ICAR '97
Monterey, CA, July 7-9, 1997

Path Planning for Mobile and Hyper-Redundant Robots using Pythagorean Hodograph Curves

Herman Bruyninckx, Dominiek Reynaerts
Katholieke Universiteit Leuven, Mechanical Engineering
Celestijnenlaan 300B, B-3001 Heverlee, Belgium
Herman.Bruyninckx@mech.kuleuven.ac.be

Abstract

Pythagorean hodograph curves (PH curves) have a number of advantages over other splines commonly used in planar robot path planning: (i) the integral of curvature is small; (ii) the curve, its length, curvature and total "bending energy" are known in closed-form; (iii) start and end positions and directions are straightforwardly usable as boundary conditions; and (iv) length can be traded off against curvature quite easily. These facts make PH curves especially interesting for the planning of car-like mobile robot paths, as well as for the calculation of the "backbone curve" in the inverse kinematics of hyper-redundant robots.

Keywords: mobile robots, hyper-redundant robots, kinematics, motion planning, splines.

1 Introduction

Car-like mobile robots and hyper-redundant ("snake" or "serpentine") robots both have particular constraints on their motion capabilities, which require special attention during motion planning.

Car-like mobile robots have only two instantaneous degrees of freedom: they cannot move transversally to their wheel axles. Moreover, they should not be given very "nervous" motion commands, since these increase the risk of slippage. The path is the geometric locus of the origin of a user-defined reference frame on the robot (Fig. 1); the tangent to the curve represents the instantaneous linear velocity of this point, which together with the curvature of the curve determines the instantaneous angular velocity of the reference frame. Hence, any given curve defines a motion for the mobile robot, and vice versa.

The instantaneous direction of motion of the car-like mobile robot is constrained by the direction of the steering wheel. Hence, the path planning curve must have a prescribed tangent direction at the end points. (These boundary conditions are called "Hermite conditions" in the spline literature, [7].) A car-like mobile robot has two complementary types of constraints on the curvature of its paths:

1. If the steering wheel direction does not change, the robot will move on a circle with radius determined by the current position of the steering wheels. Hence, in order not to have to change the steering wheel too abruptly, the robot path should start with an initial *curvature* boundary condition that corresponds to the initial position of the steering wheel.

2. Most steering mechanisms have limited range: it's impossible to make turns with a radius below a given threshold. The motion planner should already take this constraint into account, and hence must generate paths whose curvature nowhere exceeds this threshold.

The mobile robot path should be as *smooth* as possible, i.e., continuous curvature, or even continuity at higher derivatives. But also the *integral* of the curvature is important, in order to avoid excitation of the robot dynamics, and hence slippage. (For example, a sinusoidal path is infinitely smooth, but it can excite the robot's dynamics.) A third obvious "optimality" criterion is the *length* of the path. These three criteria can be mathematically modelled by the following functional:

$$\min \int_0^l \kappa^2(s)\, ds + \alpha \int_0^l ds, \qquad (1)$$

with $\kappa(s)$ the curvature at arc length s, and α a user-defined weighting factor between "bending energy" of the path on the one hand (i.e., $\int \kappa^2\, ds$) and length on the other hand (i.e., $\int ds$).

Twohundred years ago already, the analogy of this problem with the total bending energy of a thin elastic rod has been established, [2, 7, 8]. This thin rod model is the mathematical approximation of the real wooden *splines* that craftsmen have been using for many centuries. Hence, the literature on splines presents a wealth of curves with potential use in mobile robot path planning.

Snake robots typically consist of a large number of identical links ("trusses"), each with limited (discrete or continuous) motion capabilities, see e.g. [1, 6], and Fig. 2. (This paper considers *planar* snake robots only.) One of the main problems is the inverse kinematics: given the desired position and orientation of the end effector reference frame, find the corresponding position of the actuated joints in each segment. One promising inverse kinematics technique (see [1] and references therein), decouples this problem in two separate subproblems: (i) find a curve from base frame to end effector frame (this is the so-called *backbone* curve), and (ii) move the individual links as close as possible to this (imaginary) backbone curve. This paper presents a solution to the first subproblem. The criterion of minimum bending energy is also appropriate for snake robots: a minimal bending requires minimal segment motions, and reduces the risk that cables running through the trusses are damaged. Hermite boundary conditions and maximum curvature constraints are relevant; boundary curvature conditions are not. The major difference with mobile robots is that the constraint on the *length* of the backbone curve is very strict. Note however that this length is *not a constant*: typically, actuating a truss segment changes the distance between its two end plates. Moreover, the real snake robot is only able to *approximate* the backbone curve, i.e., the real position and orientation of the last segment differ from those deduced from the backbone curve.

In summary, motion planning for car-like mobile robots and hyper-redundant robots should be done with geometric curves for which not only the curve itself, but also the length, the curvature profile and the bending energy can be calculated very efficiently.

One of the most recent additions to the spline literature that satisfy most of these requirements are the Pythagorean hodograph curves (PH curves). They have been developed almost a decade ago, [5], but only recently efficient and compact computational algorithms have been published, [4]. Section 2 discusses the properties and advantages of PH splines, as well as an algorithm to compute them. Section 3 then gives a number of examples, using PH splines for robot path planning with different boundary conditions.

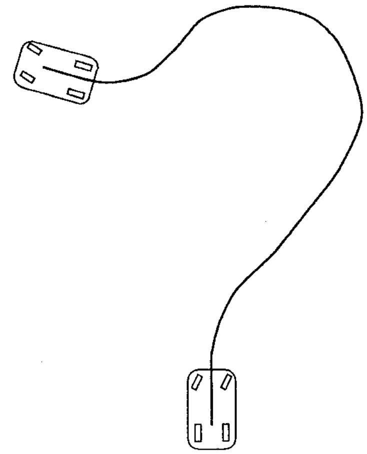

Figure 1: Car-like mobile robot motion path.

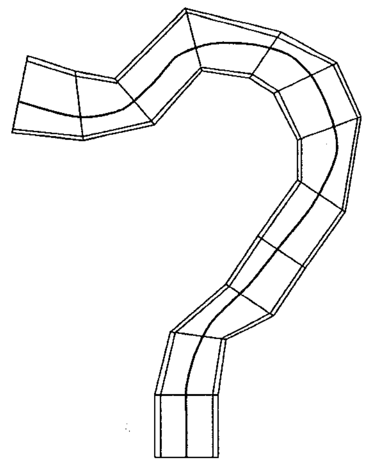

Figure 2: Hyper-redundant robot, with backbone curve.

2 Pythagorean hodograph splines

A planar curve $\boldsymbol{r}(t) = \{x(t), y(t)\}$ is a Pythagorean hodograph (PH) curve if its hodograph (i.e., the curve of *velocities* $\boldsymbol{r}'(t) = \{x'(t), y'(t)\}$) satisfies the algebraic Pythagorean *constraint*:

$$x'^2(t) + y'^2(t) \triangleq \sigma^2(t), \tag{2}$$

for some polynomial $\sigma(t)$. PH curves can be written in Bernstein-Bézier spline form:

$$\boldsymbol{r}(t) = \sum_{k=0}^{n} \boldsymbol{p}_k \binom{n}{k} (1-t)^{n-k} t^k, \quad t \in [0,1]. \tag{3}$$

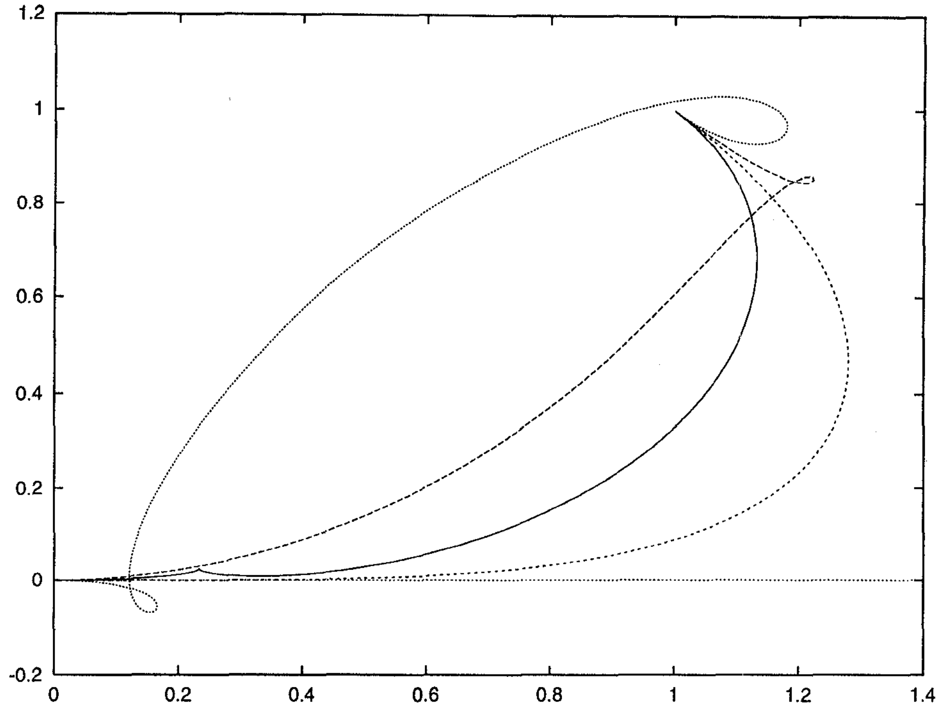

Figure 3: Four quintic PH splines solving a typical planning problem, with initial and final position ($\boldsymbol{p}_0 = (0;0), \boldsymbol{p}_1 = (1;1)$) and tangent vectors ($\boldsymbol{d}_0 = (1;0); \boldsymbol{d}_1 = (-1;1)$) as boundary conditions.

It is then well known that the control points $\boldsymbol{p}_k$ are closely related to the derivatives at the boundaries of the spline:

$$\boldsymbol{p}_0 = \boldsymbol{r}(0), \quad \boldsymbol{p}_1 = \boldsymbol{r}(0) + \frac{1}{n}\boldsymbol{r}'(0),$$
$$\boldsymbol{p}_{n-1} = \boldsymbol{r}(1) - \frac{1}{n}\boldsymbol{r}'(1), \quad \boldsymbol{p}_n = \boldsymbol{r}(1),$$

and similar relationships for the higher derivatives. In the context of mobile robot path planning, *quintic* PH splines (i.e., $n = 5$ in Eq. (3)) are an appropriate compromise between smoothness and complexity, since they allow boundary conditions up to curvature. However, it is also immediately clear from the definition of quintic PH splines that *arbitrary* boundary conditions can not be satisfied in general: a fifth order spline can accomodate boundary positions, tangents and second derivatives, and hence curvature $\kappa(t)$, since $\kappa(t) = (\boldsymbol{r}'(t) \times \boldsymbol{r}''(t))/|\boldsymbol{r}'(t)|^3$; a PH quintic spline, however, must in addition also satisfy the Pythagorean condition (2) such that not all six "degrees of freedom" of the spline can be used to satisfy the boundary conditions. It turns out that quintic PH splines can always be found for given tangent directions at the boundaries, but at the "expense" of having always *four* different solutions. However, selecting the most appropriate of these four solutions is made very easy and attractive by the following properties of PH quintic splines, [3]:

1. The curve, its length, curvature and total "bending energy" can be found in *closed-form*. Hence, it is very efficient to select the curve with the smallest functional of Eq. (1).

2. PH splines have very "smooth" profiles, i.e., the integral of the square of the curvature along the path is much smaller than, for example, cubic splines.

2.1 Algorithm for quintic PH splines

The algorithm for finding the quintic PH spline with given tangent vectors at the boundaries is given in [4], and is repeated here for convenience. It starts from the hodograph in *complex vector form*:

$$\boldsymbol{r}'(t) = \boldsymbol{k}(t - \boldsymbol{a})^2(t - \boldsymbol{b})^2, \qquad (4)$$

where $\boldsymbol{a}, \boldsymbol{b}$ and $\boldsymbol{k}$ are complex numbers. Integrating the hodograph, taking into account the boundary conditions $\boldsymbol{r}(0), \boldsymbol{d}_0 \triangleq \boldsymbol{r}'(0), \boldsymbol{r}(1)$ and $\boldsymbol{d}_1 \triangleq \boldsymbol{r}'(1)$, gives

$$r(t) = \frac{k}{30}\left[(t - a^5 - 5(t-a)^4(t-b) + 10(t-a)^3(t-b)^2\right] + c, \tag{5}$$

with

$$c = \frac{k}{30}(a^5 - 5a^4b + 10a^3b^2). \tag{6}$$

These formulas assume that the interpolation problem is reduced to a standard form where $r(0) = (0;0)$ and $r(1) = (1;0)$. a, b and k are related to the boundary conditions as follows:

$$k = \frac{d_0}{a^2b^2}, a = \frac{\mu_1}{\mu_1+1}, b = \frac{\mu_2}{\mu_2+1}, \tag{7}$$

with μ_1 and μ_2 the roots of

$$\mu^2 - \alpha\mu + \rho = 0, \quad \rho = \frac{d_0}{d_1}, \tag{8}$$

and α a solution of

$$\alpha^2 - 3(1+\rho)\alpha + 6\rho^2 + 2\rho + 6 - \frac{30}{d_1} = 0. \tag{9}$$

The length l of the curve $r(t), t \in [0,1]$, can then be found in closed form as a rational function of k, a and b:

$$l = |k|^2\left(|a|^2|b|^2 - |a|^2Re(b) - |b|^2Re(a) + \frac{|a|^2 + 4Re(a)Re(b) + |b|^2}{3} + \frac{1}{5}\right), \tag{10}$$

Where "$|\cdot|$" stands for absolute value, and "$Re(\cdot)$" for the real part of a complex number. Similarly, the curvature at parameter value t is

$$\kappa(t) = \frac{Im(\dot{\bar{r}}\ddot{r})}{|\dot{r}|^3}, \tag{11}$$

with

$$\dot{r} = k(t-a)^2(t-b)^2, \tag{12}$$

and

$$\ddot{r} = 2k(t-a)(t-b)(2t-a-b). \tag{13}$$

"$Im(\cdot)$" stands for the imaginary part of a complex number, and "$\bar{\cdot}$" is the complex conjugate. We refer to [3] for the analytical expression of the total bending energy of the curve, which is only slightly more complicated than the formulas for length and curvature. The computational costs of these closed-form equations are low.

Figure 3 gives the four solutions of a typical planning problem, with initial and final position and orientation as boundary conditions. The curve with least "bending energy" is easily selected.

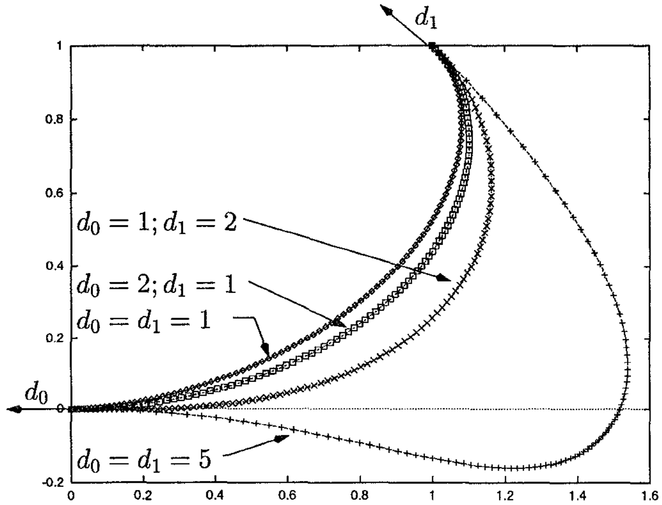

Figure 4: Influence of tangent boundary conditions on PH curves.

3 Planning with PH splines

This Section describes two complementary planning problems: (i) given desired boundary tangent *vectors*, and (ii) given desired boundary *curvatures* and tangent *directions*.

3.1 Specified boundary tangents

This problem has already largely been solved by the quintic PH spline algorithm of the previous paragraphs. One has however two degrees of planning freedom left: the *magnitudes* of the tangent vectors. Increasing this magnitude makes the curves longer; the effect on the curvature depends on the current spline. Figures 4 and 5 show the effects of changing magnitudes of the tangent directions at the boundary points on, respectively, the length and shape of the curve, and on the curvature profile.

3.2 Specified boundary curvature

It has been mentioned before already that in general no closed-form solution exists for the interpolation problem with specified curvatures at the boundaries. These paragraphs describe an iterative method that can be used to *approximate* the desired curvatures. The typical drawback of the method is that the length of the curve can increase drastically for small increases in the approximation of the curvature, see Table 1 and Figure 6.

The iterative method works as follows:

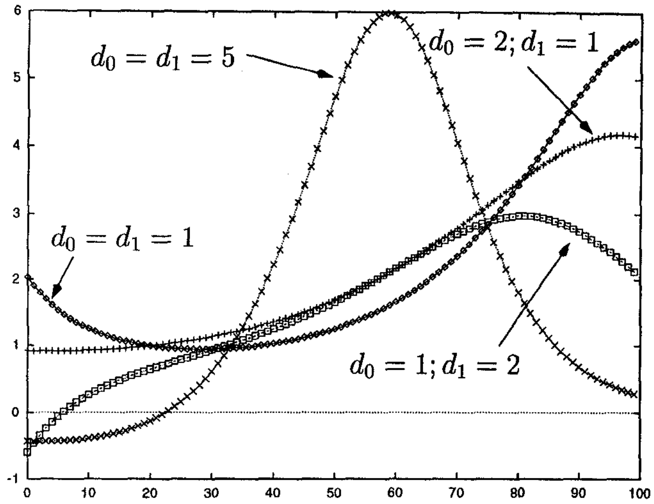

Figure 5: Curvature plots of the four interpolating PH quintic splines of Fig. 4.

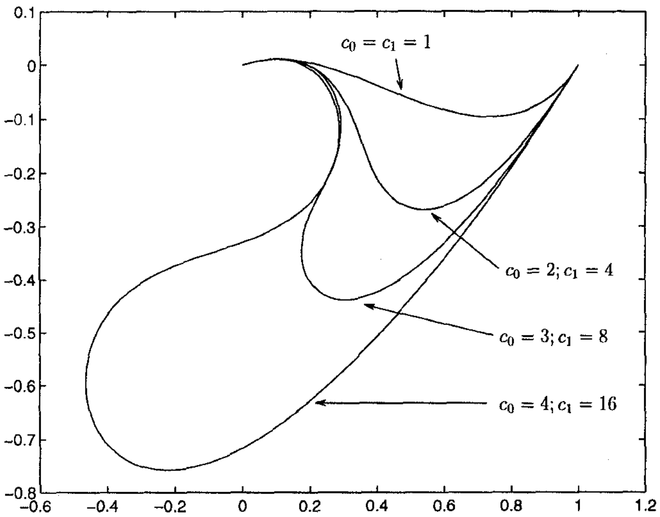

Figure 6: Iteration of boundary curvatures. The curves correspond to some values in Table 1.

1. Taking the derivative of the definition (4) of the hodograph of a PH quintic spline leads straightforwardly to the following relationships between, on the one hand, the magnitudes c_0 and c_1 of the tangent vectors at the boundary points, and, on the other hand, the boundary curvatures κ_0 and κ_1:

$$\kappa_0 = -\frac{2\,\mathrm{imag}(\frac{1}{\boldsymbol{a}} + \frac{1}{\boldsymbol{b}})}{c_0}, \tag{14}$$

$$\kappa_1 = \frac{2\,\mathrm{imag}(\frac{1}{1-\boldsymbol{a}} + \frac{1}{1-\boldsymbol{b}})}{c_1}, \tag{15}$$

where "imag($\boldsymbol{x}$)" denotes the imaginary component of the complex number $\boldsymbol{x}$. The complex numbers $\boldsymbol{a}$ and $\boldsymbol{b}$ follow from the PH algorithm. Hence,

$$\frac{\partial \kappa_0}{\partial c_0} \approx \frac{2\,\mathrm{imag}(\frac{1}{\boldsymbol{a}} + \frac{1}{\boldsymbol{b}})}{c_0^2}, \tag{16}$$

$$\frac{\partial \kappa_1}{\partial c_1} \approx -\frac{2\,\mathrm{imag}(\frac{1}{1-\boldsymbol{a}} + \frac{1}{1-\boldsymbol{b}})}{c_1^2}. \tag{17}$$

The relations are only approximations, since both $\boldsymbol{a}$ and $\boldsymbol{b}$ depend on the boundary tangent vectors too, Eqs (7)-(9).

2. The relations (16)-(17) can then be used to increase/decrease the tangent vector magnitudes c_0 and c_1, according to the fact whether the boundary curvature κ_0 and κ_1 of the actual curve are too high/too low. The exact sense of the proportionalities between κ_i and c_i is determined by the complex numbers $\boldsymbol{a}$ and $\boldsymbol{b}$.

c_0	κ_1	c_1	κ_1
1	0.6646	1	-5.1552
2	0.3323	4	-1.2888
3	0.2215	8	-0.6444
4	0.1662	16	-0.3222
16	0.0415	256	-0.0201
256	0.0026	4096	-0.0013

Table 1: Iteration on length of boundary tangent vectors to reach desired boundary curvatures of $\kappa_0 = \kappa_1 = 0$. Some of the corresponding curves are shown in Fig. 6.

Table 1 shows some iteration steps from the initial curve with $c_0 = c_1 = 1$ to a new curve with (approximately) $\kappa_0 = \kappa_1 = 0$.

A recent reference [9] solves the boundary curvature interpolation with quintic PH splines in a somewhat different way: that paper uses parts of PH spirals (i.e., PH splines that start with zero curvature and have monotonically increasing/decreasing curvature) and glues them together to satisfy the tangent or curvature boundary conditions. Not all cases have a closed-form solution however, and the optimality of the result in the sense of the functional in Eq. (1) is not considered. Nevertheless, the approach in [9] has many practical advantages, and is complementary to the approach described in this paper.

3.3 Specified length

Unfortunately, the influence of the parameters c_0 and c_1 on the length of the PH curve is much more intricate than the relationships (16)-(17) for the boundary curvatures. Hence, iterating the PH curve to a desired length is a bit more of a trial and error procedure.

4 Conclusions

This paper has discussed the advantages of quintic Pythagorean hodograph splines for the "optimal" planning of a mobile robot's motion and the calculation of the backbone curve in the inverse kinematics of planar hyper-redundant robots. The major advantages are that the PH splines can be calculated analytically, and that they can be written in Bernstein-Bézier form (hence they are easily integrated in existing CAD systems!). The optimality criterion is a weighted sum of (i) the length of the path and (ii) the integral of the curvature along the path. The problem with given tangent vectors can completely be solved analytically. The problem with prescribed boundary curvatures requires an iterative numerical approach, and is not guaranteed to converge; satisfying the curvature constraints can make the path unpractically long.

Matlab code to calculate PH quintics can be obtained from the author.

Acknowledgments

The authors are Post-Doctoral Fellows of the Fund for Scientific Research-Flanders (F.W.O.) in Belgium. The support from the Inter-Universitary Attraction Pole at PMA, K.U.Leuven is gratefully acknowledged.

References

[1] G. Chirikjian and J. Burdick. The kinematics of hyper-redundant robot locomotion. *IEEE Trans. Rob. Automation*, 11(6):781–793, 1995.

[2] L. Euler. Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes, sive solutio problematis isoperimetrici latissimo sensu accepti, additamentum II (1744). In C. Carathéodory, editor, *Opera omnia*, pages LII–LV, 298–308. Fussli, Zürich, Switserland, 1952.

[3] R. T. Farouki. The elastic bending energy of Pythagorean-hodograph curves. *Comp. Aided Geom. Des.*, 13:227–241, 1996.

[4] R. T. Farouki and C. A. Neff. Hermite interpolation by Pythagorean hodograph quintics. *Mathematics of Computation*, 64(212):1589–1609, 1995.

[5] R. T. Farouki and T. Sakkalis. Pythagorean-hodographs. *IBM Journal of Research and Development*, 34:736–752, 1990.

[6] G. J. Hamlin and A. C. Sanderson. TETROBOT modular robotics: Prototype and experiments. In *Proc. Int. Conf. Intel. Robots and Systems*, pages 390–395, Osaka, Japan, 1996.

[7] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. AK Peters, Wellesley, MA, 1993.

[8] A. E. H. Love. *A treatise on the mathematical theory of elasticity*. Dover, New York, NY, 1944.

[9] D. J. Walton and D. S. Meek. A Pythagorean hodograph quintic spiral. *Computer-Aided Design*, 28(12):943–950, 1996.