CIMCA 2001 Proceedings/ISBN 0858898470: M. Mohammadian (Ed.)

Automatic Generation of Smooth Paths Bounded by Polygonal Chains

T. Berglund[1], U. Erikson[1], H. Jonsson[1], K. Mrozek[2], and I. Söderkvist[3]

[1] Department of Computer Science and Electrical Engineering,
Luleå University of Technology,
SE-971 87 Luleå, Sweden
{Tomas.Berglund,Ulf.Erikson,Hakan.Jonsson}@sm.luth.se

[2] Q Navigator AB,
SE-971 75 Luleå, Sweden
Kent@qnav.se

[3] Department of Mathematics,
Luleå University of Technology,
SE-971 87 Luleå, Sweden
Inge.Soderkvist@sm.luth.se

Abstract

We consider the problem of planning smooth paths for a vehicle in a region bounded by polygonal chains. The paths are represented as B-spline functions. A path is found by solving an optimization problem using a cost function designed to care for both the smoothness of the path and the safety of the vehicle. Smoothness is defined as small magnitude of the derivative of curvature and safety is defined as the degree of centering of the path between the polygonal chains. The polygonal chains are preprocessed in order to remove excess parts and introduce safety margins for the vehicle. The method has been implemented for use with a standard solver and tests have been made on application data provided by the Swedish mining company LKAB.

1 Introduction

We study the problem of planning a smooth path between two points in a planar map described by polygonal chains. Here, a smooth path is a curve, with continuous derivative of curvature, that is a solution to a certain optimization problem. Today, it is not known how to find a closed form of an optimal path. Instead, we give a good approximation using B-spline functions. Even though the emphasis of this paper lies on path-planning with mine maps as input, our solution is generally applicable to any map described by polygonal chains.

1.1 Background

The Swedish mining company LKAB is using unmanned autonomous vehicles for underground ore transportation. An on-board control system guides each vehicle along precomputed paths. The paths are described in a planar map of the mine that consists of polygonal chains.

A fully loaded vehicle has a weight of about 120 tons and its maximum speed is about 20 km/h. The larger the vehicle and the faster its speed, the greater is the strain put upon its construction and the smaller is the margin for error. This puts a high demand on the smoothness of precomputed paths as the steering gear of a vehicle is worn out more quickly if there are fast changes in the curvature of its path (high speed of the turning wheel). Today, the paths are handmade and, according to LKAB, the existing path-planning is time-consuming and not always satisfactory. Several successive refinements are needed to save the vehicle and the road.

Generating smooth paths is one step towards the company's goal of lowering the ore production costs. Smooth paths allow a high speed, but more important, give low maintenance costs both for the vehicles and the road.

1.2 Related work

The path-planning problem involves planning a collision-free path for a vehicle or a robot moving amid obstacles. It is one of the main problems in robotics and has been widely studied [1, 2]. Dubins [3] was the first to study curvature-constrained shortest paths. His paths are concatenations of straight lines and arcs of circles and his theories have been extended for various problems [4, 5, 6, 7].

Dubins' paths have discontinuous curvature yielding excessive wear of the steering gear of a vehicle following them (with nonzero speed). Path-planning with bounded derivative of curvature has been studied by, for example, Boissonnat *et al.* [8] and Kostov and Degtiariova-Kostova [9]. These authors work with paths formed by a concatenation of straight line segments and arcs of clothoids. Such paths have a higher degree of smoothness than Dubins' paths, but their explicit computation tend to be difficult. Lutterkort and Peters [10, 11] present a method for computing smooth paths in polygonal channels that depends on their bound on the envelope of a B-spline function. This bound was recently extended by Reif [12].

1.3 Our contribution

Restricting the path to being a B-spline function [13], we apply the result of Lutterkort and Peters. Their envelope is suitable for our purpose, but their technique demands that the mine map is decomposed into channels. A channel is a pair of polygonal chains that are strictly monotone with respect to the same parametrization used for their corresponding B-spline function. We assume that the pairs of polygonal chains are given and solve our path-planning problem for each pair separately. Different algorithms for decomposing a simple polygon into pairs of monotone polygonal chains are presented by, for example, Keil [14] and Liu and Ntafos [15].

In order to account for a predefined safety margin and the size of the vehicle, the commonly used Minkowski sum [16] is applied to fatten the original polygonal chains. Optionally, polygon approximation techniques can be used with the objective of lowering the number of vertices of the fattened polygonal chains. A lower number of vertices decrease the time complexity of our path-planning. The generation of a safety margin is treated in Section 2.

In Section 3, we formulate our path-planning problem as an optimization problem that can be solved using standard nonlinear programming solvers. The cost function, based on experience at LKAB, has been designed to give smooth paths as well as concern about vehicle safety by a centering term. An example of our path-planning technique is shown in Section 4, followed by concluding remarks in Section 5.

2 Generating a safety margin

We use the Minkowski sum [16] on a polygonal chain to account for the size of a vehicle and to impose a safety margin. The result, in turn, needs to be a polygonal chain for use in our optimization problem. In order to decrease the time complexity of the path-planning, the number of vertices in the new polygonal chain should be minimized.

Our problem can be formulated as; Given a polygonal chain C construct another polygonal chain C', containing no point closer to C than τ and no point farther from C than $\tau + \epsilon$, having as few vertices as possible. A sketch of the problem is shown in Figure 1. We solve a restricted version of this problem where the resulting polygonal chain has its vertices at distance $\tau + \epsilon/2$ from C. Our resulting polygonal chain is the solution to a polygon approximation problem on β, that is a polygonal chain built from sufficiently frequent sample points at distance $\tau + \epsilon/2$ from C.

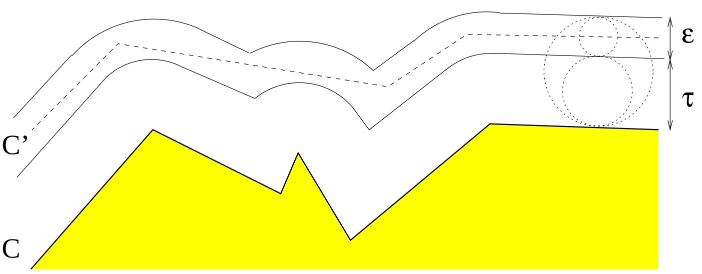


Figure 1: *The polygonal chain C' approximates C. The minimal distance between a point on C' and a point on C lies between τ and $\tau + \epsilon$.*

Let $d(p, p')$ be the Euclidean distance between the points p and p'. Furthermore, let the distance between the two polygonal chains B and B' be defined by $D(B, B') = \max_{p \in B} \min_{p' \in B'} d(p, p')$. The polygon approximation problem is; Given a polygonal chain B with n vertices and an error bound δ, find a polygonal chain B', consisting of a minimal length subsequence of the vertices of B, such that $D(B, B') \leq \delta$. A solution to this problem is presented by Iri and Imai [17]. They build a graph G by extending B with edges for all valid shortcuts from one vertex $v_i \in B$ to another vertex $v_j \in B$. A shortcut is said to be valid if all vertices $v_k \in B$, $k = i, \ldots, j$, are at distance less than δ from the straight line connecting v_i and v_j. Finding B' with minimum number of vertices is equivalent to finding the minimum number of edges in G that connect v_1 and v_n. Since G is a directed and acyclic graph, this can be done in a straightforward manner using, for example, dynamic programming techniques.

We apply the polygon approximation solution proposed by Iri and Imai [17] to our restricted problem by letting their original polygonal chain B equal our sampled polygonal chain β and their error bound δ equal $\epsilon/2$, see Figure 1. Note that there is no guarantee for our resulting polygonal chain to have less vertices than the original chain. A special case where a high number of vertices is needed is seen in Figure 2.

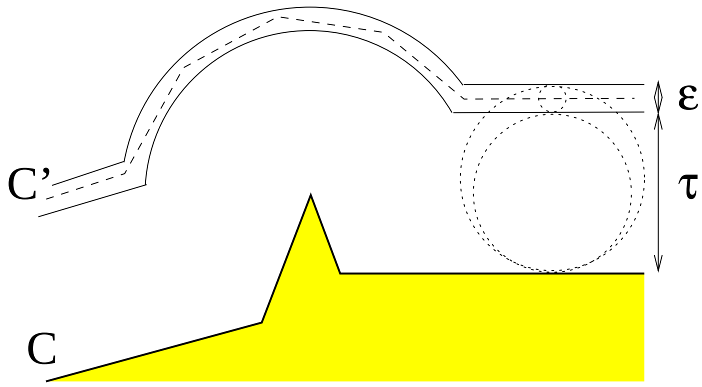


Figure 2: *The polygon approximation algorithm does not guarantee a reduction in the number of vertices of the resulting polygonal chain (dashed) when ϵ is small compared to τ.*

3 Generating smooth paths

Our paths are generated by solving a nonlinear optimization problem. The input is given as a pair of monotone polygonal chains, $\underline{c} < \overline{c}$, that correspond to the permitted region for a point-sized vehicle. We are also given start and endpoint configurations of the sought path, S and T respectively, defined by position and velocity. We look for a path combining smoothness and closeness to a center function between $\underline{c}$ and $\overline{c}$.

Our resulting path is given as a B-spline function $b(z) = \sum_{j=0}^{m} b_j N_j^d(z)$ [13]. The B-spline coefficients are found by solving our optimization problem in which $x = [b_0, \ldots, b_m]^T$ is the vector of unknowns. The B-spline basis functions N_j^d of degree d are defined by a recursion formula and a knot sequence of z-values, namely $t_0 \leq t_1 \leq \cdots \leq t_{m+d+1}$. We consider the knot sequence fixed. The Greville abscissae $t_i^* = \sum_{k=i+1}^{i+d} t_k/d$, $i = 0, \ldots, m$, are the abscissae of the vertices of the control polygon l for which $l(t_i^*) = b_i$. We denote the (signed) curvature of b by $K_s = b''/(1+(b')^2)^{\frac{3}{2}}$ and its derivative with respect to z by K_s'.

Using B-splines and the Lutterkort and Peters' envelope [11, 10], $\underline{e} \leq b \leq \overline{e}$, it is possible to ensure the continuous inequality $\underline{c} \leq b \leq \overline{c}$. The B-spline function b lies between the pair of polygonal chains $\underline{c} \leq \overline{c}$ if both $\underline{c} \leq \underline{e}$ and $\overline{e} \leq \overline{c}$ hold for the finitely many vertices of the envelope and the finitely many vertices of the polygonal chains. This is illustrated in Figure 3. We work with B-splines of degree $d = 4$ as this is the highest degree, corresponding to the most smooth B-spline function, for which the envelope is shown according to Lutterkort and Peters.

We formulate an optimization cost function combining both smoothness and centering of the path. Smoothness of a path is measured by its derivative of curvature, K_s'. The difference between the control polygon l and a center function $c = (\underline{c} + \overline{c})/2$ at discrete

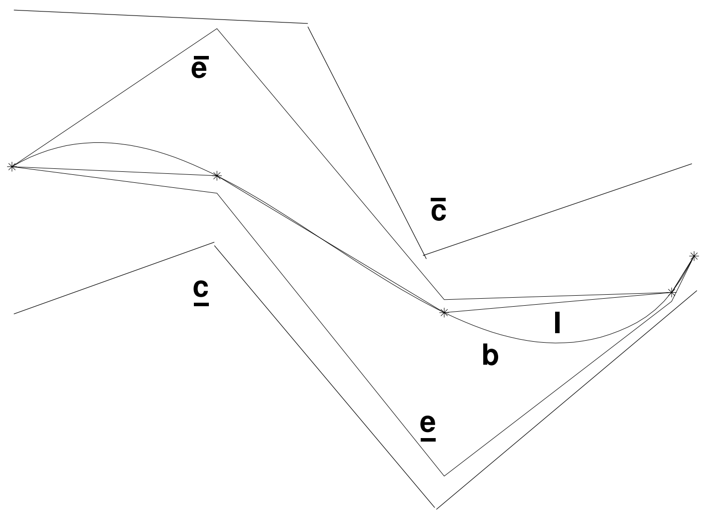


Figure 3: *Sketch of a corridor $\underline{c} \leq \bar{c}$ and a B-spline function b with control polygon l and envelope $\underline{e} \leq \bar{e}$. The B-spline function is contained in the corridor, i.e. $\underline{c} \leq \underline{e} \leq b \leq \bar{e} \leq \bar{c}$.*

values of the parameter z is a measure of the centering of the path. These two measures are combined by the given weight λ, $0 \leq \lambda \leq 1$, to form our nonlinear cost function

$$F(x) = \lambda \int_{z_S}^{z_T} (K_s')^2 \, dz + (1-\lambda) \sum_{i=0}^{m} (c(t_i^*) - b_i)^2 , \tag{1}$$

where z_S and z_T are the values of the parameter z at configurations S and T.

In order to compute (1) we have made the discretization

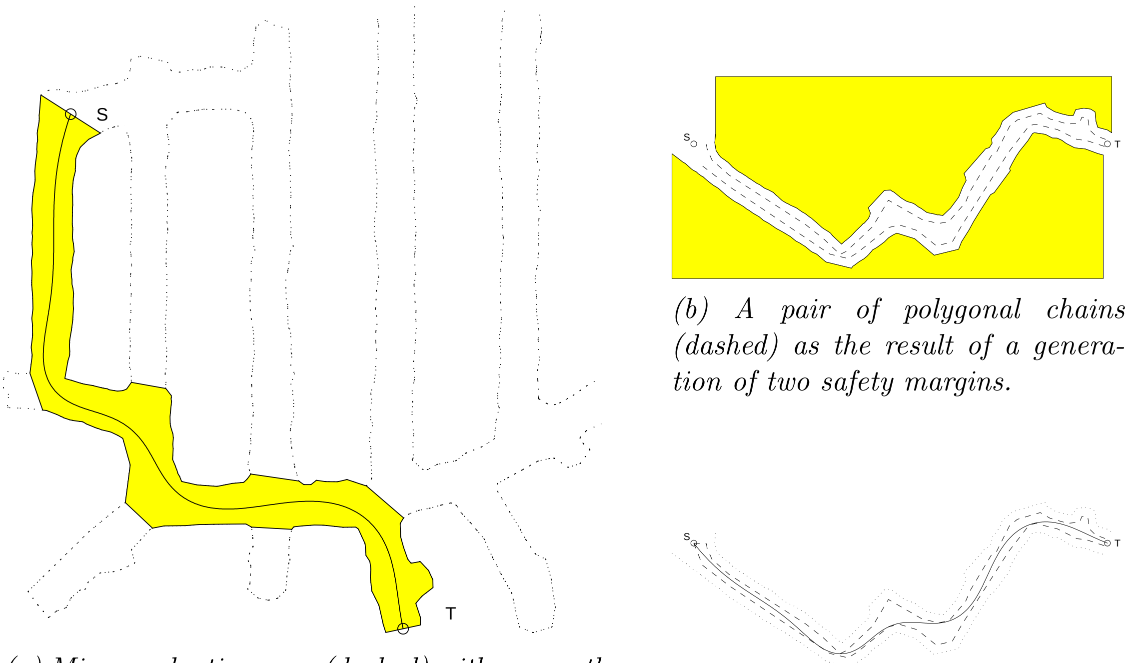
$$F_d(x) = \lambda \sum_{j=1}^{n} (K_s'(z_j))^2 (z_j - z_{j-1}) + (1-\lambda) \sum_{i=0}^{m} (c(t_i^*) - b_i)^2 , \tag{2}$$

where the evaluation points $z_S = z_0 < z_1 < \cdots < z_n = z_T$ are chosen to give a good approximation of the integral.

Our optimization problem is $\min_x F_d(x)$ s.t. $x \in \Omega$. We use a constraint set Ω in which the strict nonlinear definitions of Δ_i^+ and Δ_i^-, posed by Lutterkort and Peters, hold, see Ref. [10] for details. This yields an Ω which is a nonlinear counterpart of the linear constraint set proposed by Lutterkort and Peters. Furthermore, Ω contains constraints imposed by the start and target configurations, S and T. We have experienced trouble solving the problem resulting from the combination of Lutterkort and Peters' linear constraint set and (2), but our nonlinear problem can be solved with standard solvers. We use the center function c to produce an initial value $x = x_0$ for the solver by $b_j = c(t_j^*)$, $j = 0, \ldots, m$.

4 Path generation example

Selected parts of our proposed method to automatically design paths for vehicles have been implemented and tested on application data from mining industry. The technique is visualized through an example of a path generation.



(a) Mine production area (dashed) with a smooth path between start and target configurations, S and T.

(b) A pair of polygonal chains (dashed) as the result of a generation of two safety margins.

(c) Smooth path generated between the pair of polygonal chains.

Figure 4: *An example of a path generation.*

In Figure 4(a) a part of a mine production area (dotted), a start configuration S, and a target configuration T, are shown. The shaded area in the figure is bounded by a pair of monotone polygonal chains connecting S and T.

A safety margin is added to each of the two monotone chains. The result is simplified and shown with dashed lines in Figure 4(b).

An optimal smooth B-spline path (solid) of degree $d = 4$ having its envelope inside the simplified polygonal chains is computed and shown in Figures 4(c) and 4(a). We have used a uniform knot sequence and a design parameter λ yielding a path both smooth and to some extent centered between the walls of the mine.

5 Concluding remarks

Our main result is a proposal and an implementation of an automatic method for the generation of smooth paths. The method depends on the minimization of a specific cost function. Our cost function is a combined measure of both smoothness and safety. We define smoothness as the derivative of curvature of the path and safety as the degree of centering of the path between two polygonal chains. The resulting path is a B-spline function minimizing the cost function while still having its envelope inside a permitted region. The permitted region is generated from some original polygonal chains and accounts for the size of a vehicle and a safety margin.

The method has been tested on maps produced from sample data. Our tests show that smooth paths can be obtained in a straightforward manner by applying a standard nonlinear optimization solver on our cost function together with our constraints.

Future work include a comparison between the value of our cost function for paths produced with our method and the corresponding value for handmade paths. It would also be interesting to compare the optimality of our paths with paths that need not be described by B-spline functions only. For better understanding of the properties of the produced optimal path, further investigation of the number and placement of B-spline knot points as well as the definition of the center function is needed. Today, our proposed safety margin assumes a circular shape of the vehicle. This is a crude model as an LKAB vehicle has an extension strongly depending on the curvature of its path. Incorporating the sweep area of the vehicle in the optimization would give a more accurate modeling of the safety margin. The safety margin generation, with its proposed polygon approximation algorithm, could be improved by canceling the restriction of having vertices of the resulting polygonal chain at a specific distance from the original polygonal chain.

We conclude that, even though there is considerable work still to be done, our proposed technique is useful for automatic generation of smooth paths.

Acknowledgements

This work was supported in part by LKAB and in part by the Swedish National Board for Industrial and Technical Development (NUTEK) under contract No. P10552-1.

References

[1] J.T. Schwartz and M. Sharir. Algorithmic motion planning in robotics. In J. van Leeuwen, editor, *Algorithms and Complexity*, pages 391–430. Elsevier, 1990.

[2] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[3] 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.

[4] J.A. Reeds and L.A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2), 1990.

[5] J.-D. Boissonnat, A. Cérézo, and J. Leblond. Shortest paths of bounded curvature in the plane. In *Proc. 9th IEEE Internat. Conf. Robot. Autom.*, 1992.

[6] S. Fortune and G. Wilfong. Planning constrained motion. *Annals of Math. and AI*, 3:21–82, 1991.

[7] P. Jacobs and J. Canny. Planning smooth paths for mobile robots. In Z. Li and J. F. Canny, editors, *Nonholonomic Motion Planning*, pages 271–342. Kluwer Academic Pubishers, 1992.

[8] J.-D. Boissonnat, A. Cérézo, and J. Leblond. A note on shortest paths in the plane subject to a constraint on the derivative of the curvature. Rapport de recherche 2160, INRIA, 1994.

[9] V. Kostov and E. Degtiariova-Kostova. Suboptimal paths in the problem of a planar motion with bounded derivative of curvature. Rapport de recherche 2051, INRIA, 1993.

[10] D. Lutterkort and J. Peters. Smooth paths in a polygonal channel. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 316–321, 1999.

[11] D. Lutterkort and J. Peters. The distance between a uniform (b-)spline and its control polygon. Technical Report TR-98-013, University of Florida, 1998.

[12] U. Reif. Best bounds on the approximation of polynomials and splines by their control structure. *Computer Aided Geometric Design*, 17(6):579–589, 2000.

[13] C. de Boor. *A practical guide to splines*. Springer-Verlag, 1978.

[14] J.M. Keil. Decomposing a polygon into simpler components. *SIAM Journal on Computing*, 14(4):799–817, 1985.

[15] R. Liu and S. Ntafos. On decomposing polygons into uniformly monotone parts. *Information Processing Letters*, 27(2):85–89, 1988.

[16] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 1997.

[17] H. Imai and M. Iri. Polygonal approximations of a curve—formulations and algorithms. In G. T. Toussaint, editor, *Computational Morphology*, pages 71–86. North-Holland, 1988.