Case Study

▼ Abstract

We present in this paper a novel approach to the long-standing problem of motion planning for non-holonomic systems. Our method is built upon a parabolic partial differential equation that arises in the study of Riemannian manifold. We show how it can be brought to bear to provide a solution to a non- holonomic motion planning problem. We illustrate the method on canonical examples, namely the unicycle, the non-holonomic integrator, and the parallel parking task for a non-holonomic car model. We also briefly address computational issues pertinent to solving this particular partial differential equation, and point out the existence of fast algorithms and the fact that the problem is easily parallelizable.

▼ What makes motion planning for nonlinear systems difficult?

As an introduction to this case study, an essential concept like holonomy, must be revisited 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.

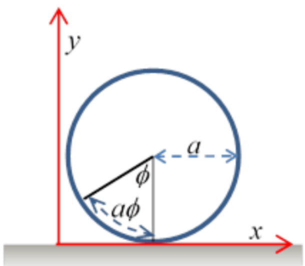
Furthermore in dynamics, systems can be classified as holonomic, non-holonomic or neither. A holonomic system's trajectory integrals depend only upon the initial and final states of the system. One can mathematically devise the path when the initial and final states are presented, therefore the system 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.

A holonomic/non-holonomic system is defined by a set of parameters which are subject to differential constraints. A system may be defined as holonomic/non-holonomic if all constraints of the system are holonomic/non-holonomic. A constraint can be outlined as a relationship between the position variables (and possibly time) as follows:

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

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

A simple example would suffice to grasp the difference between holonomic and non-holonomic constraints. Imagine a cylinder with radius a rolling in the x-direction, its orientation ϕ defined as zero as it passes the origin.



One can see immediately that its orientation is uniquely given by its position (if the point in contact with the surface doesn't slip) by $x = a\phi$, or $v - a\dot{\phi}$. Using the constraint, one can eliminate one of the dynamical variables from the equation. By measuring its position at some later time, one can determine the angle through which it turned.

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.

Setting the definitions aside, now we focus on the main idea behind this study. Motion planning for under-actuated non-linear systems is known to be a challenging problem to tackle.

A number of factors contribute to the difficulty of the problem: The nature of a non-holonomic systems 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 nonlinear case, but is considerably more challenging when combined with non-holonomic constraints.

▼ **What other approaches for motion planning had been studied in the past? Give at least three examples, and provide a reference for each method, including a complete citation (like I did above).**

Numerous ad-hoc techniques employing artificial intelligence [2] 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. In addition, there are other methods that rely on optimal control techniques [3]. 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 [4]. Alternatively, methods requiring a strong mathematical foundation doesn't scale well, such as [5], [6], or [7]. It can be concluded that these approaches cannot handle realistic scenarios involving more than a few variables.

▼ **What is the novelty of this paper?**

In this paper, a new method for motion planning of non-holonomic systems is presented. Motion planning is the idea of computing controls and trajectories of robots from an initial state to a final state. Conversely to the previously mentioned work, the method is relatively uncomplicated and seemingly easier 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 research. Authors 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 sense. Moreover, the obstacle avoidance issue and the rigorous derivation of the result is left out for future work.

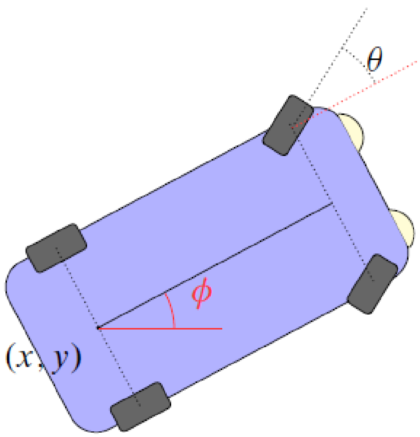
▼ **What is meant by the distribution $\Delta_0(x)$ of a robot in state x ?**

As the justification of the presented method requires introducing notions from Riemannian geometry and partial differential equations, some mathematical preliminaries must be established to set the scene. The explained approach is worked in a manifold of form $M = \mathbb{R}^n$. Firstly, we define the system linear in the control $\dot{x} = \sum_{i=1}^{p} u_i f_i(x)$ **(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 following: given two states $x_0, x_1 \in \mathbb{R}^n$, find a set of controls $u_i(t)$ that drive the above-mentioned system from x_0 to x_1 in $T > 0$ seconds. 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 product 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. A vector field f belongs to $\Delta(x)$ if $f(x) \in \Delta(x)$.

For the system at hand, the distribution is: $\Delta_0(x) = span\{f_1(x), ..., f_p(x)\}$. 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.



For example, a boxcar illustrated above is described by its position (x, y), orientation θ and the angle of its front wheels w.r.t. the current orientation ϕ.

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ \theta \\ \phi \end{pmatrix} = u_1 \underbrace{\begin{pmatrix} \cos\phi \\ \sin\phi \\ 0 \\ \frac{1}{d}\sin\theta \end{pmatrix}}_{f_1} + u_2 \underbrace{\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}}_{f_2}$$

The distribution $\Delta_0(x)$ corresponds to $\Delta_0(x) = span\{f_1(x), f_2(x)\}$

The distributions are the key to grasping the idea of the reachable space from an initial state. Following this, the definite relationship for the controllability criteria is given by Chow's Theorem.

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

▼ **In plain English, what does it mean for a curve $v(t)$ to be admissible?**

Any curve in $\mathbb{R}^n$ which can be obtained by some choice of the controls $u_i(t)$ is called an admissible curve. 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 $u_i(t)$.

- ***Definition (Admissible Curves)*** A curve $v(t) : [a, b] \to \mathbb{R}^n$ for $a < b$ is called admissible for system **(1)** if there exists continuous $u_i(t) : [a, b] \to \mathbb{R}^n, i = 1, ..., p$ so that $\frac{d}{dt}v(t) = \sum_{i=1}^{p} u_i f_i(v(t))$
- ***Lemma:*** A curve $v(t)$ is admissible if and only if $\frac{d}{dt}v(t) \in \Delta_0(v(t))$

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.

▼ **If $v(t)$ is an admissible curve with initial state $v(0) = x_0$ and final state $v(1) = x_1$, how can you find controls $u_1(t), ..., u_p(t)$ that move the robot along this curve?**

If an admissible curve $v(t)$ joining x_0 to x_1 exists, one can easily find controls $u_i(t)$ that drives the system from the initial state to the final state.

- First and foremost, one should define the matrix:

$$F_p(x) := \begin{pmatrix} f_1(x) \; f_2(x) \; ... \; f_p(x) \end{pmatrix}$$

- ***Lemma:*** Assume that system **(1)** is controllable and let $v(t)$ be an admissible curve for $\Delta_0(x)$ so that $v(0) = x_0$ and $v(1) = x_1$. Then the controls $u_i(t)$, $i = 1, ..., p$ satisfying the equation

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

drive the system from x_0 to x_1. The controls u_i doesn't have to be unique for a given $v(t)$. They are unique if and only if $\Delta_0(x)$ is of rank p everywhere.

▼ **How can the orthogonal distribution $\Delta_0^{\perp}(x)$ of a robot in state x be interpreted?**

- For the control system at hand **(1),** the distribution is defined as:

$\Delta_0(x) = span\{f_1(x), ..., f_p(x)\}$

- The distribution $\Delta_0(x)$ is assumed to have rank p for all $x \in \mathbb{R}^n$. The orthogonal distribution $\Delta_0^{\perp}(x)$ to $\Delta_0(x)$ is defined as the following:

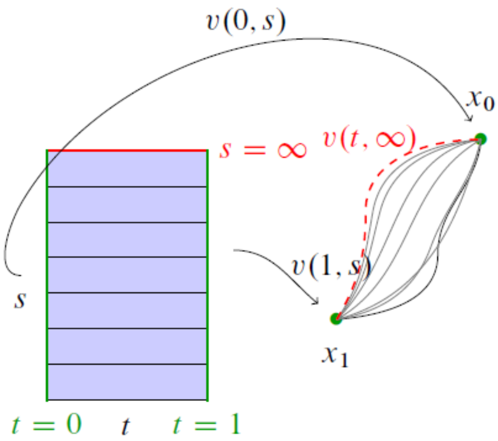
$\Delta_0^{\perp}(x) := \{v \in \mathbb{R}^n | v^T w = 0$ for all $w \in \Delta_0(x)\}$

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 (unadmissible directions) in the state x. It is like a complement for the set of axes on which the system can move.

▼ **What are the unknown functions v_i in Equation (13)? How many v_i's are there for the box car example?**

By adhering to the derivation from the paper, one can define a family of partial differential equations parametrized by $c > 0$. The solutions of these PDEs converge to an admissible path $v(t, \infty)$ joining x_0 and x_1 as $c \to \infty$.

The family of curves joining x_0 to x_1 is denoted by $v(t, s)$ and every curve in this set for each $s \geq 0$, has x_0 as the initial and x_1 as the final state. Yet, every curve may not be necessarily an admissible curve, the admissible curve is obtained as s tends to ∞.



The unknown function $v_i(t, s)$ are the ith member of the family of curves $v(t, s)$. The equation **(13)** is defined as such:

$$\frac{\partial}{\partial s} v_i(t, s) = \frac{\partial^2}{\partial t^2} v_i(t, s) + \sum_{j,k} \Gamma^i_{jk} \frac{\partial v_j}{\partial t} \frac{\partial v_k}{\partial t}$$

The system comprises partial differential equations that are coupled by second order terms, called evolution equations. This can be solved by pursuing an explicit iterative procedure where $s \to \infty$. The explicit iterative procedure yields:

$$v_i(t, s + ds) = \frac{\partial^2}{\partial t^2} v_i(t, s) + \sum_{j,k} \Gamma^i_{jk}(v(t, s)) \frac{\partial v_j(t, s)}{\partial t} \frac{\partial v_k(t, s)}{\partial t}$$

The number of the unknown functions $v_i(t, s)$ depends on the amount of the (decided) discretization steps for the evolution of the admissible curve. Let q be the number of discretization points of the curves $v(t, \cdot)$. In the boxcar example, the actual path is denoted as $v(t, 3)$ meaning that there exists $3/q + 1$ total curves $v_i(t, s)$ computed.

▼ **For each i, Equation (13) looks like a heat equation with some additional advection and reaction terms. These coupled PDEs have to be solved until steady state $s \to \infty$ or else the computed curve may not be admissible yet. Also the parameter has to be chosen very large, or else the computed curve may not be admissible yet.**

▼ **What are the initial and boundary values for this coupled system of PDEs and why?**

An important detail to touch upon is that this coupled system of PDEs can have a customized curve as an initial condition $v(t, 0)$. Considering this algorithm leads to an admissible curve as $s \to \infty$, any arbitrary curve (even inadmissible) can be selected as an initial condition. And, since every curve obtained during this procedure joins x_0 and x_1, the boundary conditions are $v(0, s_0) = x_0$ and $v(1, s_0) = x_1$.

For a large c, it is observed that solution $v(t, \infty)$ tends to an admissible curve joining initial and final state: $\frac{d}{dt} v(t, \infty) \in \Delta_0(v(t, \infty))$.

▼ **What discretisation scheme do Belabbas and Liu use in 'space' t and what scheme do they use in 'time' s? (NB: These are not actually space and time here, but in the heat equation and other diffusion-advection-reaction equations we have seen previously these variables had the interpretation of space and time.)**

In this particular method, curves are parametrized by t and s which can be interpreted as the *state* and *time* variable respectively. The discretization scheme used is finite difference approximation which approximates the partial derivatives via finite differences in the t variable.

▼ **In what sense is a trajectory computed from the PDE optimal?**

To intuitively understand what the calculated trajectory corresponds to, the definitions for length and energy functionals must be introduced.

- ***Definition (Length and Energy Functionals)***

  Let $v : [0, T] \to M$ be differentiable curve in a Riemannian manifold with inner product $G(x)$. We define the length and energy functionals as

  - $L(v) = \int_0^T \sqrt{\dot{v}^T G \dot{v}} \ dt$ and,
  - $E(v) = \frac{1}{2} \int_0^T \dot{v}^T G \dot{v} \ dt$ respectively.

The trajectory computed from the system of PDEs is optimal in the sense of the consumed energy for the control actions because the calculated admissible curve minimizes the stated functional:

$$E(v(\cdot, s)) : \int_0^1 v^T(t, s) G(v(t, s)) v(t, s) \ dt$$

which evidently gives an admissible curve that spends the least amount of energy for the system (if $c \to \infty$).

▼ **In Figure 3 (c) and (d), what is the difference between the actual path and the target path and why are they not the same?**

In the boxcar example, the difference between the trajectory of the original system using the obtained control and the target path $v(t, s = \infty)$ is caused due to the actual path being $v(t, s = 3)$. The presented technique only achieves the target path as $s \to \infty$ and $c \to \infty$. This error is mentioned to be decreasing if s or the number of points describing a curve $v(t, \cdot)$ is increased.

▼ **When was this conference paper published? Provide at least three citations of papers published since then, which cite the paper of Belabbas and Liu.**

This conference paper was published in 24-26 May 2017, in Seattle, USA at the American Control Conference. This paper was cited by the following papers which were published after it:

- Liu, S., Fan, Y., & Belabbas, M. A. (2019). Affine geometric heat flow and motion planning for dynamic systems. *IFAC-PapersOnLine*, *52*(16), 168-173.
- Liu, S., & Belabbas, M. A. (2019). A homotopy method for motion planning. *arXiv preprint arXiv:1901.10094*.
- Fan, Y., Liu, S., & Belabbas, M. A. (2019). Mid-air motion planning of floating robot using heat flow method. *IFAC-PapersOnLine*, *52*(22), 19-24.

▼ **References**

- https://www.physics.uci.edu/~tanedo/files/teaching/P3318S13/Sec_05_nonholonomic.pdf
- https://phys.libretexts.org/Bookshelves/Classical_Mechanics/Graduate_Classical_Mechanics_(Fowler)/30%3A_A_Rollir Holonomic_Constraints
- https://en.wikipedia.org/wiki/Nonholonomic_system
- [2] J.-C. Latombe, "*Robot motion planning",* Springer Science & Business Media, 2012, vol. 124.
- [3] J. Schultz, E. Johnson, and T. D. Murphey, "*Trajectory Optimization in Discrete Mechanics",* Springer International Publishing, In Press.
- [4] 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.
- [5] R. M. Murray and S. S. Sastry, "Nonholonomic motion planning: Steering using sinusoids", *IEEE Transactions on Automatic Control*, vol. 38, no. 5, pp. 700-716, 1993.
- [6] R. Brockett and L. Dai, "Non-holonomic kinematics and the role of elliptic functions in constructive controllability", in *Nonholonomic motion planning*. Springer, 1993, pp. 1–21.

- [7] J. Laumond, “*Robot motion planning and control*”, ser. Lecture notes in control and information sciences. Springer, 1998.