

Planning Smooth Paths among Obstacles using Minimum Curvature Variation B-splines

Tomas Berglund, Andrej Brodnik, *Member, IEEE*, Håkan Jonsson, Kent Mrozek, Mats Staffanson, Inge Söderkvist

Abstract—In this paper we study the problem of computing smooth planar paths in the presence of obstacles where we have an a priori knowledge about the environment. We investigate how the smoothness of a path and the total travel time required by the path are related for paths used by a four-wheel four-gear articulated vehicle. A path is considered smooth if the variation of its curvature, i.e., the integral of the square of the derivative of curvature along the path, is minimal. Paths are defined by quartic B-splines and obstacles are represented by polygonal chains. Quartic B-splines have a continuous derivative of curvature. Obstacle-avoidance is achieved by means of the envelope of the B-splines.

We present a study of eight cases based on real-world application data from the Swedish mining company Luossavaara-Kiirunavaara AB (LKAB). The results indicate that a minimum curvature variation B-spline path-planning algorithm we have developed yields paths that are substantially better than the ones used by LKAB today. Our simulations shows that the new paths are up to 39% faster to travel along than the paths currently in use. They even decrease the wear on the vehicle. Preliminary results from the production at LKAB show an overall 5-10% decrease in the total time. The total time includes both travel on the path and ore loading and unloading.

Note to Practitioners – This article was motivated by the problem of how to automatically produce drive-paths of high quality for autonomous transportation vehicles in mines. The vehicles are heavy (> 100 tonnes) but are still expected to run at speeds up to 20 km/h to be productive. To reach these speeds without destroying the steering gear and the mechanics of the vehicles, their paths must be smooth. It turns out that visual inspection is often not sufficient to distinguish a smooth path from a harmful one. We suggest a method for computing paths, requiring an a priori knowledge about the environment, that minimizes the amount of steering needed during a transport. The computed paths are also safe in that they guarantee that the vehicle will not collide with a tunnel wall. We present a study of eight cases based on real-world application data from the Swedish mining company Luossavaara-Kiirunavaara AB (LKAB). Our simulations show that our paths are up to 39% faster to travel along than the paths currently in use while they cause no more wear on the vehicles. Tests performed in the production at LKAB show an overall 5-10% decrease in the total time if not only travel but also loading and unloading is included.

Index Terms—optimization methods, mining industry, spline functions, motion-planning, vehicles, path smoothness, derivative of curvature, autonomous guided vehicles

Manuscript received THIS IS STILL MISSING; revised THIS IS STILL MISSING. This work was supported in part by the Swedish mining company LKAB and the Research Council of Norrbotten (Norrbottens Forskningsråd) under contract NoFo 04-005.

T. Berglund, A. Brodnik, H. Jonsson and M. Staffanson are with Luleå University of Technology, Department of Computer Science and Electrical Engineering, Sweden. I. Söderkvist is with Luleå University of Technology, Department of Mathematics, Sweden. A. Brodnik is also with IMFM, Ljubljana, Slovenia and University of Primorska, Faculty of Education, Slovenia. K. Mrozek is with QNavigator AB, Luleå, Sweden.

I. INTRODUCTION

PATH-PLANNING is a problem area that is very important as the level of automation increases [1–7]. Today, there are many industrial applications that require the pre-computation of paths for autonomous vehicles and robot arms to follow and move along [8–13]. In this paper, we address the problem of computing an obstacle-avoiding planar path that combines the properties of being both smooth and fast to traverse.

The work space of autonomous vehicles and robot arms are often cluttered with *obstacles* that these machines must avoid. An example is the path-planning of the autonomous vehicles used for transporting iron ore in the underground mines of the Swedish mining company Luossavaara-Kiirunavaara Aktiebolag (LKAB) [14]. It is crucial that these autonomous vehicles avoid the obstacles constituted by the mine walls in order for the production to go on. Another example is a robot arm holding a cutting tool that follows a pre-computed path. It should perform its task while staying away from other machines and it should not damage itself.

Much work has been devoted to planning paths in the presence of obstacles and there are many results yielding piecewise linear paths [3, 15]. However, practical applications involving physical machines also require that the paths are *smooth* [5, 7, 16–18]. An example of the importance of smoothness comes from the offset path of a cutting tool [19]. Its path must be curvature-continuous to avoid jumps in the acceleration that damage the driver motors. Another example comes from considering the autonomous ore transport vehicles used at LKAB. These are articulated, which means that the whole vehicle is involved in the steering. Its steering gear is worn out even by small jerks in its path and large jerks increases the risk of slippage. Moreover, such jerks easily cause the machine to drop ore on the road which eventually must be cleared before the transports may continue.

There are many definitions of smoothness. Depending on context, a path or curve in 2D as well as in 3D is referred to as being smooth if it is tangent continuous, has a continuous curvature, or even has a continuous derivative of curvature [5, 7, 16–18]. Smooth curves are of importance not only to achieve paths that do not cause unnecessary wear, but also have found application in designing curves that look aesthetically pleasing. Such applications are found in the car, airplane, virtual reality, video game, and film industry [20–22].

Here, an important aspect of the smoothness of a path is its relation to the speed at which the path can be traversed. The turning speed of the transport vehicle is limited by the speed of

the vehicle. The speed of the vehicle is therefore adjusted such that the vehicle is capable of following the path. If the vehicle has different gears with certain limitations on curvature and derivative of curvature, it is seen that the traversal time is a function of the gears along the path. Then, in turn, the gears are functions of the curvature and the derivative of curvature along the path. Therefore, the higher the smoothness, the higher the gear, and the faster the traversal of the path.

Still, obstacle-avoidance and smoothness are not enough for practical purposes. The path should also stay at a sufficient distance – the *safety margin* – from the obstacles. The smoother the path, the larger the radius of the curve. This means that a path will tend to go close to obstacles in order to have as high smoothness as possible. Once again, consider the ore transportation vehicle. A smooth path makes it go close to the walls of the mine when turning. As touching the walls is the main danger for the vehicle, it should stay at a safe distance from the walls. The computation of safety margins has been considered in the literature [23].

A. Contribution

In this paper we focus on validating the assumption that geometrically smooth paths, computed with an a priori knowledge about the environment, are fast to traverse. The Swedish mining company LKAB has already planned piecewise seventh-degree polynomial paths inside their a priori known mine. Based on mine map application data from LKAB, in the form of polygonal chains, we compute safety margins that, in turn, are two flanking polygonal chains. Inside of these margins, using a standard nonlinear programming tool, we compute geometrically smooth paths built from obstacle-avoiding quartic minimum curvature variation B-spline curves. These quartic B-splines have a continuous derivative of curvature. Using the same model of a four-gear four-wheel articulated vehicle with gear-limits as the one used by LKAB, we validate an increase in speed along the B-spline paths compared to the speed along the LKAB paths. This is done for eight scenarios. Output from our computations are piece-wise polynomial paths with a dedicated gear for each piece, traversal time, curvature, and derivative of curvature.

B. Paper outline

In the next section, we present some preliminaries. It is then followed by Section III which introduces the vehicle model used as well as a description of our method to produce a safety margin. Section IV shows a comparison between our paths and the LKAB paths, while Section V concludes this paper.

II. PRELIMINARIES

THIS section contains a description of the LKAB paths and their representation, our smoothness measure, and a motivation to the use of B-splines as a path representation. It also contains a description of the minimum curvature variation B-spline (MVB) and how it is computed followed by a short pseudocode description of our algorithm.

Our interest lies in the properties of paths which goes from an initial to a terminal posture. A path cycle is simply a combined path where we go back to the initial posture again along the same path. As path cycle is the commonly used term in industry, we also use the term although we mean a path or a half path cycle. A path is built from subpaths which are built from consecutive path segments. All paths must meet up with initial and terminal endpoint constraints. These are given as position, slope angle, curvature, and derivative of curvature. Furthermore, the paths must not intersect obstacles represented by flanking polygonal chains. These obstacles are given as input for the path-planning.

LKAB describes path segments by seventh-degree polynomials. These paths are not computed to minimize curvature variation over the whole path. Instead, for each polynomial segment, the maximal derivative of curvature is minimized. Each segment is defined in a separate coordinate system, which gives a path built from consecutive parameterizations along different room axes or cartesian coordinate systems in the plane. This representation is flexible, but the constructed curves have the drawback that there is nothing that guarantees smoothness nor a small traversal time for the whole path. We compute paths that are quartic MVB curves, which are described in Section II-C.

A. Smoothness measure

We consider the same smoothness measure as Kanayama and Hartman [24], namely the integral over the square of arc-length derivative of curvature along the path. A motivation of using this measure comes from the physics of motion. At constant speed, the centripetal (lateral) acceleration

$$\frac{mv^2}{r} = mv^2 K$$

of a vehicle is proportional to its curvature K , where m is the mass of the vehicle, r is the radius curvature, and v is its velocity. The time derivative of this acceleration is given by

$$\frac{d}{dt}(mv^2 K) = mv^2 \frac{dK}{dt} = mv^2 \frac{dK}{ds} \frac{ds}{dt} = mv^3 \frac{dK}{ds} ,$$

where s is the curve length. Thus, the variation of curvature is proportional to the jerk or change in lateral acceleration of the vehicle.

We are dealing with curves that are functions over a room axis. For a function $f(x)$ with curvature $K(x)$, $x \in [x_0, x_1]$, there is the one-to-one relation $ds = \sqrt{1 + f'(x)^2} dx$, between x and arc-length s . Therefore, letting $\dot{\nu} = d\nu/ds$ and $\nu' = d\nu/dx$ for any function ν , our smoothness measure can be written as

$$\int_{s(x_0)}^{s(x_1)} \dot{K}(s)^2 ds = \int_{x_0}^{x_1} \frac{K'(x)^2}{\sqrt{1 + f'(x)^2}} dx .$$

B. B-splines for path-planning

There are numerous ways to represent curves used for trajectory planning. A brief and helpful overview is given in [6]. There, seven different curve types are mentioned. Line segments together with arcs of circles, Cartesian polynomials,

clothoids together with anticlothoids, cubic spirals, generalized polar polynomials, B-splines, and parametric curves consisting of a sum of harmonics (sine and cosine functions). Each of the four first mentioned curve types have their obvious drawbacks. Curves built from line segments connected to arcs of circles are not smooth enough for our purposes as they are discontinuous even in their curvature. When it comes to Cartesian polynomials it is not easy to compute the maximum value of the curvature. Clothoids and anticlothoids are intrinsic curves so there are no closed form expressions of positions along them [5, 7]. Cubic spirals have the same disadvantage. The three remaining curve types do not have these obvious drawbacks. They can all be written in closed form and they can be constructed or defined in such a way that there are a sufficient number of continuous derivatives. The last mentioned parametric curves even have an infinite number of continuous derivatives.

We have chosen to work with curves built from B-spline functions. B-spline functions [25–27] have properties that make them suitable for smooth and obstacle-avoiding path-planning. A B-spline function or B-spline $B(x)$ is a piecewise polynomial of a certain degree. Much work has been devoted to planning or smoothing trajectories using B-splines [8–13]. However, this work either contains proposals of using B-splines of such a low degree that they lack a continuity in their derivative of curvature or it does not handle obstacle-avoidance. Here, we consider quartic, i.e., degree 4, uniform B-splines that are functions of x . Quartic B-splines are sufficiently smooth for our purposes as they are continuous even in their derivative of curvature.

By means of its envelope [19,28], a B-spline function can avoid obstacles represented by piecewise linear functions $\underline{c}(x) \leq \bar{c}(x)$. The envelope of $B(x)$ is a pair of piecewise linear functions $\underline{e}(x)$ and $\bar{e}(x)$, given by $B(x)$, such that $\underline{e}(x) \leq B(x) \leq \bar{e}(x)$. In order for the B-spline to avoid the obstacles it suffices to impose the constraints $\underline{c}(x) \leq \underline{e}(x)$ and $\bar{e}(x) \leq \bar{c}(x)$. This, in turn, leads to obstacle-avoidance of the B-spline $B(x)$. The constraints $\underline{c}(x) \leq \underline{e}(x)$ and $\bar{e}(x) \leq \bar{c}(x)$ can be formulated as a finite number of linear constraints as they amount to comparing the piecewise linear functions, representing the obstacles, at their respective vertices. Then, instead of having to compare the function itself, at an infinite number of points, against the obstacles, we have finitely many constraints depending only on the number of vertices of the envelope and the vertices of the piecewise linear obstacles.

B-splines are written in closed form, and they are also efficiently computed [25]. Another thing that makes B-splines attractive is the ease by which the shape of the resulting curve can be controlled. Their approximation ability and the properties of their shape depend on the number of B-spline basis functions used to define $B(x)$ [29]. The larger number of basis functions $B(x)$ is built from, the more flexible its representation. For these reasons, B-splines are also widely used in a variety of contexts such as data fitting, computer aided design (CAD), automated manufacturing (CAM), and computer graphics [21].

C. Minimum curvature variation B-spline paths

We are concerned with obstacle-avoiding minimum curvature variation B-splines (MVB) [30,31]. They are solutions to the following problem:

Problem 1: Given two flanking obstacles in the form of polygonal chains that are piecewise linear functions $\underline{c}(x) \leq \bar{c}(x)$ defined on a real interval $I = [x_0, x_1]$, a set of endpoint constraints, and n B-spline basis functions, compute a quartic uniform B-spline $B(x)$ defined on I such that

- 1) the B-spline $B(x)$ satisfies the given endpoint constraints,
- 2) $\underline{c}(x) \leq \underline{e}(x)$ and $\bar{e}(x) \leq \bar{c}(x)$, and
- 3) the B-spline $B(x)$ minimizes the cost function

$$\int_I \frac{K'(x)^2}{\sqrt{1+B'(x)^2}} dx ,$$

where $K(x) = B''(x)/(1+B'(x)^2)^{\frac{3}{2}}$ is the curvature and $\underline{e}(x) \leq \bar{e}(x)$ is the envelope of $B(x)$ respectively.

This problem is an n -dimensional nonlinear optimization problem with linear constraints that can be solved with standard optimization tools. One such tool is the MATLAB solver `fmincon` which in this case uses a medium scale optimization algorithm based on Sequential Quadratic Programming (SQL)[32]. The solution gives the values of the n B-spline coefficients and thus the optimal B-spline curve is defined. Figure 1 shows an MVB that is a solution to a setting where

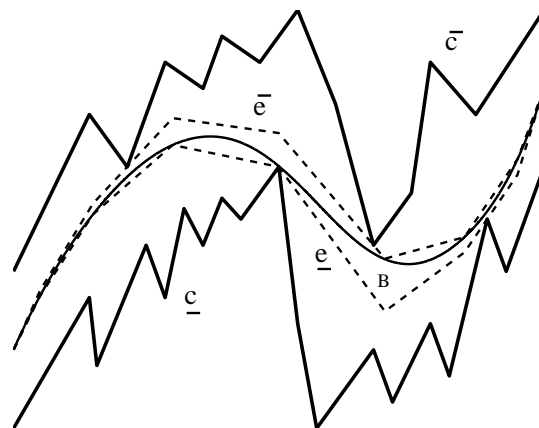


Fig. 1. A computed minimum curvature variation B-spline (MVB) B with its envelope $\underline{e} \leq \bar{e}$ between two polygonal chains $\underline{c} \leq \bar{c}$ such that $\underline{c} \leq \underline{e} \leq B \leq \bar{e} \leq \bar{c}$.

the B-spline is built from 10 basis functions.

This problem formulation only accounts for obstacles that are linear functions on a room axis. These functions are here built from connected laser point data. Such obstacles are said to be *monotone* with respect to the axis in question. A setting that can not be parameterized along a room axis in its entirety, can be dealt with by splitting it into smaller subparts, each of which is monotone to a room axis of its own, and then computing a solution in each subpart. We consider the overall, however suboptimal, solution to be formed by concatenating the smaller ones. A problem here is to choose proper split-points. At this instance, this is done manually.

D. Method overview

For clarity, we present a short pseudocode describing the main computational steps performed in this paper. These steps are taken for each path cycle comparison. Briefly, we are given a corridor-like environment, LKAB paths with prescribed path segment gear specifications, a cost function, a vehicle model, and start and endpoint constraints. Then we

- 1) compute safety margins which gives a new and narrower corridor environment.
- 2) plan an optimal MVB path which satisfies the start and endpoint constraints. This is done by computing the quartic B-spline function which minimizes the cost function while at the same time has its envelope contained inside the corridor. The computation technique is nonlinear programming with linear constraints. The result is the B-spline coefficients defining the path.
- 3) compute the highest possible gear for the vehicle on each MVB path segment. This includes the computation of possible accelerations and needed decelerations.
- 4) compare the MVB path cycle with the corresponding LKAB path cycle in different manners such as traversal time, value of the cost function, maximum curvature, maximum derivative of curvature, and path length.

III. VEHICLE MODEL AND SAFETY MARGIN

WE use a vehicle model based on the Load-Haulage-Dump vehicle Tamrock Toro 2500 [33]. It is an electrical wheel loader with articulated steering. A picture of the vehicle is seen in Figure 2. Even though it is a rough approximation, we will consider the sweep area of the vehicle being of circular shape when the safety margin is computed.

A. Gears, speed, acceleration, and time

The vehicle has four distinct gears depending on its electrical engine. The power output to the wheels of the vehicle is about 165kW on average. The vehicle weighs differently depending on the ore load in its bucket. We consider an average weight of 85 tonnes.

A mean driving speed of each gear and the mean acceleration up to each gear has been measured at LKAB according to Table I. The deceleration is 0.9 meters per second for all gears.

| Gear | Measured speed (m/s) | Acceleration (m/s ²) |
|------|----------------------|----------------------------------|
| 1 | 1.0 | 3.8 |
| 2 | 1.9 | 1.3 |
| 3 | 3.1 | 0.8 |
| 4 | 5.0 | 0.5 |

TABLE I
GEAR SPEEDS AND ACCELERATIONS.

The time it takes for a vehicle to traverse a path cycle is a sum of the time it takes for the vehicle to traverse each path segment. A path segment is the B-spline function between two knots.

Each segment has a prescribed gear chosen such that it is the highest allowed gear on the segment according to the limits given in Section III-B. The speed over a segment can contain an acceleration part, a constant speed stretch and a deceleration. In tests, the constant speed is reached on every segment. For simplicity, we assume that this is also true in theory. If a higher gear is allowed on the next segment, an acceleration is performed and vice versa. It is important that accelerations are not made before entering a segment with higher gear and that decelerations are made before entering a segment with lower prescribed gear.

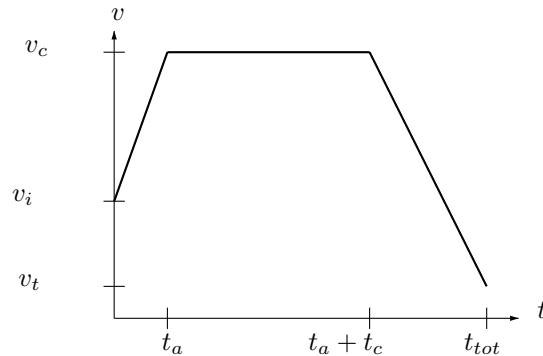


Fig. 3. Time calculation of a segment.

Let a be the acceleration, d the deceleration, and v_c the constant speed. Furthermore, let v_i , v_t , and l be the initial speed, terminal speed, and the length of the segment, respectively. As shown in Figure 3, the time $t_a = (v_c - v_i)/a$ of acceleration and the time $t_d = (v_c - v_t)/d$ of deceleration can be used to compute the arc-lengths s_a , s_d , and s_c of acceleration, deceleration, and constant speed, respectively, through

$$s_a = \frac{at_a^2}{2} + v_it_a, \quad s_d = \frac{dt_d^2}{2} + v_it_d, \quad s_c = l - s_a - s_d.$$

After some computation, the total time, t_{tot} , is shown to be

$$t_{tot} = t_a + \frac{s_c}{v_c} + t_d = \frac{(v_c - v_i)^2}{2av_c} + \frac{l}{v_c} + \frac{(v_c - v_t)^2}{2dv_c}.$$

B. Limits in maneuverability

The vehicle has limits in its maneuverability. When designing paths, the steering angle α of the vehicle is limited to $|\alpha| \leq \alpha_{\max} = 38^\circ$ and the time derivative $d\alpha/dt$ of the steering angle is limited to $|d\alpha/dt| \leq \dot{\alpha}_{\max} = 10^\circ$ per second.

A model of the vehicle is built from its technical specification (Figure 4). The distance $L = 2.55$ meters to the joint is the same for both the front and the rear wheels. The radius of curvature is denoted R . The curvature K can be calculated as

$$K = \frac{1}{R} = \frac{\tan(\alpha/2)}{L}. \quad (1)$$

Consequently each path segment must fulfill

$$|\tan^{-1}(LK)| \leq \frac{\alpha_{\max}}{2}.$$

Let v be the velocity of the vehicle. From (1) the derivative of curvature is given by

$$\frac{dK}{ds} = \frac{(1 + \tan^2(\alpha/2))}{2L} \frac{d\alpha}{ds} = \frac{(1 + L^2K^2)}{2L} \frac{d\alpha}{dt} \frac{dt}{ds},$$



Fig. 2. Side view of the Tamrock Toro 2500 electrical wheel loader; length: 14 meters, operating weight: 76 tonnes, and capacity: 25 tonnes.

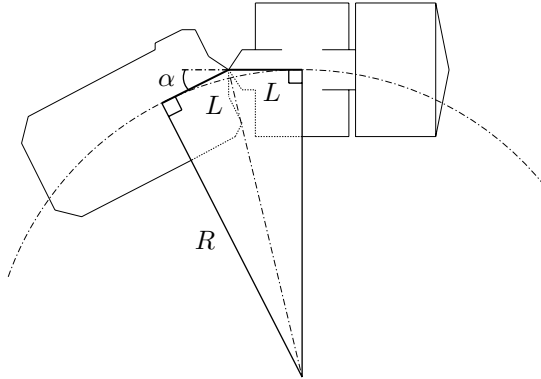


Fig. 4. Model of the wheel loader. The right part represents the front of the vehicle.

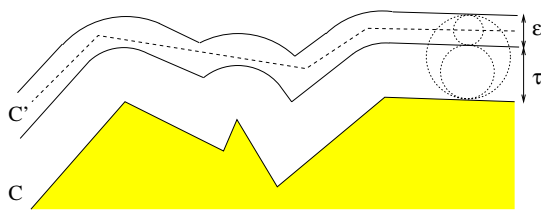


Fig. 5. The polygonal chain C' approximates C . The minimal distance between a point on C' and a point on C lies between τ and $\tau + \epsilon$.

which gives

$$\frac{dK}{ds} = \frac{(1 + L^2 K^2) d\alpha}{2Lv dt}. \quad (2)$$

Then it is possible to isolate $d\alpha/dt$ and establish the bound

$$\left| \frac{v}{(1 + L^2 K^2)} \frac{dK}{ds} \right| \leq \frac{\dot{\alpha}_{\max}}{2L}.$$

For each path segment, the gear with the highest prescribed velocity v is chosen such that this bound still holds.

C. Safety margin

In order to account for the sweep area of the vehicle and at the same time impose a safety distance between the vehicle and the mine walls, we compute a safety margin. Therefore, when we refer to a safety margin, we mean a margin that takes into account both the sweep area of the vehicle and a safety distance. This subsection contains a description of how the margin is computed.

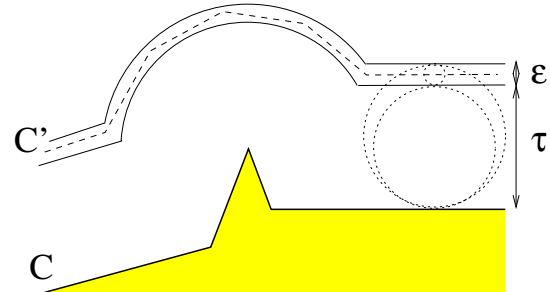


Fig. 6. The polygon approximation algorithm does not guarantee a reduction in the number of vertices of the resulting polygonal chain (dashed) when ϵ is small compared to τ .

1) *Safety margin problem formulation:* Since we consider the vehicle being of circular shape, we use the Minkowski sum [23] 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 safety margin 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:

Problem 2: 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 5. 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 .

2) *Polygon approximation:* Let $d(p, p')$ be the Euclidean distance between the points p and p' . Furthermore, let the measure $D(\cdot, \cdot)$ between the two polygonal chains V and V' be defined by $D(V, V') = \max_{p \in V} \min_{p' \in V'} d(p, p')$. The polygon approximation problem is:

Problem 3: Given a polygonal chain V with n vertices and an error bound δ , find a polygonal chain V' , consisting of a minimal length subsequence of the vertices of V , such that $D(V, V') \leq \delta$.

A solution to this problem is presented by Iri and Imai [34]. They build a graph G by extending V with edges for all valid shortcuts from one vertex $v_i \in V$ to another vertex $v_j \in V$. A shortcut is said to be valid if all vertices $v_k \in V$, $k = i, \dots, j$,

are at distance less than δ from the straight line connecting v_i and v_j . Finding V' 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 [34] to our restricted problem by letting their original polygonal chain V equal our sampled polygonal chain β and their error bound δ equal $\epsilon/2$, see Figure 5. Note that, although we minimize the number of resulting vertices, 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 6.

IV. PATH CYCLE COMPARISONS

Our algorithm permits us to take two flanking polygonal chains and produce a safety margin. From this margin, we are able to compute, by means of its envelope, an optimal or suboptimal obstacle-avoiding MVB path from a start to a terminal endpoint. This is also done in previous work [30]. Now, we want to compare our paths with corresponding paths produced in industry. The industrial paths are built from relatively small concatenated path segments produced entirely by hands. We use the same vehicle and time model, see Section III, as LKAB and compare the traversal time of MVB paths with the traversal time of the LKAB paths. The comparison is made without taking into account any control algorithm.

The LKAB paths are built from consecutive path segments. Each segment is a seventh degree polynomial over a separate room axis. This means that the LKAB path is flexible and need not be entirely defined over a certain room axis. An MVB path needs to be defined over one room axis only. The path becomes a fourth degree piecewise polynomial over the particular axis. Each piece constitutes a path segment according to Section III-A. The idea is to have the same parametrization from initial to terminal endpoint. If this is not possible, the path is split into subpaths where the optimization is performed on each subpath. This yields a suboptimal path. The LKAB path representation is more flexible than the MVB path representation. It allows the path to turn in any direction and each segment has more degrees of freedom than that of an MVB path segment. The good thing with the MVB path is that it is defined in such a manner that it yields a smooth path from initial to terminal endpoint.

We consider mining area 820:39 in the LKAB underground mine in Kiruna, Sweden. This mining area is shown in Figure 7. The figure also points out eight different path cycles, namely a1, a2, a3, a5, a6, c1, c2, and c3. A safety margin is computed such that it accounts for both the sweep area of the vehicle and a safety distance. The margin is such that it is not closer to the walls than the minimum distance from the walls to the LKAB paths. In our setting, the safety margin lies in a region between 2.25m and 2.35m from the polygonal chains describing the walls of the mine.

The MVB paths are constrained to have the same position, slope angle, curvature, and derivative of curvature at its

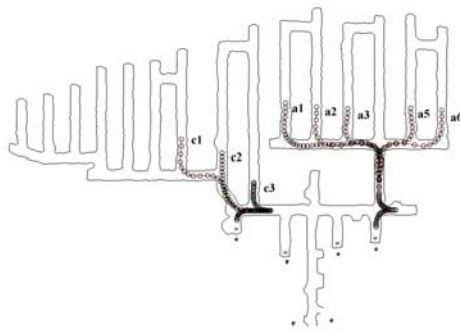


Fig. 7. Mining area 820:39 and cycles a1, a2, a3, a5, a6, c1, c2, and c3.

endpoints as the LKAB paths. The derivative of curvature and the curvature are computed numerically at 300 positions along each segment. Based on this, the highest gears possible are chosen on each segment according to methods described in Section III-B. The same method of choosing gears is used both for the LKAB and the MVB paths. We accelerate and decelerate in the same manner as LKAB and we use an average weight – 85 tonnes – in all our computations. Acceleration and deceleration are modelled according to Section III-A.

In order to produce an MVB path cycle, the cycle is divided into connected subpaths that are produced from 25 B-spline basis functions. At this instance, we have split the cycles into paths by hands according to a scheme where a split-point is introduced where the corridor has a slope change of more than about 40 degrees and, if possible, in the middle of a straight stretch of the road. This technique is thus not automated yet, but it requires only a few steps compared to producing the LKAB cycles, where each segment is defined by hands. For example, in Figure 8, 4 split-points are chosen. One at each endpoint and one on the middle of the horizontal straight stretch.

For each path, knots are placed uniformly. For the optimization, the MATLAB solver `fmincon` is used with termination criteria `tolfun = tolcon = tolx = 10-8`. The computation of function and gradient values depend on the integration routine `coteglob` with accuracy 10^{-8} [35]. The appropriate gears of the vehicle are chosen for each segment after the optimization.

We choose to present only one path cycle comparison in detail, namely the one for cycle a1; see Figure 8a. Figure 8b shows how velocities (gears) are chosen along the LKAB and the MVB path. Computations starts from the initial position I in the upper left corner and end at the terminal position T at the bottom of Figure 8a (the detour close to T is required because the LHD-vehicle connects to an power outlet there).

From Figure 8b it is seen that the MVB path cycle is slightly longer than the LKAB path (184.7 compared to 183.2 as seen in Table II) but that it sustains a much higher speed during most of the travel. Here, negative numbers means that the vehicle is reversing and positive numbers that it is driving forward.

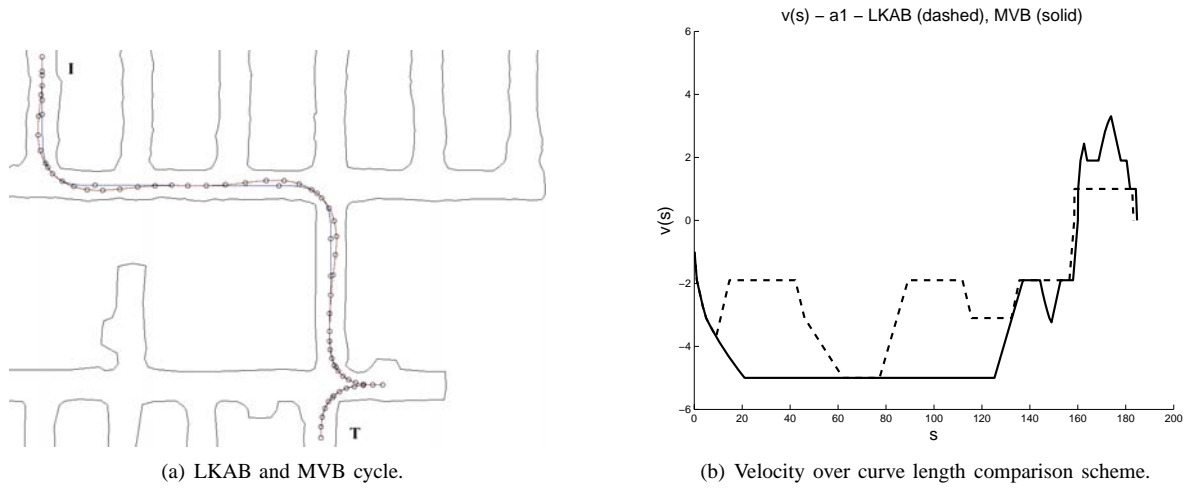


Fig. 8. Cycle a1: LKAB (dotted) and MVB (solid).

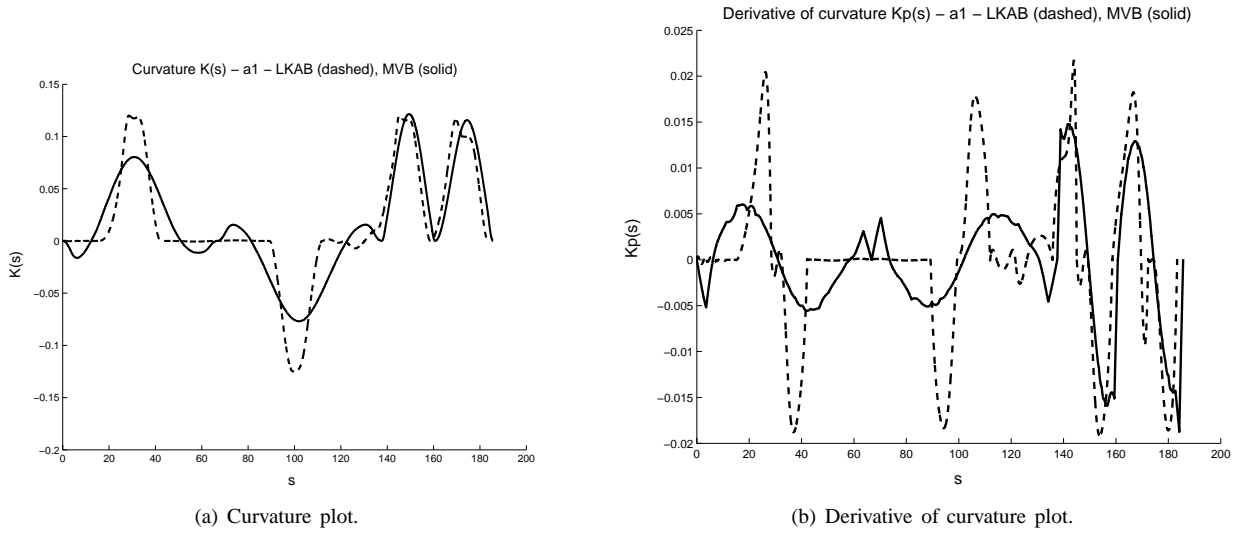


Fig. 9. Cycle a1: LKAB (dotted) and MVB (solid).

Figure 9 shows a comparison between the curvature and a comparison between the derivative of curvature for the LKAB and the MVB paths over path cycle a1. It is seen that neither the magnitude of curvature nor the magnitude of derivative of curvature of the MVB path is larger than for the LKAB path.

We get similar results for the other seven path cycles. We present corresponding results in Table II. The columns of the table show the following properties:

- t the time in seconds it takes for the vehicle to traverse a path from an initial to a terminal endpoint given in seconds.
- S_{cost} the smoothness cost $\int \dot{K}(s)^2 ds$ over the path with metric rad^2/m^3 , where rad stands for radians and m stands for meters. This is the measure of our optimization.
- K_{max} : the maximal magnitude of the curvature along the path expressed in radians per meter. Together with \dot{K}_{max} , K_{max} limits the speed of the vehicle.
- \dot{K}_{max} the maximal magnitude of the derivative of curvature

along the path expressed in radians per square meter. s the length of the path in meters.

The rows of the Table II are divided into groups depending on what cycle they belong to. Each cycle has a row for LKAB, MVB, and a difference in percents going from the LKAB value to the MVB value. The very last row contains a mean of all the percentage differences. It is seen that an MVB path is more smooth and faster to traverse than the LKAB path at the expense of being slightly longer. The computation time for an optimization increases with the number of B-spline basis functions. An optimization with 25 basis functions takes about 30 minutes on a Pentium IV 2 GHz machine with 2 Gb RAM memory. As seen in Figure 7, the cycles presented here mostly consists of turns. Longer cycles also include straight stretches. For these stretches, the benefit of using MVB paths is not significant as they can be traversed on the highest gear. Then, the time differences remain the same, but the relative differences decrease. It is in the turns where the MVB paths come to their best.

| Cycle | | t | S_{cost} | K_{max} | \dot{K}_{max} | s |
|-------------|-----------------|--------------|--------------|---------------|-----------------|----------------|
| a1 | LKAB | 90.8 | 0.01358 | 0.125 | 0.02171 | 183.2 |
| | MVB | 56.63 | 0.007523 | 0.1214 | 0.01876 | 184.7 |
| | Diff (%) | 37.63 | 44.62 | 2.914 | 13.58 | -0.8301 |
| a2 | LKAB | 84.95 | 0.01369 | 0.125 | 0.02171 | 155.9 |
| | MVB | 52.89 | 0.007727 | 0.1214 | 0.01876 | 157.3 |
| | Diff (%) | 37.74 | 43.58 | 2.914 | 13.58 | -0.8684 |
| a3 | LKAB | 76.95 | 0.01374 | 0.125 | 0.02171 | 130 |
| | MVB | 58.11 | 0.01033 | 0.1214 | 0.01885 | 131.6 |
| | Diff (%) | 24.48 | 24.82 | 2.912 | 13.16 | -1.245 |
| a5 | LKAB | 75.7 | 0.01373 | 0.131 | 0.02171 | 129.6 |
| | MVB | 54.08 | 0.00968 | 0.1214 | 0.01876 | 130.3 |
| | Diff (%) | 28.56 | 29.49 | 7.365 | 13.58 | -0.5315 |
| a6 | LKAB | 82.52 | 0.01178 | 0.12 | 0.02171 | 151.8 |
| | MVB | 50.11 | 0.007129 | 0.1214 | 0.01876 | 152.1 |
| | Diff (%) | 39.28 | 39.48 | -1.136 | 13.58 | -0.2491 |
| c1 | LKAB | 60.41 | 0.01077 | 0.1278 | 0.02194 | 112.1 |
| | MVB | 44.06 | 0.005326 | 0.1142 | 0.01787 | 112.8 |
| | Diff (%) | 27.08 | 50.55 | 10.6 | 18.54 | -0.6826 |
| c2 | LKAB | 40.3 | 0.004429 | 0.1278 | 0.02194 | 79.7 |
| | MVB | 27.2 | 0.001884 | 0.1142 | 0.01787 | 80.75 |
| | Diff (%) | 32.51 | 57.45 | 10.6 | 18.54 | -1.32 |
| c3 | LKAB | 42.58 | 0.006361 | 0.1278 | 0.02194 | 63.56 |
| | MVB | 29.9 | 0.004048 | 0.1142 | 0.02188 | 63.42 |
| | Diff (%) | 29.78 | 36.36 | 10.6 | 0.2729 | 0.2281 |
| Mean | Diff (%) | 32.13 | 40.79 | 5.847 | 13.10 | -0.6873 |

TABLE II

MEASURE COMPARISONS FOR CYCLES A1, A2, A3, A5, A6, C1, C2, AND C3.

V. CONCLUSIONS AND FUTURE WORK

USING application data from the Swedish mining company LKAB, we have shown that quartic minimum curvature variation B-splines are suitable as paths for four-wheel Load-Haulage-Dump vehicles. In the 8 real-world cases we have studied these smooth B-spline curves yield paths that are faster to drive along than paths used by LKAB today. We conclude that, the higher the smoothness, the higher the gear and therefore, the shorter the travel time.

We are able to plan planar paths that are up to 39 percents faster to drive along than the paths currently in use at LKAB without increase in wear. Moreover, preliminary results from an internal project at LKAB show an overall 5-10% increase of the speed in which iron ore is transported out of the mine during production. In this ongoing study they include the entire transportation cycle (including not only transportation but also loading, dumping, maintenance, etc). Altogether, this shows that a minimum curvature variation B-spline constitutes a suitable path representation if traversal time is considered.

Our computation of B-spline paths relies on the definition of initial and terminal constraints, an a priori knowledge about the surrounding, as well as a division of the paths in parts that are functions along a local room axis. The choice of initial and terminal constraints and the choice of cutting points can be made more automatic than they are today. The placing of knot points defining the B-splines could also be made more automatic. Through a good choice of knot placement, the B-spline paths would have sufficient degrees of freedom where needed, while at the same time being fast to compute.

From a practical point of view, there seems to be a trade-off

between short travel times and high wear of the vehicle. Our smooth paths inflict less wear on the vehicle. However, they also makes it possible to drive at a higher speed, which in turn increases the wear. This relation would be of interest to investigate in order to further optimize the transportation cycle. Also, short path segments allow high granularity in the choice of gear since low gears need not be used for a longer time than necessary. This also contributes to the lowering of the travel time and is a great time benefit of the B-spline paths. In real life, however, frequent changes of gears yield high wear on the gearbox of the vehicle which brings down its expected life-time. Future work should take short segments and frequent gear changes into account. It is possible to use the method of computing a safety margin according to the one described in this paper. Still, the safety margin does not take the real sweep area of the vehicle into account. This is a difficult task, but would give a more realistic setting and in practice enable the computation of even smoother paths. It is clear that our paths can be applied in other contexts than mining. It would also be of interest to investigate whether minimum curvature variation B-splines could be computed in three dimensions and used for smooth obstacle-avoiding path-planning in, for instance, underwater and air space applications.

REFERENCES

- [1] J.-C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [2] J. Schwartz and M. Sharir, "Algorithmic motion planning in robotics," in *Algorithms and Complexity*, J. van Leeuwen, Ed. Elsevier, 1990, pp. 391–430.
- [3] K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, Eds., *Algorithmic Foundations of Robotics*. Wellesley, MA: A. K. Peters, 1995.
- [4] Y. Hwang and N. Ahuja, "Gross motion planning - A survey," *ACM Computing Surveys*, vol. 24, no. 3, pp. 219–291, Sept. 1992. [Online]. Available: <http://www.acm.org/pubs/toc/Abstracts/0360-0300/136037.html>
- [5] T. Fraichard and A. Scheurer, "From reeds and shepp's to continuous curvature paths," *IEEE Trans. Robot. Automat.*, vol. 20, no. 6, pp. 1025–1035, Dec. 2004.
- [6] P. Gallina and A. Gasparetto, "A technique to analytically formulate and to solve the 2-dimensional constrained trajectory planning problem for a mobile robot."
- [7] S. Fleury, P. Soueres, J.-P. Laumond, and R. Chatila, "Primitives for smoothing mobile robot trajectories," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 1, Las Vegas, USA, May 1993, pp. 832–839.
- [8] A. Piazzi and A. Visioli, "Global minimum-jerk trajectory planning of robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 47, no. 1, pp. 140–149, Feb. 2000.
- [9] K. Komoriya and K. Tanie, "Trajectory design and control of a wheel-type mobile robot using b-spline curve," in *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, Tsukuba, Japan, Sep 1989, pp. 398–405.
- [10] A. Piazzi, C. Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic g^2 -splines for the iterative steering of vision based autonomous vehicles," *IEEE Trans. Intell. Transport. Syst.*, vol. 3, no. 2, pp. 27–36, Mar. 2002.
- [11] G. Song and N. Amato, "Randomized motion planning for car-like robots with c-prm*," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, Maui, Hawaii, USA, Oct 2001, pp. 37–42.
- [12] A. Simon and J. Becker, "Vehicle guidance for an autonomous vehicle," in *IEEE/IEE/JSAT Int. Conference on Intelligent Transportation Systems*, Oct 1999, pp. 429–434.
- [13] M. Yamamoto, M. Iwamura, and A. Mohri, "Quasi-time-optimal motion planning of mobile platforms in the presence of obstacles," in *ICRA*, 1999, pp. 2958–2963.
- [14] Luossavaara-Kiirunavaara Aktiebolag (LKAB), <http://www.lkab.com>.
- [15] J. E. Goodman and J. O'Rourke, Eds., *Handbook of Discrete and Computational Geometry*. CRC Press LLC, 1997.

- [16] 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.*, vol. 79, pp. 497–516, 1957.
- [17] 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," INRIA, Rapport de recherche 2160, 1994. [Online]. Available: <http://www.inria.fr/RRRT/RR-2160.html>
- [18] P. Jacobs and J. Canny, "Planning smooth paths for mobile robots," in *Nonholonomic Motion Planning*, Z. Li and J. F. Canny, Eds. Kluwer Academic Publishers, 1992, pp. 271–342.
- [19] D. Lutterkort and J. Peters, "Smooth paths in a polygonal channel," in *Proceedings of the Conference on Computational Geometry (SCG '99)*. New York, N.Y.: ACM Press, June 13–16 1999, pp. 316–321.
- [20] D. Nieuwenhuisen and M. O. (2002), "Motion planning for camera movements in virtual environments," Utrecht University: Information and Computing Sciences, Utrecht, the Netherlands, Tech. Rep. UU-CS-2003-004, 2003.
- [21] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A practical Guide*. Academic Press, Boston, 1988.
- [22] H. P. Moreton, "Minimum curvature variation curves, networks, and surfaces for fair free-form shape design," Ph.D. dissertation, University of California at Berkeley, 1992.
- [23] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 1997. [Online]. Available: <http://www.cs.ruu.nl/geobook/>
- [24] Y. Kanayama and B. I. Hartman, "Smooth local path planning for autonomous vehicles," in *Proc. of the 1989 IEEE International Conference on Robotics and Automation (Vol. 3)*, Scottsdale, AZ, 1989, pp. 1265–1270.
- [25] C. de Boor, *A practical guide to splines*. Springer-Verlag, 1978.
- [26] M. Cox, "The numerical evaluation of B-splines," *J. Institute of Mathematics and its Applications*, vol. 10, pp. 134–149, 1972.
- [27] P. Dierckx, *Curve and Surface Fitting with Splines*. New York: Clarendon Press, 1995.
- [28] U. Reif, "Best bounds on the approximation of polynomials and splines by their control structure," *Computer Aided Geometric Design*, vol. 17, no. 6, pp. 579–589, 2000.
- [29] T. Berglund, T. Strömberg, H. Jonsson, and I. Söderkvist, "Epi-convergence of minimum curvature variation b-splines," Department of Computer Science and Electrical Engineering, Luleå University of Technology, Sweden, Tech. Rep. 2003:14, ISSN 1402-1536, 2003.
- [30] T. Berglund, U. Erikson, H. Jonsson, K. Mrozek, and I. Söderkvist, "Automatic generation of smooth paths bounded by polygonal chains," in *Int. Conf. on Computational Intelligence for Modelling Control and Automation (CIMCA'2001)*, M. Mohammadian, Ed., Las Vegas, USA, July 2001, pp. 528–535.
- [31] T. Berglund, H. Jonsson, and I. Söderkvist, "An obstacle-avoiding minimum variation b-spline problem," in *Int. Conf. on Geometric Modeling and Graphics (GMAG03)*, London, England, July 2003.
- [32] P. Gill, W. Murray, and M. Wright, *Practical Optimization*. Academic Press, New York, 1981.
- [33] S. Tamrock, *Toro 2500 Electric Technical specification L002500E-1*, retrieved 16 Dec. 2003 from the World Wide Web: <http://www.toro.sandvik.com/>.
- [34] H. Imai and M. Iri, "Polygonal approximations of a curve—formulations and algorithms," in *Computational Morphology*, G. T. Toussaint, Ed. Amsterdam, Netherlands: North-Holland, 1988, pp. 71–86.
- [35] T. Espelid, "Doubly adaptive quadrature routines based on newton-cote rules," Department of Informatics, Tech. Rep. 229, May 2002.



Andrej Brodnik THIS IS STILL MISSING



Håkan Jonsson THIS IS STILL MISSING



Kent Mrozek THIS IS STILL MISSING



Mats Staffanson THIS IS STILL MISSING



Tomas Berglund THIS IS STILL MISSING



Inge Söderkvist THIS IS STILL MISSING