

Case Study: PDE-Based Motion Planning for Robots

Mehmet Batu Özmeteler

Technische Universität Dortmund, Germany

Abstract—This case study aims to utilize a proposed novel method for motion planning to implement a canonical example for robots. The guiding approach for this work tackles the motion planning problem for systems with holonomic and non-holonomic constraints by building the method upon a parabolic PDE that arises in the study of Riemannian manifold. The underlying method defines a metric for an optimal path, initializes a trajectory which connects the start and the end point (which doesn't have to respect constraints), and deforms the initial track via a homotopy into an admissible trajectory by numerically solving a system of coupled PDEs. The controls enabling the motion can be retrieved simply from the admissible path which in turn is practically optimal. By adhering to this procedure, this study performs its application for the two-link planar arm with additional holonomic constraints on the motion trajectory. Results are illustrated along with the benchmark example to demonstrate the viability of the implementation.

I. INTRODUCTION

A. Motion Planning for Non-holonomic Systems

The principal objective of motion planning is to compute a trajectory attained by a sequence of controls, which satisfies certain dynamic and obstacle avoidance constraints to achieve an admissible, collision-free motion of the system from the source to destination. For linear systems, the problem is unambiguous; explicit solutions that steer the system from start to end do exist. However for the non-linear case, the situation is way more complicated.

Before moving on to why, an essential concept like holonomy, must be visited to induce a better understanding of the presented material. In robotics, holonomy refers to the existence of restrictions among translational axes. If a robot is holonomic with respect to N dimensions, it's capable of moving (in any direction) in any of those N physical dimensions available to it. If it's non-holonomic, it's restricted to the directions which it can move in. A train in a 1D space would be considered holonomic, yet a differential wheeled robot would be non-holonomic in a 3D space.

Furthermore in dynamics, a system is denoted by a set of parameters which are subject to differential constraints. The system may be defined as holonomic/non-holonomic if all constraints of the system are holonomic/non-holonomic. A constraint on a dynamical system that can be integrated to eliminate one of the variables, reducing the number of degrees of freedom in a system, is called a holonomic constraint. A constraint that cannot be integrated in this way is called a non-holonomic constraint. For a sphere rolling on a rough plane, the no-slip constraint turns out to be non-holonomic while for a two-link planar arm, the dynamic constraints are holonomic.

Constraints can be outlined as a relationship between the position variables (and possibly time) as follows:

- **Holonomic:** $f(q_1, \dots, q_n, t) = 0$
- **Non-Holonomic:** $f(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n, t) = 0$

where n is the number of generalized coordinates that describe the system.

A holonomic system's trajectory integrals depend only upon the initial and final states of the system. One can mathematically devise the path when the initial and final states are presented, therefore the system is said to be integrable. However, a non-holonomic system is non-integrable because the states depend on the path taken to achieve it. More precisely, a non-holonomic system's current state depends on the intermediate values of its trajectory through parameter space; there is a continuous closed mechanism for transforming the system from one state to another.

Setting the definitions aside, now one can explain why motion planning for under-actuated non-linear systems is known to be a challenging problem. A number of factors contribute to the difficulty of the problem: The nature of a non-holonomic system is that of implicitly dependent parameters. The non-holonomic character of the system doesn't allow for any explicit solution to exist for controls driving a system from an initial state to a final state since the system is non-integrable. Computational and robustness issues arise from the nonlinearities, and the need to avoid obstacles which is not unique to the non-linear case, but is considerably more complex when combined with non-holonomic constraints.

B. Literature Review

Numerous ad-hoc techniques employing artificial intelligence [3] have been developed to deal with these challenges. Since these methods are practically "black boxes", it is hard to determine the quality of the control obtained or even to verify its validity. Therefore, these methods need to be adjusted to meet the requirements of each application [4].

In addition, there are other methods that rely on optimal control techniques [5]. Especially for these complex techniques, obstacle avoidance is a critical concern to be taken care of since it is not a part of the optimal control problem as a whole. Unfortunately, these methods being complex is also a problem for the implementation [6].

Alternatively, methods requiring a strong mathematical foundation don't scale well, such as [7], [8], or [9]. It can be concluded that these approaches cannot handle realistic scenarios involving more than a few variables. Other approaches include probabilistic road maps [10], graph search algorithms [11] and artificial potential fields [12].

While the above methods are proven to be robust for linear systems with holonomic constraints in dynamic environments, the extension of most of the mentioned techniques above to non-holonomic systems are strenuous.

C. A New Method for Motion Planning using PDEs

In this case study, a PDE-based novelty method [1] is followed to implement an extended example which solves the motion planning problem of a two-link planar arm with additional holonomic path constraints. Conversely to the previously referred work, the methodology is relatively uncomplicated and seemingly easy to understand intuitively. Also, the implementation is remarked to be a straightforward procedure which is the main discussion topic along with the analysis of results for this case study.

Authors of this approach utilize partial differential equations and Riemannian geometry as the workhorse to take on the problem of motion planning for non-holonomic systems. Based on this technique, it is possible to solve numerically a system of coupled PDEs, leading to trajectory solutions that are almost optimal in a certain metric. Moreover, the obstacle avoidance issue and the rigorous derivation of the result is addressed in their subsequent work [2].

II. PROBLEM STATEMENT AND METHODOLOGY

A. Mathematical Preliminaries

As the justification of the pursued method requires introducing notions, a mathematical background should be provided to set the scene. The explained approach is worked in a (at least locally) differentiable manifold of form $M = \mathbb{R}^n$. Firstly, one defines the system linear in the control:

$$\dot{x} = \sum_{i=1}^p u_i f_i(x) \quad (1)$$

where the f_i are differentiable vector fields in \mathbb{R}^n : $f_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Also, the system is assumed to be under-actuated, meaning $p < n$. The objective is described as the following: given an initial state and a final state $x_0, x_1 \in \mathbb{R}^n$, find a set of controls $u_i(t)$ that drive the above-mentioned system from x_0 to x_1 in $T > 0$ seconds while also respecting a set of holonomic, non-holonomic and obstacle avoidance constraints. In order to ensure the existence of such controls, one needs to look into the controllability criteria to prove that x_1 is reachable from x_0 . The evolution of the system trajectory is represented as a sum of the products of the input magnitudes with vector fields which has its elements in terms of state variables. To devise the concept of accessible motions to the system, distributions are considered.

A distribution $\Delta(x)$ is a vector subspace of \mathbb{R}^n or, in general, of $T_x M$ which depends on x [1]. A vector field f belongs to $\Delta(x)$ if $f(x) \in \Delta(x)$. For the system at hand (1), the distribution is: $\Delta_0(x) = \text{span}\{f_1(x), \dots, f_p(x)\}$. It represents the set of possible infinitesimal motions the system can perform in state x . In other words, a distribution is a vector space that is dependent on the current state which

demonstrates all the possible infinitely small movements that the system can accomplish. The distributions are the key to grasping the idea of the reachable space from an initial state. Using the Lie bracket operation with the initial distribution iteratively,

$$\Delta_i(x) := \Delta_{i-1}(x) \oplus \text{span}\{[f_1, f_2](x) \mid f_1, f_2 \in \Delta_{i-1}(x)\} \quad (2)$$

the reachability analysis is performed. Consequently, the definite relationship for the controllability criteria is given by Chow's Theorem.

Theorem (Chow): Consider the control system (1) and the associated distribution $\Delta_0(x) := \text{span}\{f_1(x), \dots, f_p(x)\}$. If $\lim_{i \rightarrow \infty} \Delta_i(x) = \mathbb{R}^n$ for all $x \in \mathbb{R}^n$, then the system is controllable.

As the controllability issue is addressed, one can now investigate the theoretical guarantee that solutions exist. According to the problem statement, the planned path must result from an array of selected controls which is often denoted as a curve. A curve is a piecewise differentiable function which connects the start and the end points of a path respectively. Any curve in \mathbb{R}^n which can be obtained by some choice of the controls $u_i(t)$ is called an admissible curve.

Definition (Admissible Curves): A curve $v(t) : [a, b] \rightarrow \mathbb{R}^n$ for $a < b$ is called admissible for system (1) if there exists continuous $u_i(t) : [a, b] \rightarrow \mathbb{R}$ $i = 1, \dots, p$ so that

$$\frac{d}{dt}v(t) = \sum_{i=1}^p u_i(t) f_i(v(t)) \quad (3)$$

Particularly, a curve is said to be admissible if the state trajectory results in it when it is expressed by any arbitrary set of controls $u_i(t)$. Hence, the lemma ensues

Lemma (Admissibility): A curve $v(t)$ is admissible if and only if

$$\frac{d}{dt}v(t) \in \Delta_0(v(t)) \quad (4)$$

In conclusion, it is clear that if a system is controllable, for any pair of $x_0, x_1 \in \mathbb{R}^n$, there exists a horizontal curve $v(t)$ connecting the initial state to the final state. Lastly, one should also describe the length of a curve by employing Riemannian geometry. The length of a curve $x(t)$ is defined with respect to a norm on the tangent bundle $T_x M$ of M [2]. When the state-space is a manifold M , a Riemannian metric is defined as an inner product on $T_x M$ which depends smoothly on $x \in M$ [1]. The length of a curve $p(t)$ is then given by:

$$L(x) := \int_0^T \sqrt{\dot{x}^\top(t) G(x(t)) \dot{x}(t)} dt \quad (5)$$

where G is an inner product defining a Riemannian metric. It is a well-known fact that the energy functional depends on the time parametrization, but the length does not. However, their minimizers are the same [13] which is why the length is primarily discussed.

B. The Method MotionSketch

Having established the basics, now one can move on to how the proposed underlying method works. The method MotionSketch [2] contains the three following steps:

- 1) Encode the constraints of the motion planning problem (obstacles, holonomic, non-holonomic and dynamical constraints) into a Riemannian inner product.
- 2) Provide a curve in configuration space between the initial and final desired states. This curve, which we call the sketch, does not need to meet the holonomic, non-holonomic and dynamical constraints, but is required to avoid obstacles. Numerically solve the geometric heat flow (GHF), defined below, equation with the sketch as initial condition.
- 3) Extract the controls from the solution of the GHF.

The starting point to elaborate on these steps is the following: assuming the control system (1) in \mathbb{R}^n , one has the distribution Δ_0 with constant rank over the state space, which represent the physical directions of motion for the system. To represent all the possible directions in the space which may or may not be admissible, the concept of orthogonal distribution is introduced. The distribution $\Delta_0^\perp(x)$ is defined as the orthogonal distribution to Δ_0 :

$$\Delta_0^\perp(x) := \left\{ v \in \mathbb{R}^n \mid v^\top w = 0 \text{ for all } w \in \Delta_0(x) \right\} \quad (6)$$

The orthogonal distribution is a vector space that contains the vector fields which are orthogonal to every vector field in the distribution of the system. Recalling the fact that a distribution represents the the set of possible infinitesimal motions the system can perform in state x , the orthogonal distribution interprets to the set of infinitesimal motions which the system can not perform (inadmissible directions) in the state x . It is like a complement for the set of axes on which the system can move.

By adhering to the derivation of holonomic and non-holonomic constraints from [2], it is known that performing an admissible motion means that the direction of motion δx must be orthogonal to the undesirable directions of motion that are characterized by the holonomic and non-holonomic constraints. In this regard, the desirable directions of motion is computed by finding a matrix $F_f(x)$ that is orthogonal to $F_c(x)$ which represents the inadmissible directions. Following this, the matrix F is written as such:

$$F(x) = \begin{pmatrix} | & | \\ F_c(x) & F_f(x) \\ | & | \end{pmatrix} \quad (7)$$

Then, the matrix G is defined to encode the constraints into a Riemannian inner product.

$$G(x) = F(x)DF^\top(x) \quad (8)$$

where $D = \text{diag}(\underbrace{[k \cdots k]_{n-p}}_{n-p}, \underbrace{[1 \cdots 1]_p}_p)$, n being the total number of

degrees of freedom in the space while p being the degrees of freedom of the system.

The main idea behind this technique is to initialize a trajectory called a sketch which does not need to meet the constraints (except the obstacle avoidance constraints) and deform this curve *via a homotopy* into an admissible trajectory from which controls realizing the transfer can be obtained. A fixed end-points homotopy [2] between the two curves $x_1(t)$ and $x_2(t)$ with the same end-points (i.e., $x_1(0) = x_2(0)$ and $x_1(T) = x_2(T)$) is a differentiable function $v(s, t) : [0, \infty) \times [0, T] \rightarrow M$ with the properties:

$$\begin{aligned} v(s, 0) &= x_1(0) & \text{for all } s \geq 0 \\ v(s, T) &= x_1(T) & \text{for all } s \geq 0 \end{aligned}$$

Relying on the information above, one defines the geometric heat flow (GHF) equation which is a parabolic PDE whose solutions converge to an admissible path $v(t, \infty)$ joining x_0 and x_1 as $k \rightarrow \infty$.

$$\frac{\partial}{\partial s} v_i(t, s) = \frac{\partial^2}{\partial t^2} v_i(t, s) + \sum_{j,k} \Gamma_{jk}^i \frac{\partial v_j}{\partial t} \frac{\partial v_k}{\partial t} \quad (9)$$

where Γ_{jk}^i are the Christoffel symbols of $G(x)$.

$$\Gamma_{jk}^i(x) := \frac{1}{2} \sum_l g^{il} \left(\frac{\partial g_{lj}}{\partial x_k} + \frac{\partial g_{lk}}{\partial x_j} - \frac{\partial g_{jk}}{\partial x_l} \right) \quad (10)$$

This family of PDEs that are coupled by their second terms called evolution equations, can be solved using the explicit iterative procedure

$$v_i(t, s+ds) := \frac{\partial^2}{\partial t^2} v_i(t, s) + \sum_{j,k} \Gamma_{jk}^i(v(t, s)) \frac{\partial v_j(t, s)}{\partial t} \frac{\partial v_k(t, s)}{\partial t} \quad (11)$$

where the partial derivative are approximated via finite differences in the t variable. The solution procedure is illustrated below.

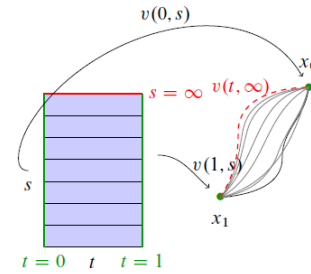


Fig. 1: An Illustration of the Explicit Iterative Procedure to Solve the GHF [1]

The curves $v(t, \cdot) : [0, 1] \rightarrow \mathbb{R}^n$ act as the state-variable, and s as the time-variable for the curve evolution. For large values of k , one can observe that the solution $v(t, \infty)$ tends to an admissible curve joining v_0 to v_1 [1], that is

$$\frac{d}{dt} v(t, \infty) \in \Delta_0(v(t, \infty)) \quad (12)$$

Finally, one can obtain the controls from the admissible curve $v(t, \infty)$ using

$$\begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_p(t) \end{pmatrix} = F_p^\dagger(v(t))\dot{v}(t) \quad (13)$$

III. IMPLEMENTATION AND RESULTS

A. Benchmark Example

The main goal of this case study is to demonstrate that the proposed novelty method works for a benchmark example and to extend it to an application of interest. The benchmark example considers a two link planar arm which has 2 degrees of freedom being the joint angles (θ_1, θ_2) . The actuators are servo drives which are located on the joints. The motors apply a certain commanded torque that rotates the joint which results in the motion of the end-effector. The motion planning problem for this particular case aims to figure out an admissible sequence of controls to steer the end-effector of the robot from an initial state x_i to a final state x_f . The holonomic constraints that describe the dynamics of the motion is of the subsequent form:

$$\begin{aligned} q_1(x) &= L_1 \cos(\theta_1) + L_2 \cos(\theta_2) - x = 0 \\ q_2(x) &= L_1 \sin(\theta_1) + L_2 \sin(\theta_2) - y = 0 \end{aligned} \quad (14)$$

where L_1 and L_2 are assumed to be 1. By taking the differential of the two constraints, one obtains

$$\begin{aligned} \frac{\partial q_1}{\partial x} &= (-1, 0, -L_1 \sin \theta_1, L_2 \sin \theta_2)^\top \\ \frac{\partial q_2}{\partial x} &= (0, -1, L_1 \cos \theta_1, L_2 \cos \theta_2)^\top \end{aligned} \quad (15)$$

to form the F_c and F_f matrices where $F_f(x)^\top F_c(x) = 0$ is satisfied.

$$F_c = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \sin \theta_1 & -\cos \theta_1 \\ \sin \theta_2 & -\cos \theta_2 \end{pmatrix}, \quad F_f = \begin{pmatrix} -\sin \theta_1 & -\sin \theta_2 \\ \cos \theta_1 & \cos \theta_2 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (16)$$

Using $G(x) = F(x)DF^\top(x)$ yields

$$G(x) = \begin{pmatrix} \sin^2 \theta_1 + \sin^2 \theta_2 + k & -\frac{\sin 2\theta_1}{2} - \frac{\sin 2\theta_2}{2} & (k-1) \sin \theta_1 & (k-1) \sin \theta_2 \\ -\frac{\sin 2\theta_1}{2} - \frac{\sin 2\theta_2}{2} & \cos^2 \theta_1 + \cos^2 \theta_2 + k & -(k-1) \cos \theta_1 & -(k-1) \cos \theta_2 \\ (k-1) \sin \theta_1 & -(k-1) \cos \theta_1 & k+1 & k \cos(\theta_1 - \theta_2) \\ (k-1) \sin \theta_2 & -(k-1) \cos \theta_2 & k \cos \theta_1 - \theta_2 & k+1 \end{pmatrix} \quad (17)$$

where $k = 300$. Later, the above defined geometric heat flow equation is solved using the pdepe function in MATLAB. The curve and the time interval is discretized with 250 points, initial and boundary conditions are set in separate functions. The simulation was executed for a

total time of 4 seconds. Here are the initial and boundary conditions for the benchmark example exhibited in the table below.

Boundary Condition - Initial			
x	y	θ_1	θ_2
$\frac{\sqrt{2}}{2}$	$1 - \frac{\sqrt{2}}{2}$	$\frac{\pi}{2}$	$-\frac{\pi}{4}$
Boundary Condition - Final			
x	y	θ_1	θ_2
$\frac{\sqrt{2}}{2}$	$1 + \frac{\sqrt{2}}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{4}$
Initial Condition			
x	y	θ_1	θ_2
$\frac{\sqrt{2}}{2}$	$1 - \frac{\sqrt{2}}{2} + (\sqrt{2}t)$	$\frac{\pi}{2}$	$-\frac{\pi}{4} + (\frac{\pi}{2}t)$

TABLE I: Table of Initial and Boundary Conditions for the Benchmark Example

The comparison of the target and the actual path is illustrated in the successive figures.

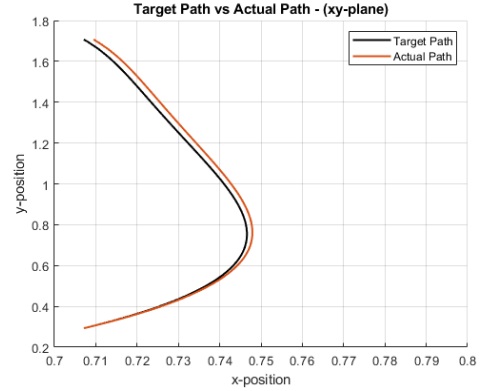


Fig. 2: Results for the Benchmark Example (xy-plane)

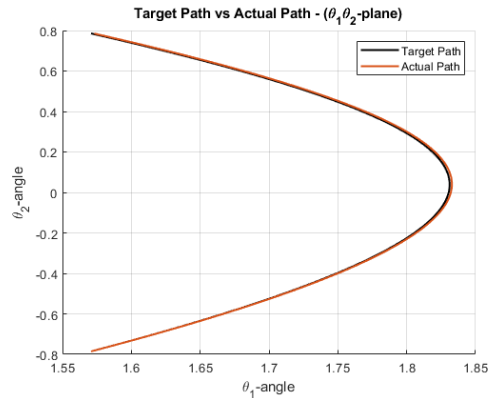


Fig. 3: Results for the Benchmark Example $(\theta_1 \theta_2$ -plane)

The difference between the target path and the actual path $v(t, \infty)$ decreases as s increases. Additionally, the error diminishes as the number of points set to describe a curve $v(t, \cdot)$ rises. As this implementation considers finite discretization points and s , the actual and the target path differ.

B. Two-link Planar Arm with Additional Holonomic Path Constraints

The application of interest for this case study was to implement additional path constraints for the benchmark example. Two constraints on the path (q_3, q_4) were considered separately which restrict the motion of the end-effector to be vertical and circular respectively. The constraints are

$$\begin{aligned} q_3(x) &= x - x_i = 0 \\ q_4(x) &= (x - x_c)^2 + (x - y_c)^2 - r^2 = 0 \end{aligned} \quad (18)$$

where $(x_c, y_c, r) = (0, 1, 1)$. The differentials yield

$$\begin{aligned} \frac{\partial q_3}{\partial x} &= (1, 0, 0, 0)^\top \\ \frac{\partial q_4}{\partial x} &= (2x, 2y - 2, 0, 0)^\top \end{aligned} \quad (19)$$

Abiding by the process, F_c , F_f , matrices are attained to compute the G matrix for each case. Due to their size, these matrices are too large to be explicitly written out. Their derivation will be present in the uploaded material. The same initial and boundary conditions as in the benchmark example were used for each of the cases.

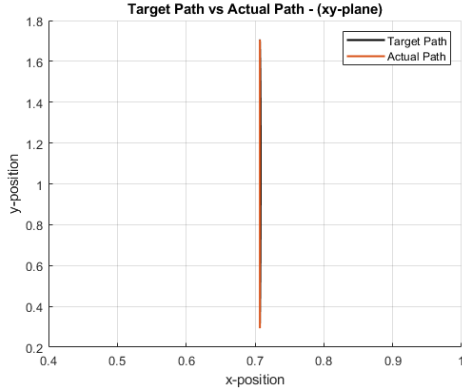


Fig. 4: Results for the Vertical Motion Example (xy-plane)

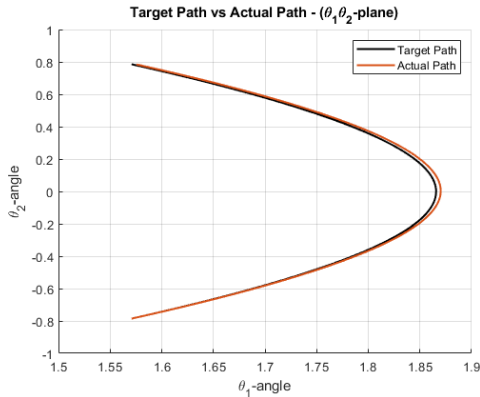


Fig. 5: Results for the Vertical Motion Example $(\theta_1, \theta_2\text{-plane})$

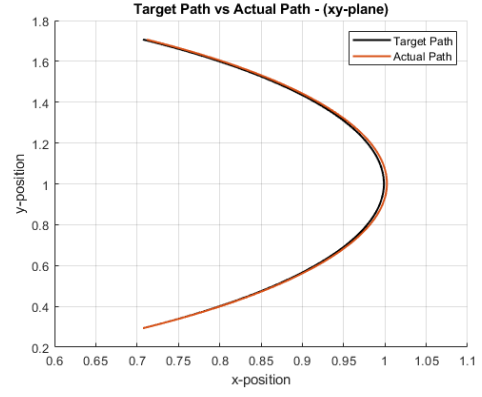


Fig. 6: Results for the Circular Motion Example (xy-plane)

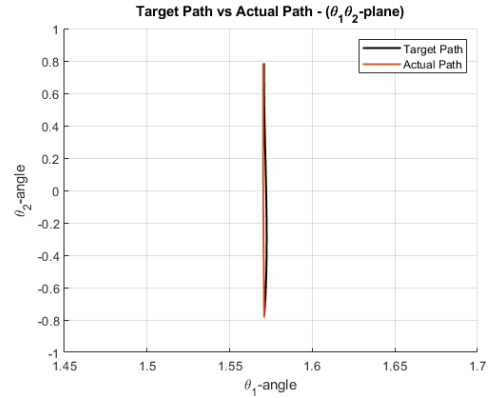


Fig. 7: Results for the Circular Motion Example $(\theta_1, \theta_2\text{-plane})$

The error magnitudes are presented in Table II.

Average Error of the Actual Path				
Case	x	y	θ_1	θ_2
Benchmark	$3.2 \cdot 10^{-4}$	$4.3 \cdot 10^{-3}$	$3.7 \cdot 10^{-3}$	$6.3 \cdot 10^{-4}$
Vertical Motion	$4.1 \cdot 10^{-4}$	$4.4 \cdot 10^{-3}$	$3.8 \cdot 10^{-3}$	$7.2 \cdot 10^{-4}$
Circular Motion	$2.5 \cdot 10^{-4}$	$4.2 \cdot 10^{-3}$	$3.7 \cdot 10^{-3}$	$5.6 \cdot 10^{-4}$

TABLE II: Error Magnitudes

IV. CONCLUSION

In this case study, a proposed novel method for motion planning of non-holonomic systems were discussed and summarized. The method termed MotionSketch was explained in detail with the underpinning PDE problem. As an extension to the work of M. Belabbas and S. Liu [1], [2] the technique was implemented with additional constraints for an application of interest. The results were illustrated in comparison and the viability of the method was shown with the similarity of error magnitudes. For further improvement of the results, discretization steps and the "time" variable s can be increased [1]. Future work could be done on how to implement additional non-holonomic or obstacle constraints.

ACKNOWLEDGEMENT

This case study was done as a part of the Mathematical Simulation Techniques course at Technische Universität Dortmund. The guidance and the support of the instructor and the teaching assistants are acknowledged.

REFERENCES

- [1] Belabbas, Mohamed Ali, and Shenyu Liu. "New method for motion planning for non-holonomic systems using partial differential equations." 2017 American Control Conference (ACC). IEEE, 2017.
- [2] Liu, Shenyu, and Mohamed Ali Belabbas. "A homotopy method for motion planning." arXiv preprint arXiv:1901.10094 (2019).
- [3] Latombe, Jean-Claude. Robot motion planning. Vol. 124. Springer Science Business Media, 2012.
- [4] Zhang, Jian. "AI based Algorithms of Path Planning, Navigation and Control for Mobile Ground Robots and UAVs." arXiv preprint arXiv:2110.00910 (2021).
- [5] Schultz, Jarvis, Elliot Johnson, and Todd D. Murphey. "Trajectory optimization in discrete mechanics." Differential-Geometric Methods in Computational Multibody System Dynamics. Springer International Publishing (2015).
- [6] C. R. Hargraves and S. W. Paris, "Direct Trajectory Optimization Using Nonlinear Programming and Collocation", Journal of Guidance, Control, and Dynamics, vol. 10, no. 4, pp. 338–342, 1987.
- [7] Hargraves, Charles R., and Stephen W. Paris. "Direct trajectory optimization using nonlinear programming and collocation." Journal of guidance, control, and dynamics 10.4 (1987): 338-342.
- [8] Brockett, Roger W., and Liyi Dai. "Non-holonomic kinematics and the role of elliptic functions in constructive controllability." Nonholonomic motion planning. Springer, Boston, MA, 1993. 1-21.
- [9] Laumond, Jean-Paul, ed. Robot motion planning and control. Vol. 229. Berlin: Springer, 1998.
- [10] Kavraki, Lydia E., et al. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces." IEEE transactions on Robotics and Automation 12.4 (1996): 566-580.
- [11] Chen, Pang C., and Yong K. Hwang. "SANDROS: a dynamic graph search algorithm for motion planning." IEEE Transactions on Robotics and Automation 14.3 (1998): 390-403.
- [12] Khatib, Oussama. "Real-time obstacle avoidance for manipulators and mobile robots." Autonomous robot vehicles. Springer, New York, NY, 1986. 396-404.
- [13] Jost, Jürgen, and Jèurgen Jost. Riemannian geometry and geometric analysis. Vol. 42005. Berlin: Springer, 2008.