3.3. Generating polygon

Conclusion

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 Positionning”, 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.

(a) (b)

Figure 10: Set of clothoids passing through (P_0,P_1,P_2,P_3,P_4) with heading conditions $\phi0=90^o$ and $\phi4=90^o$. 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 condition $\phi_0=90^o$, $\phi_1=50^o$ 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 posture of order two with two intermediate points. The curvature profile is piecewise cubic and the jerk is C^1 continuous.

(a) (b)

Figure 12: Set of IS3 passing through (P_0,P_1,P_2,P_3) with the 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}.

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 curvature constraints

Because most mobile robots have a limited radius of curvature, it is necessary to provide trajectories with curvature within 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 the robot.

Previous path generation method did not address this problem, the extrema of curvature being usually impossible to compute explicitly. We propose a method that, given an intrinsic spline of order one or three, deforme 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 used 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 picewise intrinsic spline. The algorithm is applied on the new spline and iterated until the extrema of curvature are below the threshold. Figure 6 shows the different stages of the algorithm. It should be noticed that the extrema of curvature are not necessary the same duting the deformation.

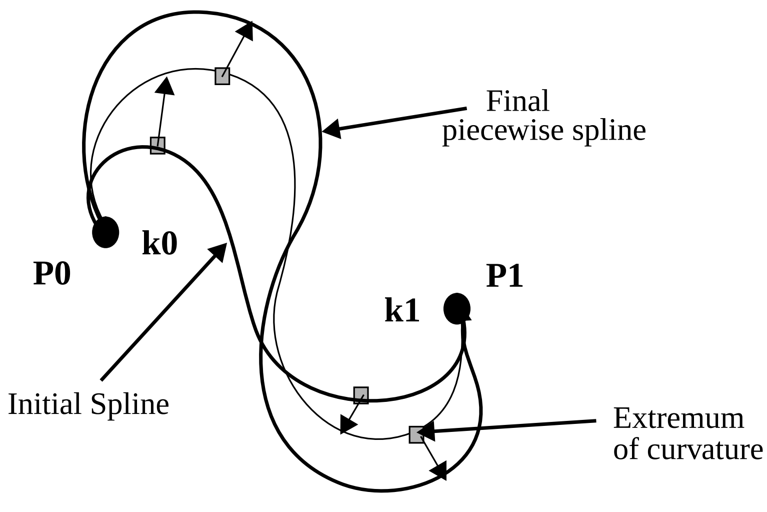


Figure 6: IS3 spline deformed such that its curvature is bounded by a given value. The small arrows indicate how the extremum of curvature are moved. The final spline is a piecewise IS3.

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.

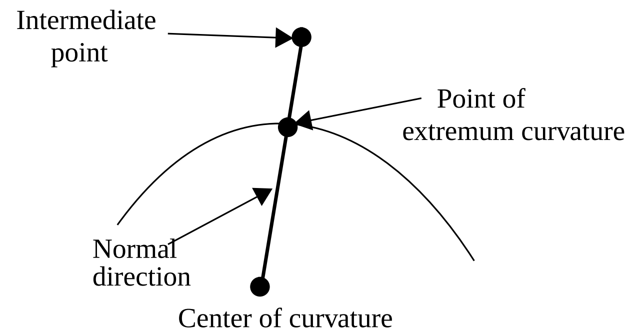


Figure 7: Movement of a point of extremum curvature outside the curvature of the curvature. The intermediate point is used to fit a new [piecewise spline.

By moving the points outside the curvature, we create a longer trajectory which allow to decrease 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 that is fit between two postures of order one. The curvature is linear and the jerk is constant.

(a) **(b)**

Figure 8: Clothoid drawn between P_0 (0,0) and P_1 (0,1) with $\phi_0=90^o$ and $\phi_1= -135^o$. 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.

Figure3 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..

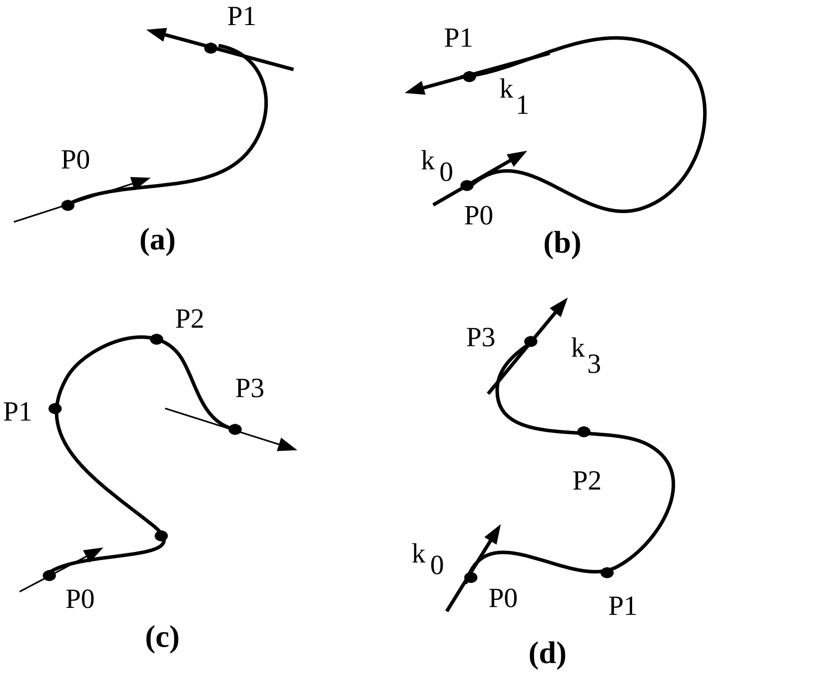


Figure 3: a) Geometric contraint consisting in two posutures of order one. b) Two postures of order two.c) Two postures of order one with intermediate points. d) Two postures of order two with intermediate points.

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

2.3. Trajectory generation

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

To explain the principle of the algorithm, we will make 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 tome the criterion C. The algorithm consists in deforming iteratively the curve $T^0(u)$ such that it minimizes its 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 equation 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) that applying this deformation iteratively, 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 an example of regularization techniques, widely used in computer vision.

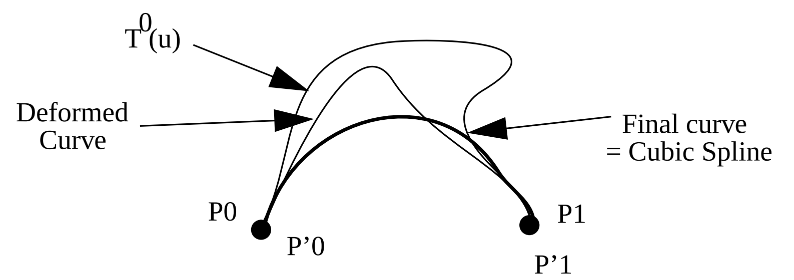


Figure 4: Deformation of a curve from $T^0(u)$ to a cubic spline solution.

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

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

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

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 the discrete curvature $\{k_i\}$, $(i=0,q)$ (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 of a curve from a line to a 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.

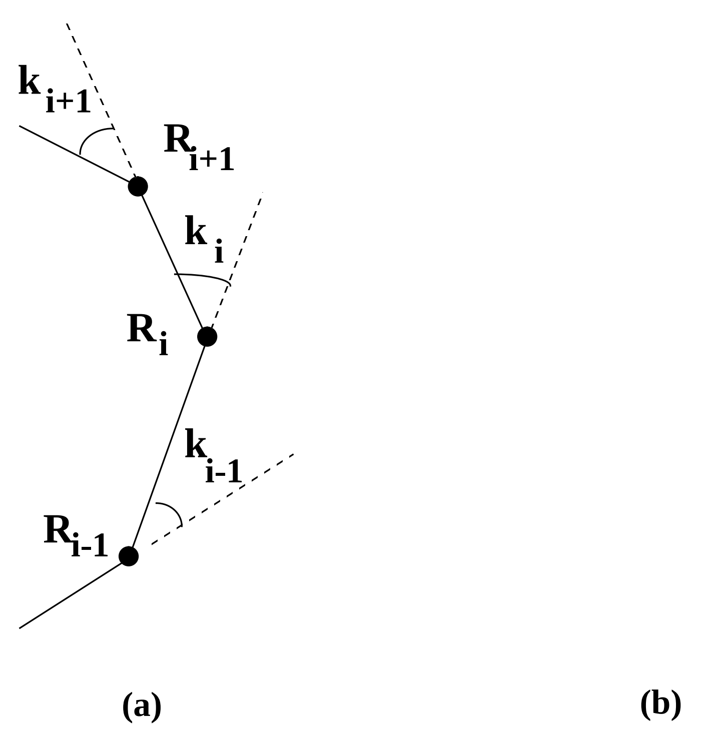


Figure 5: a) Definition of the discretized curvature k_i. b) Deformation from line to clothoid.

While previous numerical methods attempted to find the coefficients of the polynomial k(s), our method provides direcly a discrete trajectory that can be used by the tracking module.Futhermore, if symetric postures are used then tions. A generating polygon can be used to handle these spline in the same manner than B-spline. Futhermore, 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.

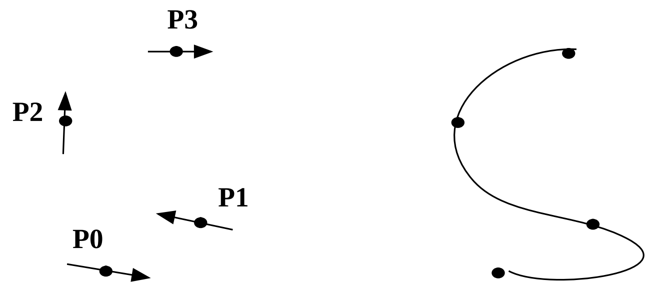
In the next section, we will consider Section2 descibes the algorithm and Section 3 gives some examples of trajectories.

2. Trajectory generation

2.1. Path constraints

Once the Path Planning problem solve, a set of postures is generated that provide 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 ith 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.

(a)

(b)

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

Intrrinsic 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

Trajectory Generation with curvature constraint based on energy minimization

H. DELINGETTE, M. HEBERT, K. IKEUCHI

The Robotics Institute , Carnegie Mellon University, Pittsburgh PA 15213

Abstract [1]

Trajectories for mobile robots have to be smooth and at the same time have to meet certain geometric constraints. Previous work We present an algorithm that generates trajectories that

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 geometrically define the trajectories that will be used by the Tracking module. Therefore, the trajectory generation problem is of purely geometric nature; furthermore, it 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 the 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 and 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 and 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 :were solution which include circles.

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

Circular arcs and lines[5] have first been used to generate trajectories despite the fact that the curvature profile generated is not continuous. Quintic polynomia[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 and 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 and al[8]. develloped 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] allow a continuous smooth trajectory and minimizes the variation of jerk but is rather difficult to compute. Nelson[9] chose curves with closed-form expression such as polar splines to join a pair 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 create trajectories that are curvature continuous and that meet given end condi-

1. 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.