A Path Planner Enabling Second-Order Continuity Across Multiple Motion Modes of the Generalized Bicycle Model

Transportation Research Record
2020, Vol. XX(X) 1–16

DOI: 10.1177/ToBeAssigned
journals.sagepub.com/home/trr
SAGE

Abstract

In the area of mobile robotics, the generalized bicycle model stands out for its agility and diverse motion capabilities. However, a gap exists in current path planning methodologies, which often overlook the model's potential for multiple motion modes. This oversight can lead to impractical paths and collision risks in real-world scenarios, particularly in complex or constrained environments. To bridge this gap, our study introduces a novel path planning algorithm that not only utilizes the generalized bicycle model but also ensures second-order (G2) continuity. This continuity is critical for smooth and safe robot navigation, as it directly influences the consistency of velocity and steering control commands.

Our approach begins with establishing rigorous continuity conditions for path planning, ensuring G2 continuity that is vital for seamless motion mode transitions. We then systematically derive conditions for each motion mode and define clear criteria for transitioning between them. This comprehensive framework forms the basis of our path planning algorithm, which integrates the principles of the Rapidly-exploring Random Tree (RRT) concept. This algorithm is tailored to generate optimal paths that adhere to G2 continuity while accommodating the unique capability of mode-changing inherent in the generalized bicycle model.

The efficacy of our proposed method is demonstrated through a series of tests in various challenging environments, particularly highlighting its performance in scenarios involving narrow passages. Our findings indicate a substantial enhancement in path planning for mobile robots, showcasing our algorithm's ability to navigate complex environments with improved safety and efficiency. This advancement marks a significant contribution to the field of robotic path planning, offering a more dynamic and reliable approach to autonomous navigation.

Introduction and Review

Mobile robots have become an integral part of modern factories, significantly enhancing efficiency and flexibility. They autonomously handle and transport objects, navigate complex environments, and perform repetitive tasks. Mobile robots are also crucial in healthcare and in hazardous environments, assisting with a range of tasks from delivering medication to conduct search and rescue. The versatile functionality of mobile robots is enabled by their underlying design. In the following sections, we will provide an overview of the various forms of robot mobility, along with an in-depth discussion of the mechanisms that facilitate their movement, as detailed in existing literature

Existing models of mobile robots

Numerous vehicle models have been suggested, each possessing distinct characteristics. For example Figure 1 shows the generazlied model featuring two wheels, as a simplified representation to study the dynamics of a vehicle's motion. It abstracts the vehicle's behavior using two wheels, one at the front (f) and one at the rear (r), each with their own velocity vectors (V) and steering angles (θ). Based on their controllabilities of wheel steering and driving, models in the literature can be categorized in Table 1.

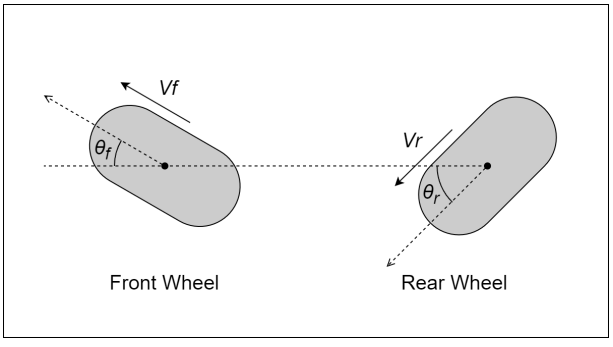


Figure 1. Generalized Bicycle Model. This type of vehicle includes two steerable and drivable wheels.

Table 1 classifies the studies in the literature according to the controllability of the models depicted in Figure 1. The "Bicycle Model", recognized for its front wheel steering and rear wheel drive, stands out as one of the most studied variants. This particular model is frequently represented by car-like mobile robots, as described by Rajamani(2).

Table 1. Vehicle research with different controllability

	Front Drive	Rear Drive	Double Drive
Front Steer	H. Chen (*1*)	R. Rajamani(*2*)	A. Bonci (*3*)
Rear Steer	L. Li (*4*)	T. A. Tamba (*5*)	A. Bonci (*3*)
Double Steer	J. Yuan (*6*)		M. Kokot (*7*)

Alternatively, some vehicles, especially those resembling forklifts, adopt a reverse configuration with the front wheel for propulsion and the rear wheel for steering, as investigated by Li et al.(*4*). Moreover, Chen et al. have highlighted potential issues such as steering deadlocks that can arise when both steering and driving functions are allocated to the front wheel(*1*), or both to the rear(*5*).

Yuan et al.'s work is particularly relevant to our exploration of dual-wheel steering systems, especially within the context of Tractor-Trailer Mobile Robots (TTMR). In (*6*), they developed control laws for managing the driving speed and steering angles of the tractor, along with the steering angles of the trailer, closely paralleling the dual-wheel steering with front wheel drive design in Table 1. Their subsequent study (*8*) advanced this framework by integrating multiple steerable trailers, thus evolving the design into a more complex structure that simulates a system with several steering wheels all governed by a single front driving wheel. Although we have not come across a model that combines dual-wheel steering with rear wheel driving, this setup could theoretically be envisioned as a TTMR operating in reverse. While there are studies like those by Morales et al. and Widyotriatmo et al. that delve into the reverse movements of TTMR(*9*, *10*), they do not extend to cover dual-wheel steering functionality. This omission marks it as a relatively uncharted domain in the research.

In the context of double-wheel drive integrated with either front or rear wheel steering, this setup effectively represents an all-wheel-drive bicycle model. Bonci et al.(*3*) have not only introduced but also thoroughly examined the dynamics of this particular configuration. Despite their contributions, the authors acknowledge that this field of study still holds many opportunities for further development and exploration.

Kokot et al.(*7*) address the less-explored model of double steering and driving in Table 1, suggesting its analysis through the Generalized Bicycle Model (GBM). Equipping all wheels of GBM robots with both steering and driving capabilities is theorized to significantly enhance their agility and flexibility, making them comparable to omni-directional mobile robots equipped with Mechanum wheels. Yet, unlike the omni-directional counterparts, GBM robots circumvent the maintenance complexities and the increased costs associated with sophisticated wheel designs. Most importantly, the control strategies and analytical approaches developed for GBM robots can be adapted to other models by accounting for specific wheel constraints. Therefore, our research concentrates on exploring GBM-based mobile robots, dissecting their models, and effectively leveraging their features in practical implementations.

Methods in mobile robot path planning

Navigation is a fundamental capability of mobile robots. The key technologies involved include simultaneous localization and mapping (SLAM), path planning, and path following, among others. SLAM plays a critical role in real-time localization of the vehicle and mapping of the surroundings, while path planning is tasked with creating routes that navigate around obstacles. Conversely, path following ensures the robot adheres to a set path while navigating around dynamic impediments. Although SLAM has become quite refined and stable, path planning and following remain central areas of interest for numerous researchers. Effective navigation depends on precisely defined paths and robust control strategies. A well-considered path not only improves tracking accuracy but also reduces the complexity of control tasks. As such, this study will primarily concentrate on advancing path planning.

Path planning involves a range of strategies, including graph search techniques (Dijkstra and A*), as well as sampling-based methods (RRT and PRM)(*11*). Among these existing methods, RRT's random nature and node expansion capabilities make it particularly adept at integrating the motion modes and constraints specific to the GBM. Consequently, this research adopts the RRT algorithm as the primary method for planning with GBM-based vehicles. Moreover, the continuity of the path is of paramount importance as it seeks to improve the controllability of local trajectories(*12*) and lessen the demand on the vehicle's control mechanisms. A smoother path effectively diminishes control oscillations and reduces the margin of tracking errors(*7*). For example, Huh and Chang(*13*) studied the various definitions of path continuity, highlighting that G_2 continuity signifies a path with continuous curvature, implying continuous acceleration and a reduction in the burden on the dynamic system. To address this concern, the authors parameterized each set of path points using a second-order polynomial. They employed a membership function from Fuzzy Control Theory to manage the overlap of adjacent sets of points, enabling their combination into a third-order polynomial and thus ensuring G_2 continuity. In addition, Yang et al.(*14*) introduced a spline-based RRT planning method. The authors incorporated conditions such as the maximum curvature and the maximum angle for node expansion, thereby constraining the range of expanded RRT nodes. They further smoothed the path points using a third-order Bezier curve, ensuring G_2 continuity. This approach not only diminishes the control burden on the robot but also satisfies the constraints of non-holonomic motion.

Vehicle kinematics has also been incorporated into path planning to ensure the selected global path appropriately accounts for the vehicle's motion behavior. For instance, in the study by Ghosh et al.(*15*), the authors integrated the kinematic constraints of non-holonomic mobile robots into the bi-directional RRT method, ensuring that the kinematic relations between adjacent nodes are maintained during node expansion. However, when connecting two RRT trees, the authors noted potential issues with kinematic continuity. To address this, they employed the Parametrized Trajectory Generator (PTG)(*16*) method to reconstruct a trajectory for the connection. Despite the consideration of kinematic constraints, this method only guarantees G_1 continuity, indicating continuous tangential orientation. Additionally, Wang et al.(*17*) also introduced a bi-directional RRT planning method that considers kinematic constraints. While the authors incorporated kinematic constraints into the node expansion process, there is a lack of careful consideration for continuity in the candidates of discrete control inputs. Consequently, this approach can only ensure continuous curvatures within a single node period and does not guarantee overall planning time continuity, ultimately achieving only G_1 continuity. Besides, Kang et al.(*18*) introduced a dynamic smoothing method that considers kinematic constraints. The authors, based on kinematic limitations, modified the connection between points by combining arcs and lines as a rewiring technique. Despite incorporating kinematic constraints, the junctions between arc and line segments cannot ensure curvature continuity but only tangential direction continuity, resulting in the satisfaction of only G_1 continuity.

Remaining challenges and the objectives of this work

The aim of integrating kinematic constraints into path planning is to derive a path that aligns with the robot's motion capabilities. Yet, current methodologies typically achieve only G_1 continuity. Research, such as that by Kokot et al.(*7*), has shown that navigating a robot along a G_1 continuous path can induce oscillatory behaviors, which are notably reduced when following a G_2 continuous path. This finding highlights the critical need for research focused on developing paths with G_2 continuity, particularly for enhancing stability and efficiency in robot navigation within the framework of the GBM.

Creating paths that capitalize on a robot's agility is essential for enhancing operational efficiency and adaptability in varied environments. Therefore, this study employs the Generalized Bicycle Model (GBM) as the basis for examining motion modes and developing a path planner that ensures smooth (G_2) continuity and enable mode-changing within GBM. In what follows, Section 2 explores the GBM's motion modes and the constraints of transitioning between these modes, crucial for understanding robot maneuverability. Section 3 introduces our proposed path planning methodology along with a continuity analysis, ensuring that robots move smoothly and efficiently. Section 4 describes the specific case environments used to test our path planning approach, highlighting practical applications and challenges. Finally, Section 5 summarizes the key contributions of this study, discusses ongoing considerations in the field of robotics, and outlines directions for future research, emphasizing the real-world impact of improved robotic path planning.

Motion Modes

The generalized bicycle model (GBM) features several distinct motion modes. In the research by Kokot(*7*), the author showed that adjusting the orientation of the GBM robot under the same path can have varying effects on its motion behavior, potentially causing it to move closer to the wall. Consequently, the author designed motion modes for GBM based on the robot's orientation. However, the study in (*7*) lacks explicit mention of conditions governing the transition between different motion modes during movement. For example, the crab mode, enabling parallel translation, may transit to the tangential mode, aligning the robot's orientation tangent to the path.

We intend to utilize these modes as a foundation for developing an overarching global planner. The definitions based only on orientation may overlook the continuous nature of wheel controls during mode transitions. To address these issues, our research defines motion modes based on wheel states, thoroughly examines motion behavior, establishes switch conditions, and ultimately clarifies the intricate relationship between motion modes and path planning. Before we delve into the path planner algorithm, it's imperative to fully articulate each mode.

Definitions of motion modes

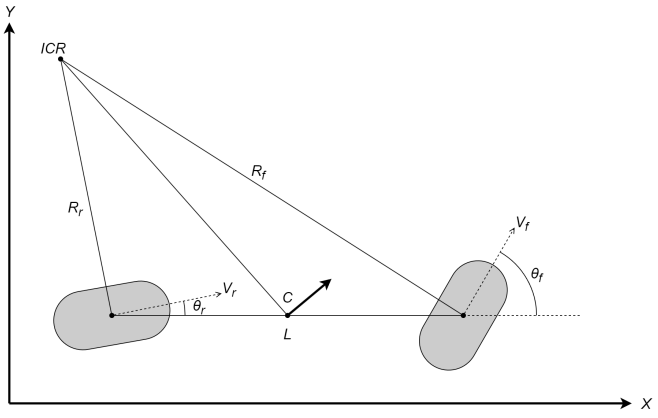


Figure 2. The triangular relations for a standard bicycle model.

Given the robot's characteristics of low-speed motion, our analysis is based on kinematics. We assume the presence of two wheels positioned at the front and rear of the vehicle's center point, as depicted in Figure 2, with the wheelbase denoted as L. The standard analysis of the bicycle model based on triangular relations between the turning radius and the wheelbase of the vehicle show that

$$\begin{aligned} \frac{L}{\sin(\theta_f - \theta_r)} &= \frac{R_f}{\sin(\frac{\pi}{2} + \theta_r)} = \frac{R_r}{\sin(\frac{\pi}{2} - \theta_f)} \\ \frac{V_f}{R_f} &= \frac{V_r}{R_r} = \omega \end{aligned} \tag{1}$$

θ_f and θ_r in Eq.(1) represents the steering angle of the front and rear wheel, respectively; V_f and V_r are the translational velocity of the centers of the front and rear wheels. ω is the pure rolling angular velocity of wheels; R_f and R_r are the distance of the front and rear wheel to the instantaneous center of rotational (ICR). Following the relationships, we consider the variables $\{V_f, V_r, \theta_f, \theta_r\}$ as control inputs, and accordingly define the motion modes for GBM robots in the following sections.

Ackerman and tangential modes Figure 3 illustrates an Ackermann steering geometry with its kinematic relationship expressed by Eq.(2), where $R_f \neq 0$ or $R_r \neq 0$ according to Eq.(1).

$$\begin{aligned} \frac{\cos(\theta_f)}{\cos(\theta_r)} &= \frac{R_r}{R_f} = \frac{V_r}{V_f} \\ V_r &= V_f \cdot \frac{\cos(\theta_f)}{\cos(\theta_r)} \end{aligned} \tag{2}$$

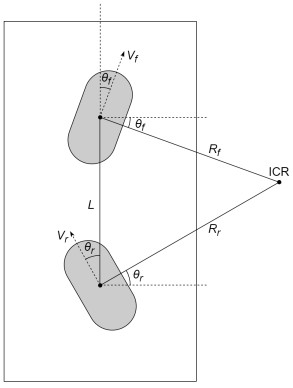
The Ackermann steering geometry mode adjusts the vehicle's orientation based on the position of the ICR. A clear illustration is when the line connecting the ICR and the center of the vehicle is perpendicular to the vehicle (refer to Figure 3b), resulting in the orientation being tangent to the path during motion. This specific scenario is referred to as the "Tangential" mode, with its conditions defined by Eq.(3).

$$\begin{aligned} V_f(t) &= V_r(t) \\ \theta_f(t) &= -\theta_r(t) \\ t &\geq 0 \end{aligned} \tag{3}$$

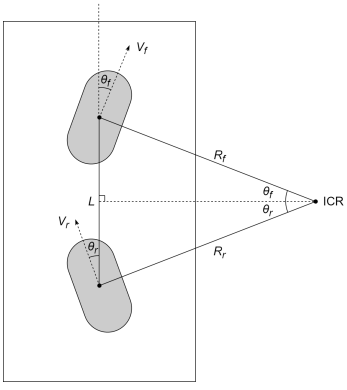
The Ackermann mode may have three scenarios depending on the steering geometry, namely:

- Two wheels with parallel steering angles: $\theta_f = \theta_r$
- Both wheels with steering angles at 90 degrees: $\theta_f = \theta_r = \pm\frac{\pi}{2}$
- One wheel with a steering angle at 90 degrees and the other without, indicating $R_f \neq 0 \;\&\; R_r = 0$ or $R_f = 0 \;\&\; R_r \neq 0$.

In response to the first two scenarios, we introduce the "Crab" and "Differential Drive" modes. However, the third situation results in the GBM becoming immobile, and thus, it should be actively avoided during planning and control phases.



(a) General Ackermman steering geometry visualization.



(b) Specific condition of Ackermann steering geometry, defined as "Tangential" mode.

Figure 3. Ackermann mode.

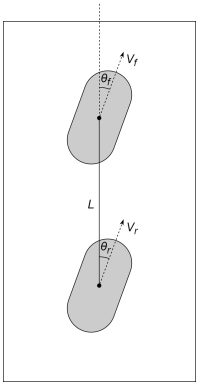


Figure 4. Crab mode

Crab mode Figure 4 depicts the Crab mode of GBM robots, characterized by parallel steering angles of the wheels that deviate from the traditional Ackerman steering geometry. This unique configuration enables a special motion control mechanism where the robot propels the wheels at a uniform velocity, enabling translational movement without changing its heading angle, reminiscent of a crab's sideway

locomotion. Consequently, the detailed definition of the "Crab" mode is articulated in Eq.(4).

$$\begin{aligned} V_f(t) &= V_r(t) \\ \theta_f(t) &= \theta_r(t) \\ t &\geq 0 \end{aligned} \tag{4}$$

Differential drive mode When both steering angles of a GBM reach $\pm\frac{\pi}{2}$, as depicted in Figure 5, the robots can behave similar to differential drive mobile robots by driving the wheels with different velocities. As a result, this configuration is called "Differential drive" mode, and is defined as Eq.(5).

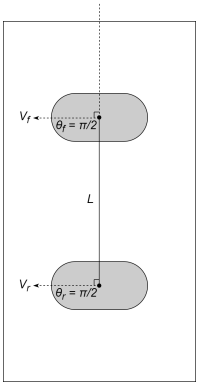


Figure 5. Differential drive mode

$$\begin{aligned} \theta_f(t) &= \pm\frac{\pi}{2} \\ \theta_r(t) &= \pm\frac{\pi}{2} \\ t &\geq 0 \end{aligned} \tag{5}$$

Switch conditions of motion modes In this section we define the transition relationships between the modes, as depicted in Figure 6. Central to this figure is the Ackermann steering geometry, which serves as a pivot for transitioning between the modes. The 'Crab' mode is indicated where both front (θ_f) and rear (θ_r) wheel angles are aligned, facilitating lateral movement without a change in orientation. This contrasts with the 'Differential Drive' mode, where θ_f and θ_r are opposed by an angle of $\pi/2$, allowing rotation about the center without lateral translation. The 'Tangential' motion mode is a subset of Ackermann steering where the velocity at the front (V_f) equals the velocity at the rear (V_r), and the wheel angles are parallel, enabling the model to execute smooth, curvilinear paths. The switch between modes can be achieved by ensuring the continuity of wheel control states. Taking Crab and Ackermann Steering Geometry modes as an example, the boundary conditions for both modes are $V_f = V_r$ and $\theta_f = \theta_r$. Therefore, the vehicle can seamlessly switch its behavior by adjusting its wheel states to meet these conditions.

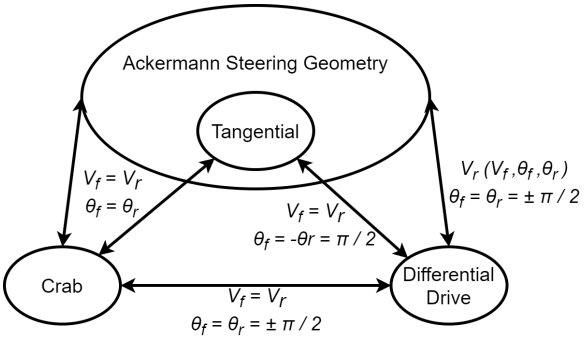


Figure 6. Relationship of motion modes.

Summary and visualization of GBM modes Building upon the defined motion modes, we illustrate their respective behaviors in Figure 7, with the Ackermann steering geometry mode exemplified by the intuitive Tangential mode.

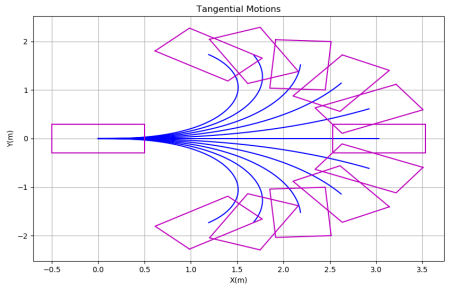
Figures in 7a, 7b, and in 7c provide graphical representations of the trajectories followed by a robot employing the tangential, crab, and differential steering principles, respectively. The figures showcase a series of paths generated by varying the steering angles. The lines trace the robot's motion when the wheels turn at different steering angles. We can see a spectrum of motions of each mode. For example in Fig.7a, the overlay of these paths captures the robot's capability to navigate complex maneuvers, illustrating the control over direction and the ability to negotiate around obstacles with minimal changes to its orientation.

Exploring motion modes in path planning considerations

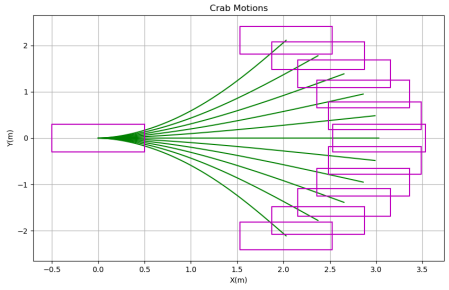
Not all paths are equally viable across all modes of transport; certain routes may be less suitable or inaccessible for specific methods of travel. While many path planning methods are tailored for non-holonomic vehicles, accommodating a single motion mode such as 'Ackermann' or 'Differential Drive,' it becomes imperative to develop planners that consider diverse kinematics, especially when dealing with mobile robots like GBM robots and omni-directional robots. Traditional path planning methods often fall short when applied to these robots, primarily due to the undefined nature of their motion modes, posing significant challenges to their effectiveness.

To illustrate this challenge more clearly, Figure 8 depicts a comparison of robots with varying sizes operating in different modes. It becomes apparent that robots following the same path occupy different amounts of space, depending on their mode, which we refer to as the 'coverage area'. This coverage area represents the space that must be free from obstacles. Therefore, a larger coverage area implies greater space requirements, making it less preferable compared to a robot with a smaller coverage area.

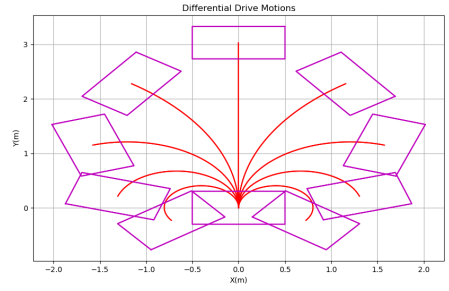
The path depicted in Figure 8 consists of three linear segments and two turns. To delve deeper into this, we focus



(a) Tangential mode motions.



(b) Crab mode motions.



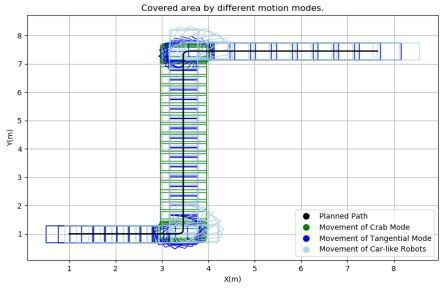
(c) Differential drive mode motions.

Figure 7. Visualization of the GBM motions. Purple rectangles represent poses of GBM vehicle.

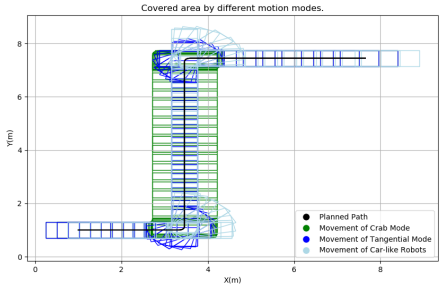
solely on the path segments—straight lines and corners—as shown in Figure 9. The coverage areas associated with these segments are comprehensively detailed in Table 2 and Table 3.

Observing the figures and the tables, it's evident that Car-like robots require a significant amount of space to navigate corners due to the unsteerable rear wheel. In contrast, the GBM robot in Tangential mode does not face the same constraint. Additionally, when comparing the GBM robot in Crab mode to the other two modes, both Tangential and Car-like movements necessitate adjustments to the robot's orientation during turning, requiring more space. However, Crab mode enables movement without altering the robot's heading, potentially enhancing coverage when the orientation is not parallel to the path.

As conventional path planning methods are not designed to accommodate multiple motion modes, these issues remain unaddressed, impacting both collision detection and obstacle avoidance during planning. Therefore, our objective is to directly incorporate the consideration of motion modes into our path planning process. This approach not only resolves the issue of collision detection with obstacles but also enables the explicit definition of which mode to use in different path segments, thereby eliminating confusion in mode selection.



(a) Covered areas with different movements and a shorter robot length.



(b) Covered areas with different movements and a longer robot length.

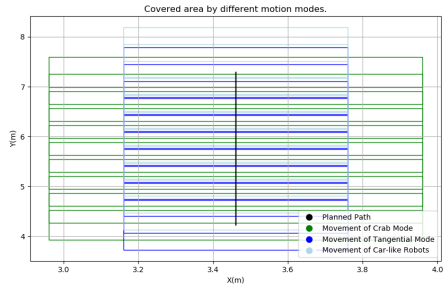
Figure 8. Visualization of covered areas by different movements.

Table 2. Coverage Area Comparison Across Various Modes and Robot Sizes During Straight Line Navigation (Refer to Figure 9a).

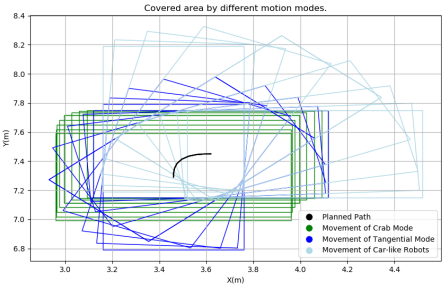
	Length × Width $= (1.0m \times 0.8m)$	Length × Width $= (1.5m \times 0.8m)$
Crab	3.7168 (m^2)	5.5568 (m^2)
Tangential	2.4480 (m^2)	2.7480 (m^2)
Car-like Robots	2.4480 (m^2)	2.7480 (m^2)

Methodology

Based on the discourse in Section , this section proposes a path planner that takes into consideration the motion modes of GBM kinematics and satisfies the G_2 continuity requirement. The motion mode constraints are contingent



(a) Straight lines navigation.



(b) Corners navigation.

Figure 9. Extracted path segments for robots navigation.

Table 3. Coverage Area Comparison Across Various Modes and Robot Sizes During Corner Navigation (Refer to Figure 9b).

	Length × Width $= (1.0m \times 0.8m)$	Length × Width $= (1.5m \times 0.8m)$
Crab	0.8630 (m^2)	1.2369 (m^2)
Tangential	1.0433 (m^2)	1.8116 (m^2)
Car-like Robots	1.3531 (m^2)	2.2306 (m^2)

upon the wheel states, leading us to formulate the path planner utilizing the Rapidly-exploring Random Tree (RRT) method. Our approach draws inspiration from the work of Wang et al. in (*17*), incorporating node expansion with control inputs while simultaneously addressing the constraints. The control inputs encompass the steering angles and velocities of the wheels. In addition, to adhere to the G_2 continuity requirement, we conduct a thorough analysis of the kinematic model, deducing the essential conditions for the path planner. The subsequent section will commence by elucidating the continuity analysis, followed by an in-depth discussion on the design of polynomial functions for G_2 paths, and ultimately, the comprehensive introduction of our RRT-based path planner.

Continuity Analysis

The kinematic model of GBM, as articulated by Kelly in (*19*), is expressed in Eq.(6) and (7). Here, the position vectors and velocities of the wheels are represented as $(x_i, y_i)i = f, r$ and $(Vix, V_{iy})_{i=f,r}$, while (V_x, V_y, ω) denote the linear and angular velocities of the vehicle.

- Forward Kinematics:

$$v_c^w = \begin{bmatrix} V_{fx} \\ V_{fy} \\ V_{rx} \\ V_{ry} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -y_f \\ 0 & 1 & x_f \\ 1 & 0 & -y_r \\ 0 & 1 & x_r \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} = H_c^v \dot{x_v^w} \tag{6}$$

- Inverse Kinematics:

$$\dot{x_v^w} = \left[(H_c^v)^T (H_c^v) \right]^{-1} (H_c^v)^T v_c^w \tag{7}$$

Additionally, the wheel velocity can be alternatively expressed using Eq.(8).

$$\begin{aligned} V_{fx} &= V_f \cdot \cos(\theta_f) \\ V_{fy} &= V_f \cdot \sin(\theta_f) \\ V_{rx} &= V_r \cdot \cos(\theta_r) \\ V_{ry} &= V_r \cdot \sin(\theta_r) \end{aligned} \tag{8}$$

Combining Eq.(6 - 8), a substitution for the inverse kinematics can be derived as:

$$\begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{L} & 0 & \frac{-1}{L} \end{bmatrix} \begin{bmatrix} V_f \cos\theta_f \\ V_f \sin\theta_f \\ V_r \cos\theta_r \\ V_r \sin\theta_r \end{bmatrix} \tag{9}$$

This shows how one can obtain the velocities of the GBM model given the control of velocity and the steering angle of front and rear wheels.

G_2 continuity refers to the uninterrupted progression of curvature, ensuring a smooth transition at the juncture of differentiable segments(*13*). To investigate the continuity of a path by wheel command, let us first define the curvature of a path as Eq.(10).

$$\kappa = \frac{|\ddot{x}\dot{y} - \dot{x}\ddot{y}|}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}} \tag{10}$$

In Eq.(10), the curvature depends on the 1st and 2nd-order derivatives of the robot poses in the global coordinate system, which can be calculated using the fundamental kinematics of vehicles and their derivatives as presented in Eq.(11) and (12).

$$\begin{aligned} \dot{x} &= V_x \cos\theta - V_y \sin\theta \\ \dot{y} &= V_x \sin\theta + V_y \cos\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{11}$$

$$\begin{aligned} \ddot{x} &= \dot{V}_x \cos\theta - V_x\omega \sin\theta - \dot{V}_y \sin\theta - V_y\omega \cos\theta \\ \ddot{y} &= \dot{V}_x \sin\theta + V_x\omega \cos\theta + \dot{V}_y \cos\theta - V_y\omega \sin\theta \\ \ddot{\theta} &= \dot{\omega} \end{aligned} \tag{12}$$

Additionally, obtaining the 1st-order derivative of (V_x, V_y, ω) is necessary for curvature calculation. Hence, we differentiate Eq.(9) and present the result in Eq.(13).

$$
\begin{aligned}
\dot{V_x} &= \frac{1}{2}(\dot{V_f}\cos\theta_f - V_f\dot{\theta_f}\sin\theta_f + \dot{V_r}\cos\theta_r - V_r\dot{\theta_r}\sin\theta_r) \\
\dot{V_y} &= \frac{1}{2}(\dot{V_f}\sin\theta_f + V_f\dot{\theta_f}\cos\theta_f + \dot{V_r}\sin\theta_r + V_r\dot{\theta_r}\cos\theta_r) \\
\dot{\omega} &= \frac{1}{L}(\dot{V_f}\sin\theta_f + V_f\dot{\theta_f}\cos\theta_f - \dot{V_r}\sin\theta_r - V_r\dot{\theta_r}\cos\theta_r)
\end{aligned} \tag{13}
$$

Integrating Eq.(9-13), a new expression for curvature can be computed as shown in Eq.(14). In this equation, continuous curvature is achievable as long as the wheel states are designed to be continuous, and the robots avoid zero velocity, thereby preventing the occurrence of a zero denominator in the equation.

$$
\begin{aligned}
\kappa &= \frac{|\ddot{x}\dot{y} - \dot{x}\ddot{y}|}{(\dot{x}^2+\dot{y}^2)^{\frac{3}{2}}} \\
&= \frac{1}{\frac{1}{8}\left(V_f^2 + V_r^2 + 2V_fV_r\cos(\theta_f-\theta_r)\right)^{\frac{3}{2}}} \\
&\times \left| \frac{-1}{4L}\left(V_f^3\sin\theta_f + V_fV_r^2\sin\theta_r - V_f^2V_r\sin\theta_r - V_r^3\sin\theta_r\right)\right. \\
&+ \frac{-1}{4L}\left(2V_f^2V_r\sin\theta_f\cos(\theta_f-\theta_r) - 2V_fV_r^2\sin\theta_r\cos(\theta_f-\theta_r)\right) \\
&+ \frac{1}{4}\left(-V_f^2\dot{\theta_f} - V_r^2\dot{\theta_r} + (V_f\dot{V_r} - V_r\dot{V_f})\sin(\theta_f-\theta_r)\right) \\
&+ \left.\frac{1}{4}\left(-V_fV_r(\dot{\theta_f}+\dot{\theta_r})\cos(\theta_f-\theta_r)\right)\right|
\end{aligned} \tag{14}
$$

Based on the aforementioned derivations, G_2 continuous paths can be achieved through path planning that considers appropriate continuous wheel states. Consequently, in our proposed path-finding algorithm, 2nd-order polynomial functions are formulated for the wheel states to meet the G_2 continuity requirement, facilitating a seamless control of GBM robots. Further elaboration on the specifics of these polynomial functions will be provided in the subsequent section.

Design of Polynomial Functions for G_2 Continuous Paths

The analysis of continuity underscores the necessity for wheel states to be governed by polynomial functions that uphold continuous first-order derivatives. Moreover, within each polynomial function, a fixed time period T is taken into consideration to regulate the wheel states. Consequently, we employ a 2nd-order polynomial function, as outlined in Equation (15), to describe the function $f_N(t)$ and its derivatives $f'_N(t)$, with N denoting the number of nodes in a path during planning. This variable also signifies the enumeration of polynomials originating from the initial state.

$$
\begin{aligned}
f_N(t) &= a_Nt^2 + b_Nt + c_N \quad && t\in[0,T) \\
f'_N(t) &= 2a_Nt + b_N && t\in[0,T)
\end{aligned} \tag{15}
$$

Moreover, the connection between polynomial functions across adjacent time intervals is expressed as:

$$
\begin{aligned}
f_N(t=T) &= f_{N+1}(t=0) \\
f'_N(t=T) &= f'_{N+1}(t=0)
\end{aligned} \tag{16}
$$

Eq.(16) describes the continuity between two adjacent polynomials, which can be utilized to calculate the parameters, b_N and c_N. For example, for the two adjacent polynomials f_0 and f_1, we have :

$$
\begin{aligned}
f_0(t) &= a_0t^2 + b_0t + c_0 \\
f'_0(t) &= 2a_0t + b_0 \\
f_1(t) &= a_1t^2 + b_1t + c_1 \\
f'_1(t) &= 2a_1t + b_1
\end{aligned} \tag{17}
$$

Because f_0 represents the initial polynomial, hence $f_0(t=0)=0$ and $f'_0(t=0)=0$ is assumed, resulting in $b_0=0$ and $c_0=0$. Then Eq.(16) shows that

$$
\begin{aligned}
f_0(t=T) = f_1(t=0) &\Longrightarrow a_0T^2 = c_1 \\
f'_0(t=T) = f'_1(t=0) &\Longrightarrow 2a_0T = b_1
\end{aligned} \tag{18}
$$

Thus, by iteratively applying the continuity conditions to the N polynomials, the parameters b_N and c_N can be derived through the recursive relation outlined in Equation (19).

$$
\begin{aligned}
b_0 &= 0, \quad c_0 = 0 \\
b_N &= 2a_{N-1}T + b_{N-1} && N>0 \\
c_N &= a_{N-1}T^2 + b_{N-1}T + c_{N-1} && N>0
\end{aligned} \tag{19}
$$

However, it is noteworthy that the parameter a_N remains independent of the preceding polynomials; its determination is governed by the boundary conditions. These conditions are derived from both the kinematic constraints of GBM and the capabilities of the wheel actuators. To determine the value of a_N, we start by defining the boundary conditions for the polynomial, as indicated in Equation (20).

$$
\begin{aligned}
f_{\min,N} \le f_N(t) \le f_{\max,N} \\
f'_{\min,N} \le f'_N(t) \le f'_{\max,N}
\end{aligned} \tag{20}
$$

Subsequently, the first-order derivative of the polynomial is designed as a symmetric piece-wise linear function, as described by Equation (21) and illustrated in Figure 10.

$$
f'_N(t) = \begin{cases} 2a_Nt, & 0\le t\le \frac{T}{2} \\ -2a_N(t-T), & \frac{T}{2} < t \le T \end{cases} \tag{21}
$$

The integration of the first-order derivative, representing the function's change, facilitates a straightforward control of the variation in $f(t)$. Consequently, it becomes more manageable to meet the boundary conditions outlined in Equation (20). Ultimately, by leveraging these boundary conditions, the range of a_N can be determined, as expressed in Equation (22).

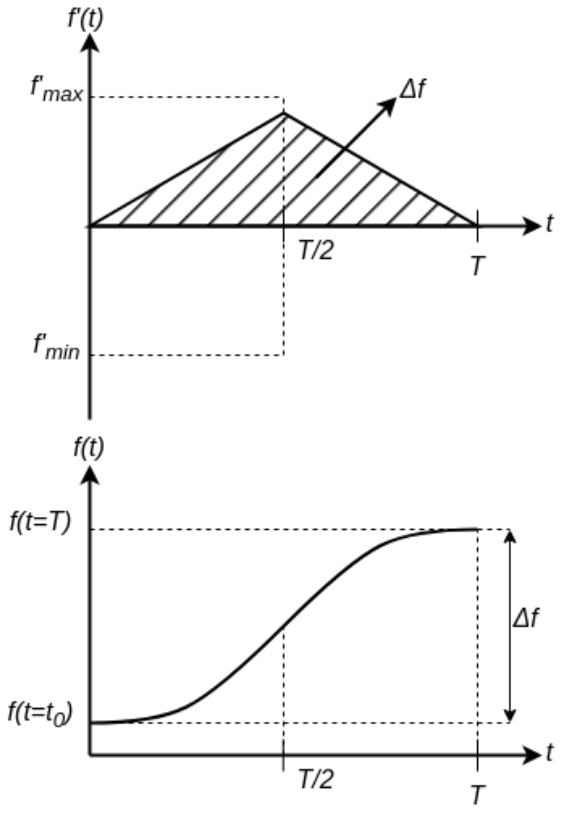


Figure 10. The calculation of polynomial's extreme values.

$$\begin{aligned} f'_{\min,N} \le a_N T \le f'_{\max,N} \\ f_{\min,N} \le f_{N-1}(T) + \frac{a_N T^2}{2} \le f_{\max,N} \end{aligned} \tag{22}$$

On the other hand, as the symmetric function is defined by two sub-equations, leading f_N to involve two sub-equations as well. Therefore, we reformulate f_N as expressed in Eq.(23).

$$f_N(t) = \begin{cases} a_N t^2 + b_N^{\text{1st}} t + c_N^{\text{1st}} & 0 \le t \le \frac{T}{2} \\ -a_N (t - \frac{T}{2})^2 + b_N^{\text{2nd}} (t - \frac{T}{2}) + c_N^{\text{2nd}} & \frac{T}{2} \le t \le T \end{cases} \tag{23}$$

For the parameters $b_N^{\text{1st}}, c_N^{\text{1st}}, b_N^{\text{2nd}}, c_N^{\text{2nd}}$, Eq.(19) can still be used, and each parameter is computed as shown in Eq.(24).

$$\begin{aligned} b_0^{\text{1st}} &= 0, \quad c_0^{\text{1st}} = 0 \\ b_0^{\text{2nd}} &= a_0 T, \quad c_0^{\text{2nd}} = \frac{T^2}{4} a_0 \\ b_N^{\text{1st}} &= -a_{N-1} T + b_{N-1}^{\text{2nd}} && N > 0 \\ c_N^{\text{1st}} &= -\frac{T^2}{4} a_{N-1} + \frac{T}{2} b_{N-1}^{\text{2nd}} + c_{N-1}^{\text{2nd}} && N > 0 \\ b_N^{\text{2nd}} &= a_N T + b_N^{\text{1st}} && N > 0 \\ c_N^{\text{2nd}} &= \frac{T^2}{4} a_N + \frac{T}{2} b_N^{\text{1st}} + c_N^{\text{1st}} && N > 0 \end{aligned} \tag{24}$$

By discretizing a_N, the corresponding parameters of $b_N^{\text{1st}}, c_N^{\text{1st}}, b_N^{\text{2nd}}, c_N^{\text{2nd}}$ can be computed, and feasible polynomial candidates for the wheel states can be established. The resulting polynomial functions, denoted as $F = \{f_0, f_1, \ldots, f_N\}$, allow the derivation of paths that are guaranteed to satisfy the G_2 continuity requirement. Implementing this design in the path planner facilitates the generation of G_2 continuous paths. Consequently, we integrate this design for the control inputs, specifically wheel states, into our proposed RRT-based planner to ensure continuity during path generation.

Path Planning

By adopting a continuous wheel input commands, from previous design approach, our proposed an RRT path planning method, which incorporates the diverse motion modes of GBM kinematics and meets the G_2 continuity requirement. This algorithm has three key components: random state sampling, node selection, and node expansion, as detailed in Algorithm 1. Subsequent sections will provide a step-by-step elucidation of the planner.

Algorithm 1 Proposed RRT-based path planning algorithm.

T.init(X_{init})
while Distance($X_{\text{goal}}, X_{\text{new}}$) $> d_{\text{lim}}$ **do**
  $x_{\text{sample}} \leftarrow$ **RandomStateSampling**()
  $X_{\text{near}} \leftarrow$ **NodeSelection**(T, x_{sample})
  $x_{\text{new}} \leftarrow$ **NodeExpansion**($T, x_{\text{sample}}, X_{\text{near}}$)
  T.add(x_{new})
end while

Random State Sampling Our algorithm begins with a randomly sampled state X_{sample} within the obstacle-free areas of the known environment. Additionally, to encourage the path to extend toward the destination, a probability P_{sample} is introduced to decide whether to consider the goal position as the sampled state. The specific sampling method is outlined in Algorithm 2.

Algorithm 2 RandomStateSampling

Ensure: x_{sample}: Sample Point.
if RandomProbability() $> P_{\text{sample}}$ **then**
  $x_{\text{sample}} \leftarrow x_{\text{goal}}$
else
  $x_{\text{sample}} \leftarrow$ **SampleInObstacleFreeAreas()**
end if
return x_{sample}

Node Selection During the node selection phase, we establish a sorted node set X_{sorted} by organizing each node in the RRT tree based on its distance to the sampled state X_{sample}. Subsequently, k nodes are chosen iteratively in

sequence from X_{sorted} . If a node $x_{\text{sorted},i}$ meets the condition where the angle between the velocity direction in the world coordinate system and the position angle relative to the node is below a predefined threshold θ_{thres}, as described in Equation (25), it is included in the neighboring node set X_{near}. The pseudocode for this node selection is presented in Algorithm 3.

$$\left|\arctan\frac{V_{\text{sorted},iy}}{V_{\text{sorted},ix}}+\theta_{\text{sorted},i}-\arctan\frac{y_{\text{sample}}-y_{\text{sorted},i}}{x_{\text{sample}}-x_{\text{sorted},i}}\right|\leq\theta_{\text{thres}} \tag{25}$$

Algorithm 3 NodeSelection

Require: T: Tree; x_{sample}: Sample Point.
Ensure: X_{near}: k Nearest Nodes.

$X_{\text{sorted}} \leftarrow$ **Distance**(T, x_{sample})
$X_{\text{near}} \leftarrow \emptyset$
for $x_{\text{sorted},i} \in X_{\text{sorted}}$ **do**
  $\theta_{\text{velocity}} \leftarrow$ **VelocityOrientaion**$(x_{\text{sorted},i})$
  $\theta_{\text{position}} \leftarrow$ **PositionOrientaion**$(x_{\text{sorted},i}, x_{\text{sample}})$
  if $||\theta_{\text{velocity}} - \theta_{\text{position}}|| \leq \theta_{\text{thres}}$ **then**
    X_{near}.add$(x_{\text{sorted},i})$
  end if
  if $|X_{\text{near}}| \geq k$ **then**
    Break
  end if
end for
return X_{near}

Node Expansion Utilizing information from the closest nodes set X_{near}, we compute potential polynomial functions for control inputs, taking into account the constraints of motion modes and kinematics. Following this, we calculate several predictive states from these candidates and assess them based on their distances to the sampled state. The optimal node, identified through this evaluation, becomes the outcome of the node expansion process. The comprehensive concept is detailed in Algorithm 4.

Algorithm 4 NodeExpansion

Require: T: Tree; x_{sample}: Sample Point; X_{near}: k Nearest Nodes;
Ensure: x_{new}: New node.

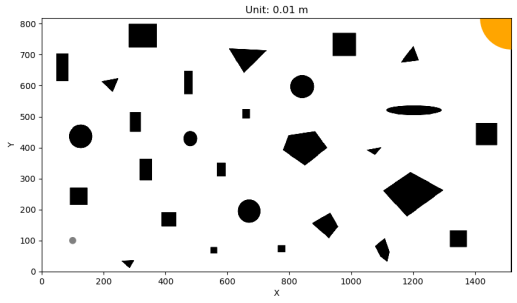
$J \leftarrow \infty$
$x_{\text{new}} \leftarrow$ **null**
for $x_{\text{near},i} \in X_{\text{near}}$ **do**
  $G \leftarrow$ **GetMotionModeConstraints**$(x_{\text{near},i})$
  $\Theta_{f,i}, \Theta_{r,i} \leftarrow$ **DiscreteSteerPolynomials**$(x_{\text{near},i}, G)$
  $V_{f,i}, V_{r,i} \leftarrow$ **DiscreteVelocityPolynomials**$(x_{\text{near},i}, G)$
  $X_{\text{new},i} \leftarrow$ **ComputePose**$(x_{\text{near},i}, \Theta_{f,i}, \Theta_{r,i}, V_{f,i}, V_{r,i})$
  for $x_j \in X_{\text{new},i}$ **do**
    $J_j \leftarrow ||x_{\text{sample}} - x_j||_2$
    if $J_j < J$ **then**
      $J \leftarrow J_j$
      $x_{\text{new}} \leftarrow x_j$
    end if
  end for
end for
return x_{new}

Case Studies and Result Discussions

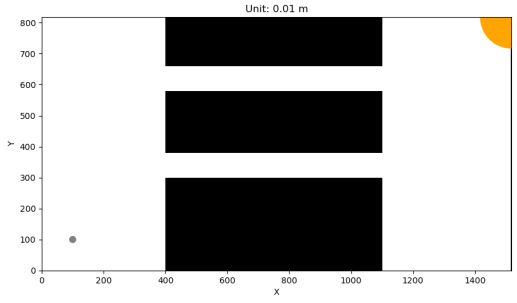
In this section, we will establish test scenarios to emphasize the impact of motion modes on path planning. Initially, we employ the spline-based RRT (SRRT) method (*14*) to acquire global paths. Subsequently, we pinpoint the issue arising from the oversight of motion modes. Following that, we test our proposed method in the same scenarios and conduct a thorough analysis of the results.

To begin with, we introduce the settings of the environments and the robot. The robot and environment assumptions are detailed as follows. The robot has dimensions of $1\text{m} \times 0.6\text{m}$, with two steerable wheels positioned at the front and back of the center points and a wheelbase of 0.8m. The wheel capabilities include a maximum speed of 0.3m/s, acceleration of 0.15m/s^2, and angular speed of $45°/\text{s}$. These specifications play a crucial role in the path planning process.

Two distinct environments as shown in Figure 11 are considered. The first scenario involves a cluttered environment with various irregular obstacles. The second scenario features a narrow passage with a width of 0.8m that imposes constraints on the robot's pose during traversal. In both cases, the overall environment size is $15.18\text{m} \times 8.18\text{m}$. The paths are required to start from $(1, 1)$ and end near $(15, 8)$, with a tolerance of 1m.



(a) Environment 1: Cluttered Environment.

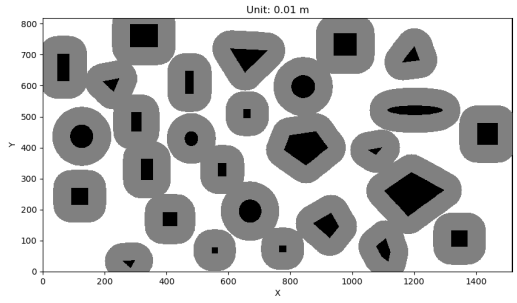


(b) Environment 2: Narrow Passages.

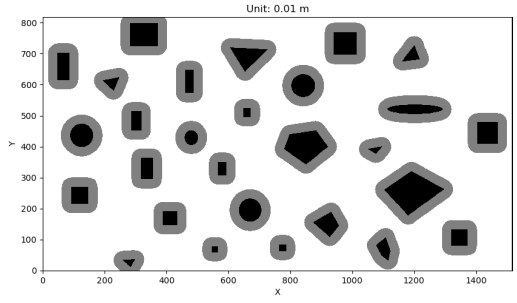
Figure 11. Environments for the test scenarios. The grey point on the bottom left represents the start point, while the orange range on the top right signifies the acceptable range of the destination.

Planned Paths Using Spline-Based RRT Method

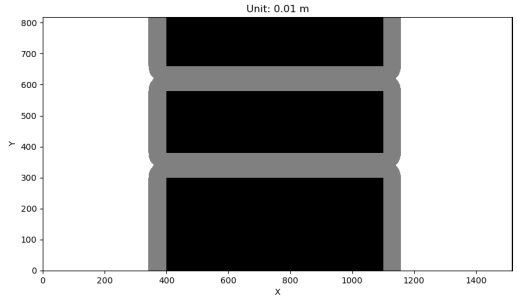
As discussed in Section , neglecting the consideration for motion modes can impact obstacle detection, as the planned path is generated without factoring in orientation, making it challenging to define or predict the robot's pose on the path. Consequently, without accounting for motion modes or kinematic models, a common approach in path planning for obstacle avoidance involves inflating the areas around obstacles, referred to as an inflation layer. However, the inflation layer alone cannot fully address the issue, as the planned path may not be suitable for different motion modes unless specific conditions are taken into account. Besides, designing the width of the inflation layer also poses challenges. Typically, the width is set to at least the radius of the circle that encircles the robot, which is 0.583m in this paper. However, using this width for inflation may mistakenly identify the inflated passage as obstacle regions, as depicted in Figure 12. Consequently, a reduced width of 0.3m is employed in these scenarios. In the following section, we utilize the SRRT to illustrate this issue.



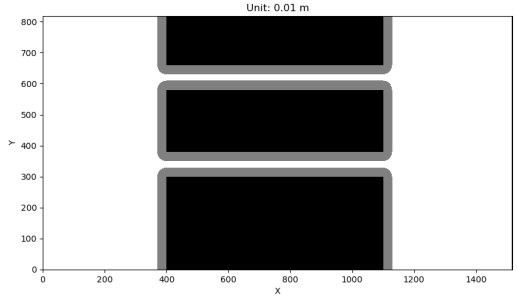
(a) Large inflation.



(b) Small inflation.

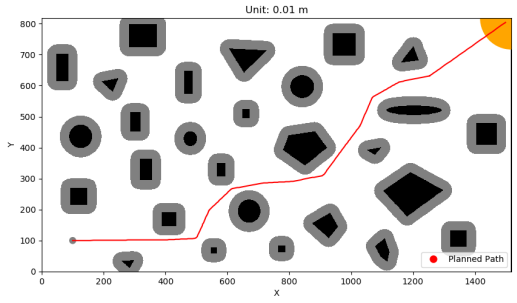


(c) Large inflation.

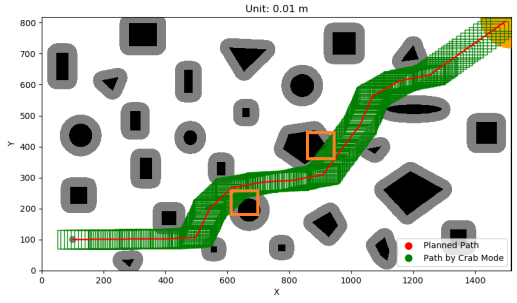


(d) Small inflation.

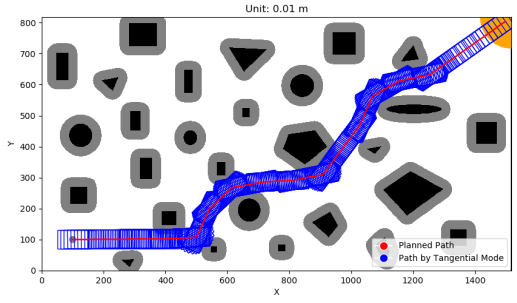
Figure 12. Obstacle inflation with different widths. Black areas represent the original obstacles, while the grey areas signifies the inflation layers.



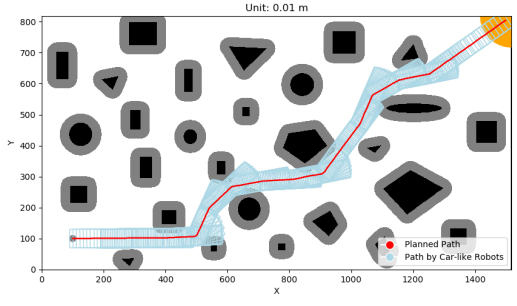
(a) The planned paths by using SRRT in the cluttered environment.



(b) The robot poses executing the Crab movements along the planned path. The collision between the robot and the obstacle is highlighted by the orange box.

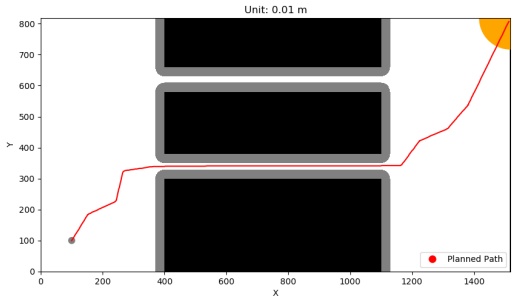


(c) The robot poses executing the Tangential movements along the planned path.

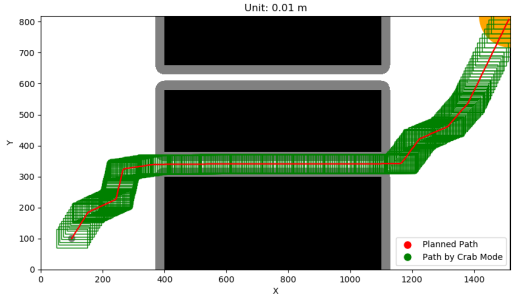


(d) The robot poses executing the Car-like movements along the planned path.

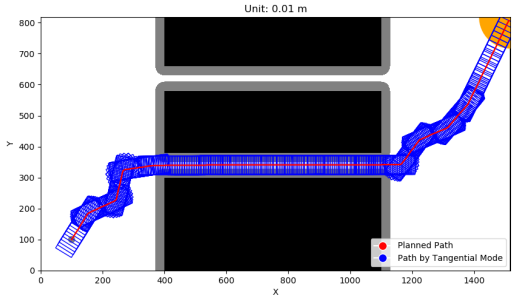
Figure 13. Graphical results by the SRRT(*14*) in the cluttered environment.



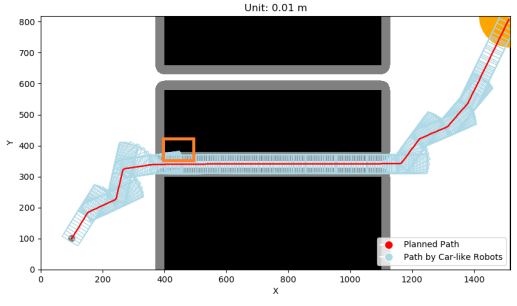
(a) The planned paths by using SRRT in the narrow passage environment.



(b) The robot poses executing the Crab movements along the planned path.



(c) The robot poses executing the Tangential movements along the planned path.

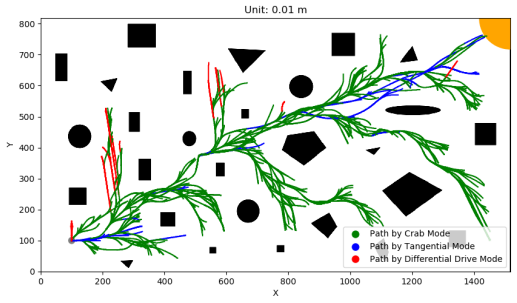


(d) The robot poses executing Car-like movements along the planned path. The collision between the robot and the obstacle is highlighted by the orange box.

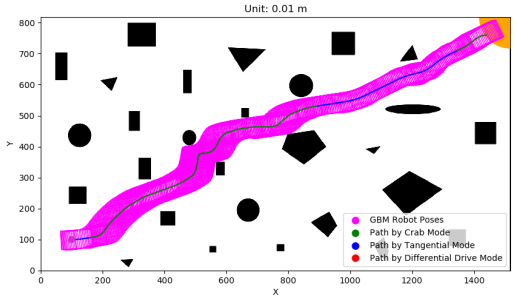
Figure 14. Graphical results by the SRRT(*14*) in the narrow passage.

Scenario 1: Cluttered Environment In the cluttered environment, Figure 13a illustrates the planned path generated using SRRT. Meanwhile, Figure 13b, 13c, and 13d depict the robot poses with different motion modes considered. It is evident that by incorporating inflation layers, SRRT can calculate an obstacle-free global path. This path allows the robot with Tangential mode and Car-like behavior to navigate without colliding with obstacles. However, when the robot operates in Crab mode along the same path, collisions with obstacles occur, as indicated by the orange boxes in Figure 13b. This issue stems from the lack of consideration for motion modes, kinematic models, or more specifically, the robot's orientation. Inflation layers prove ineffective in preventing collisions if a proper width fully encircling robots is not applied.

Scenario 2: Narrow Passage In the case of the narrow passage, Figure 14 displays the SRRT-planned path and the robot poses with different motion modes along the path. However, a similar issue to the one mentioned earlier arises. As depicted in Figure 14b and 14c, the robot successfully navigates the passage with Crab and Tangential modes. On the contrary, the robot employing Car-like movement collides with obstacles, as highlighted by the orange box in Figure 14d. As discussed in Section , various motion modes or behaviors result in different area coverage for robots, thereby influencing obstacle detection. To address this, consideration of motion modes and kinematic models becomes imperative. Consequently, we integrate motion modes constraints into our proposed pathfinding method, enhancing obstacle detection and leveraging the agility and flexibility of GBM robots effectively.



(a) The planned paths in the cluttered environment. The red line represents the Differential drive mode, the green one represents the Crab mode, and the blue one represents the Tangential mode.



(b) The robot poses on the planned path. The purple rectangles indicate the robot's poses.

Figure 15. Graphical results in cluttered environment scenario.

Planned Paths Using Proposed Method

In comparison to SRRT, our proposed method, which integrates the constraints of motion modes and the kinematic model, enables the prediction of robot poses along the planned path. This prediction facilitates direct collision detection with obstacles, eliminating the need for inflation layer design. Additionally, as illustrated in Figure 6, our approach in these tests considers the following motion modes: Crab, Tangential, and Differential drive. We employ the Tangential mode to represent Ackermann steering geometry, chosen for its ease in achieving the mode switch condition between Crab and Tangential modes.

Scenario 1: Cluttered Environment The paths generated by our method in the cluttered environment are illustrated in Figure 15, while the corresponding continuity analysis and planned control inputs can be found in Figure 16 and 17, respectively.

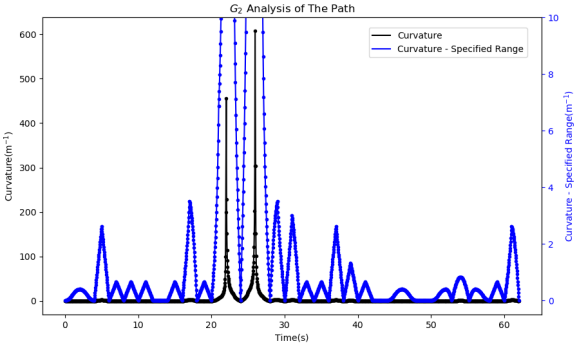


Figure 16. G_2 continuity. Due to scaling constraints, we utilize double axes to depict both the original and a specified range of curvature. It is evident that the curvature remains continuous at all times, thereby achieving G_2 continuity.

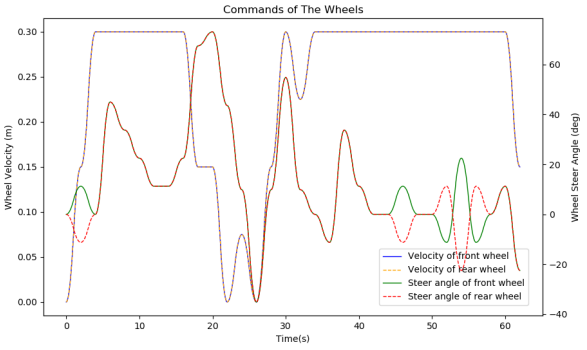
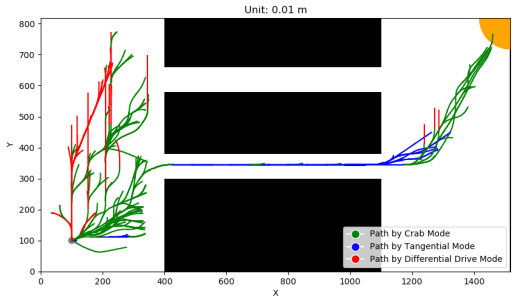


Figure 17. The control inputs corresponding to the planned path. At $t = 4$ seconds, the robot switched from the Tangential to the Crab motion mode, and a similar behavior can be observed again after $t = 44$ seconds.

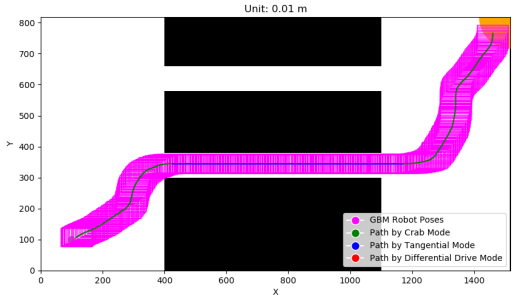
Examining the planned control inputs illustrated in Figure 17, we can observe that from $t = 0$ to 4 seconds, the robot operates in Tangential mode, characterized by opposite steering angles and identical wheel speeds. Afterward, the steering angles converge to 0, and subsequently, both steering angles remain consistent, signifying a transition to Crab mode. This mode switch occurs at $t = 4$ seconds. Furthermore, the continuity analysis in Figure 16 confirms that the curvature remains continuous throughout, satisfying the G_2 continuity requirement. The overall polynomial functions of the control inputs planned for the global path are listed in Table 4.

Scenario 2: Narrow Passage For the scenario of narrow passage, Figure 18 showcases the planned paths. Additionally, Figure 19 provides insights into the continuity analysis, while Figure 20 displays the control inputs planned for the scenario.

From the planned paths in Figure 18a, the robot traversed the narrow passage using the Tangential mode, represented by the blue lines. The corresponding control inputs in Figure 20 were recorded from $t = 16$ to 40 seconds. During this period, the robot maintained the steering angles of its two wheels at 0, effectively operating at the boundary between Crab and Tangential modes. Additionally, similar to Scenario 1, the planned path continues to satisfy G_2 continuity, as demonstrated in Figure 19. Furthermore, the overall polynomial function of the control inputs is also documented and listed in Table 5.



(a) The planned paths in the narrow passage. The red line represents the Differential drive mode, the green one represents the Crab mode, and the blue one represents the Tangential mode.



(b) The robot poses on the planned path. The purple rectangles indicate the robot's poses.

Figure 18. Graphical results in narrow passage scenario.

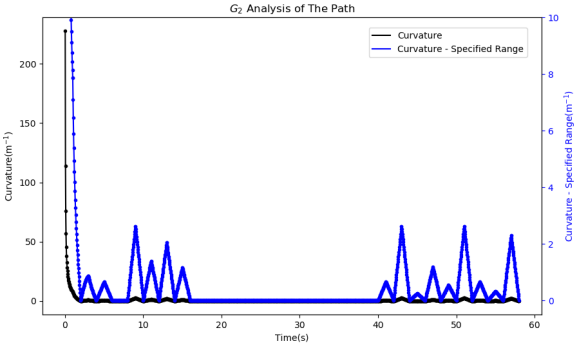


Figure 19. G_2 continuity. Due to scaling constraints, we utilize double axes to depict both the original and a specified range of curvature. It is evident that the curvature remains continuous at all times, thereby achieving G_2 continuity.

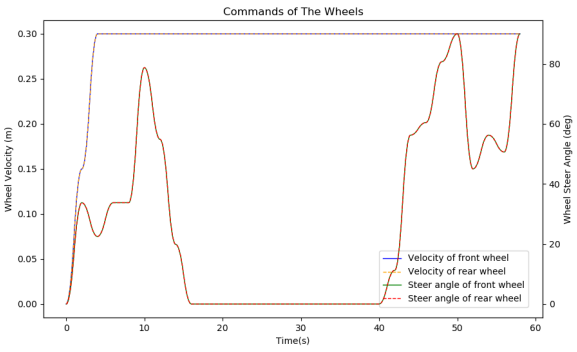


Figure 20. The control inputs corresponding to the planned path. The robot operates within the boundary of Crab and Tangent modes from $t = 26$ to 40 seconds.

In both test cases, our proposed method showcases its ability to compute a global path by integrating the diverse motion modes of GBM robots. Throughout the planning phase, the RRT tree expands its nodes, considering the current state, obstacles, and available modes. This highlights the flexibility and agility of GBM robots, effectively addressing the previously mentioned obstacle detection issue. Additionally, our planning method utilizes smooth 2nd-order polynomials for control inputs, ensuring controllability. This is further substantiated by our analysis of G_2 continuity.

Conclusion and Future Work

This work initially reviewed existing research on vehicle types and highlighted the potential for further development of GBM robots. Subsequently, we defined detailed motion modes for GBM robots, underscoring their significance in path planning. We employed a general path planning method that considers various modes, highlighting their impact on both area coverage and obstacle detection. As a result, we integrate motion modes into the RRT planner by designing control inputs with 2nd-order polynomials. We also establish continuity analysis to validate the planned path. In two distinct scenarios, our proposed method successfully computes global paths and meet the G_2 continuity requirement.

However, several issues still exist, warranting further study. For example, incorporating asymptotic optimality considerations into path planners based on RRTs will be a future focus, aiming to compute the best path while accounting for motion modes. Additionally, while bi-directional expansion is commonly used in RRT methods to enhance efficiency, its application in our method poses challenges due to the need for continuity in both path and control inputs. Moreover, path tracking methods are essential for GBM robots, given the likelihood of encountering control errors or the need to avoid dynamic obstacles, which can limit reliance on information from the planned path. Despite these challenges, we believe that this work contributes significantly and will propose solutions to address these issues in the future.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Chen, H., H. Yang, X. Wang, and T. Zhang. Formation control for car-like mobile robots using front-wheel driving and steering. *International Journal of Advanced Robotic Systems*, Vol. 15, No. 3, 2018, p. 1729881418778228.
2. Rajamani, R. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.
3. Bonci, A., S. Longhi, and G. A. Scala. Towards an all-wheel drive motorcycle: Dynamic modeling and simulation. *IEEE Access*, Vol. 8, 2020, pp. 112867–112882.
4. Li, L., K. Song, and H. Xie. Active Disturbance Rejection Path-following Control for Self-driving Forklift Trucks with Geometry based Feedforward. In *2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI)*. IEEE, 2020, pp. 187–192.
5. Tamba, T. A., B. Hong, and K.-S. Hong. A path following control of an unmanned autonomous forklift. *International Journal of Control, Automation and Systems*, Vol. 7, 2009, pp. 113–122.
6. Yuan, J., F. Sun, and Y. Huang. Trajectory generation and tracking control for double-steering tractor–trailer mobile robots with on-axle hitching. *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 12, 2015, pp. 7665–7677.
7. Kokot, M., D. Miklić, and T. Petrović. Path continuity for multi-wheeled AGVs. *IEEE Robotics and Automation Letters*, Vol. 6, No. 4, 2021, pp. 7437–7444.
8. Yuan, J., S. Yang, and J. Cai. Consistent path planning for on-axle-hitching multisteering trailer systems. *IEEE Transactions on Industrial Electronics*, Vol. 65, No. 12, 2018, pp. 9625–9634.
9. Morales, J., J. L. Martínez, A. Mandow, and A. J. García-Cerezo. Steering the last trailer as a virtual tractor for reversing vehicles with passive on-and off-axle hitches. *IEEE Transactions on Industrial Electronics*, Vol. 60, No. 12, 2013, pp. 5729–5736.
10. Widyotriatmo, A., Y. Y. Nazaruddin, M. R. F. Putranto, and R. Ardhi. Forward and backward motions path following controls of a truck-trailer with references on the head-truck and on the trailer. *ISA transactions*, Vol. 105, 2020, pp. 349–366.
11. Sanchez-Ibanez, J. R., C. J. Perez-del Pulgar, and A. García-Cerezo. Path planning for autonomous mobile robots: A review. *Sensors*, Vol. 21, No. 23, 2021, p. 7898.
12. Simba, K. R., N. Uchiyama, and S. Sano. Real-time smooth trajectory generation for nonholonomic mobile robots

using Bézier curves. *Robotics and Computer-Integrated Manufacturing*, Vol. 41, 2016, pp. 31–42.

13. Huh, U.-Y. and S.-R. Chang. A G2 continuous path-smoothing algorithm using modified quadratic polynomial interpolation. *International Journal of Advanced Robotic Systems*, Vol. 11, No. 2, 2014, p. 25.
14. Yang, K., S. Moon, S. Yoo, J. Kang, N. L. Doh, H. B. Kim, and S. Joo. Spline-based RRT path planner for non-holonomic robots. *Journal of Intelligent & Robotic Systems*, Vol. 73, No. 1-4, 2014, pp. 763–782.
15. Ghosh, D., G. Nandakumar, K. Narayanan, V. Honkote, and S. Sharma. Kinematic constraints based Bi-directional RRT (KB-RRT) with parameterized trajectories for robot path planning in cluttered environment. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8627–8633.
16. Blanco, J. L., M. Bellone, and A. Gimenez-Fernandez. TP-space RRT–kinematic path planning of non-holonomic any-shape vehicles. *International Journal of Advanced Robotic Systems*, Vol. 12, No. 5, 2015, p. 55.
17. Wang, J., B. Li, and M. Q.-H. Meng. Kinematic Constrained Bi-directional RRT with Efficient Branch Pruning for robot path planning. *Expert Systems with Applications*, Vol. 170, 2021, p. 114541.
18. Kang, Y., Z. Yang, R. Zeng, and Q. Wu. Smooth-RRT: Asymptotically Optimal Motion Planning for Mobile Robots under Kinodynamic Constraints. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8402–8408.
19. Kelly, A. *Mobile robotics: mathematics, models, and methods*. Cambridge University Press, 2013.

Table 4. The polynomials of the control inputs referred to Figure 17. The computation of RRT nodes assumes a period of 2 seconds. Additionally, the modes listed in the first column represent the mode in which the robot moves at the end of each node.

Nodes	Polynomials of θ_f	Polynomials of θ_r	Polynomials of V_f	Polynomials of V_r
0(Tangential)	(0.098175,0.000000,0.000000) (-0.098175,0.196350,0.098175)	(-0.098175,-0.000000,-0.000000) (0.098175,-0.196350,-0.098175)	(0.075000,0.000000,0.000000) (-0.075000,0.150000,0.075000)	(0.075000,0.000000,0.000000) (-0.075000,0.150000,0.075000)
1(Crab/Tangential)	(-0.098175,0.000000,0.196350) (0.098175,-0.196350,0.098175)	(0.098175,-0.000000,-0.196350) (-0.098175,0.196350,-0.098175)	(0.075000,0.000000,0.150000) (-0.075000,0.150000,0.225000)	(0.075000,0.000000,0.150000) (-0.075000,0.150000,0.225000)
2(Crab)	(0.392699,0.000000,0.000000) (-0.392699,0.785398,0.392699)	(0.392699,0.000000,0.000000) (-0.392699,0.785398,0.392699)	(0.000000,0.000000,0.300000) (-0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (-0.000000,0.000000,0.300000)
3(Crab)	(-0.098175,0.000000,0.785398) (0.098175,-0.196350,0.687223)	(-0.098175,0.000000,0.785398) (0.098175,-0.196350,0.687223)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
4(Crab)	(-0.098175,0.000000,0.589049) (0.098175,-0.196350,0.490874)	(-0.098175,0.000000,0.589049) (0.098175,-0.196350,0.490874)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
5(Crab)	(-0.098175,0.000000,0.392699) (0.098175,-0.196350,0.294524)	(-0.098175,0.000000,0.392699) (0.098175,-0.196350,0.294524)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
6(Crab)	(0.000000,0.000000,0.196350) (0.000000,0.000000,0.196350)	(0.000000,0.000000,0.196350) (0.000000,0.000000,0.196350)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
7(Crab)	(0.098175,0.000000,0.196350) (-0.098175,0.196350,0.294524)	(0.098175,0.000000,0.196350) (-0.098175,0.196350,0.294524)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
8(Crab)	(0.392699,0.000000,0.392699) (-0.392699,0.785398,0.785398)	(0.392699,0.000000,0.392699) (-0.392699,0.785398,0.785398)	(-0.075000,0.000000,0.300000) (0.075000,-0.150000,0.225000)	(-0.075000,0.000000,0.300000) (0.075000,-0.150000,0.225000)
9(Crab)	(0.049087,0.000000,1.178097) (-0.049087,0.098175,1.227185)	(0.049087,0.000000,1.178097) (-0.049087,0.098175,1.227185)	(0.000000,0.000000,0.150000) (0.000000,0.000000,0.150000)	(0.000000,0.000000,0.150000) (0.000000,0.000000,0.150000)
10(Crab)	(-0.257709,0.000000,1.276272) (0.257709,-0.515418,1.018563)	(-0.257709,0.000000,1.276272) (0.257709,-0.515418,1.018563)	(-0.075000,0.000000,0.150000) (0.075000,-0.150000,0.075000)	(-0.075000,0.000000,0.150000) (0.075000,-0.150000,0.075000)
11(Crab)	(-0.294524,0.000000,0.760854) (0.294524,-0.589049,0.466330)	(-0.294524,0.000000,0.760854) (0.294524,-0.589049,0.466330)	(0.037500,0.000000,0.000000) (-0.037500,0.075000,0.037500)	(0.037500,0.000000,0.000000) (-0.037500,0.075000,0.037500)
12(Crab)	(-0.392699,0.000000,0.171806) (0.392699,-0.785398,-0.220893)	(-0.392699,0.000000,0.171806) (0.392699,-0.785398,-0.220893)	(-0.037500,0.000000,0.075000) (0.037500,-0.075000,0.037500)	(-0.037500,0.000000,0.075000) (0.037500,-0.075000,0.037500)
13(Crab)	(0.392699,0.000000,-0.613592) (-0.392699,0.785398,-0.220893)	(0.392699,0.000000,-0.613592) (-0.392699,0.785398,-0.220893)	(0.075000,0.000000,0.000000) (-0.075000,0.150000,0.075000)	(0.075000,0.000000,0.000000) (-0.075000,0.150000,0.075000)
14(Crab)	(0.392699,0.000000,0.171806) (-0.392699,0.785398,0.564505)	(0.392699,0.000000,0.171806) (-0.392699,0.785398,0.564505)	(0.075000,0.000000,0.150000) (-0.075000,0.150000,0.225000)	(0.075000,0.000000,0.150000) (-0.075000,0.150000,0.225000)
15(Crab)	(-0.392699,0.000000,0.957204) (0.392699,-0.785398,0.564505)	(-0.392699,0.000000,0.957204) (0.392699,-0.785398,0.564505)	(-0.037500,0.000000,0.300000) (0.037500,-0.075000,0.262500)	(-0.037500,0.000000,0.300000) (0.037500,-0.075000,0.262500)
16(Crab/Tangential)	(-0.085903,0.000000,0.171806) (0.085903,-0.171806,0.085903)	(-0.085903,0.000000,0.171806) (0.085903,-0.171806,0.085903)	(0.037500,0.000000,0.225000) (-0.037500,0.075000,0.262500)	(0.037500,0.000000,0.225000) (-0.037500,0.075000,0.262500)
17(Crab)	(-0.098175,0.000000,0.000000) (0.098175,-0.196350,-0.098175)	(-0.098175,0.000000,0.000000) (0.098175,-0.196350,-0.098175)	(0.000000,0.000000,0.300000) (-0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (-0.000000,0.000000,0.300000)
18(Crab)	(0.392699,0.000000,-0.196350) (-0.392699,0.785398,0.196350)	(0.392699,0.000000,-0.196350) (-0.392699,0.785398,0.196350)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
19(Crab)	(-0.196350,0.000000,0.589049) (0.196350,-0.392699,0.392699)	(-0.196350,0.000000,0.589049) (0.196350,-0.392699,0.392699)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
20(Crab/Tangential)	(-0.098175,0.000000,0.196350) (0.098175,-0.196350,0.098175)	(-0.098175,0.000000,0.196350) (0.098175,-0.196350,0.098175)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
21(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
22(Tangential)	(0.098175,0.000000,0.000000) (-0.098175,0.196350,0.098175)	(-0.098175,-0.000000,-0.000000) (0.098175,-0.196350,-0.098175)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
23(Crab/Tangential)	(-0.098175,0.000000,0.196350) (0.098175,-0.196350,0.098175)	(0.098175,-0.000000,-0.196350) (-0.098175,0.196350,-0.098175)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
24(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
25(Tangential)	(-0.098175,0.000000,0.000000) (0.098175,-0.196350,-0.098175)	(0.098175,-0.000000,-0.000000) (-0.098175,0.196350,0.098175)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
26(Tangential)	(0.294524,0.000000,-0.196350) (-0.294524,0.589049,0.098175)	(-0.294524,-0.000000,0.196350) (0.294524,-0.589049,-0.098175)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
27(Tangential)	(-0.294524,0.000000,0.392699) (0.294524,-0.589049,0.098175)	(0.294524,-0.000000,-0.392699) (-0.294524,0.589049,-0.098175)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
28(Crab/Tangential)	(0.098175,0.000000,-0.196350) (-0.098175,0.196350,-0.098175)	(-0.098175,-0.000000,0.196350) (0.098175,-0.196350,0.098175)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
29(Crab)	(0.098175,0.000000,0.000000) (-0.098175,0.196350,0.098175)	(0.098175,0.000000,0.000000) (-0.098175,0.196350,0.098175)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
30(Crab)	(-0.294524,0.000000,0.196350) (0.294524,-0.589049,-0.098175)	(-0.294524,0.000000,0.196350) (0.294524,-0.589049,-0.098175)	(-0.075000,0.000000,0.300000) (0.075000,-0.150000,0.225000)	(-0.075000,0.000000,0.300000) (0.075000,-0.150000,0.225000)

Table 5. The polynomials of the control inputs referred to Figure 20. The computation of RRT nodes assumes a period of 2 seconds. Additionally, the modes listed in the first column represent the mode in which the robot moves at the end of each node.

Nodes	Polynomials of θ_f	Polynomials of θ_r	Polynomials of V_f	Polynomials of V_r
0(Crab)	(0.294524,0.000000,0.000000) (-0.294524,0.589049,0.294524)	(0.294524,0.000000,0.000000) (-0.294524,0.589049,0.294524)	(0.075000,0.000000,0.000000) (-0.075000,0.150000,0.075000)	(0.075000,0.000000,0.000000) (-0.075000,0.150000,0.075000)
1(Crab)	(-0.098175,0.000000,0.589049) (0.098175,-0.196350,0.490874)	(-0.098175,0.000000,0.589049) (0.098175,-0.196350,0.490874)	(0.075000,0.000000,0.150000) (-0.075000,0.150000,0.225000)	(0.075000,0.000000,0.150000) (-0.075000,0.150000,0.225000)
2(Crab)	(0.098175,0.000000,0.392699) (-0.098175,0.196350,0.490874)	(0.098175,0.000000,0.392699) (-0.098175,0.196350,0.490874)	(0.000000,0.000000,0.300000) (-0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (-0.000000,0.000000,0.300000)
3(Crab)	(0.000000,0.000000,0.589049) (0.000000,0.000000,0.589049)	(0.000000,0.000000,0.589049) (0.000000,0.000000,0.589049)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
4(Crab)	(0.392699,0.000000,0.589049) (-0.392699,0.785398,0.981748)	(0.392699,0.000000,0.589049) (-0.392699,0.785398,0.981748)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
5(Crab)	(-0.208621,0.000000,1.374447) (0.208621,-0.417243,1.165825)	(-0.208621,0.000000,1.374447) (0.208621,-0.417243,1.165825)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
6(Crab)	(-0.305262,0.000000,0.957204) (0.305262,-0.610524,0.651942)	(-0.305262,0.000000,0.957204) (0.305262,-0.610524,0.651942)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
7(Crab/Tangential)	(-0.173340,0.000000,0.346680) (0.173340,-0.346680,0.173340)	(-0.173340,0.000000,0.346680) (0.173340,-0.346680,0.173340)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
8(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
9(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
10(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
11(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
12(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
13(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
14(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
15(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
16(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
17(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
18(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
19(Crab/Tangential)	(0.000000,0.000000,0.000000) (0.000000,0.000000,0.000000)	(-0.000000,-0.000000,-0.000000) (-0.000000,-0.000000,-0.000000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
20(Crab)	(0.098175,0.000000,0.000000) (-0.098175,0.196350,0.098175)	(0.098175,0.000000,0.000000) (-0.098175,0.196350,0.098175)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
21(Crab)	(0.392699,0.000000,0.196350) (-0.392699,0.785398,0.589049)	(0.392699,0.000000,0.196350) (-0.392699,0.785398,0.589049)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
22(Crab)	(0.036816,0.000000,0.981748) (-0.036816,0.073631,1.018563)	(0.036816,0.000000,0.981748) (-0.036816,0.073631,1.018563)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
23(Crab)	(0.176408,0.000000,1.055379) (-0.176408,0.352816,1.231787)	(0.176408,0.000000,1.055379) (-0.176408,0.352816,1.231787)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
24(crab/differential)	(0.081301,0.000000,1.408194) (-0.081301,0.162602,1.489495)	(0.081301,0.000000,1.408194) (-0.081301,0.162602,1.489495)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
25(Crab)	(-0.392699,0.000000,1.570796) (0.392699,-0.785398,1.178097)	(-0.392699,0.000000,1.570796) (0.392699,-0.785398,1.178097)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
26(Crab)	(0.098175,0.000000,0.785398) (-0.098175,0.196350,0.883573)	(0.098175,0.000000,0.785398) (-0.098175,0.196350,0.883573)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
27(Crab)	(-0.049087,0.000000,0.981748) (0.049087,-0.098175,0.932660)	(-0.049087,0.000000,0.981748) (0.049087,-0.098175,0.932660)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)
28(crab/differential)	(0.343612,0.000000,0.883573) (-0.343612,0.687223,1.227185)	(0.343612,0.000000,0.883573) (-0.343612,0.687223,1.227185)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)	(0.000000,0.000000,0.300000) (0.000000,0.000000,0.300000)