Bounded-Curvature Path Normalization

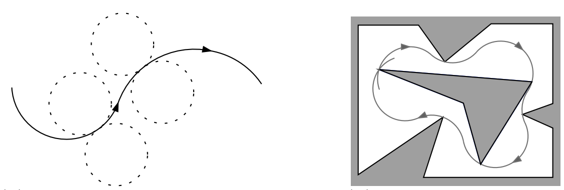
Jonathan Backer* David Kirkpatrick†

Abstract

We present a new normal form for bounded-curvature paths that admits a combinatorial discription of such paths and a bound on the algebraic complexity of the underlying geometry.

1 Introduction

We are interested in paths of bounded curvature between two configurations that avoid polygonal obstacles. Such paths approximate the motion of vehicles with a restricted turning radius such as cars and bicycles. We assume that the vehicle's turning radius is one without loss of generality by scaling the obstacles. In this case, a bounded-curvature path is a path that can be sandwiched between two unit radius circles at every point (see Figure 1(a)), which restricts how quickly the path turns left or right. This restriction makes the instantaneous configuration of a point tracing a bounded-curvature path a combination of the point's position and *orientation*. It also complicates bounded-curvature motion planning because the only path between two similar configurations may be arbitrarily long and complicated (see Figure 1(b)).



(a) Wedged between circles. (b) Cycling behaviour.

Figure 1: Bounded-curvature paths.

Rather than trying to find the shortest bounded-curvature path (optimization), we focus on finding any bounded-curvature path (feasibility) because optimization is NP-hard in general [9] and the best general feasibility algorithm is exponential in time and space [7]. Despite the NP-hardness of optimization, there are efficient incomplete approximation algorithms [8, 10]; these algorithms are *incomplete* because they may not find a bounded-curvature path even if one exists. Unfortunately, the NP-hardness of optimization reduction is clearly feasible and highly structured; in particular, the reduction construction prohibits path cycling and uses infinitesimally small holes through which all paths must pass as checkpoints.

The problem posed by cycling is that it may be required to incrementally change a vehicle's orientation (see Figure 1(b)) resulting in a long and complicated path. Fortune and Wilfong resolve this problem with a closure operation at the computational cost of exponential space and time. The normal form presented in this paper is a first step towards an alternative approach: finding a feasibility algorithm whose runtime is proportional to the simplest description of any feasible path (i.e. output sensitive).

2 Normal Forms

All of the complete bounded-curvature motion planning results either analyze the structure that paths must take (implicit normalization) [2, 4, 1, 3] or perturb a given path to give it structure (explicit normalization) [6, 7]. This normalization reduces the space of considered paths and allows a systematic exploration of the path space.

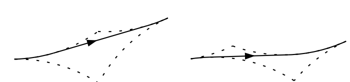


Figure 2: Wedged between cones.

Dubins' seminal work [6] characterizes shortest bounded-curvature paths in the absence of obstacles. He first notices that short bounded-curvature paths are contained between two cones (see Figure 2); hence shortest paths are made of arcs of unit radius circles (C-segments) and straight line segments (S-segments). Using length reducing perturbations, Dubins shows that the *structure* of shortest paths (i.e. the sequence of C- and S- segments of which it is made) must be either CSC or CCC, where each segment may have zero length. Wang and Agarwal [10] argue that the shortest path amidst obstacles is a sequence of Dubins type paths between obstacle contacts.

The normalization of Fortune and Wilfong [7] essentially pushes a shortest path against obstacles so that

*Department of Computer Science, University of British Columbia, backer@cs.ubc.ca

†Department of Computer Science, University of British Columbia, kirk@cs.ubc.ca

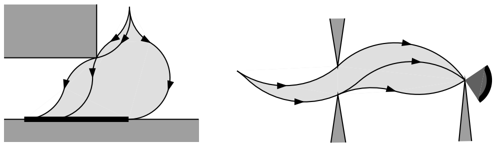


Figure 3: Feasible contact intervals.

every C-segment touches the boundary. This allows us to represent the space of feasible paths as intervals of feasible obstacle contact (see Figure 3). It also allows us to view a normal path as a sequence of jumps between obstacle contacts, where a *jump* is a subpath whose structure is CSC. By analyzing how jumps get from one interval of contact to another, Fortune and Wilfong polynomially upper bound the number of distinct intervals of obstacle contact after merging overlapping intervals. Fortune and Wilfong then reduce the feasibility problem to the first-order theory of the reals by closing the set of intervals with respect to jumps. However, the compact representation of merged intervals prevents path recovery.

3 Normalization

Our path normalization eliminates degrees of freedom from path segments until the path is completely fixed. Each path segment has at most two degrees of freedom because the path is C^1continuous; that is, if we represent the path with underlying unit radius circles and lines, each with only two degrees of freedom, the path segment end-points are the points of tangency between the lines and circles.

Fortune and Wilfong's normalization eliminates a degree of freedom from each C-segment by pushing it into contact with an obstacle. We generalize their notion of supporting a segment by an obstacle in order to remove a degree of freedom from S-segments. Specifically, we say that a $C-$ or $S-$segment is *supported* by an obstacle if it either touches the obstacle (see right Figure 6) or reaches the obstacle with a length π C-segment (see left Figure 6); we view the later as being tangent to the Minkowski sum of the obstacle with a radius two circle. We take a Fortune and Wilfong path and cause S-segments to become supported by using perturbations: continuous path deformations that maintain the path's obstacle support, homotopy class, and structure.

Each perturbation rotates or translates a unit radius circle underlying the path, which causes the path segments to stretch, shrink, or directly contact an object. These effects eventually lead to increased support: when a C-segment degenerates at length 0 or π, an adjacent segment becomes supported; when a S-segment degenerates at length 0, we consider it trivially supported.

We call a maximal contiguous sequence of supported segments a *chain* because perturbing one segment in the chain perturbs the whole chain (see Figure 4). We call a segment with two obstacle supports *anchored* because it has no degree of freedom; we call a chain with at least one anchored segment *anchored* because none of the segments in the chain can be perturbed. After our normalization, a path is a sequence of anchored chains joined by unsupported S-segments.

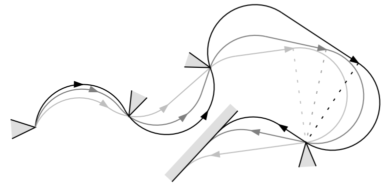


Figure 4: Perturbing a chain.

3.1 Chains

An unanchored chain can be perturbed until either the chain becomes anchored, the jump preceding the chain becomes a hop, or the jump trailing the chain becomes a hop. These events cause the chain to either become anchored or merge with another chain, both of which reduce the number of unanchored chains. Our normalization always shifts the first unanchored chain on the path in the direction that extends the preceding S-segment. This criterion was selected because the maximum S-segment length is always achieved at anchored hops (i.e. no local maxima) and bounded by the contact geometry.

3.2 Jumps

Our perturbations *shift* a C-segment about its support by either rotating it about a corner or sliding it along a wall. We examine the effects of these perturbations at the jump level because we maintain the Fortune and Wilfong property that a path is a sequence of jumps.

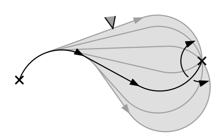


Figure 5: Shifting a C-segment.

Shifting a C-segment continuously deforms at least two jumps: jumps that use part of the C-segment and jumps that start or end at one of the C-segment's endpoints (because we have to keep the path C^1 continuous). If the affected jump has an unsupported S-segment, it can continuously deform until either a segment bumps into an obstacle or degenerates (see Figure 5). If the former happens, we simplify our analysis by considering the jump a sequence of two jumps; if the

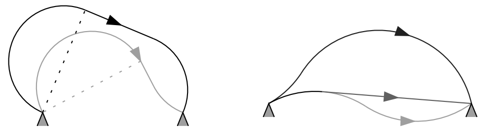
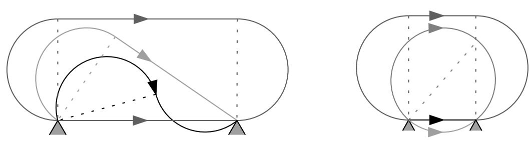


Figure 6: Hops (omitting reflections).

latter happens, we end up with a jump where every segment is supported, which we call a *hop* (see Figure 6).

Hops have a functional relationship between initial and final configuration, which causes perturbations to ripple along the chains to which they belong. When a hop is perturbed, it continuously deforms until either a hop segment degenerates or bumps into an obstacle, which results in an anchored hop completely determined by the arrangement of obstacles (see Figure 7). The continuous deformations of jump and hop perturbations guarantee that the result is homotopic to the original.



Further than 2 away. Less than 2 away.

Figure 7: Anchored hops (omitting reflections).

3.3 Hops

We now illustrate ruler and compass constructions of hops to show that they deform continuously and their domain of definition is bounded by anchored hops. In Figures 8 and 9, the obstacle contacts are marked with crosses and C-segment centers are marked with discs. Recall that S-segments can be supported by touching a π length C-segment, passing through a corner, or being degenerate. Figure 8 shows how the first two types of hops are calculated geometrically by intersecting a circle with a line tangent to another circle and parallel with the hop S-segment; Figure 9 shows how the latter type of hop is calculated by intersecting a circle with a circle.

4 Algebraic Complexity

Normal paths have a purely combinatorial description: they are made of anchored chains joined by S-segments, where anchored chains are completely determined by the sequence of hops emanating from an anchored hop. Fortune and Wilfong [7] represent corner co-ordinates as quotients of integers, where each integer has absolute value less than 2^m. This and our combinatorial perspective of paths allows to answer the question of "How close can two distinct paths be?" By addressing this issue, we can approximate real arithmetic with floating-point arithmetic of sufficient precision to compute the relative order of configurations of different paths at a contact. Such a basic calculation seems fundamental to a feasibility algorithm.

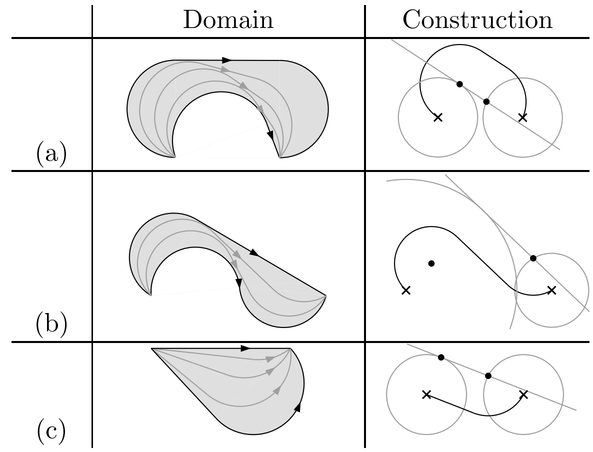


Figure 8: Hops with non-zero S-segments.

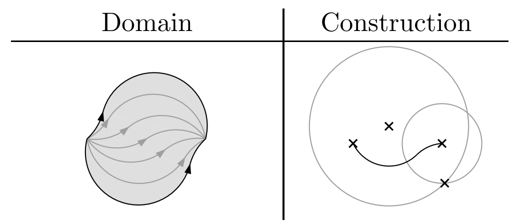


Figure 9: Hops with degenerate S-segments.

Given an arithmetic expression composed of $+, -, \times$, $/$, $\sqrt{}$ and integers, Burnikel et al. [5] describe how to construct a quotient of irreducible polynomials with integer coefficients such that the result of the arithmetic expression corresponds to a root of the quotient; the magnitude of a non-zero root is bounded above and below by the polynomial's degree and coefficients. Rather than construct the polynomial, Burnikel et al. describe how to recursively compute root bounds from the arithmetic expression. Their technique shows that the centers of circles underlying C-segments of normal paths must be at least $2^{\Theta(-k^n m)}$ apart for some constant k, where n is the number of contacts on a path. What makes the expression doubly exponential in n is that every hop involves an intersection with a circle, but k is a constant because we only intersect a fixed number of circles per hop.

5 Discussion

An advantage of our normalization is that a normal path has a combinatorial description in terms of the anchored hops that it contains, the sequence of obstacle contacts that it passes through, and the type of hops it makes between contacts. This contrasts with the Fortune and Wilfong perspective where each obstacle contact is parameterized by a real number. Our combinatorial description allows us to consider extending a path with a finite set of hops, whereas Fortune and Wilfong use an algebraic description of the bounded-curvature and obstacle constraints to consider continuous set of extensions. As such we move away from the continuous intervals of contacts of Fortune and Wilfong to relevant countable subsets. So in this sense, our normalization provides a complete discretization of the path space.

By completely describing paths structurally, we are able to analyze the algebraic complexity of the arithmetic involved, which is essential if we want to maintain completeness and use a Turing equivalent model of computation. Specifically, it tells us to what precision we would have to substitute arbitrary precision arithmetic for real arithmetic.

One potential pitfall of our normalization is that it may ill-condition the arithmetic underlying the geometric path description by pushing paths towards extremities. That is, such paths may be hard to represent with floating-point approximations to real arithmetic. However, this is a danger faced by any constraint based normalization, and any complete algorithm has to consider these types of paths to ensure feasibility.

References

[1] Pankaj K. Agarwal, Therese Biedl, Sylvain Lazard, Steve Robbins, Subhash Suri, and Sue Whitesides. Curvature-constrained shortest paths in a convex polygon (extended abstract). In *SCG '98: Proceedings of the fourteenth annual symposium on Computational geometry*, pages 392–401, New York, NY, USA, 1998. ACM Press.

[2] Pankaj K. Agarwal, Prabhakar Raghavan, and Hisao Tamaki. Motion planning for a steering-constrained robot through moderate obstacles. In *STOC '95: Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 343–352, New York, NY, USA, 1995. ACM Press.

[3] Sergey Bereg and David Kirkpatrick. Curvature-bounded traversals of narrow corridors. In *SCG '05: Proceedings of the twenty-first annual symposium on Computational geometry*, pages 278–287, New York, NY, USA, 2005. ACM Press.

[4] Jean-Daniel Boissonnat and Sylvain Lazard. A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacles (extended abstract). In *SCG '96: Proceedings of the twelfth annual symposium on Computational geometry*, pages 242–251, New York, NY, USA, 1996. ACM Press.

[5] Christoph Burnikel, Stefan Funke, Kurt Mehlhorn, Stefan Schirra, and Susanne Schmitt. A separation bound for real algebraic expressions. In *Lecutre Notes in Computer Science: European Symposium on Algorithms 2001*, volume 2161, pages 254–. Springer-Verlag, January 2001.

[6] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Amer. J. Math.*, 79:497–516, 1957.

[7] Steven Fortune and Gordon Wilfong. Planning constrained motion. *Ann. Math. Artificial Intelligence*, 3(1):21–82, 1991. Algorithmic motion planning in robotics.

[8] Paul Jacobs and John Canny. Planning smooth paths for mobile robots. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pages 2–7, 1989.

[9] John Reif and Hongyan Wang. The complexity of the two dimensional curvature-constrained shortest-path problem. In *WAFR '98: Proceedings of the third workshop on the algorithmic foundations of robotics on Robotics : the algorithmic perspective*, pages 49–57, Natick, MA, USA, 1998. A. K. Peters, Ltd.

[10] Hongyan Wang and Pankaj K. Agarwal. Approximation algorithms for curvature-constrained shortest paths. In *SODA '96: Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 409–418, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics.