MATHEMATICAL MODELING AND COMPUTING, Vol. 12, No. 2, pp. 693–708 (2025)

Motion planning of sensorless differential drive mobile robot using clothoid

Zakaria W. Z. E. W.[1], Ramli A. L. A.[2], Reif U.[3]

[1] *School of Industrial Technology, Universiti Sains Malaysia (USM), 11800 Pulau Pinang, Malaysia*
[2] *School of Mathematical Sciences, Universiti Sains Malaysia (USM), 11800 Pulau Pinang, Malaysia*
[3] *Technische Universität Darmstadt, Schlossgartenstrasse, 64385, Darmstadt, Germany*

(Received 20 March 2025; Revised 23 June 2025; Accepted 26 June 2025)

Mobile robots have the ability to navigate safely and cost-effectively from one location to another. In sensorless motion planning, the absence of sensor data poses challenges for continuous replanning to ensure that the robot follows the desired trajectory. Our research addresses this gap by leveraging computational methods for curve generation within the domain of Computer-Aided Geometric Design (CAGD). Specifically, we propose a clothoid-based, sensorless motion planning algorithm for differential drive mobile robots, assuming the known initial point, end point, and initial direction. Differential drive mobile robots, commonly used in various vehicles and robotics applications, are valued for their reliability and ease of use. These robots control movement through differential speed adjustments between the left and right wheels, with curvature determining the respective wheel speeds. In this paper, we detail the process of generating the clothoid curve and connecting it across two or more points, ensuring that each consecutive clothoid follows the direction in which the previous clothoid ends. We also present simulation results to assess the effectiveness of the proposed technique. The curve formulation and computational approach are tailored to this specific problem, utilizing MATLAB's Remote API capabilities and CoppeliaSim's robot simulation software for implementation.

Keywords: *differential drive; clothoid; robot simulation; motion planning; mobile robot.*
2010 MSC: 65D17, 70B15, 51N05, 68U07 **DOI:** 10.23939/mmc2025.02.693

1. Introduction

Intelligent mobile robots must be equipped with the ability to operate in challenging environments. For decades, research and engineering efforts have focused on developing advanced navigation systems to facilitate the transportation of mobile robots between various locations. In robotics, "motion planning" refers to the process of breaking down a desired movement task into discrete motions that satisfy specific movement constraints while potentially optimizing certain aspects of the motion. The study of robotic motion planning has been extensively explored, with numerous methods devised in this domain [1, 2]. Various techniques have been investigated, including potential fields [3] and randomized procedures [4], as well as approaches based on geometric design, such as Bezier curves [5], graph theory [6], and game theory [7]. Motion planning techniques for Differential Drive Mobile Robots (DDMRs) encompass foundational theories, practical applications, and innovations in autonomous navigation. Potential field methods, first introduced by Khatib et al. [8], guide robots using artificial forces but are challenged by local minima, with recent research integrating machine learning to address this issue [9]. Bug algorithms, developed by Lumelsky et al. [10], offer simple obstacle-avoidance strategies that have been optimized for dynamic environments [11]. Grid-based methods like A* excel in static settings, with adaptations enhancing real-time performance and scalability [12]. Sampling-based techniques, including PRM and RRT, perform well in high-dimensional spaces, with recent improvements focused

This research is supported by the Ministry of Higher Education Malaysia through Fundamental Research Grant Scheme (FRGS/1/2021/STG06/USM/02/6).

on sampling efficiency [13]. Optimal control methods such as Dynamic Programming and Model Predictive Control minimize trajectory costs, with AI-enhanced approaches improving performance in dynamic scenarios [14]. These techniques contribute to advancing DDMR motion planning, with hybrid methods and machine learning driving continuous innovation. On the other hand, modern robots are typically equipped with various sensors, including cameras (or visual sensors in general) [15, 16], sonar sensors [17, 18], tactile sensors [19], and others, to perceive and understand their environment. There is also extensive research on robots without sensors, as demonstrated in the studies conducted by Khalil et al. [20], Xia et al. [21], Li et al. [22] and Yuan et al. [23]. Sensorless motion planning refers to planning and controlling a robot's motion without relying on external sensors for feedback. This approach is achieved by modeling the system's dynamics and employing mathematical algorithms to plan and control its motion. Therefore, sensorless motion planning applications are beneficial when there is a desire to minimize reliance on sensors. In this paper, we use the strength of Computer-Aided Geometric Design (CAGD) for motion planning without relying on sensor assistance. Conventionally, motion planning assumes that the target and/or the robot have a known, at least partially, posture (position and orientation). Motion plans are then calculated using only the pose data that is available. While motion planning for mobile robots has been extensively studied, much of the focus remains on sensor-based approaches, leaving sensorless strategies largely unexplored. This gap is significant as sensorless methods offer advantages in cost reduction, simplicity, and robustness in environments where sensors may be unreliable or unavailable. This paper aims to address this gap by investigating sensorless path planning techniques, providing novel insights and solutions that contribute to more efficient and reliable motion planning for differential drive mobile robots. Motion planning techniques using CAGD, including lines, circles, polynomial curves, and Cornu spiral or clothoid curves, have long been used in traditional automatic parking systems and are extensively discussed in the review by Zhou [24]. More recently, there has been increased interest in clothoid-based path planning algorithms, as evidenced by the works of Merrel et al. [25] through Silva et al. [26]. However, the calculation of Fresnel integrals presents a substantial computational challenge in commonly used clothoid-based path planning methods. Clothoid-based trajectories, characterized by gradual curvature changes, offer significant advantages over traditional methods like circular arcs, particularly in ensuring smooth transitions between trajectory segments and improving motion stability and control. This smoothness is crucial for differential drive mobile robots, where continuous curvature enhances precision and reduces path-following errors. However, despite these benefits, clothoid techniques have been insufficiently explored in sensorless motion planning, with most prior work focusing on simpler models or relying heavily on sensors. This study addresses this gap by demonstrating the effectiveness of clothoid-based motion planning for smoother, more reliable control in sensorless environments. Due to their naturally appealing curvatures, Cornu spirals or clothoids have gained popularity as representations for free-form curves. However, Cornu spirals are inherently transcendental curves and can only be approximated, making accurate modeling challenging. Bertolazzi et al. [27] proposed an effective technique for approximating Cornu spirals within a specified tolerance using a series of arcs. Specifically, we employ a method known as biarcs, which utilizes two arc segments, to interpolate a collection of two data points. To interpolate a G^1 continuous biarc curve between two points while taking into account the defined start, joint, and end tangents, a trigonometric solution is constructed. According to Bertolazzi et al. [28], G^1 Hermite interpolation is a fitting method that permits a curve to interpolate two provided points in a plane while preserving designated tangent directions. In this study, we offer a new approach that uses a clothoid curve to solve the G^1 Hermite interpolation problem. Like in Binninger et al. [29], this algorithm fits a piecewise clothoid that fulfills the Hermite G^1 interpolation conditions by evaluating tangents and curvatures at control points. Currently, a lot of academics are interested in employing trajectories based on clothoids, wherein they use Cornu spirals to connect several segments. Recent years have seen a surge in the study of smooth clothoid interpolation for parametric curves. The trajectory-following algorithm proposed by Horváth et al. [30] is meant for autonomous vehicles. The route is made up of clothoid curves connecting straight parts. The goal of the suggested method is to simulate a driver's movements while driving with greater accuracy. The project's main objective is to automate Sinha [31] obstacle-free trajectory design process using motion planning for AGVE's AGVs (Automated Guided Vehicles). Clotoid curves are used to produce feasible trajectories that result in pathways free of obstacles. Kala et al. [32] discusses and evaluates several enhancements to a clothoid local path template (LPT) that was previously developed for assessing motorist intentions and aiding wheelchair navigation. Additionally, this LPT holds potential for other mobile robotics applications, including intuitive teleoperated control, rapid collision testing during path planning, and obstacle avoidance algorithms. The LPT comprises a substantial but fixed set of paths in the immediate vicinity of the mobile robot, with path lengths adjusted to ensure collision-free navigation using an efficient look-up table. Experimental results have indicated that in congested situations, there is a lack of sufficient clothoid pathways, resulting in challenges for providing accurate navigation assistance and intention estimation. This article explores the utilization of clothoidal pathways instead of clothoid paths, significantly improving the capability to navigate intricate paths through dense terrain. Moreover, by incorporating motion prediction estimations of these objects, the LPT is modified to handle dynamic obstacles with unpredictable shapes. Girbés et al. [33] also discusses attitude planning, which is the skillful movement of an airplane in a straight line to alter its height and direction, for example, to prevent an impending collision. The authors provide a technique to create smooth roads inside a global planning framework by combining straight lines with various 3D curves. Their goal is to identify a curve that can be used as a smooth reference trajectory for more complicated maneuver planning, obstacle avoidance in fixed-wing aircraft, and UAVs (unmanned aerial vehicles). They describe a novel clothoid-based 3D curve in this work that has the following desirable qualities for path-planning problems: scalability, symmetry, smoothness, and monotonicity. It also allows for real-time calculations for a variety of applications. Silva et al. [26] have developed a clothoid-based road network model for autonomous cars. This approach is applied to path generation in urban settings involving roundabouts, intersections, double-lane roadways, and curving lanes. The model generates a kinematically viable path with minimal curvature rate change and bounded piecewise linear continuous curvature by using clothoids as control points for online interpolation. The geometry of the road lanes is accurately represented by this path. Another noteworthy outcome of this study is the creation of a lane change path planner utilizing clothoids. This planner has a low curvature derivative, a set longitudinal trip distance, and is made to manage non-parallel configuration changes. The potential of smooth clothoid interpolation to produce subtle curvature transitions, which improve stability and control in robotic and autonomous systems, has drawn a lot of interest lately. Its natural, human-like motion improves interaction in shared environments, while its ability to generate smooth paths minimizes mechanical strain and increases energy efficiency. Clothoids are particularly valuable in applications like autonomous driving, where they reduce lateral accelerations for a smoother ride. Additionally, they are highly adaptable to complex environments, making them ideal for advanced motion planning tasks in cluttered or dynamic settings. In this paper, our focus revolves around two main objectives. Firstly, we aim to formulate clothoid-based trajectories for motion planning in differential drive robots, with particular emphasis on the Pioneer 3-DX model. The Pioneer 3-DX is well-known in mobile robotics research for its robust differential drive system and adaptability, making it an ideal platform for testing our approach. While previous studies using this robot have often relied on sensor feedback, our research explores the application of clothoid paths in a sensorless context. The fixed wheel configuration of the Pioneer 3-DX presents a unique challenge for maintaining continuous curvature, which allows us to demonstrate how clothoid trajectories can enhance path-following accuracy and reduce motion errors. This formulation can be adapted to other similar robots by adjusting specific parameters, offering a broader application of our findings. By bridging the gap between sensorless planning and advanced trajectory models, this study contributes to more precise and reliable motion planning techniques for differential drive robots. Secondly, we seek to observe the robot's movement through simulation. This allows us to validate our planned path and also identify any potential limitations that may arise during the execution. By combining theoretical formulation and practical simulation, we aim to provide a comprehensive understanding of clothoid-based motion planning for differential drive robots. Referring to Bertolazzi et al. [27] and Bertolazzi et al. [28], we suggest a clothoid-based method for point interpolation in route planning for differential drive mobile robots. This method has assigned tangent directions. We create a smooth road made up of a line, a circle, and a clothoid path by building a path based on Cornu spiral arcs that interpolate two or more points. Next, in order to move the robot in accordance with the created clothoid, we have developed an algorithm to compute the wheel velocities and the trip time based on curvature Broman et al. [34]. We assess this method's precision and effectiveness in the CoppeliaSim simulator as well.

2. Mobile robots

In this section, we present details regarding the mobile robot employed in the CoppeliaSim simulation. We will delve into the various components responsible for the locomotion of the mobile robot.

2.1. Pioneer 3-DX

The Pioneer 3-DX robot is a differential drive robot equipped with two wheels and two motors, making it an ideal choice for laboratory or classroom settings. It features a front SONAR sensor, a single battery, and wheel encoders. Pioneer research robots are widely popular in educational and research environments worldwide, establishing them as leading intelligent mobile robots in the field [35].

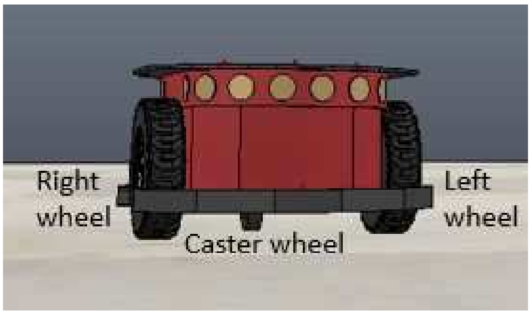


Fig. 1. A Pioneer 3-DX robot on CoppeliaSims.

The model depicted in Figure 1 illustrates the simulated version of the robot. It is equipped with two wheels, each of which can be controlled individually, enabling precise movement control. Through adjustments to the wheel velocities, the robot is able to perform various actions and manoeuvres. The MATLAB remote API function creates a connection with the robot and generates the movement. Table 1 presents the corresponding movements of the robot based on the velocity of its wheels.

Table 1. The movement of pioneer 3-DX robot.

Velocity of left wheel, V_L and right wheel, V_R	Operations
$V_L < V_R$	Counter-clockwise circular movement
$V_L > V_R$	Clockwise circular movement
$V_L - V_R = 0$	Stop

It should be noted that when implementing clothoid movement, a specific parameter is determined based on the speed. However, if the robot needs to rotate at a specific point without incorporating clothoid movement, this can be achieved by setting the left wheel velocity, V_L equal in magnitude but opposite in direction to the right wheel velocity, V_R.

2.2. Differential drive mobile robot

The Pioneer 3-DX robot is a two-wheeled, two-motor differential drive robot that is perfect for use in lab or classroom environments. Wheel encoders, a single battery, and a front SONAR sensor are its features. In academic and research settings, pioneer research robots are highly favoured. A platform with a front caster and two rear coaxial driving wheels makes up the differential drive robot, which maintains isostatic balance. A DC motor that is driven individually by a given voltage powers each driving wheel. The differential wheeled mobile robot may move in a straight line or follow varied trajectories, like curves or circles, by varying the power supplied to the motors. Creating a precise mathematical model is necessary in order to create control systems that work well for these robots [36]. Figure 2 illustrates the kinematic concept of a differential drive mobile robot. To apply this concept to any two-wheeled robot, additional features like sensors are not utilized. The following assumptions are made for this model: (1) The mass center of the robot is located at the geometric center of the body frame. (2) The robot's movement does not rely on sensors. (3) The robot operates on a solid ground surface, with both wheels maintaining continuous contact with it.

In the given kinematic model, the coordinates of the robot platform's center of mass are represented by (x_c, y_c), where x_c represents the position on the x-axis, y_c represents the position on the y-axis, V_R represents the velocity of the right wheel, V_L represents the velocity of the left wheel, V_c represents the linear velocity, and ω_c represents the angular velocity of the robot. The wheels are positioned at a distance of $2d$ apart, where d is the length between the center of the robot and each wheel. Varying the velocities of the wheels results in different directions and operations. The movement of the robot is generated through the remote API function in MATLAB, establishing a connection with the robot. The characteristics of the Pioneer 3-DX in CoppeliaSim simulator are obtained from Technology [35] and summarized in Table 1 in meters, as CoppeliaSim uses meters as its unit of measurement.

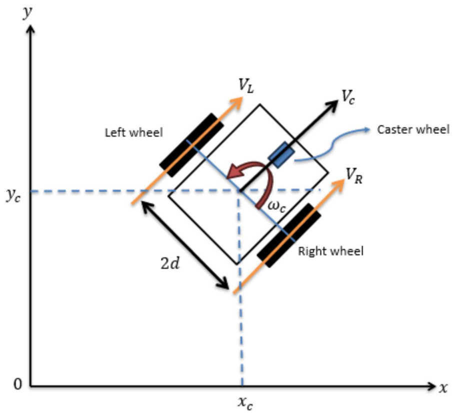


Fig. 2. The kinematics schematic of differential drive mobile robot.

Table 2. Pioneer 3-DX robot dimensions.

Characteristics	Dimensions (m)
Length	0.455
Height	0.237
Distance between the two wheels	0.381
Wheel's height	0.195
Caster wheel's height	0.052

3. Methodology used and results

In this section, clothoid formulation will be described in subsection 3.1 to 3.3. Following that, the settings of our robot will be defined, and motion planning on clothoid paths will be discussed in subsection 3.4. The calculation of wheel velocity will be addressed in subsection 3.5. Finally, in subsection 3.6, the robot's movement will be simulated in CoppeliaSim, and the accuracy will be observed.

3.1. G^1-Interpolation with piecewise clothoids

Clothoid, also referred to as the Cornu spiral or Euler's spiral, was initially explored by Johann Bernoulli in approximately 1696. It is utilized to describe diffraction phenomena arising from the edge of a half-plane [37].

A clothoid is a planar curve whose curvature depends linearly on its arc length. The prototype $c(t) = [x(t), y(t)]^T$, $t \in \mathbb{R}$, is given by the Fresnel integrals

$$x(t) := \int_0^t \cos\frac{\pi s^2}{2}\,ds, \quad y(t) := \int_0^t \sin\frac{\pi s^2}{2}\,ds. \tag{1}$$

The general form is obtained by translation, scaling, reflection, and rotation of this prototype (see Figure 3),

$$C(t) = P_0 + a\,S^k r_{\phi_0} c(t), \quad t \in \mathbb{R}, \tag{2}$$

where $P_0 \in \mathbb{R}^2$ is a point in the plane, $a \in \mathbb{R} \setminus \{0\}$ is a scaling factor.

$$S := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{3}$$

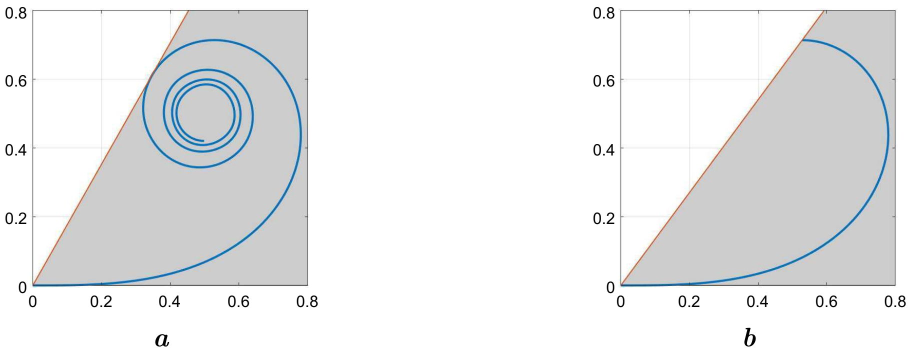


Fig. 3. Clothoid c (***a***) and restriction to range of interest $t \in [0, \sqrt{2}]$ (***b***).

is the reflection at the x-axis, the exponent $k \in \{0, 1\}$ is a switch disabling or enabling reflection, and

$$r_{\phi_0} := \begin{bmatrix} \cos\phi_0 & -\sin\phi_0 \\ \sin\phi_0 & \cos\phi_0 \end{bmatrix} \tag{4}$$

is the rotation with angle $\phi_0 \in \mathbb{R}$. The *tangent angle* of C at t is

$$\phi(t) = \phi_0 + \pi t^2/2 \tag{5}$$

i.e., $C'(t) = [\cos\phi(t), \sin\phi(t)]^T$, while its curvature is $\kappa(t) = \pi t$.

3.2. The two-point problem

Given two points $P_0, P_1 \in \mathbb{R}^2$ and a tangent angle ϕ_0, we want to determine a clothoid C such that $C(0) = P_0$, $\phi(0) = \phi_0$, and $C(A) = P_1$ for some parameter $A > 0$. This is equivalent to determining a and A such that

$$c(A) = a^{-1}P, \quad P := S^k r_{-\phi_0}(P_1 - P_0),$$

and then setting $C(t) := P_0 + a\, S^k r_{\phi_0} c(t)$. We assume that $k \in \{0, 1\}$ is chosen such that P lies in the upper half plane. This leads us to the definition of the two-point interpolation problem in normal form:

Given $P \in \mathbb{R} \times \mathbb{R}_{>0}$, find positive parameters A and a such that

$$c(A) = a^{-1}P. \tag{6}$$

To make the solution unique and to avoid extra rotation of the curve, we further demand that A attains the smallest value within the set of possible solutions.

Applying the *slope function*

$$\sigma(x, y) := \frac{y}{x}$$

to (6) removes the unknown a and yields the necessary condition

$$S(A) := \sigma(c(A)) = \sigma(P). \tag{7}$$

The slope of the clothoid $c(t)$, $t > 0$, is bounded by $0 < c(t) \leqslant \sigma_0$, where $\sigma_0 \approx 1.765$. Hence, the problem is not solvable if $\sigma(P) > \sigma_0$. For practical purposes, however, we restrict the range of admissible slopes further to $\sigma(P) \leqslant 1.35$, meaning that the tangent angle of c, which starts at $\phi(0) = 0$, is bounded by approximately π. In other words, we disregard cases requiring more than a U-turn. Since the function $\sigma(c(t))$ is monotone increasing, we may solve Eq. (7) by a bisection strategy. However, it is much more efficient to use the precomputed approximation

$$T(s) := \frac{29\sqrt{s}}{21} - \frac{s^2}{9} \approx S^{-1}(s), \quad s \in (0, 1.35],$$

of the inverse of S. Then

$$A := T(\sigma(P))$$

is a good approximation for the solution of (7). More precisely, the error

$$\Delta(s) := S^{-1}(s) - T(s), \quad s \in (0, 1.35]$$

is bounded in modulus by 0.01, see Figure 4. It may be improved by one or two rounds of the fixed-point iteration

$$A \leftarrow A - \frac{\sigma(c(A)) - \sigma(P)}{A + A^4/3},$$

which works pretty well since the denominator $A + A^4/3$ is a good approximation of the derivative $S'(t) = D\sigma(c(t)) \cdot c'(t)$, as used by Newton's method. The first application reduces the maximal error to 4×10^{-4}, and the second one to 2×10^{-5}, what should be good enough for all practical purposes, see Figure 5.

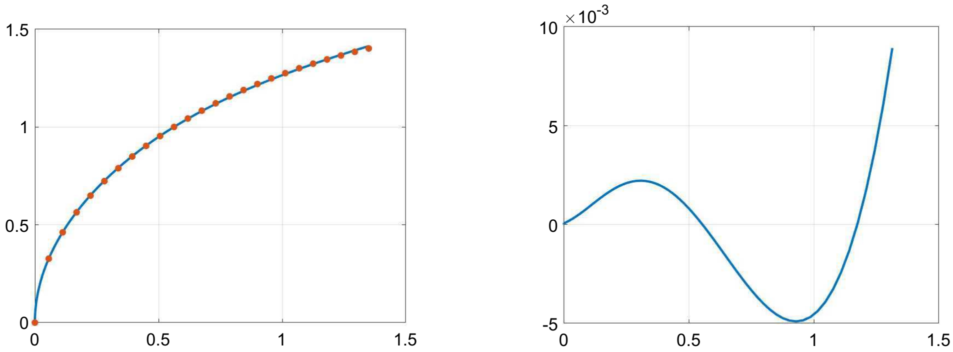


Fig. 4. Function S^{-1} with dots indicating the values of the approximation T (left) and error Δ (right).

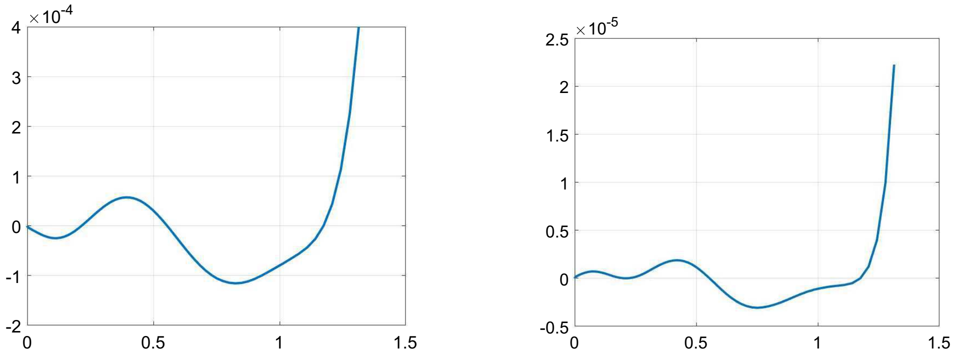


Fig. 5. Error Δ after one (left) and two (right) rounds of the fixed-point iteration.

When A is found, the value of the scaling factor a is determined by

$$a := \frac{|P|}{|c(A)|}.$$

3.3. The multi-point problem

Now, given $n+1$ points $P_0, \dots, P_n$, we seek a sequence of n clothoids forming a G^1-curve C that interpolates these points. More precisely, we want to determine parameters $A_i, \dots, A_n$ such that

$$C_i(0) = P_{i-1}, \quad C_i(A_i) = P_i$$

for $i = 1, \dots, n$ and

$$C_i'(0) = C_{i-1}'(A_{i-1})$$

for $i = 2, \dots, n$. Then, setting $t_i := \sum_{j=1}^{i} A_j$, the sought composite curve $C\colon [0, t_n] \to \mathbb{R}^2$ is defined by

$$C(t) = C_i(t - t_{i-1}), \quad t \in [t_{i-1}, t_i].$$

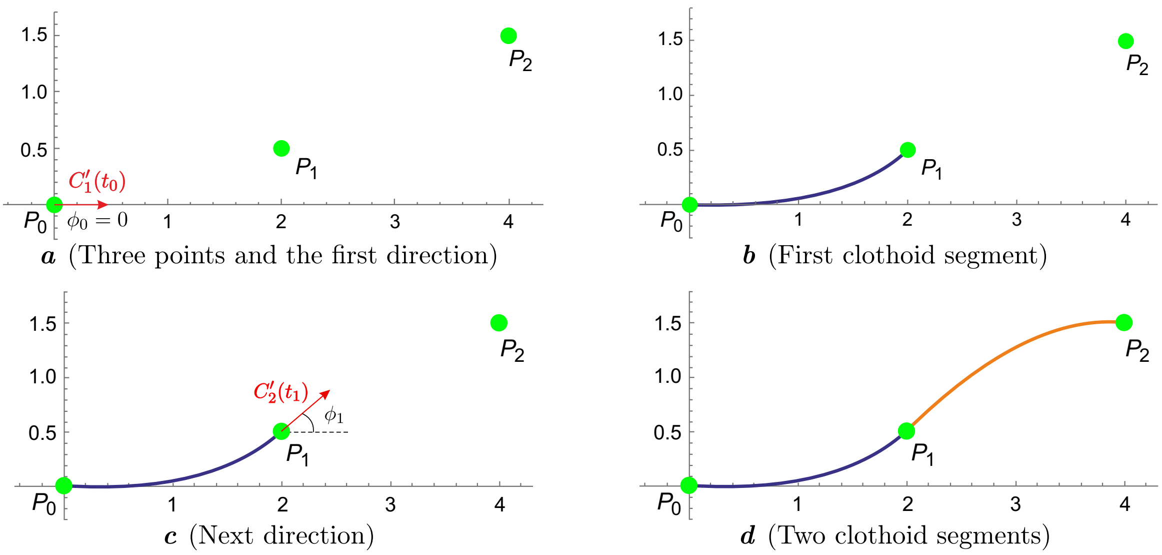
The algorithm is a simple iteration over the segments proceeding from $i = 1$ to $i = n$. At step i, we solve the two-point problem with data $C_i(0) = P_{i-1}$, $C_i(A_i) = P_i$. For $i = 1$, the initial tangent angle $\phi_0(0)$ is undetermined and can be chosen by the user. Otherwise, for $i \geqslant 2$, it is set to the end tangent angle of the previous segment to ensure G^1-continuity, i.e.,

$$\phi_i(0) := \phi_{i-1}(A_{i-1}).$$

3.4. Clothoid on differential drive mobile robot

In this paper, we address the challenge of motion planning for a sensorless differential drive mobile robot tasked with navigating a predefined set of points using clothoid curves and Computer-Aided Geometric Design (CAGD) principles. Our objective is to devise an algorithm capable of autonomously determining the robot's position and orientation without external sensor input and subsequently generating trajectories that incorporate clothoid curves and leverage CAGD knowledge to enable smooth and accurate traversal of the specified points. This problem is of paramount importance in scenarios where sensor deployment is constrained, such as specific industrial environments or specialized robotics applications. Solving this problem necessitates a comprehensive grasp of geometric design principles to achieve precise and efficient motion planning. Our research pivots around the establishment of a connection between a line and a circle, with a core focus on ensuring that the robot maintains its direction from the previous path constructed by another clothoid as it continues to move. This choice stems from the robot's primary objective: to follow a predefined path traced by this curve. At each specified point along this path, the robot is programmed to come to a stop, potentially to execute various tasks, such as item retrieval. Consequently, we provide a detailed outline of the robot's movement, carefully adhering to the following settings:

1. Initially, both wheels of the robot have zero velocity before any motion commences.
2. During the initial robot's movement, we set the robot to move straight briefly before executing a turn.
3. After completing each segment, the robot is set to stop and then continues with a the same behavior, adhering to the setting stated in 1 and 2.
4. The robot is prohibited from rotating at any specific point.



a (Three points and the first direction)

b (First clothoid segment)

c (Next direction)

d (Two clothoid segments)

Fig. 6. Illustration of clothoid interpolation.

A clothoid curve can be used to join two points in four different ways [38]: a line to a circle with one spiral, a circle to another circle with one spiral, a circle to another circle generating an S-curve with two spirals, and a circle to another circle forming a C-curve with two spirals. Based on this rationale, the most appropriate choice of clothoid is a single spiral connected to a circle and a line. To achieve this connection, certain conditions must be satisfied. Next, we visualize the construction of the clothoid from three points in the differential drive mobile robot setting.

Given any two points, $P_0 = (x_0, y_0)$ and $P_1 = (x_1, y_1)$, our goal is to interpolate these points with a specified tangent direction at P_0. In this example, we set ϕ_0 as zero, placing it on the x-axis and looking toward the direction of $(1, 0)$ (Refer to first figure in Figure 6).

We will obtain the first curve by substituting $k = 0$ and $\phi_0 = 0$ into (2) and obtaining $C(t)$ from $t = 0$ until A (refer to second figure in Figure 6). Then, we will obtain the next ϕ_1 as in Eq. (5) (refer to third figure in Figure 6).

As the robot moves towards the same tangent direction, the two consecutive clothoids are connected with a point continuity, C^0 and tangent continuity, G^1. Technically, the robot will stop at the endpoint of the first curve and start with different velocities for the subsequent curve.

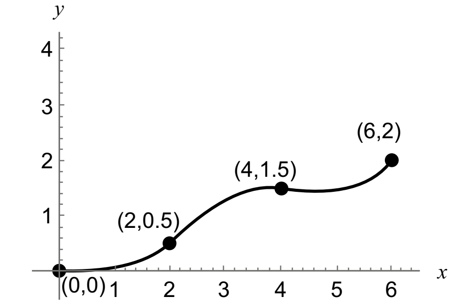


Fig. 7. Clothoid interpolation of 3 clothoid segments.

The procedure can be repeated if there is a next point. We show the continuation of the clothoid connecting to point (6,2) as illustrated in Figure 7.

3.5. Velocity and time for differential drive mobile robot simulation

Creating a path that connects the desired points is computationally easier. However, using this information in a robot simulator needs additional assessment. Assuming the robot starts facing a certain direction, it can reach the next point by following a clothoid curve with set parameters. The robot's movement can be managed by varying the speeds of its left and right wheels. In this case, the relationship between the left wheel velocity (V_L), right wheel velocity (V_R), and the overall velocity of the robot's center can be described as follows:

$$V_c = \frac{(V_L + V_R)}{2}. \tag{8}$$

When turning clockwise,

$$V_L = V_c + d\omega_c, \quad V_R = V_c - d\omega_c. \tag{9}$$

When turning counter-clockwise,

$$V_L = V_c - d\omega_c, \quad V_R = V_c + d\omega_c. \tag{10}$$

By referring to the evaluation of the Bezier curve formula and the curvature in [34], we adapt for a clothoid to represent the linear velocity, V_c, and angular velocity, ω_c. ω_c as follows:

$$V_c = \sqrt{(x'[A])^2 + (y'[A])^2} \tag{11}$$

and

$$\omega_c = \frac{x'[A]y''[A] - y'[A]x''[A]}{(x'[A])^2 + (y'[A])^2}. \tag{12}$$

We select $d = 0.1905$ in accordance with the specifications of the Pioneer-3DX robot [35]. The time, t, for each velocity is determined as follows:

$$t = \frac{V_c}{D}, \tag{13}$$

where D is the Euclidean distance between two points.

3.6. Robot simulation

In this section, we introduce a robot simulation that builds on the results from our method. The simulation is run on a computer featuring an Intel(R) Core (TM) i3-8100 CPU @ 3.60GHz. Figure 8 shows the movement of Pioneer-3DX on a simulator in clockwise movement from a start point to an end point. The movement is achieved by varying the velocities of both wheels.

We substituted various values of linear velocity, V_c, and angular velocity, ω_c, into Eqs. (9) and (10) to calculate the velocities for both wheels. This simulation is implemented in CoppeliaSim, where we input the values of V_L, V_R, and time t corresponding to their speeds in MATLAB. The aim of our simulation is to determine whether the robot can navigate the projected path without relying on additional elements like sensors. We also track the points the robot passes through. Figure 9 shows

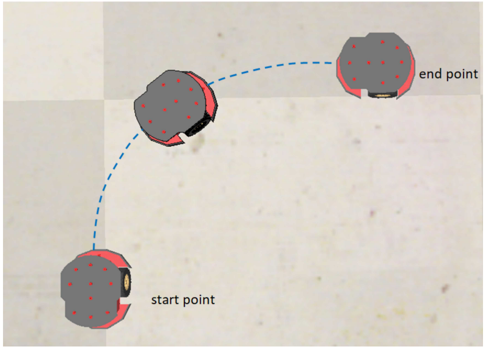


Fig. 8. An example of Pioneer-3DX movement on CoppeliaSim.

the path generated from the simulation, comparing it with the interpolated curve derived from our method, which constructs the curve using clothoids.

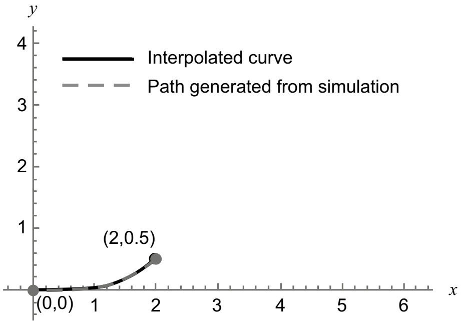


Fig. 9. One segment of clothoid.

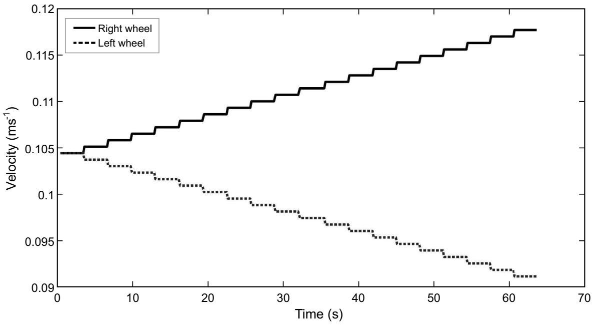


Fig. 10. Illustration of velocities for Figure 9.

In the robot simulation depicted in Figure 9, the robot moves from (0,0) to (2,0.5) in the left direction. Using our method in subsection 3.4, we obtained the first clothoid segment. Subsequently, we calculated the velocities and corresponding times for both the left and right wheels, as shown in Figure 10. Figure 10 demonstrates a decreasing pattern for the left wheel velocities and an increasing pattern for the right wheel velocities, reflecting the leftward movement of the robot.

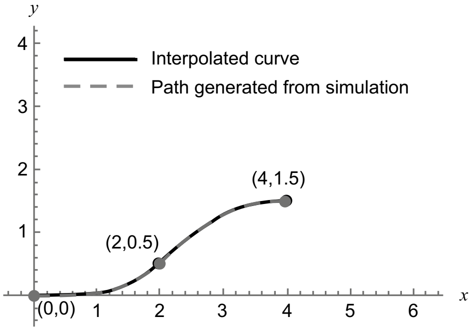


Fig. 11. Clothoid interpolation of 2 clothoid segments.

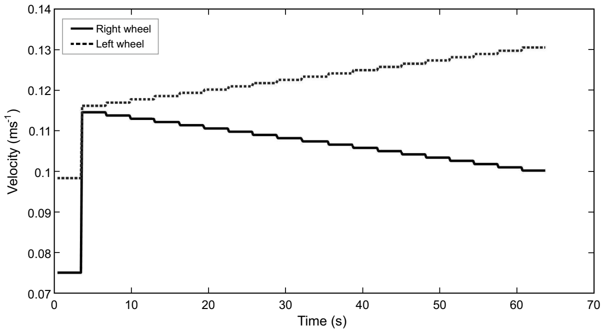


Fig. 12. Illustration of velocities for second clothoid segment in Figure 11.

The time required to complete the clothoid in the simulator is 68.771 seconds. It is clear that the path generated by the simulation closely aligns with the expected interpolated curve. The endpoint of the simulation is $(1.9944, 0.4944)$, while the endpoint of the interpolated curve is $(2.0000, 0.5000)$, yielding a minimal error of 0.0082. Next, we run our method with 3 points or 2 clothoid segments. We set P_1 as (4,1.5) in the right direction. The resulting simulation is depicted in Figure 11.

The dashed paths shown in Figure 11 illustrate the trajectories produced by the simulation, highlighting the varying directions of the second segment of the clothoid and allowing for a comparison of the simulation results. The simulation results align closely with the interpolated result. In Figure 12, the corresponding list of velocities for the second segments of the dashed path shown in Figure 11 is illustrated. As the second clothoid segment turns to the right from the first segment, the right wheel velocity (V_R) decreases, while the left wheel velocity (V_L) increases, as indicated in Figure 12.

The path generated from the simulation (dashed path) in Figure 11 closely follows the interpolated curve, with an error difference of less than 0.05 at P_1, as shown in Table 3. The total time taken to complete the entire path is 141.3586 seconds. Note that the error at P_0 for the simulation is not necessarily the same value as in the previous example, even though the clothoid segment is the same. This discrepancy occurs because the simulation itself may introduce its own error every time we run it.

Table 3. The comparison error between interpolated and simulation coordinate for path Figure 11.

Points	**Interpolated coordinate**	**Simulation coordinate**	**Error**
P_0	$(2, 0.5)$	$(2.0063, 0.5010)$	0.0063
P_1	$(4, 1.5)$	$(4.0134, 1.4923)$	0.0154

Next, we test our algorithm on three clothoid segments. Figure 13 shows the interpolated curve along with the path generated from the robot simulation.

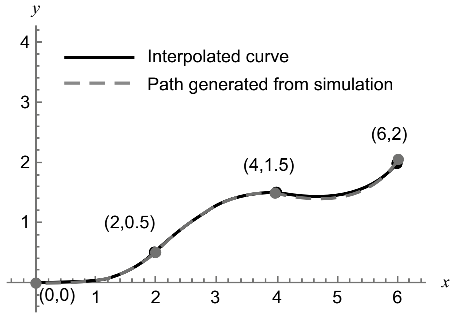


Fig. 13. The three segments of the clothoid.

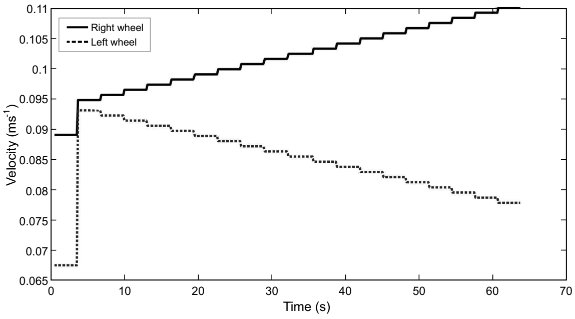


Fig. 14. Illustration of velocities for third clothoid segment in Figure 11.

Figure 14 illustrates the velocity list for the left and right wheels, V_L and V_R, respectively, during the third segment of the dashed path depicted in Figure 13. The error for each point is computed and displayed in Table 4.

As observed, the simulation path closely follows the shape of the interpolated curve. The first two clothoid segments exhibit a similar error with two segments clothoid in Figure 11 of approximately 0.0050 between the interpolated and simulated path after two segments. Furthermore, the error decreases after three clothoid segments, with an error of 0.0284 indicating a good fit. The simulation took approximately 215.5731 s to complete. Further analysis was conducted by comparing the clothoid with a circular arc in our paper Wan et al. [39]. The results of the comparison are shown in Figure 15.

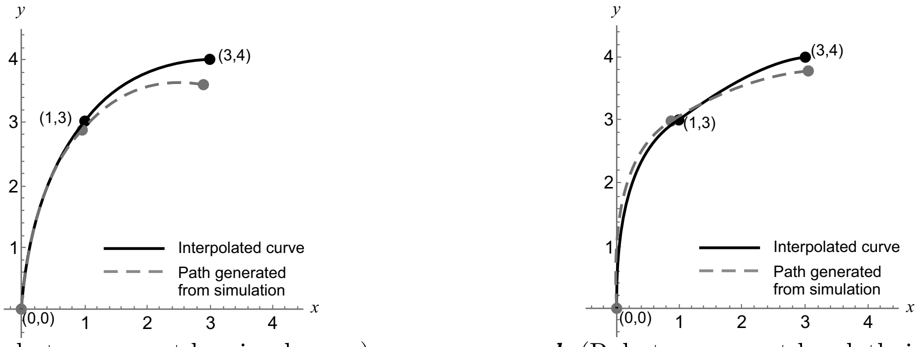
Table 4. The comparison error between interpolated and simulation coordinate for path Figure 13.

Points	**Interpolated coordinate**	**Simulation coordinate**	**Error**
P_0	$(2, 0.5)$	$(2.0653, 0.5164)$	0.0673
P_1	$(4, 1.5)$	$(4.0462, 1.5401)$	0.0611
P_2	$(6, 2)$	$(6.0113, 2.0175)$	0.0208

Figure 15 illustrates that the simulation path for the second segment of the clothoid exhibits a smaller gap between the interpolated and simulated paths compared to the second circular arc. The errors between the interpolated and simulated paths for both the circular arc and the clothoid shown in Figure 15, are presented in Table 5.

Table 5. The comparison error between interpolated and simulation coordinate for path Figure 15.

Interpolated	Cirular arc		Clothoid	
	Simulation coordinate	**Error**	**Simulation coordinate**	**Error**
$(1, 3)$	$(0.8894, 2.7591)$	0.1221	$(0.8827, 2.9809)$	0.1187
$(3, 4)$	$(2.9007, 3.5996)$	0.4125	$(3.0552, 3.7852)$	0.2217



a (Robot movement by circular arc) *b* (Robot movement by clothoid)

Fig. 15. Comparison between simulation of robot movement by circular arc [39] and clothoid.

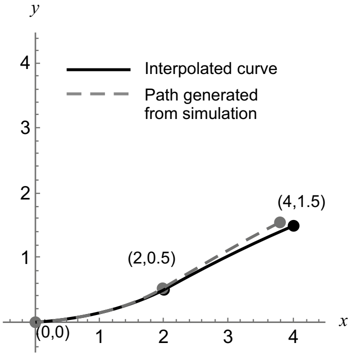


Fig. 16. Recreation of path in Figure 11 using circular arc interpolation and simulation.

Table 5 shows that the error in the clothoid is smaller than that in the circular arc for both end points (1,3) and (3,4). This is consistent with the results in Figure 15, where the clothoid exhibits a closer alignment with the interpolated path compared to the circular arc simulation. We also have done a same path for circular arc using path in Figure 11. The result of circular arc is shown in Figure 16.

As observed in Figure 16, the interpolation of the second circular arc employs a larger radius than the first arc to achieve a straighter line. The simulation results reveal that the trajectory of the second arc deviates from the assigned interpolated end point (4,1.5), shifting to (3.7957,1.5428), in contrast to the behavior of the clothoid depicted in Figure 11. This discrepancy results in a larger error for the second circular arc, measuring 0.208, in comparison to the clothoid's error of only 0.0054, as indicated in Table 4.

4. Discussion

Our approach introduces an algorithm that improves trajectory continuity by allowing the robot to move smoothly through each segment. After the robot stops at the end of one segment, we recalculate the velocity and time for the next segment, which makes it easier to determine the movement parameters. While the robot starts aligned horizontally, the algorithm can adjust to different starting angles by aligning the coordinate system with the x-axis, so no additional transformations are needed. A limitation of this approach is that the robot must come to a full stop after each segment, which can result in slight positional errors due to inertia. Future work will aim to enhance continuity to G^2, which will help the robot maintain a consistent velocity and reduce these errors. During development, we faced difficulties in finding the right velocities and times for the wheels in the simulation. In real-world applications, a different method will be needed to determine these parameters. By analyzing the relationship between velocity, time, and clothoid curvature, as discussed by Broman et al. [34], we can improve the simulation by using only the necessary parameters and avoiding the need for initial robot rotations. Our simulation results show that the robot closely follows the interpolated curve, especially with clothoid segments. However, errors can increase when using multiple arcs. These errors can be due to factors like wheel slippage [40], skidding [41], friction, and the robot's physical characteristics. Additionally, variations in results may arise from the computer's processing power and potential communication issues between Matlab and CoppeliaSim. When comparing our results to the circular arc method from Zakaria et al. [39], we found that the circular arc method leads to larger errors, particularly when transitioning between arcs. The clothoid technique, on the other hand, provides better performance and accuracy, especially for additional segments.

5. Conclusion

In this paper, we have presented a methodology for generating paths using clothoids within a simulation framework for differential drive mobile robots. Our approach allows users to create interpolated paths by defining initial coordinates and orientation, ensuring G^1 continuity between connected curves. The provided code implementation offers a practical advantage for users seeking to guide their robots along clothoid trajectories by specifying the speed and duration for the left and right wheels. Through experimental analysis, we have demonstrated that the simulation closely adheres to the intended curve and successfully interpolates the specified points, even when incorporating two or three clothoid segments. Several promising avenues for future research are worth highlighting. First, implementing our code on a physical robot would be beneficial to identify any additional limitations or challenges that may occur in real-world settings. Conducting experiments in real environments could provide valuable insights into the practical performance and applicability of our approach. Secondly, further exploration of factors that affect robot simulation, such as wheel slipping and skidding, would be beneficial. By investigating these factors, we can identify methods to reduce the error between simulated and interpolated paths. This research could lead to improved techniques and strategies for enhancing the accuracy and reliability of robot simulations in relation to real-world behaviour. Thirdly, when constructing paths using two clothoid segments, there may be a lack of curvature continuity between the arcs from a mathematical perspective. This discontinuity could potentially cause speed inconsistencies in robot movements, as noted in studies like Othman et al. [42], Zakaria et al. [43], and Wahrburg et al. [44]. Investigating alternative curve types that ensure curvature continuity may lead to smoother robot motions and improve the overall quality of generated paths. Further research in this area could contribute to the development of advanced path planning algorithms and more fluid trajectory generation for mobile robots. These proposed areas for future research hold the potential to significantly enhance the practicality and performance of our method, expanding its applicability to real-world scenarios while addressing limitations related to curvature continuity. By exploring these avenues, we can further refine and improve our approach, making it more robust and adaptable for a broader range of practical applications.

Acknowledgements

This research is supported by the Ministry of Higher Education Malaysia through Fundamental Research Grant Scheme (FRGS/1/2021/STG06/USM/02/6).

[1] Sun H., Zhang W., Yu R., Zhang Y. Motion planning for mobile robots – Focusing on deep reinforcement learning: A systematic review. IEEE Access. **9**, 69061–69081 (2021).

[2] Xiao X., Liu B., Warnell G., Stone P. Motion planning and control for mobile robot navigation using machine learning: a survey. Autonomous Robots. **46** (5), 569–597 (2022).

[3] Liu C., Zhai L., Zhang X. Research on local real-time obstacle avoidance path planning of unmanned vehicle based on improved artificial potential field method. 2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI). 1–6 (2022).

[4] Janoš J., Von'asek V., Pěnička R. Multi-goal path planning using multiple random trees. IEEE Robotics and Automation Letters. **6** (2), 4201–4208 (2021).

[5] Arslan Ö., Tiemessen A. Adaptive Bézier degree reduction and splitting for computationally efficient motion planning. IEEE Transactions on Robotics. **38** (6), 3655–3674 (2022).

[6] Zhang M., Sutcliffe M., Nicholson P. I., Yang Q. Efficient Autonomous Path Planning for Ultrasonic Non-Destructive Testing: A Graph Theory and K-Dimensional Tree Optimisation Approach. Machines. **11** (12), 1059 (2023).

[7] Hang P., Lv C., Xing Y., Huang C., Hu Z. Human-like decision making for autonomous driving: A noncooperative game theoretic approach. IEEE Transactions on Intelligent Transportation Systems. **22** (4), 2076–2087 (2020).

[8] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research. **5** (1), 90–98 (1986).

[9] Zhu K., Zhang T. Deep reinforcement learning based mobile robot navigation: A review. Tsinghua Science and Technology. **26** (5), 674–691 (2021).

[10] Lumelsky V. J., Stepanov A. A. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. Algorithmica. **2** (1), 403–430 (1987).

[11] Singh R., Ren J., Lin X. A review of deep reinforcement learning algorithms for mobile robot path planning. Vehicles. **5** (4), 1423–1451 (2023).

[12] Hewawasam H. S., Ibrahim M. Y., Appuhamillage G. K. Past, present and future of path-planning algorithms for mobile robot navigation in dynamic environments. IEEE Open Journal of the Industrial Electronics Society. **3**, 353–365 (2022).

[13] Wijayathunga L., Rassau A., Chai D. Challenges and solutions for autonomous ground robot scene understanding and navigation in unstructured outdoor environments: A review. Applied Sciences. **13** (17), 9877 (2023).

[14] Wang B., Liu Z., Li Q., Prorok A. Mobile robot path planning in dynamic environments through globally guided reinforcement learning. IEEE Robotics and Automation Letters. **5** (4), 6932–6939 (2020).

[15] Alshorman A. M., Alshorman O., Irfan M., Glowacz A., Muhammad F., Caesarendra W. Fuzzy-based fault-tolerant control for omnidirectional mobile robot. Machines. **8** (3), 55 (2020).

[16] Cimurs R., Suh I. H., Lee J. H. Information-based heuristics for learned goal-driven exploration and mapping. 2021 18th International Conference on Ubiquitous Robots (UR). 571–578 (2021).

[17] Li C., Guo S., Guo J. Study on obstacle avoidance strategy using multiple ultrasonic sensors for spherical underwater robots. IEEE Sensors Journal. **22** (24), 24458–24470 (2022).

[18] Ngwenya T., Ayomoh M., Yadavalli S. Virtual obstacles for sensors incapacitation in robot navigation: A systematic review of 2D path planning. Sensors. **22** (18), 6943 (2022).

[19] Costanzo M. Control of robotic object pivoting based on tactile sensing. Mechatronics. **76**, 102545 (2021).

[20] Khalil A., Mukhopadhyay S., Rehman H. Comparative Performance Analysis of PI Controller and MRAC for a Differential Drive Robot. 2022 5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA). 1–6 (2022).

[21] Xia S., Zhao M., Adhivarahan C., Hou K., Chen Y., Nie J., Wu E., Dantu K., Jiang X. Anemoi: A Low-cost Sensorless Indoor Drone System for Automatic Mapping of 3D Airflow Fields. ACM MobiCom '23: Proceedings of the 29th Annual International Conference on Mobile Computing and Networking. 77, 1–16 (2023).

[22] Li Y., Li Y., Zhu M., Xu Z., Mu D. A nonlinear momentum observer for sensorless robot collision detection under model uncertainties. Mechatronics. **78**, 102603 (2021).

[23] Yuan H., Goh K., Andras P., Luo W., Wang C., Gao Y. Steering angle sensorless control for four-wheel steering vehicle via sliding mode control method. Transactions of the Institute of Measurement and Control. **46** (3), 453–462 (2024).

[24] Zhou C., Huang B., Fränti P. A review of motion planning algorithms for intelligent robots. Journal of Intelligent Manufacturing. **33** (2), 387–424 (2022).

[25] Merrell B., Droge G. Clothoid-based moving formation control using virtual structures. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 11567–11572 (2020).

[26] Silva J. A. R., Grassi V. Clothoid-based global path planning for autonomous vehicles in urban scenarios. 2018 IEEE International Conference on Robotics and Automation (ICRA). 4312–4318 (2018).

[27] Bertolazzi E., Frego M. Interpolating clothoid splines with curvature continuity. Mathematical Methods in the Applied Sciences. **41** (4), 1723–1737 (2018).

[28] Bertolazzi E., Frego M. G^1 fitting with clothoids. Mathematical Methods in the Applied Sciences. **38** (5), 881–897 (2015).

[29] Binninger A., Sorkine-Hornung O. Smooth interpolating curves with local control and monotone alternating curvature. Computer Graphics Forum. **41** (5), 25–38 (2022).

[30] Horváth E., Pozna C. R. Clothoid-based trajectory following approach for self-driving vehicles. 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI). 000251–000254 (2021).

[31] Sinha S. Automatic generation of obstacle-free trajectories for AGVs. Dissertation (2022).

[32] Kala R., Warwick K. Dynamic distributed lanes: motion planning for multiple autonomous vehicles. Applied Intelligence. **41**, 260–281 (2014).

[33] Girbés V., Vanegas G., Armesto L. Clothoid-based three-dimensional curve for attitude planning. Journal of Guidance, Control, and Dynamics. **42** (8), 1886–1898 (2019).

[34] Broman E., Rossing I. A system for procedural camera movements for navigation in astrographics. Dissertation (2020).

[35] IA Technology. Pioneer3-DX. https://www.generationrobots.com/media/Pioneer3DX-P3DX-RevA.pdf (2011).

[36] Scaglia G. J. E., Serrano M. E., Godoy S. A., Rossomando F. Linear algebra-based controller for trajectory tracking in mobile robots with additive uncertainties estimation. IMA Journal of Mathematical Control and Information. **37** (2), 607–624 (2020).

[37] Weisstein E. Cornu spiral. https://mathworld.wolfram.com/CornuSpiral.html (2023).

[38] Meek D. S., Walton D. J. An arc spline approximation to a clothoid. Journal of Computational and Applied Mathematics. **170** (1), 59–77 (2004).

[39] Wan Zakaria W. Z. E., Ramli A., Misro M. Y., Wahab M. N. A. Motion Planning of Differential Drive Mobile Robot Using Circular Arc. Engineering Letters. **32** (1), 160–167 (2024).

[40] Há V. T., Thuong T. T.. Thanh N. T., Vinh V. Q. Research on Some Control Algorithms to Compensate for the Negative Effects of Model Uncertainty Parameters, External Interference, and Wheeled Slip for Mobile Robot. Actuators. **13** (1), 31 (2024).

[41] Gao X., Yan L., Gerada C. Modeling and analysis in trajectory tracking control for wheeled mobile robots with wheel skidding and slipping: Disturbance rejection perspective. Actuators. **10** (9), 222 (2021).

[42] Othman N. A., Reif U., Ramli A., Misro M. Y. Manoeuvring speed estimation of a lane-change system using geometric Hermite interpolation. Ain Shams Engineering Journal. **12** (4), 4015–4021 (2021).

[43] Zakaria W. Z. E. W., Ramli A., Ali J. M. Bezier curves interpolation with endpoint constraints on road map. AIP Conference Proceedings. **2184** (1), 060060 (2019).

[44] Wahrburg A., Guida S., Enayati N., Zanchettin A. M., Rocco P. Extending dynamic movement primitives towards high-performance robot motion. 2020 IEEE 16th International Workshop on Advanced Motion Control (AMC). 52–58 (2020).

Планування руху безсенсорного мобільного робота з диференціальним приводом за допомогою клотоїди

Закарія В. З. Е. В.[1], Рамлі А. Л. А.[2], Райф У.[3]

[1] *Школа промислових технологій, Університет наук Малайзії (USM), 11800 Пулау Пінанг, Малайзія*
[2] *Школа математичних наук, Університет наук Малайзії (USM), 11800 Пулау Пінанг, Малайзія*
[3] *Технічний університет Дармштадту, Шлоссґартенштрассе, 64385, Дармштадт, Німеччина*

Мобільні роботи мають здатність безпечно та економічно переміщатися з однієї локації в іншу. У плануванні руху без сенсорів відсутність даних із сенсорів створює виклики для неперервного перепланування, щоб гарантувати, що робот дотримується бажаної траєкторії. Наше дослідження заповнює цю прогалину, використовуючи обчислювальні методи для генерації кривих у галузі комп'ютерно-допоміжного геометричного дизайну (CAGD). Зокрема, пропонується алгоритм планування руху без сенсорів для мобільних роботів з диференціальним приводом на основі клотоїди, припускаючи відомі початкову та кінцеву точки, а також початковий напрямок. Мобільні роботи з диференціальним приводом, що широко використовуються в різних транспортних засобах та робототехнічних додатках, цінуються за їхню надійність та простоту використання. Ці роботи контролюють рух шляхом диференціального регулювання швидкості між лівим і правим колесами, причому кривизна визначає відповідні швидкості коліс. У цій статті детально описується процес генерації клотоїдної кривої та її з'єднання через дві або більше точок, гарантуючи, що кожна послідовна клотоїда дотримується напрямку, в якому закінчується попередня клотоїда. Також подано результати моделювання для оцінки ефективності запропонованої методики. Формулювання кривої та обчислювальний підхід адаптовані до цієї конкретної проблеми, використовуючи можливості віддаленого API MATLAB та програмне забезпечення для симуляції роботів CoppeliaSim для реалізації.

Ключові слова: *диференціальний привід; клотоїда; симуляція роботів; планування руху; мобільний робот.*