IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '91, Nov. 3-5, 1991, Osaka, Japan. IEEE Cat. No. 91TH0375-6

Trajectory Generation with Curvature Constraint based on Energy Minimization*

H. Delingette, M. Hebert, K. Ikeuchi

The Robotics Institute

Carnegie Mellon University

Pittsburgh, PA 15232

Abstract

The trajectory generation problem for mobile robots consists in providing a set of trajectories that are "smooth" and meet certain boundary conditions. We present a method to generate curvature continuous trajectories for which the curvature profile is a polynomial function of arc length. An algorithm based on the deformation of a curve by energy minimization allows to solve general geometric constraints which was not possible by previous methods. Furthermore, we are able to take into account the limitation of radius of curvature of the robot by controlling the extrema of curvature along the path.

1. Introduction

The trajectory generation problem is a key aspect of the more general Motion Planning problem for mobile robots. Once the Path Planning module have found a satisfactory global path among obstacles, the Trajectory Planning module have to define the trajectories that will be used by the Tracking module. The trajectory generation problem is of purely geometric nature and can be defined as providing a set of trajectories that are "smooth" and meet certain boundary conditions.

The notion of "smoothness" is an ambiguous one. First, smoothness of a trajectory relates to the smoothness of its curvature profile $k(s)$ (s is the length along the curve). Most mobile robots or autonomously guided vehicles are controlled by the velocities of their wheels which are related to the radius of curvature of the vehicle. Therefore, suitable trajectories should have a smooth curvature profile $k(s)$ in order to guaranty smooth variations of wheels velocities.

Second, the smoothness of a trajectory is a relative concept and is defined through the use of a smoothness criterion. Authors have used different types of smoothness criterion in order to derive trajectories. Kanayama et al.[1] used the square of curvature and the square of the derivative of curvature as cost functions:

$$C = \int k^2 ds = \int \left(\frac{d\varphi}{ds}\right)^2 ds$$

$$C = \int \left(\frac{d^2\varphi}{ds^2}\right)^2 ds$$

where φ is the polar angle of the tangent vector. The trajectories that minimizes these criterion are clothoids and cubic spirals. Takahashi et al[2] used the jerk or derivative of acceleration:

$$C = \int [x_{ttt}^2 + y_{ttt}^2]\, dt$$

which led to quintic polynomials. For a different purpose, Horn[3] studied the curves that minimize the square of curvature with fixed end points:

$$C = \int k^2 ds = \int \frac{(x_u y_{uu} - y_u x_{uu})^2}{(x_u^2 + y_u^2)^{5/2}} du$$

He found that the optimal curves were those having the intrinsic equation:

$$k^2 = \mu \cdot \cos(\varphi - \varphi_0)$$

Bruckstein[4] pointed out that the previous cost function was scale dependant and proposed to use:

$$C = L \cdot \int k^2 ds = \int ds \cdot \int k^2 ds$$

where L is the length of the curve. He found that curves of equation:

$$k^2 = \mu \cdot \cos(\varphi - \varphi_0) + \nu$$

were solution which include circles.

Circular arcs and lines[5] have first been used to generate trajectories despite the fact that the curvature profile generated is not continuous. Quintic polynomial[2]l and B-splines[6] are easy to compute and can provide curvature continuity along the curve. But their curvature profile is complex, not necessary smooth which make them difficult to follow. Clothoids on the contrary are easy to track because their curvature profile is a straight line but are difficult to compute because no closed-form expression of the coordinates (x,y) is available. Pairs of clothoids[7] have been used to join two straight lines and provide the minimum length curve for a maximum jerk. Shin et al[8]. developed a method to create piecewise-clothoids trajectories that guaranty continuity of curvature; but its complexity and some numerical considerations limit its applicability. Cubic spirals introduced by Kanayama and Hartman[1] provide continuous smooth trajectories and minimize the variation of jerk but is rather difficult to compute. Nelson[9] chose curves with closed-form expression such as polar splines to join pairs of segment.

Clothoids, cubic spirals and more generally curves with a polynomial curvature profile $k(s)$ are of great interest for trajectory generation because they provide a simple curvature profile. Nonetheless current numerical methods make them expensive to compute and therefore limit their applicability. In this paper, we propose an original method to build such curves; the generality and efficiency of this method allow to

*. This research was supported in part by NASA under Grant NAGW 1175, and in part by DARPA through ARPA order No 4976. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, expressed or implied of those agencies.

create trajectories that are curvature continuous and that meet given end conditions. Furthermore, it is possible by adding control points to provide trajectories such that their maximum curvature is below a given value; therefore, we can take into account the limitation of radius of curvature of a mobile robot.

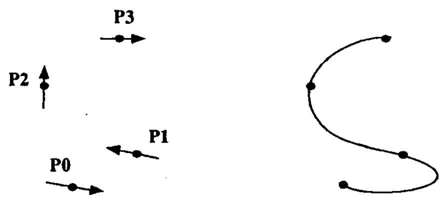
Section2 describes the algorithm and Section 3 gives some examples of trajectories.

2. Trajectory generation

2.1. Path constraints

Once the Path Planning problem solved, a set of postures is generated that provides the geometric constraints for the Trajectory Generation module. These constraints can be of different nature and we set the following definition: a *posture of order p* ($p>0$) is defined by the set $(x,y,\phi,\phi^{(1)},..\phi^{(p-1)})$ where (x,y) are the coordinates of a point and where $\phi^{(i)}$ is the i^{th} derivative of the heading with respect to the arc length. For example, a posture of order two corresponds to the data of a point, a heading and a curvature. A posture of order zero corresponds to the data of a point (x,y). A posture of order p is therefore a set of 2+p real numbers. In practice, postures of order p with $p<3$ are used to generate trajectories.

We can represent path constraints by an ordered set of n postures $(Q_0,Q_1,..,Q_n)$. Figure 1 shows an example of path constraints with a trajectory that matches the constraints.



(a) (b)

Figure 1: a) Path constraints made of four postures of order one. b) Resulting trajectory.

2.2. Intrinsic Splines

Given geometric constraints, we use curves with polynomial curvature profile to generate trajectories. We set the following definition: we call *Intrinsic Splines* of degree n, ISn, curves whose curvature profile $k(s)$ is a polynomial of degree n. Their parametric expression is:

$$x(u) = x_0 + \int_0^u \cos(a_0 + a_1 \cdot s + .. + a_{n+1} \cdot s^{n+1})\, ds$$

$$y(u) = y_0 + \int_0^u \sin(a_0 + a_1 \cdot s + .. + a_{n+1} \cdot s^{n+1})\, ds \qquad (1)$$

Clothoids correspond to IS1 while cubic spirals correspond to IS2. Figure 2 shows examples of intrinsic splines of degree one and three.

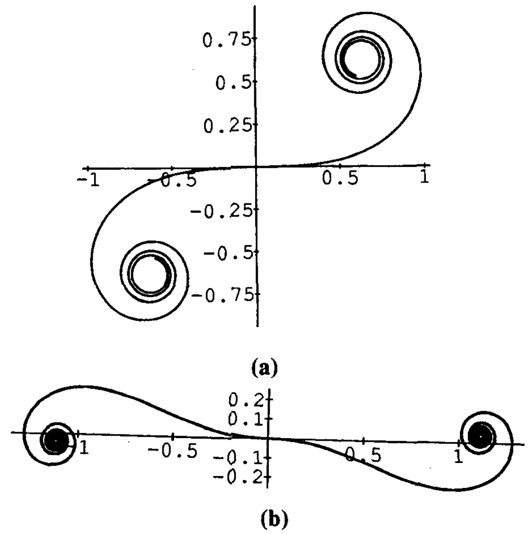


Figure 2: a) Clothoid: Curve of equation $k=s$. b) Intrinsic spline of degree 3: curve of intrinsic equation $k=s^3-s$

Intrinsic splines can be used to solve end conditions in the same way that polynomial splines. But while end conditions are defined in terms of first and second derivatives for polynomial splines, they are defined in terms of heading and curvature for the intrinsic splines. More precisely, given two postures of order p, there exists at most one intrinsic spline of order $n=2p-1$ that matches the constraints. The following table shows the analogy between on one hand, IS1 (Clothoids) and cubic splines, and on the other hand, IS3 and quintic splines.

Constraints	Intrinsic Spline	Constraints	Polynomial Spline
(P_0,ϕ_0) (P_1,ϕ_1)	IS1 (Clothoids)	(P_0,P'_0) (P_1,P'_1)	Cubic Spline
(P_0,ϕ_0,k_0) (P_1,ϕ_1,k_1)	IS3	(P_0,P'_0,P''_0) (P_1,P'_1,P''_1)	Quintic Spline
(P_0,ϕ_0) (P_1) .... (P_{n-1}) (P_n,ϕ_n) (ϕ, k) continuous	Piecewise IS1	(P_0,P'_0) (P_1) .... (P_{n-1}) (P_n,P'_n) (P', P'') continuous	Piecewise Cubic Spline
(P_0,ϕ_0,k_0) (P_1) .... (P_{n-1}) (P_n,ϕ_n,k_n) (ϕ, k, k', k'') continuous	Piecewise IS3	(P_0,P'_0,P''_0) (P_1) .... (P_{n-1}) (P_n,P'_n,P''_n) $(P', .., P^{(4)})$ continuous	Piecewise Quintic Spline

Figure 3 shows the four types of geometric constraints that can be matched using the intrinsic splines; by combining these four types, it is possible to solve most of the path constraints encountered for trajectory generation.

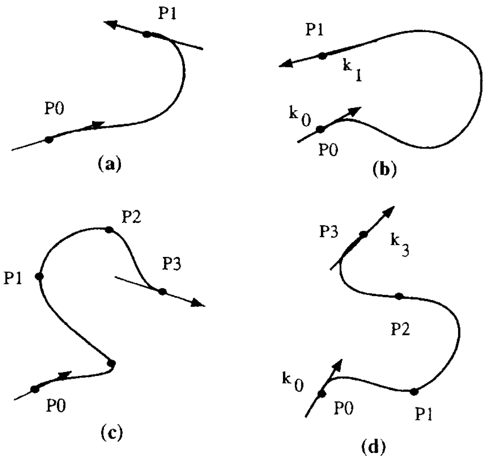


Figure 3: a) Geometric constraints consisting in two postures of order one. b) Two postures of order two. c) Two postures of order one with intermediate points. d) Two postures of order two with intermediate points.

Intrinsic splines of degree one and three are therefore sufficient for generating most trajectories. An algorithm based on the deformation of a string is used to compute the splines.

2.3. Trajectory generation

The lack of closed-form expression for intrinsic splines has been a serious limitation for their applicability since the usual numerical method such as Newton-Raphson algorithm or Simpson's approximation perform poorly. Our method has the advantage to be fully parallelizable and to solve more general conditions than the previous methods.

To explain the principle of the algorithm, we will use an analogy with cubic splines. Let $T^0(u)$ be a curve such that it meets the end-conditions (P_0,P'_0) and (P_1,P'_1) (Figure 4). If we use the smoothness criterion:

$$C = \int [x_{uu}^2 + y_{uu}^2]\, du$$

then the cubic spline is the only curve that meet the same end-conditions that $T^0(u)$ and that minimizes at the same time, the criterion C. The algorithm consists in deforming iteratively the curve $T^0(u)$ such that it decreases the cost C. A parallel can be drawn with the deformation of a clamped string as it tries to reach its stable position through the minimization of its potential energy. The deformation $D(T^0(u))$ of the curve $T^0(u)$ is given by the Euler-Lagrange equations derived from the criterion C:

$$D(T^0(u)) = D\left(\begin{bmatrix} x^0(u) \\ y^0(u) \end{bmatrix}\right) = \begin{bmatrix} x^0_{uuuu} \\ y^0_{uuuu} \end{bmatrix}$$

The curve $T^0(u)$ is transformed into the curve $T^0(u)+\alpha.D(T^0(u))$ (α is a constant). By iterating the deformation process, the curve will converge toward the cubic spline for which $D(T(u))=0$. The convergence is guaranteed by the convexity of the cost function C.This method is related to the regularization techniques, widely used in computer vision.

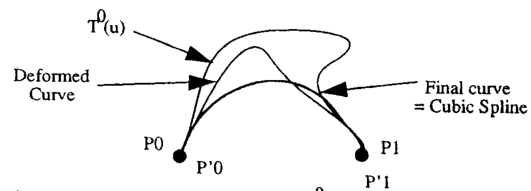


Figure 4: Deformation of a curve from $T^0(u)$ to a cubic spline solution.

In a similar manner, we can deformed a given curve $T(u)$ such that it converges toward an intrinsic spline. The deformation functional $D(T(u))$ is then defined as:

$$D_1(T(u)) = \frac{d^2s}{du^2}\vec{t} + \frac{ds}{du}\cdot\left(\frac{d\phi}{du}(u) - \frac{\left(\int_{-u_0}^{u_0}\frac{d\phi}{dt}(u+t)\,dt\right)}{2\cdot u_0}\right)\vec{n}$$

$$D_2(T(u)) = \frac{d^2s}{du^2}\vec{t} + \frac{-1}{3}\cdot\frac{ds}{du}\cdot\left(\frac{d^3\phi}{du^3} - \frac{\int_{-u_0}^{u_0}\frac{d^3\phi}{du^3}(u+t)\,dt}{2\cdot u_0}\right)\vec{n}$$

where $\vec{t}$ and $\vec{n}$ are the tangent and normal of the curve, s is the arc-length, ϕ is the polar angle of the tangent and u_0 is a constant. $D_1(T(u))$ leads to intrinsic spline of degree one, while $D_1(T(u))$ leads to IS3. In practice, we use discrete curve defined by a set of knots $\{R_i = (x_i,y_i)\}$ $(i=0,q)$ and we use discrete curvatures $\{k_i\}$, $(i=0,q)$ (Figure 5.a). The curves are initialized as straight lines and then deformed by moving the knots $\{Ri\}$; the deformation is stopped when the displacement of the knots is less than a threshold. Figure 5.b shows the deformation from a line to a clothoid..

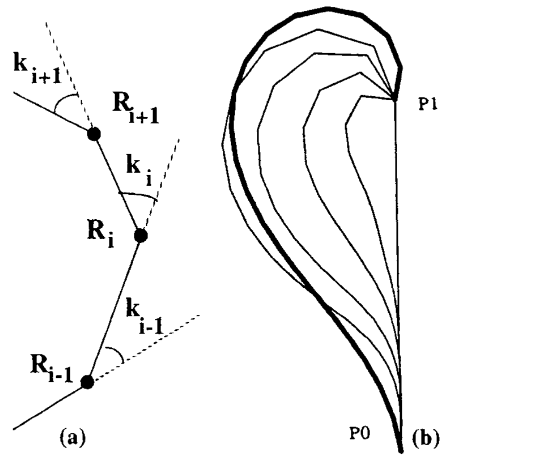


Figure 5: a) Definition of the discretized curvature k_i. b) Deformation from line to clothoid.

The convergence rate depends on the number of knots and of the degree of the intrinsic spline. We use a minimum of 12 knots to defined each curve

While previous numerical methods attempted to find the coefficients of the polynomial $k(s)$, our method provides directly a discrete trajectory that can be used by the tracking module. Furthermore, if symmetric postures are used then clothoid will degenerate into circles and IS3 into cubic spirals without modification of the algorithm. Finally, the coefficients of the polynomials $k(s)$ can be easily extracted once convergence has been reached.

The algorithm does not currently allow to generate curves that enroll around a point like a spiral.

2.4. Trajectories with bounded curvature

Because most mobile robots have a limited radius of curvature, it is necessary to provide trajectories with a curvature limited to certain bounds. Without this constraint, the mobile robot would be unable to follow the prescribe trajectory which could result in serious deviation from the path and eventually in a collision with an obstacle. Another reason to control the extrema of curvature along the path is to avoid the robot to slow down at high curvature points and therefore to guaranty a minimum speed of advance.

We propose a method that, given an intrinsic spline of order one or three, deforms the trajectory until its extrema of curvature are below a given value, while keeping the continuity of curvature. This method consists in adding intermediate points and moving these points outside the curvature of the curve. The result is a piecewise intrinsic spline.

More precisely, let $T(u)$ be a trajectory obtained from the previous method. Because the curvature profile is either a line or a cubic, there are at most two extrema of curvature. If the curvature exceeds a certain threshold, then these points are moved and then considered as intermediate points. A new spline is fit that meets the previous end-conditions and that goes through the intermediate points. The result is a piecewise intrinsic spline. The algorithm is applied on the new spline and iterated until the extrema of curvature are below the threshold. Figure 6 shows different stages of the algorithm.

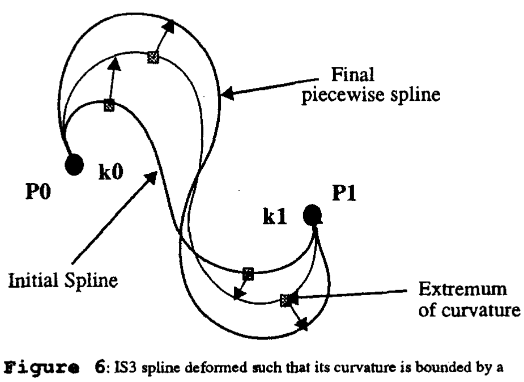


Figure 6: IS3 spline deformed such that its curvature is bounded by a given value. Small arrows indicate how the extremum of curvature are moved. The final spline is a piecewise IS3.

Figure7 shows how the extrema are moved along the normal of the curve. The distance of which the point is moved is proportional to the curvature.

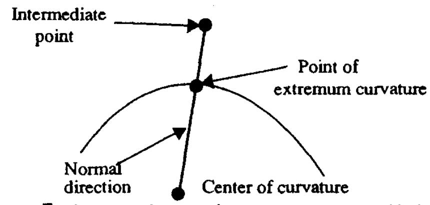


Figure 7: Movement of a point of extremum curvature outside the curvature of the curve. The intermediate point is used to fit a new piecewise spline.

By moving those points outside the curvature, we create a longer trajectory which decreases curvature since the amount of turning is spread along the curve. This method performs well if the maximum curvature allowed is not too small (otherwise trajectories tend to be extremely long).

3. Examples

3.1. Free Trajectories

Figure 8 is an example of a clothoid spline fit between two postures of order one. The curvature is linear and the jerk is constant.

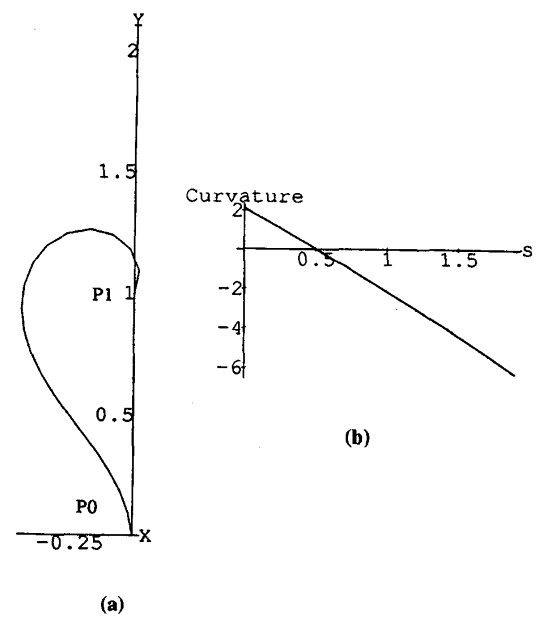


Figure 8: Clothoid drawn between P_0 (0,0) and P_1 (0,1) with ϕ_0=90° and ϕ_1= -135°. a) Curve b) Curvature profile.

Figure 9 is an example of a piecewise clothoid fit between two postures of order one and three intermediate points. The curvature is piecewise linear and the jerk is therefore not continuous at intermediate points.

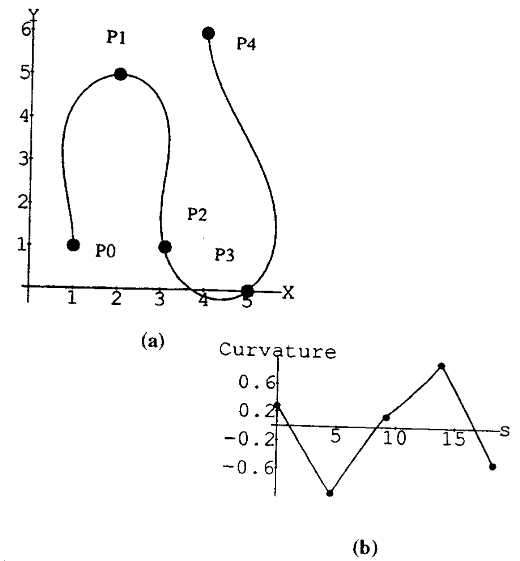


Figure 10: Set of clothoids passing through (P_0,P_1,P_2,P_3,P_4) with heading conditions $\phi 0=90^o$ and $\phi 4=90^o$. a) Curve b) Curvature profile

Figure 11 is an example of IS3 used to fit two postures of order two. The curvature is a cubic polynomial and jerk a parabola.

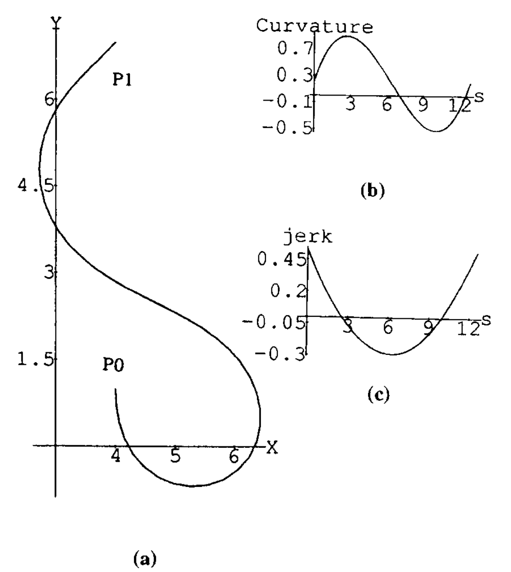


Figure 11: IS3 passing through P_0 (4,1) and P_1 (4,7) with heading conditions $\phi_0=90^o$, $\phi_1=50^o$ and curvature condition $k_0 = 0.2$, $k_1 = 0.2$. a) Curve b) Curvature profile c) Jerk profile.

Figure 12 is an example of IS3 used to fit two postures of order two with two intermediate points. The curvature profile is piecewise cubic and the jerk is C^1 continuous.

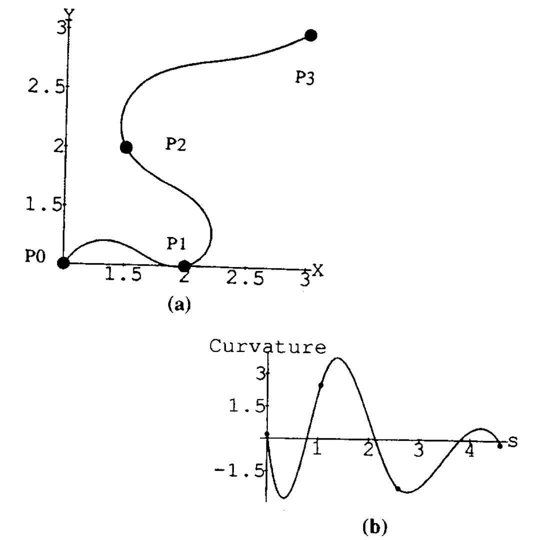


Figure 12: Set of IS3 passing through (P_0,P_1,P_2,P_3) with heading conditions $\phi_0=50^o$, $\phi_3=25^o$ and curvature conditions $k_0=0.2$, $k_3= -0.2$. a) Curve b) Curvature profile.

3.2. Trajectories with bounded curvature

Figure 13 illustrates the effect of limiting extremum of curvature: we use the same end conditions than figure 9 and 11 but add a constraint on maximum curvature.

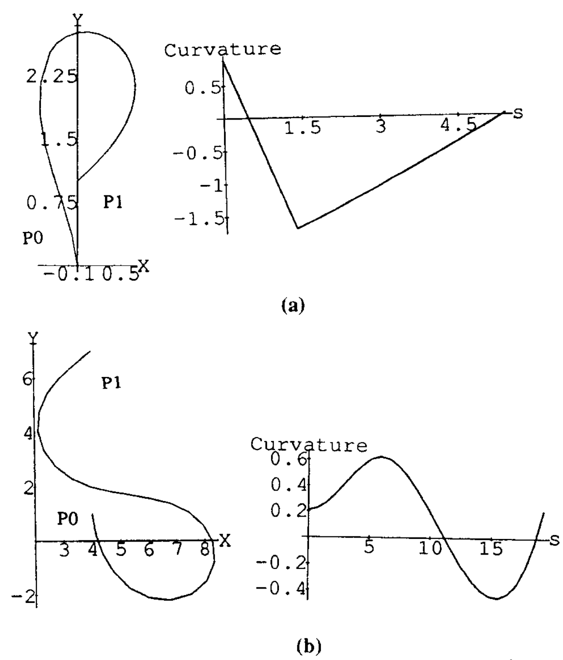


Figure 13: a) Same as figure 9 but with curvature limited to 2 m^{-1}. b) Same as figure 11 but with curvature limited to 0.6 m^{-1}.

3.3. Generating polygon

In order to avoid obstacles, mobile robots have to use rather complex paths and therefore need numerous postures to describe trajectories. While previous examples show that it is possible to use piecewise clothoids or IS3 to link two postures with intermediate points, this has two major drawbacks.

First, most of the time, the path is described in term of postures of order one; the Path Planning module prescribes a set of points where the robot should pass with a corresponding heading. Such conditions cannot be solved with piecewise splines.

Second, piecewise intrinsic spline are numerically inefficient. Each point has an influence on the shape of the whole curve; the movement of one intermediate point would modify the curve globally.

Instead, we propose a method similar to Bezier-Spline: we fit an IS3, between two postures the continuity of curvature being insured by the use of generating polygon.

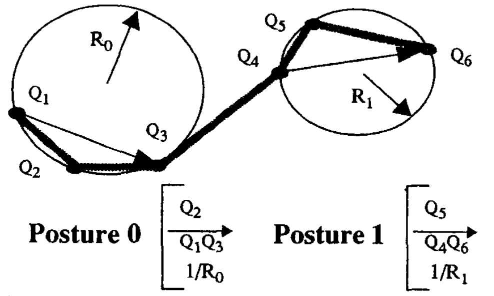
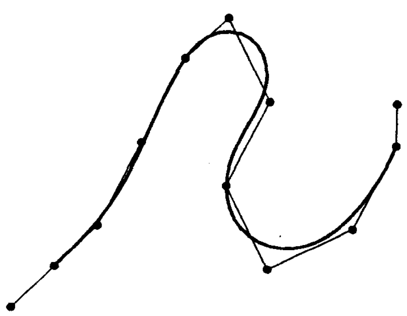
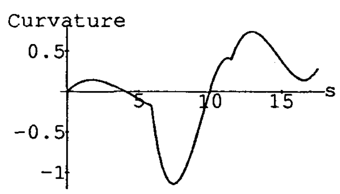


Figure 14: Two postures of order two are defined from 6 control points of the generating polygon.



(a)



(b)

Figure 15: a) Example of trajectory generated from a twelve points polygon and composed of three IS3. b) Curvature profile.

An IS3 needs two postures of order two to be defined. The Figure 14 shows how six control points (Q_1,Q_2,Q_3,Q_4,Q_5,Q_6) define two postures. The spline is constraint to pass through Q_2 and Q_5 with given headings and curvatures: the headings are determined by the direction of (Q_1Q_3) and (Q_4Q_6) while the curvatures are the inverses of the radius of the two circles passing through the first three and last three points. A polygon of $6+3p$ control points will generate p IS3 splines, the continuity of heading and curvature being systematically guaranteed. Figure 15.a is an example of a trajectory made of three IS3 generated from a twelve points polygon. Figure 15.b shows that curvature is only C^0 continuous along the path and presents some C^1 discontinuities where splines meet. The use of piecewise IS3 would have provided a trajectory with C^2 continuity of curvature.In addition to its efficiency, the use of generating polygon is an intuitive and natural way of defining a trajectory.

Conclusion

In this paper, we have shown that Intrinsic Splines, curves with polynomial curvature profile, can be used to generate suitable smooth trajectories for mobile robots. A general and efficient algorithm allow to handle these splines similarly to polynomial splines; use of generating polygons makes the trajectory generation more efficient and intuitive. The parallel structure of the algorithm would allow real-time performance on dedicated hardware and would extend the applicability of the Intrinsic Splines to solve different problems such as geometric modeling.

References

[1]Y. Kanayama and B. Hartman, "Smooth Local Path Planning for Autonomous Vehicles", Proc. Int. Conf. on Robotics and Automation, pp 1260-1264, Scottdale, Arizona, 1989.

[2]A. Takahashi, T. Hongo and Y. Ninomiya, "Local Path Planning and Motion Control for AGV in Positioning", Proc. IROS, pp 392-395, Tsukuba, Japan, 1989.

[3]B. K. Horn, "The curve of least energy", ACM Trans. Math. Soft. 9, pp441-460, 1983.

[4]A. Bruckstein, "On Minimal Energy trajectories", Journ. of Comp. Vision, Graphics and Image Processing Vol 49, pp283-296, 1990.

[5]T. Hongo, H. Arakawa, G. Sugimoto, K. Tange and Y. Yamamoto, "An Automatic Guidance System of a Self-Controlled Vehicle", Proc. IECON. 1985, MIT Press, 1985.

[6] K. Komoriya and K. Tanie, "Trajectory Design and Control of a Wheel-type Mobile Robot Using B-spline Curve", Proc. IROS, pp 398-405, Tsukuba, Japan, 1989.

[7]Y. Kanayama and N. Miyake, "Trajectory Generation for Mobile Robots", Third Int. Symp. on Robotics Research, pp 333-340, Gouvieux, France, 1985.

[8]D. Shin and S. Singh, "Path Generation for Robot Vehicles Using Composite Clothoid Segments", Carnegie-Mellon Un. Tech. Rep. CMU-RI-TR-90-31, 1990.

[9] W. L. Nelson, "Continuous Curvature Paths for Autonomous Vehicles", Proc. Int. Conf. on Robotics and Automation, pp 1265-1270, Scottdale, Arizona, 1989.

[10]H. Delingette, M. Hebert, K. Ikeuchi, "Energy functions for regularization algorithms", Proc. SPIE Symp. on Geometric Methods for Computer Vision, San Diego, July 1991.