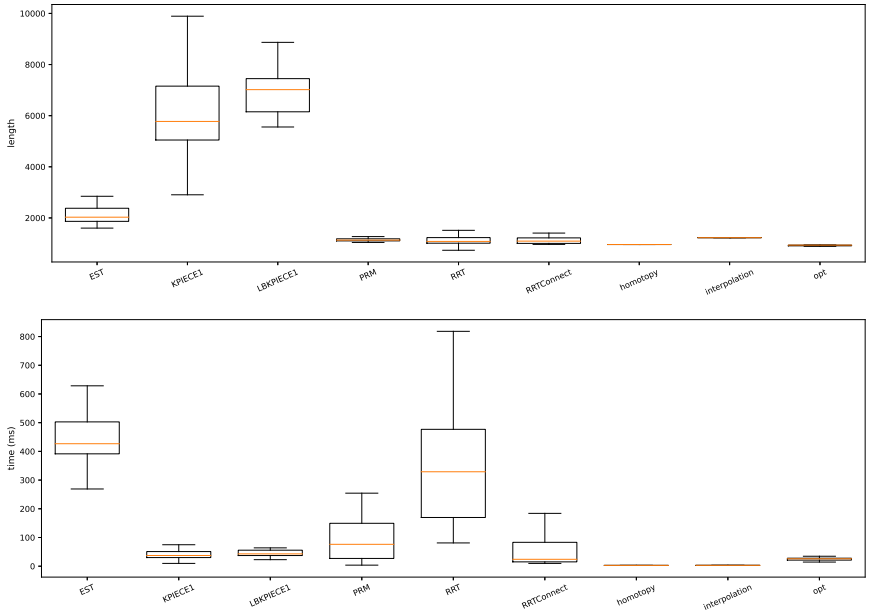

Fig. 4. Scenario 2 benchmark. *Top:* Reeds-Shepp path length. *Bottom:* computation time in milliseconds

The main computational cost of the homotopic approach lies in the Newton–Raphson corrector steps required to trace the solution curve, which scale with the complexity of the repulsion field and are therefore sensitive to the number of obstacles. This explains the increase in computation time from Scenario 1 to Scenario 3, where the reported 581 ms corresponds to the aggregate of the median times of the three pipeline stages.

Table 5 breaks down the computation time of the proposed method by pipeline stage. In the dense Scenario 3 the homotopy stage dominates, as the number of obstacles directly raises the cost of each Newton–Raphson corrector evaluation, whereas the randomized optimization stage exhibits the largest variance owing to its stochastic nature.

5 Conclusions and Future Work

This paper extended homotopy-based path planning to car-like robots through a two-stage methodology. The first stage formulates and solves a system of nonlinear equations directly in $SE(2)$, coupling position, orientation, and obstacle avoidance via a multi-point rigid-body collision model, convex-polygon barrier functions, and an adaptive guide-function repulsion field that requires no per-obstacle parameter tuning.

The second stage converts the resulting configuration sequence into a kinematically feasible trajectory by chaining shortest Reeds-Shepp curves, followed by a randomized shortcutting pass that monotonically reduces path length.

Benchmarking against six sampling-based planners across three scenarios of increasing difficulty shows that the proposed method consistently produces the shortest paths while maintaining a 100% success rate.

Its computation time is competitive in constrained and dense environments, and the deterministic structure of the homotopy and interpolation stages makes its behavior fully reproducible, a diagnostic advantage absent in stochastic planners.

As future work, the post-processing stage will be replaced by embedding the kinematic constraints of the vehicle directly into the homotopy system, seeking to produce Reeds-Shepp-compatible trajectories without a separate interpolation step.

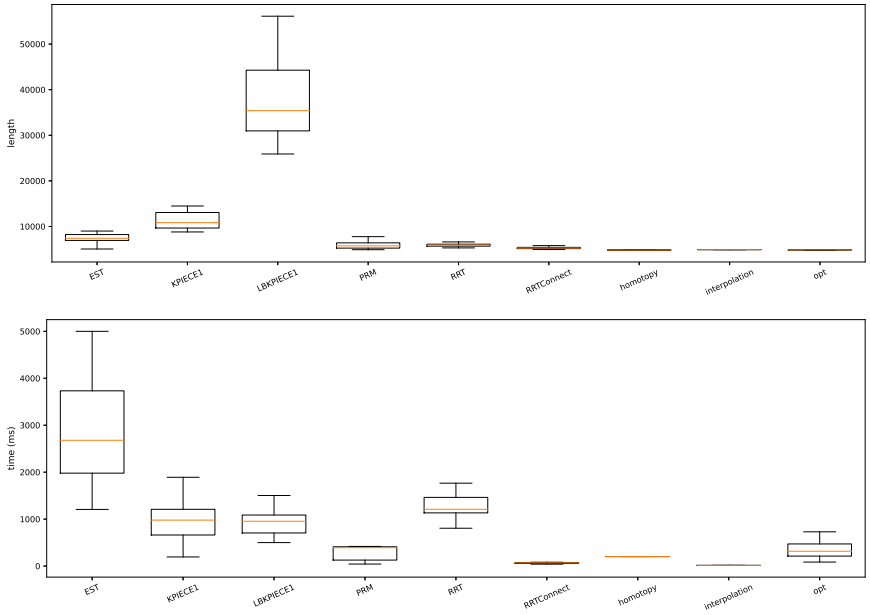

Fig. 5. Scenario 3 benchmark. *Top:* Reeds-Shepp path length. *Bottom:* computation time in milliseconds

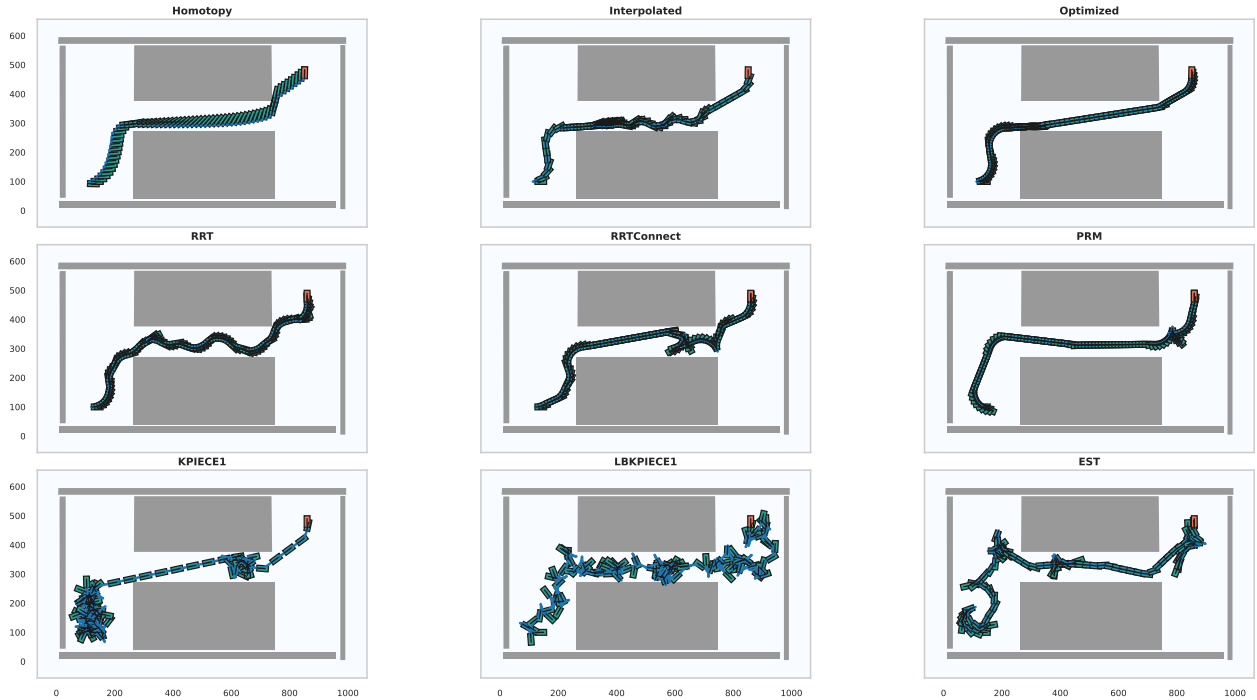

Fig. 6. Scenario 2 example trajectories

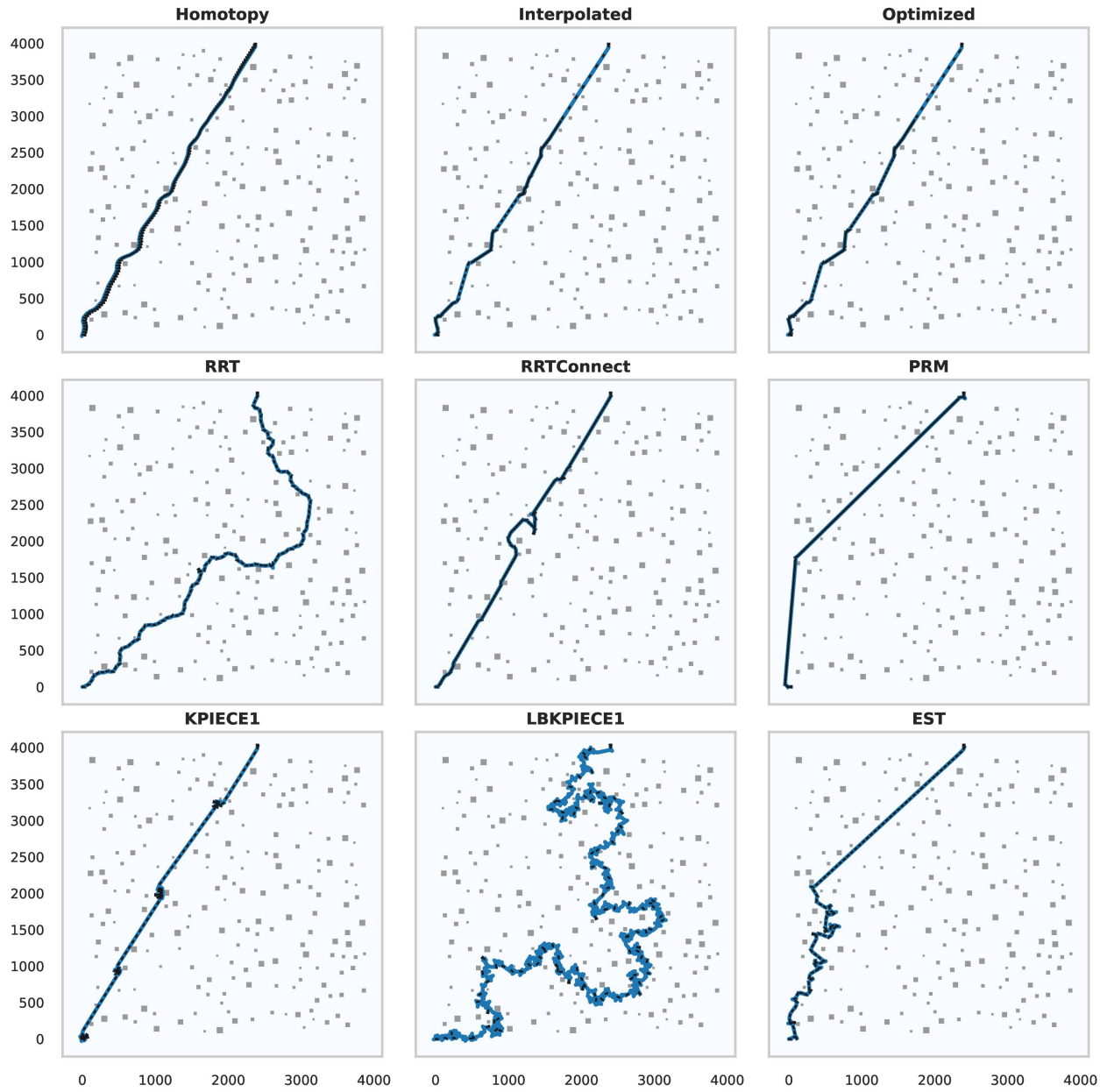

Fig. 7. Scenario 3 example trajectories

Two further directions are envisaged.

First, the planner will be validated on a real car-like platform, in order to quantify the effect of localization noise, actuation limits, and the overall simulation-to-reality gap on the trajectories produced.

Second, the formulation will be extended to environments with dynamic obstacles, for instance, by making the barrier functions time-dependent or by re-solving the continuation problem online as the obstacle configuration evolves.

Table 6. Median path length and median computation time across the three scenarios. For the proposed method, time is the sum of the median computation times of the homotopy, interpolation, and optimization stages

		Ours(Opt)	RRT	RRTConnect	PRM	KPIECE	LBKPIECE	EST
Sc. 1	Length	**1104.49**	1477.17	1392.21	1329.54	6568.39	9155.70	2412.50
	Time (ms)	42.41	504.56	21.01	101.96	33.20	17.36	256.94
Sc. 2	Length	**982.50**	1145.09	1144.19	1103.15	5884.34	6737.68	2025.43
	Time (ms)	30.85	378.27	74.86	99.65	47.16	50.85	447.25
Sc. 3	Length	**4873.55**	5978.24	5337.93	5966.53	11246.70	39000.55	7485.20
	Time (ms)	581.04	1300.56	70.01	306.90	1242.09	902.30	1111.49

Acknowledgments

This work was supported by the Office of Naval Research Global ONRG, Award No. N62909-24-1-2001, and by IIxM Secretariat of Science, Humanities, Technology, and Innovation (SECIHTI) project 745. Yu'ban Yoshua Martinez Vasquez and Gerardo Diaz-Arango gratefully acknowledge the financial support provided by the SECIHTI through academic scholarships under contract numbers 2042110 and 480421, respectively.

References

1. **Allgower, E. L., Georg, K. (2003).** Introduction to Numerical Continuation Methods. SIAM.
2. **Diaz-Arango, G., Vázquez-Leal, H., Hernandez-Martinez, L., Sanz Pascual, M. T., Sandoval-Hernandez, M. (2017).** Homotopy path planning for terrestrial robots using spherical algorithm. IEEE Transactions on Automation Science and Engineering. DOI: 10.1109/TASE.2016.2638208.
3. **Hsu, D., Latombe, J.-C., Motwani, R. (1997).** Path planning in expansive configuration spaces. Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 2719–2726.
4. **Kavraki, L. E., Švestka, P., Latombe, J.-C., Overmars, M. H. (1996).** Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation, Vol. 12, No. 4, pp. 566–580.
5. **Khatib, O. (1986).** Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research, Vol. 5, No. 1, pp. 90–98.
6. **Kuffner, J. J., LaValle, S. M. (2000).** RRT-Connect: An efficient approach to single-query path planning. Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), Vol. 2, pp. 995–1001.
7. **Latombe, J.-L. (2006).** Robot Motion Planning. Kluwer Academic Publishers, Boston.
8. **Laumond, J. P., Jacobs, P. E., Taiex, M., Murray, R. M. (1994).** A motion planner for nonholonomic mobile robots. IEEE Transactions on Robotics and Automation, Vol. 10, No. 5, pp. 577–593.
9. **Laumond, J.-P., Sekhavat, S., Lamiraux, F. (1998).** Guidelines in nonholonomic motion planning for mobile robots. In Robot Motion Planning and Control. Springer, pp. 1–53.
10. **LaValle, S. M. (1998).** Rapidly-exploring random trees: A new tool for path planning. Technical Report TR 98-11, Computer Science Department, Iowa State University.
11. **LaValle, S. M. (2006).** Planning Algorithms. Cambridge University Press. Available: `lavalle.pl/planning`.
12. **LaValle, S. M., Kuffner, J. J. (2001).** Randomized kinodynamic planning. The International Journal of Robotics Research, Vol. 20, No. 5, pp. 378–400.

13. **Lynch, K. M. (2017).** Modern Robotics. Cambridge University Press.

14. **Oliveros-Muñoz, J. M., Jiménez-Islas, H. (2013).** Hyperspherical path tracking methodology as correction step in homotopic continuation methods. Chemical Engineering Science, Vol. 97, pp. 413–429.

15. **Reeds, J. A., Shepp, L. A. (1990).** Optimal paths for a car that goes both forwards and backwards. Pacific Journal of Mathematics, Vol. 145, No. 2, pp. 367–393.

16. **Sucan, I. A., Kavraki, L. E. (2009).** Kinodynamic motion planning by interior-exterior cell exploration. Proc. Robotics: Science and Systems (RSS), pp. 449–464.

17. **Sucan, I. A., Kavraki, L. E. (2012).** A sampling-based tree planner for systems with complex dynamics. IEEE Transactions on Robotics, Vol. 28, No. 1, pp. 116–131.

18. **Sucan, I. A., Moll, M., Kavraki, L. E. (2012).** The open motion planning library. IEEE Robotics & Automation Magazine, Vol. 19, No. 4, pp. 72–82.

19. **Vázquez-Leal, H., et al. (2013).** Exploring collision-free path planning by using homotopy continuation methods. Applied Mathematics and Computation, Vol. 219, No. 14, pp. 7514–7532.

20. **Yamamura, K. (1992).** Simple algorithms for tracing solution curves. Proc. IEEE Int. Symp. Circuits and Systems (ISCAS), Vol. 6, pp. 2801–2804.

Article received on 16/04/2026; accepted on 18/07/2026.
**Corresponding author is Yu'ban Yoshua Martinez Vasquez.*