Proceedings of the 36th
Conference on Decision & Control
San Diego, California USA • December 1997

Visual Path Following by Recursive Spline Updating

R. Frezza† S. Soatto‡ G. Picci†

† Dipartimento di Elettronica e Informatica, Università di Padova, Italy
‡ Washington University, St. Louis - MO and Università di Udine, Italy
frezza@dei.unipd.it

Abstract

In this paper we address the problem of tracking an unknown contour by commanding the translational and rotational velocities of a non-holonomic vehicle moving on a plane. A TV camera provides measurements of the distance of the contour from the x axis of the moving frame (fixed with the vehicle). We discuss an on-line path planning control strategy which leads the vehicle to land softly at a desired point ahead on the unknown curve and to stay on the curve once it has been reached. This strategy could be seen as an explicit model-based predictive control scheme.

1 Introduction

An autonomous vehicle moving on a plane must follow a contour described by some unknown curve Γ on the plane. The contour may describe the boundary of some unknown obstacle or one of the borders of an unknown road that the vehicle must follow. In most applications it is particularly important that the vehicle avoids overshooting and does not impact with the border. A TV camera provides measurements of the distance of the curve from the x axis of the moving frame (which we assume to coincide with the optical axis of the camera) at N points placed at fixed distances $\{x_1, \ldots, x_N\}$ from the origin of the vehicle-fixed coordinate system $\{x, y\}$ (the origin conventionally coincides with the optical center of the camera). See fig. 1.

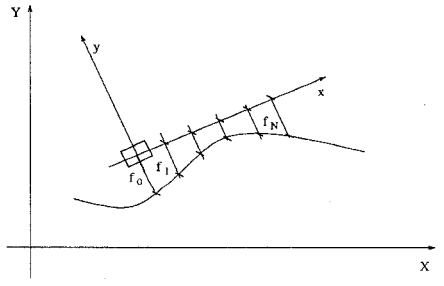

Figure 1: *Path–following of an unknown trajectory in a plane.*

This is a prototype problem of autonomous navigation and has been studied rather intensively in the past, but mostly in an ad hoc context. A solution based on an approximate spline-based stochastic state-space model of the dynamics of the unknown contour as seen by an observer sitting on the moving vehicle-fixed frame has been proposed in [4]. An Extended Kalman Filter built from this model serves to estimate the approximate contour on-line. The contour estimate is then tracked by applying a suitable state feedback control law.

This idea has been shown to work well for tracking unknown smooth contours in simulations. There are, however, fundamental control issues on the problem which were not fully understood. One basic issue is for example understanding the controllability of the moving contour model and describing explicitely the manifold of steerable variables of the system. This has been clarified recently by Frezza and Soatto [5] and Ma et al. [7]. This paper takes a further step in this direction and studies path planning and tracking control strategies which automatically adapt to the unknown shape of the contour and to the current position of the vehicle.

From the sensor data the on-board computer must reconstruct on-line a current local model of the chunk of curve seen on the image plane. The reconstruction should be continuously updated based on both the current measurements and on some a priori model of the contour.

Assume that the controller drives the vehicle by imposing the translational (v) and angular velocity (ω) of the camera-fixed frame. Then the on-board local model of the environment changes depending on the imposed motion. In particular the contour model permits to predict at each instant t a set of N future distances $\{\hat{f}_1, \ldots, \hat{f}_N\}$ of the vehicle from the contour corresponding to the chosen control actions. This framework is clearly reminiscent of "predictive control" [8], altough it does not necessarily rely on building an actual predictor.

2 The model

In this section we describe a simple model for a vehicle following an unknown road. The kinematic model is a single wheel that rolls without slipping. The road is described as a parametrized planar curve that can be represented locally as a continuous function.

0-7803-3970-8/97 $10.00 © 1997 IEEE

In this paper we restrict ourselves to consider a *planar* road, which we represent in an *inertial* reference frame $\{O, X, Y\}$ as a parametrized curve

$$\Gamma = \{(X(s), Y(s)) \in \mathbb{R}^2,\ s \in [0, S] \subset \mathbb{R}\} \quad (1)$$

where s is the curve parameter, for instance arc-length. We will assume that Γ is of class at least $\mathbf{C}^1$, i.e. that it is continuous along with its tangent. We model the vehicle as a wheel that is allowed to roll without slipping. This can be represented as a *moving frame* $\{o, x, y\}$ that rotates about the normal to the road-plane, but can *only translate along one independent direction.* Without loss of generality we let such a direction coincide with the $x-$axis, so that the instantaneous inertial velocity of the vehicle, in the moving frame, is represented by

$$\widehat{v} = \left(\begin{bmatrix} v \\ 0 \end{bmatrix}, \widehat{\omega} \right) \in se(2) \quad (2)$$

where $\widehat{\omega} = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}$. We assume to be able to control the longitudinal speed v and the steering angle ω. Such a restriction on the *velocity* of the wheel does not impose limitations on the *positions* it can reach. Constraints on the velocity of a system that cannot be integrated into constraints on position are called *non-holonomic*, and there is a vast literature on controllability, stabilization and path planning for such systems [9, 10].

In the moving frame, the road is represented as a contour $\Gamma(t)$ that changes over time under the action of the motion of the vehicle:

$$\Gamma(t) = \{(x(l,t), y(l,t)) \in \mathbb{R}^2,\ l \in [0, L] \subset \mathbb{R}\}. \quad (3)$$

In order to simplify the representation, we will assume that – locally at time t – $\Gamma(t)$ satisfies the conditions of the implicit function theorem, so that we can let $x(l,t) = l\ \forall\, t$ and $l \in [0, L]$. Consequently the contour can be represented as a function

$$y = \gamma(x, t)\ x \in [0, L]. \quad (4)$$

Such a representation breaks down, for instance, when the road winds-up or self-intersects or when the vehicle is oriented orthogonal to it.

2.1 Measurement process

When we drive a car we measure the *perspective projection* of the 3-D world onto a 2-D surface, such as our retina or the CCD surface of a video-camera, which can be modeled as a plane. We choose a *camera reference-frame* centered in the center of projection, with the $x-$axis orthogonal to the retinal line. For the sake of simplicity, we consider the optical center of the camera to coincide with the center of the wheel. What we can measure is then the perspective projection

$$\pi(\begin{bmatrix} x \\ y \end{bmatrix}) = \frac{y}{x} + n \quad (5)$$

up to a white, zero-mean Gaussian noise n. In practice it is computationally prohibitive to measure the projection of the whole contour

$$\{\pi\left(\begin{bmatrix} x \\ \gamma(x,t) \end{bmatrix}\right)\ \ \forall x \in [0, L]\} \quad (6)$$

while it is more convenient to process few regions of interest and localize the position of the projection of the contour at a few, controlled locations on the image [2]:

$$\{\pi\left(\begin{bmatrix} x \\ \gamma(x,t) \end{bmatrix}\right)\ \ x \in [x_1, \ldots, x_N]\}. \quad (7)$$

Note that the positions x_i can be considered control parameters that can therefore be chosen according to some optimality criterion.

In the remainder of the paper, we will assume that we can measure directly pairs of coordinates

$$(x_i, \gamma(x_i, t))\ \ i = 1 \ldots N$$

on the road-plane from the image coordinates.

2.2 Local evolution of the contour

The controls v, ω act on the vehicle's moving frame generating a vector field. A point which is stationary in inertial coordinates has coordinates on the moving frame (x, y) that evolve according to

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \widehat{\omega}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} v \\ 0 \end{bmatrix}. \quad (8)$$

In particular, points on the contour, that is pairs of the form $(x, y) = (x, \gamma(x,t))$, evolve according to

$$\frac{dy}{dt} = -\omega x = \frac{\partial \gamma}{\partial x}(\omega\gamma - v) + \frac{\partial \gamma}{\partial t}. \quad (9)$$

The above is a Riccati-type Partial Differential Equation that can be interpreted as governing the evolution of the surface $\{\gamma(x,t), x \in [0, L], t \in [0, t_f]$. If we couple the above dynamics with the measurements from the previous section, we end up with a distributed dynamical system:

$$\begin{cases} \frac{\partial \gamma}{\partial t} = -\omega x - \frac{\partial \gamma(x,t)}{\partial x}(\omega\gamma(x,t) - v) \\ y^i(t) = \gamma(x^i, t) + n^i(t) \quad i = 1 \ldots N. \end{cases} \quad (10)$$

Our goal is that of using the inputs v, ω to control the evolution of $\gamma(x,t)$ in order to drive the vehicle along the contour.

2.3 Local representation of the moving contour

Consider a local representation of the contour around the point $x = 0$ via the moments

$$\begin{cases} \xi_1(t) \doteq \gamma(0,t) \\ \xi_2(t) \doteq \frac{\partial \gamma}{\partial x}(0,t) \\ \xi_3(t) \doteq \frac{\partial^2 \gamma}{\partial x^2}(0,t) \\ \vdots \doteq \vdots \end{cases} \tag{11}$$

The first two moments ξ_1 and ξ_2 encode a notion of "relative pose" between the vehicle and the contour. In particular ξ_1 could be interpreted as an approximation of the distance from the vehicle to the contour, and ξ_2 as the relative orientation between the two (of course the approximation becomes more accurate as the vehicle gets closer to parallel to the tangent to the contour at $x = 0$). The terms ξ_k $k > 2$ encode curvature and higher terms, which characterize the "shape" of the contour, which is an invariant property in the Euclidean plane and therefore does not depend upon the particular choice of the reference frame.

We can then derive the dynamics of such a representation. This can be done easily using the above definitions and the dynamics of the contour in the viewer's reference (10):

$$\begin{cases} \dot{\xi}_1 = \xi_2(v - \omega\xi_1) \\ \dot{\xi}_2 = \xi_3(v - \omega\xi_1) - \omega(\xi_2^2 + 1) \\ \dot{\xi}_3 = \xi_4(v - \omega\xi_1) - 3\xi_2\xi_3\omega \\ \dot{\xi}_4 = \xi_5(v - \omega\xi_1) - 4\omega\xi_1\xi_4 - 3\omega\xi_3^2 \\ \vdots = \vdots \end{cases} \tag{12}$$

The chain of derivatives does not close in general. The above system tells us, however, that this happens when the contour Γ can be computed as the solution of a finite-dimensional differential equation with appropriate boundary values.

2.4 Conventional control obtained by feedback linearization

The relative pose variables ξ_1 and ξ_2 satisfy

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = G(\xi_1, \xi_2, \xi_3) \begin{bmatrix} \omega \\ v \end{bmatrix} \tag{13}$$

which can be solved for $[\omega, v]$ assigning any desired dynamics. Choosing, for example, to regulate ξ_1 and ξ_2 to zero exponentially by assigning

$$\begin{cases} \dot{\xi}_1 = \xi_2 \\ \dot{\xi}_2 = -\alpha\xi_1 - \beta\xi_2 \end{cases} \tag{14}$$

where α and β are positive real numbers, one obtains the following feedback control law

$$\begin{bmatrix} \omega \\ v \end{bmatrix} = \begin{bmatrix} (\xi_3 + \alpha\xi_1 + \beta\xi_2)/(1 + \xi_2^2) \\ (1 + \xi_1\omega) \end{bmatrix}. \tag{15}$$

Clearly, *once on it*, the contour is tracked exactly applying the control law $\omega = \xi_3 v$. This can be seen from (12) by imposing $\dot{\xi}_1 = \dot{\xi}_2 = 0$ with the initial conditions $\xi_1(0) = 0$, $\xi_2(0) = 0$ which imply that the vehicle is on the contour.

Of course the method described does not result in a practical control law since it demands exact knowledge of the shape of the contour at all times. The control law depends on ξ_3 which is the local curvature of the contour and, in general, it is not measured. Furthermore, in this kind of problems it is natural to assign the dynamics of the pose variables with respect to the arc-length (lateral dynamics) of the trajectory followed by the vehicle and then parametrize the arc-length with the desired velocity (longitudinal dynamics). In practice, to achieve robustness, it is also necessary to implement a "predictive" control action.

All these considerations motivate the introduction of a novel control strategy which will be described in the following sections.

It is first necessary to model the contour $\gamma(x,t)$ $\forall\, t$.

2.5 Model of the contour and the "Splinator"

In order to render the model (10) finite-dimensional, we can restrict the contour γ to belong to a particular class. For instance, Frezza and Picci [4] have proposed the set of cubic B-splines [1]:

$$\gamma(x,t) = \sum_{i=1}^{N+2} B_i(x) p_i(t) \quad x \in [0, L],\ t \geq 0 \tag{16}$$

where $B_i(x)$ are cubic polynomials with support on $[x_{i-1}, x_{i+3}]$. The coefficients $p_i(t)$ can be chosen either to *interpolate* or to *approximate* the nodes (x_i, y_i), $i = 1 \ldots N$. If we organize the measurements y_i into an N-dimensional vector $\mathbf{y}$, we can write the measurement equation in (10) as

$$\mathbf{y} = \mathcal{B}(x)\mathbf{p}(t) + \mathbf{n}(t) \tag{17}$$

where $\mathcal{B}$ is as an $N \times (N+2)$ tri-diagonal matrix and $\mathbf{p}$ an N-dimensional vector. Substituting equation (16) into (10), we obtain a model of the form

$$\begin{cases} \mathcal{B}(x)\dot{\mathbf{p}}(t) = \omega\mathbf{x} + \mathcal{B}_x(x)\mathbf{p}(t)(\omega\mathcal{B}(x)\mathbf{p}(t) - v) \\ \mathbf{y} = \mathcal{B}(x)\mathbf{p}(t) + \mathbf{n}(t). \end{cases} \tag{18}$$

Ideally, one would like to invert the matrix $\mathcal{B}$ and write a filter having as state the coefficients $\mathbf{p}(t)$. If

this was possible, the estimates of the coefficients $\hat{\mathbf{p}}(t)$ could be used to approximate the whole contour:

$$\gamma(x,t) \approx \sum_{i=1}^{N+2} B_i(x)\hat{p}_i(t). \tag{19}$$

However, in order to invert $\mathcal{B}$ one needs to specify two additional conditions. Frezza and Picci [4] have introduced a nonlinear, finite-dimensional filter that estimates the coefficients $\mathbf{p}(t)$ on-line using two additional stochastic conditions. Such a filter is called the "Splinator" in [4].

2.6 Planning a connecting contour

Suppose at time $t = 0$ the vehicle is at a given position and orientation relative to the contour, which results in initial conditions $\xi_i(0) = \xi_{i_0}\ \forall\ i$. We could design a piece of contour γ_n (*connecting contour*) that starts at the current position of the vehicle with the tangent pointing along the $x-$direction, and ends at a point $(x_c, \gamma(x_c, 0))$ on the contour with the same tangent. Overall the contour γ_n must satisfy the minimal set of conditions:

$$\begin{array}{cc} \gamma_n(0) = 0 & \frac{\partial \gamma_n}{\partial x}(0) = 0 \\ \gamma_n(x_c) = \gamma(x_c, 0) & \frac{\partial \gamma_n}{\partial x}(x_c) = \frac{\partial \gamma}{\partial x}(x_c, 0). \end{array} \tag{20}$$

The simplest curve that satisfies the four above conditions is a polynomial of degree 3. For instance, we can consider a *cubic Bezier curve*: $\alpha_0 P_{0,3}(x) + \ldots + \alpha_3 P_{3,3}(x)$ where $P_{i,j}(x) = \begin{pmatrix} j \\ i \end{pmatrix} x^i (x_c - x)^j$ are (unscaled) Bernstein polynomials of degree $j = 3$. After the coefficients α_i are chosen to satisfy the conditions (20), we have

$$\gamma_n(x) = \frac{\left(x_c \frac{\partial \gamma}{\partial x}(x_c, 0) - 3\gamma(x_c, 0)\right)(x - x_c)x^2 + \gamma(x_c, 0)x^3}{x_c^3}. \tag{21}$$

Then, if we consider the *composite contour*

$$\gamma_c(x, 0) = \begin{cases} \gamma_n(x)\ x \in [0, x_c) \\ \gamma(x, 0)\ x \in [x_c, L] \end{cases} \tag{22}$$

and we apply the control $\omega = \xi_3 v$ to track it exactly, we may hope to be able to converge onto the contour and then follow it with no error.

This strategy is bound to failure for several reasons. First the composite contour is not a feasible path for the vehicle since continuity of the second-derivative (and therefore of the control) is not guaranteed at x_c. Second, while the connecting contour is being planned, the vehicle may have moved, so that the initial conditions (20) are violated. More in general, such a control strategy does not tolerate measurement noise, errors due to computation delays, uncertainty in the model of the road etc.

2.7 The "Splinator" as a controller

Consider the evolution of the spline model (18), after having added two extra conditions:

$$\begin{cases} \gamma(0,t) \doteq \sum_i B^i(0)p^i(t) = 0 \\ \frac{\partial \gamma}{\partial x_0} = \sum_i \dot{B}^i(0)p^i(t) = 0. \end{cases} \tag{23}$$

If we augment $\mathcal{B}$ with the two extra rows resulting from the above constraints, we can invert the model (18) and end up with

$$\begin{cases} \dot{\mathbf{p}}(t) = \bar{\mathcal{B}}^{-1}(x)\omega\mathbf{x} + \bar{\mathcal{B}}^{-1}(x)\bar{\mathcal{B}}_x(x)\mathbf{p}(t)(\omega\bar{\mathcal{B}}(x)\mathbf{p}(t) - v) \\ \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} = \bar{\mathcal{B}}(x)\mathbf{p}(t) + \begin{bmatrix} \mathbf{n}(t) \\ 0 \end{bmatrix} \end{cases} \tag{24}$$

where $\bar{\mathcal{B}}$ denotes the augmented matrix $\mathcal{B}$. We now let $\hat{\mathbf{p}}(t)$ be the current estimate of the state of the above model obtained, for instance, through and Extended Kalman Filter. We call such a filter the "Splinator control". The reason why it is a controller follows by defining

$$\gamma_c(x,t) \doteq \bar{\mathcal{B}}(x)\hat{\mathbf{p}}(t) \tag{25}$$

to be the control spline, and then considering the control action

$$\omega(t) = v(t)\frac{\partial^2 \gamma_c}{\partial x^2}(0,t) = v(t)\bar{\mathcal{B}}_{xx}(0)\hat{\mathbf{p}}(t). \tag{26}$$

Observe that this control law is feasible, since the connecting contour is at least $\mathbf{C}^2$ at the origin and, by construction, it passes through the current position of the vehicle with the correct direction. The effective trajectory of the vehicle is the envelope of the contours γ_c, which is updated at each time-instant.

While the control action just defined depends upon the curvature of the road at the control points x_i, which are positive. Therefore, in this sense, the splinator implements a "predictive" control action.

In figure 2 we report the trajectory generated by a P-D linear controller, a Ljapunov-based controller and the controller based on the splinator trying to follow a trajectory with a cusp. It can be seen that the Splinator control exhibits some "predictive" action. We are in the process of performing thorough experimentation with the Splinator controller.

References

[1] C. De Boor, A Practical Guide to Splines, Springer–Verlag, 1978.

[2] E. D. Dickmanns and V. Graefe, Dynamic monocular machine vision, *Machine Vision and Application*, vol. 1, pp. 223-240, 1988.

[3] E. D. Dickmanns and V. Graefe, Applications of dynamic monocular machine vision, *Machine Vision and Application*, vol. 1, pp. 241-261, 1988.

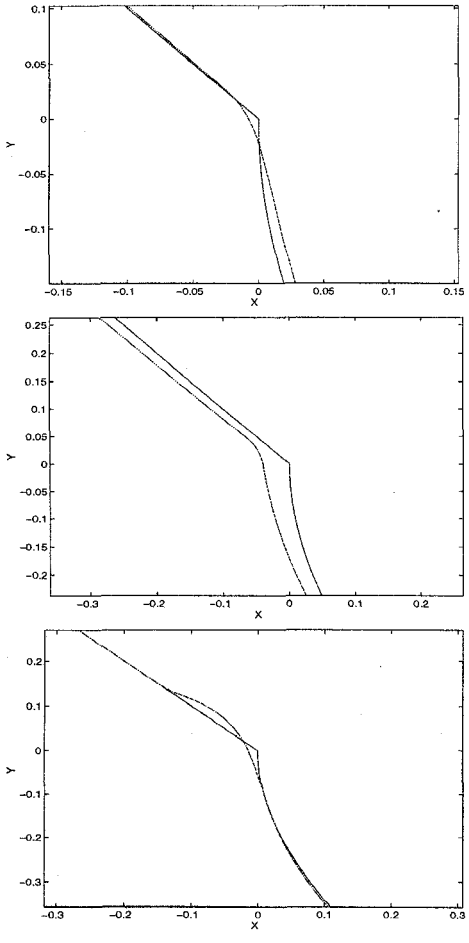

Figure 2: *Convergence of a linear H_∞ controller (top), a Ljapunov-based controller (center) and the spline-controller (bottom) following a curved road with a cusp. The spline-controller exhibits a "predictive" behavior.*

[4] R. Frezza and G. Picci, On line path following by recursive spline updating, In *Proceedings of the 34th IEEE Conderence on Decision and Control,* volume 4, pages 4047–4052, 1995.

[5] R. Frezza and S. Soatto, Autonomous navigation by controlling shape, Presented at the MTNS'96, St. Louis, June 1996.

[6] B.K. Ghosh and E.P. Loucks, A perspective theory for motion and shape estimation in machine vision, *SIAM J. on Control and Optimiz.*, june 1995.

[7] Yi Ma, J. Kosecka and S. Sastry, Vision guided navigation for nonholonomic mobile robot, unpublished technical report, 1997.

[8] M. Morari, Some Control Problems in the Process Industries, in *Essays on Control: Perspectives in the Theory and its Applications*, pp. 55–78, (survey paper persented at the second ECC Conference, Groningen, 1993), Birkhauser.

[9] R. Murray and S. Sastry, Nonholonomic motion planning: steering using sinusoids, *IEEE Transactions on Automatic Control*, 38 (5), pages 700–716, May, 1993.

[10] R. Murray, Z. Li and S. Sastry, *A Mathematical Introduction to Robotic Manipulation.* CRC Press Inc., 1994.