VISION GUIDED NAVIGATION FOR A NONHOLONOMIC MOBILE ROBOT

by

Yi Ma

Memorandum No. UCB/ERL M97/34

16 May 1997

VISION GUIDED NAVIGATION FOR A NONHOLONOMIC MOBILE ROBOT

by

Yi Ma

Memorandum No. UCB/ERL M97/34

16 May 1997

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

Visual servoing, *i.e.* the use of the vision sensor in feedback control has been of increasing interest. Work has been done by applications in autonomous driving, manipulation, mobile robot navigation and surveillance. However, theoretical aspects of the problem have not received much attention. The problem of estimation of necessary quantities from the vision measurements has been considered separately from the design of the control strategies. Instead of addressing the estimation and control problems separately, we attempt to characterize the types of control tasks which can be achieved using purely the quantities directly measurable in the image, bypassing the estimation phase. We consider a navigation task of a nonholonomic ground mobile base cast as a problem of tracking an arbitrarily shaped continuous ground curve. The control problem is formulated as one of controlling the shape of the curve in the image plane. We study the controllability of the system characterizing the dynamics of the image curve, and show that the shape of the image curve is controllable only up to its linear curvature parameters. We present stabilizing control laws for tracking piecewise analytic curves, and propose to track arbitrary curves by approximating them by piecewise linear curvature curves. Simulation results are given for these control schemes. The observability of the curve dynamics by using direct measurements from vision sensors as the outputs is studied and an Extended Kalman Filter is proposed to dynamically estimate the image quantities needed for the feedback controls from the actual noisy images.

Acknowledgment

First of all, I would like to express my gratitude to my research advisor, Professor Shankar Sastry, for initially giving me this very interesting research project. It is his support and guidance that make this project possible. Through the duration of the whole project, I have been constantly encouraged by his inspiration and enthusiasm. He read various versions of the manuscript with exceptional thoroughness and patience and made extremely valuable comments on everything from typos to issues of mathematical completeness and accuracy.

I am also very grateful to Dr. Jana Kosecka, my project partner, for providing me valuable references and helpful discussions. Her solid background and knowledge in computer vision are always a source of advice and guidance. She and Professor Sastry are the coauthors of a paper that we submitted to CDC'97 which this report is primarily based on.

I wish to thank Professor Jitendra Malik for serving as the second reader for my M.S. research report. It is his Computer Vision and Artificial Intelligence lectures brought me into this research field.

Two people in particular contributed enormously and deserve special mention. Joseph Yan and John Koo read the next-to-final version of the manuscript and provided good suggestions for final improvements and checked the grammar. I also would like to thank all the people in Cory 301 for providing such a positive as well as humorous atmosphere in the office and make the intensive life at Berkeley so enjoyable, Stanley Brown, William Huang, M. Cenk Cavusoglu and Ye Xia. I am also very grateful to the other people in Professor Sastry's group, Tomlin Claire and George Pappas, for always being helpful.

This research project has been partially supported by the California PATH program and by Army Research Office (ARO) under the MURI grant DAAH04-96-1-0341.

Finally, I would like to thank my parents and my brother for always more concerning about my health than supporting my study and research. This paper is dedicated to them.

Y.M.

Contents

List of Figures

1 Introduction

"Which world is true, the one you see or the one you think?"

Sensing of the environment and subsequent control are pertinent components of a successful navigation of an autonomous mobile agent. In spite of the fact that there has been an increased interest in the use of visual servoing in the control loop, the sensing and control problems have been often studied separately. The literature in computer vision has mainly concentrated on the process of estimating the necessary information about the state of the agent in the environment and the structure of the environment, e.g., [7] [9] [19] [22]. The control issues are often not relevant or are addressed separately. On the other hand control approaches typically assume the full specification of the environment and the task as well as the availability of the state estimate of the agent.

The dynamic vision approach proposed by Dickmanns, Mysliwetz and Graefe [2] [3] [4] makes the connection between the estimation and control tighter by setting up a dynamic model of the evolution of the curvature of the road in a driving application. However the curvature estimates are used only for estimation of the state of the vehicle with respect to the road frame where the control objective is formulated. Control for steering along a curved road directly using the measurement of the projection of the road tangent and it's optical flow has been previously considered by Raviv and Herman [16]. Stability and robustness issues have not been addressed, and no statements have been made as to what extent these cues are sufficient for general road scenarios. A visual servoing framework proposed in [5] [17] by Espiau *et al* addresses the control issues directly in the image plane and outlines the dynamics of certain simple geometric primitives. Further extensions of this approach for nonholonomic mobile platforms has been made by Pissard-Gibollet and Rives [15]. Generalization of the curve tracking and estimation problem outlined in Dickmanns for arbitrarily shaped curves addressing both the estimation of the shape parameters as well as control has been explored in [6] by Frezza and Picci. They used an approximation of an arbitrary curve by a spline, and proposed a scheme for recursive estimation of shape parameters of the curve and designed control laws for tracking the curve.

For a theoretical treatment of the problem, the understanding of the dynamics of the image of an arbitrary ground curve is crucial. In a computer vision seminar given at Berkeley in October 1996, Dr. Stefano Soatto [20] formulated the problem of tracking as that of controlling shape of the ground curve in the image plane. In spite of the fact that the system characterizing the image curve seems to be infinite-dimensional, we show that for linear curvature curves the system is of finite dimension. Since the control problem is formulated as one of controlling the image curve dynamics, we prove that the controllabilty distribution has dimension 3 and show that the system characterizing the image curve dynamics is fully controllable only up to the linear curvature term regardless of the kinematics of the mobile robot base.

The controllability results indicate that the parameters characterizing the images of linear curvature curves (to be defined in Section 2.2.3) can be controlled using the driving and steering inputs. We show that the dynamics of the images of linear curvature curves can be transformed to a canonical chained-form, which already has existing point-to-point steering control scheme in Murray and Sastry [13] [14].

We formulate the task of tracking ground curves as a problem of controlling the image curves in the image plane. We design stabilizing feedback control laws for tracking any piecewise analytic curves (for general treatments of stabilizing trajectory tracking control of nonlinear systems, one could refer to, e.g., [8] [23]). We also propose to approximate general curves by piecewise linear curvature curves. We present how to compute the image parameters for such approximating virtual curves so as to obtain the appropriate controls to track them. Simulation results are given for these control schemes.

We also study the observability of the curve dynamics from the direct measurements of the vision sensors. Based on the sensor models, an extended Kalman filter is proposed to dynamically estimate on-line the image quantities needed for the feedback control from the actual noisy images. We thus obtain a complete closed-loop vision-guided navigation system for non-holonomic systems.

Report Outline

Section 2 derives the dynamics of image curves, *i.e.* how the shape of the image of a ground curve evolves in the image plane. Section 3 studies controllability issues for the dynamic systems obtained in Section 2. Section 4 shows how to formulate specific control tasks for the mobile robot in the image plane. Corresponding control designs and their simulation results are also presented in the same section. Section 5 develops an extended Kalman filter to estimate on-line the image quantities needed for the feedback control. Observability issues of the sensor models are also presented. Some simulations and animations are done for the entire closed-loop vision-guided navigation system in Section 6. Section 7 concludes the report with some discussions and directions of future work.

2 Curve Dynamics

"The introduction of numbers as coordinates...is an act of violence..."
— H. Weyl, *Philosophy of Mathematics*
and Natural Science, 1949

We derive equations of motion for the image curve under motions of a ground-based mobile robot. We begin with a unicycle model for the mobile robot and consider the general cases later.

2.1 Mobile Robot Kinematics

Consider the case where $g_{fm}(t) \in SE(3)$ is a one parameter curve (parameterized by time) representing a trajectory of a unicycle: more specifically, the rigid body motion of the *mobile frame* F_m attached to the unicycle, relative to a fixed *spatial frame* F_f, as shown in the Figure 1.

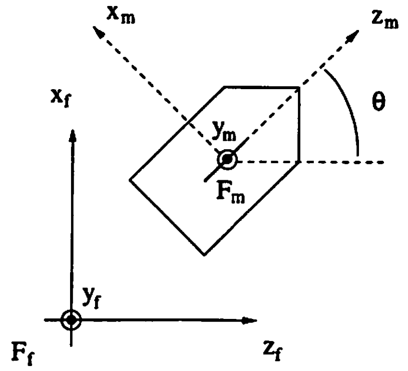

Figure 1: Model of the unicycle mobile robot.

Let $p_{fm}(t) \in R^3$ be the position vector of the origin of frame F_m from the origin of frame F_f. The *homogeneous* coordinates[1] of a point q attached to the unicycle frame F_m in spatial

[1] A systematical introduction of using homogeneous coordinates to describe rigid body motion is given in [13].

frame F_f are given by

$$q_f(t) = g_{fm}(t)q_m(t) = \begin{pmatrix} R_{fm}(t) & p_{fm}(t) \\ 0 & 1 \end{pmatrix} q_m(t) \tag{1}$$

where $q_f, q_m \in R^4$ are the homogeneous coordinates of the point q relative to frames F_f and F_m respectively, and

$$R_{fm}(t) = \begin{pmatrix} \cos\theta(t) & 0 & \sin\theta(t) \\ 0 & 1 & 0 \\ -\sin\theta(t) & 0 & \cos\theta(t) \end{pmatrix} \tag{2}$$

represents the rotation of the mobile frame F_m with respect to F_f and the rotation angle θ is defined in the counter-clockwise sense about the y_f-axis, as shown in Figure 1. For the unicycle kenimatics, $\theta(t)$ and $p_{fm}(t)$ satisfy:

$$\begin{aligned} \dot{p}_{fm} &= \begin{pmatrix} v\sin\theta \\ 0 \\ v\cos\theta \end{pmatrix} \\ \dot{\theta} &= \omega \end{aligned} \tag{3}$$

where the steering input ω controls the angular velocity, $\dot{\theta}$; the driving input v controls the linear velocity along the direction of the wheel. From (1) (2) and (3), we thus have

$$\dot{g}_{fm} = \begin{pmatrix} \dot{R}_{fm}(t) & \dot{p}_{fm}(t) \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} -\omega\sin\theta & 0 & \omega\cos\theta & v\sin\theta \\ 0 & 0 & 0 & 0 \\ -\omega\cos\theta & 0 & -\omega\sin\theta & v\cos\theta \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{4}$$

We then express the velocity of the point q in the (instantaneous) mobile frame F_m:

$$\dot{q}_m = g_{fm}^{-1}\dot{g}_{fm}q_m = \begin{pmatrix} 0 & 0 & \omega & 0 \\ 0 & 0 & 0 & 0 \\ -\omega & 0 & 0 & v \\ 0 & 0 & 0 & 0 \end{pmatrix} q_m. \tag{5}$$

Now, suppose a monocular camera mounted on the mobile robot which is facing downward with a tilt angle $\phi > 0$ and the camera is elevated above the ground plane by distance d, as shown in Figure 2. The *camera coordinate frame* F_c chosen for the camera is such that the z-axis of F_c is the optical axis of the camera, the x-axis of F_c and x_m-axis of F_m coincide, and the optical center of the camera coincides with the origins of both F_m and F_c.[2]

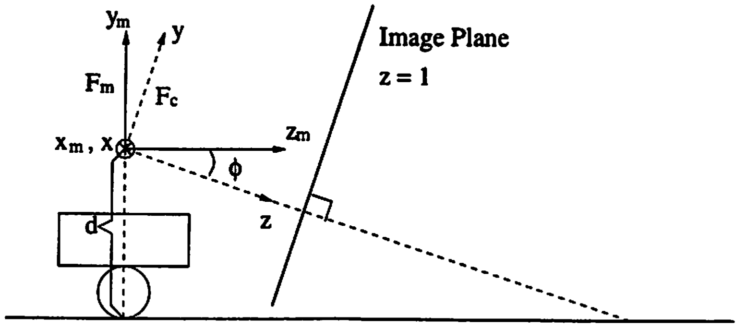

Figure 2: The side-view of the unicycle mobile robot with a camera facing downward with a tilt angle $\phi > 0$.

Then, through the *adjoint transformation* which transforms the twists from the mobile frame F_m to the camera frame F_c, the velocity of a point q attached to the camera frame F_c is given in the (instantaneous) camera frame by

$$\dot{q}_c = g_{mc}^{-1} g_{fm}^{-1} \dot{g}_{fm} g_{mc} q_c = \begin{pmatrix} 0 & \omega \sin\phi & \omega\cos\phi & 0 \\ -\omega\sin\phi & 0 & 0 & v\sin\phi \\ -\omega\cos\phi & 0 & 0 & v\cos\phi \\ 0 & 0 & 0 & 0 \end{pmatrix} q_c \tag{6}$$

where $g_{mc} = \begin{pmatrix} R_\phi & 0 \\ 0 & 1 \end{pmatrix}$ and $R_\phi \in SO(3)$ represents the rotation of the camera frame F_c relative to the mobile frame F_m (by the tilt angle ϕ).

Now extracting the individual coordinates of q_c in the camera frame F_c and (6) can be rewritten as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} 0 \\ \sin\phi \\ \cos\phi \end{pmatrix} v + \begin{pmatrix} y\sin\phi + z\cos\phi \\ -x\sin\phi \\ -x\cos\phi \end{pmatrix} \omega. \tag{7}$$

[2] Without loss of generality, we assume the camera is in such a position that such a choice of coordinate frame is possible.

For a unit focal length camera, the image plane is $z = 1$ in the camera coordinate frame, as shown in Figure 2.

2.2 Image Curve Dynamics Analysis

In this section, we consider a planar curve Γ on the ground, and study how the shape of the image of the curve Γ evolves under the motion of the mobile robot. For the rest of this paper, we make the following assumptions:

Assumption 1 *The ground curve Γ is analytic.*

Assumption 1 means Γ can be locally expressed by its convergent Taylor series expansion.

Assumption 2 *The ground curve Γ is such that it can be parameterized by y in the camera coordinate frame F_c.*

Assumption 2 guarantees that the task of tracking the curve Γ can be solved using a smooth control law. For example, if the curve is orthogonal to the direction of the heading of the mobile robot, such as the curve Γ_2 shown in Figure 3, it can not be parameterized by y.

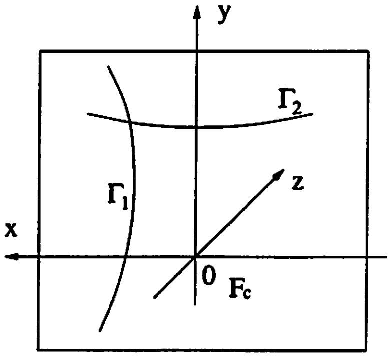

Figure 3: An example showing that a ground curve Γ_2 cannot be parameterized by y, while the curve Γ_1 can be.

Obviously, in this case, if the mobile robot needs to track the curve Γ_2, it has to make a decision as to which direction to track the curve: turning right or turning left. This decision cannot be made using smooth control laws.

2.2.1 Relations between Orthographic and Perspective Projections

According to Assumption 2, at any time t, the curve Γ can be expressed in the camera coordinate frame as $(\gamma_x(y,t), y, \gamma_z(y,t))^T$. Since Γ is a planar curve on the ground, $\gamma_z(y,t)$ is given by

$$\gamma_z(y,t) \quad = \quad \frac{d + y\cos\phi}{\sin\phi}. \tag{8}$$

which is a function of only y. Thus only $\gamma_x(y,t)$ changes with time and determines the dynamics of the ground curve. In order to determine the dynamics of the image curve we consider both *orthographic* and *perspective* projection cases and show that under certain conditions they are equivalent.

The orthographic projection image curve of Γ in the image plane $z = 1$ given by $(\gamma_x(y,t), y, 1)^T$ is denoted by $\tilde{\Gamma}$, as shown in Figure 4.

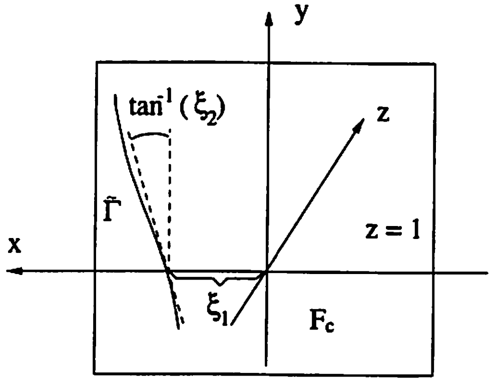

Figure 4: The orthographic projection of a ground curve on the $z = 1$ plane. Here $\xi_1 = \gamma_x$ and $\xi_2 = \frac{\partial \gamma_x}{\partial y}$.

$\tilde{\Gamma}$ is not yet the real image curve which is usually meant by the perspective projection image of the curve Γ on the image plane $z = 1$. The perspective projection image curve, denoted by Λ, is given in the image plane coordinates by

$$\begin{aligned} X(y,t) &= \frac{\gamma_x}{\gamma_z} = \frac{\gamma_x(y,t)\sin\phi}{d+y\cos\phi} \\ Y(y,t) &= \frac{y}{\gamma_z} = \frac{y\sin\phi}{d+y\cos\phi} \end{aligned}. \tag{9}$$

Notice, in equation (9), that $Y(y,t)$ is a function of only y and the derivative of $Y(y,t)$

with respect to y is

$$\frac{\partial Y(y,t)}{\partial y} = \frac{d \sin\phi}{(d + y\cos\phi)^2} > 0 \tag{10}$$

so long as $\phi > 0$ and $y \neq -d/\cos\phi$. Using the inverse function theorem, locally, the image curve Λ can be re-parameterized by Y when $\frac{\partial Y(y,t)}{\partial y} \neq 0$. Λ can then be represented by $(\lambda_X(Y,t), Y)^T$ in the image plane coordinates, where the function $\lambda_X(Y,t)$ can be directly measured. However, since, as we will soon see, for the given ground curve Γ, it is easier to get an explicit expression for the dynamics of its orthographic image $\tilde{\Gamma}$ than the perspective projection image Λ. Thus, it will be helpful to find out the relations between these two image curves $\tilde{\Gamma}$ and Λ, *i.e.* relations between the two functions γ_x and λ_X.

First, let us simplify the notation. Define

$$\begin{aligned} \xi_{i+1} &\equiv \frac{\partial^i \gamma_x(y,t)}{\partial y^i} \qquad i = 0,1,2,\ldots \\ \zeta_{i+1} &\equiv \frac{\partial^i \lambda_X(Y,t)}{\partial Y^i} \qquad i = 0,1,2 \end{aligned} \tag{11}$$

and

$$\begin{aligned} \xi^i &\equiv (\xi_1, \xi_2, \ldots, \xi_i)^T \in R^i \qquad \xi \equiv \xi^\infty \\ \zeta^i &\equiv (\zeta_1, \zeta_2, \ldots, \zeta_i)^T \in R^i \qquad \zeta \equiv \zeta^\infty. \end{aligned} \tag{12}$$

If $\gamma_x(y,t)$ is an analytic function of y, $\gamma_x(y,t)$ is completely determined by the vector ξ evaluated at any y; similarly for $\lambda_X(Y,t)$. Thus, the relations between $\tilde{\Gamma}$ and Λ are given by the relations between ξ and ζ for the case of analytic curves.

Lemma 1 (Equivalence of ξ, ζ Coordinates) *Consider the orthographic projection image curve $\tilde{\Gamma} = (\gamma_x(y,t), y, 1)^T$ and the perspective projection image curve $\Lambda = (\lambda_X(Y,t), Y)^T$, with ξ and ζ defined in (11) and (12). Assume that the tilt angle $\phi > 0$ and $y \neq -d/\cos\phi$. Then for any fixed y,*

$$\zeta^n = A_n(y)\xi^n \qquad \forall n \in \mathcal{N} \tag{13}$$

where $A_n(y) \in R^{n\times n}$ is a $n \times n$ nonsingular lower triangular matrix.

Proof We prove this lemma by using mathematical induction. For $n = 1$, from (9), $\zeta^1 = \frac{\sin\phi}{d+y\cos\phi}\xi^1$, so the lemma is true for $n = 1$. Now suppose that the lemma is true for all $n \leq k$, *i.e.*

$$\zeta^n = A_n(y)\xi^n \qquad n = 1, 2, \ldots, k \tag{14}$$

where all $A_n(y)$ is a nonsingular lower triangular matrix. Clearly, in order to prove that for $n = k + 1$ the lemma is still true, it suffices to prove that ζ_{k+1} is a linear combination of ξ^{k+1}, *i.e.*

$$\zeta_{k+1} = \sum_{i=1}^{k+1} \beta_i(y)\xi_i \tag{15}$$

and in order for $A_{k+1}(y)$ to be nonsingular, $\beta_{k+1}(y)$ needs to be non-zero. Differentiating (14) with respect to y, we have

$$\begin{aligned} \frac{\partial\zeta^k}{\partial Y}\frac{\partial Y(y,t)}{\partial y} &= A_k'(y)\xi^k + A_k(y)\frac{\partial\xi^k}{\partial y} \\ \Rightarrow \quad \frac{\partial\zeta^k}{\partial Y} &= \frac{A_k'(y)}{\frac{\partial Y(y,t)}{\partial y}}\xi^k + \frac{A_k(y)}{\frac{\partial Y(y,t)}{\partial y}}\frac{\partial\xi^k}{\partial y}. \end{aligned} \tag{16}$$

where the last entry of the column vector $\frac{\partial\zeta^k}{\partial Y}$ is ζ_{k+1} and

$$\frac{\partial\xi^k}{\partial y} = (\xi_2, \xi_3, \ldots, \xi_{k+1})^T. \tag{17}$$

Therefore, according (16), ζ_{k+1} is a linear combination of ξ^{k+1} and, since $A_k(y)$ is a $k \times k$ nonsingular lower triangular matrix, $A_k(y)_{kk} \neq 0$,[3] the coefficient $\beta_{k+1}(y) = \frac{A_k(y)_{kk}}{\frac{\partial Y(y,t)}{\partial y}}$ is non-zero. □

Example We calculate the matrix $A_4(y) \in R^{4\times 4}$ to be

$$\zeta^4 = \begin{pmatrix} \frac{\sin\phi}{d+y\cos\phi} & 0 & 0 & 0 \\ -\frac{\cos\phi}{d} & \frac{d+y\cos\phi}{d} & 0 & 0 \\ 0 & 0 & \frac{(d+y\cos\phi)^3}{d^2\sin\phi} & 0 \\ 0 & 0 & 3\frac{(d+y\cos\phi)^4\cos\phi}{d^3\sin^2\phi} & \frac{(d+y\cos\phi)^5}{d^3\sin^2\phi} \end{pmatrix}\xi^4. \tag{18}$$

Lemma 1 tells us that under certain conditions, the dynamics of the system ξ for the orthographic projection image curve and that of ζ for the perspective projection image

[3] $A_k(y)_{kk}$ is the (k,k) entry of the matrix $A_k(y)$.

curve are algebraically equivalent. We may obtain either one of them from the other. ζ are quantities that we can directly measure from the perspective projection image Λ. Our ultimate goal is to design feedback control laws exclusively using these image quantities. However, as we will soon see, it is much easier to analyze the curve's dynamics in terms of ξ, the quantities in the orthographic projection image. It also turns out to be easier to design feedback control laws in terms of ξ. For these reasons, in the following sections, we choose system ξ (*i.e.* the orthographic projection image) to study our problem and design control laws since it simplifies the notation.

2.2.2 Dynamics of General Analytic Curves

While the mobile robot moves, a point attached to the spatial frame F_f moves in the opposite direction relative to the camera frame F_c. Thus, from (7), for points on the ground curve $\Gamma = (\gamma_x(y,t), y, \gamma_z(y))^T$, we have

$$\dot{\gamma}_x(y,t) = -(y \sin\phi + \gamma_z \cos\phi)\omega. \tag{19}$$

Also, by chain rule

$$\begin{aligned} \dot{\gamma}_x(y,t) &= \frac{\partial \gamma_x}{\partial t} + \frac{\partial \gamma_x}{\partial y}\dot{y} \\ &= \frac{\partial \gamma_x}{\partial t} + \frac{\partial \gamma_x}{\partial y}(-(v \sin\phi - \gamma_x \omega \sin\phi)). \end{aligned} \tag{20}$$

The shape of the orthographic projection image $\tilde{\Gamma} = (\gamma_x(y,t), y, 1)^T$ then evolves in the image plane $z = 1$ according to the following *Riccati-type* partial differential equation [4]

$$\frac{\partial \gamma_x}{\partial t} = -(y \sin\phi + \gamma_z \cos\phi)\omega + \frac{\partial \gamma_x}{\partial y}(v \sin\phi - \gamma_x \omega \sin\phi). \tag{21}$$

Using the notation ξ from (11) and the expression (8) for γ_z, this partial differential equation can be transformed to an infinite-dimensional dynamic system ξ through differentiating

[4] This equation is called a Riccati-type PDE since it generalizes the classical well-known Riccati equation for the motion of a homogeneous straight line under rotation around the origin [6] [7].

equation (21) with respect to y repeatedly:

$$\begin{pmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \vdots \\ \dot{\xi}_i \\ \vdots \end{pmatrix} = - \begin{pmatrix} \xi_1\xi_2 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_1\xi_3 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_1\xi_4 \sin\phi + 3\xi_2\xi_3 \sin\phi \\ \vdots \\ \xi_1\xi_{i+1} \sin\phi + g_i(\xi_1, \ldots, \xi_i) \\ \vdots \end{pmatrix} \omega + \begin{pmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \\ \vdots \\ \xi_{i+1} \sin\phi \\ \vdots \end{pmatrix} v \tag{22}$$

where $g_i(\xi_1, \ldots, \xi_i)$ are appropriate functions (polynomials) of only $\xi_1, \ldots, \xi_i$.

In the general case, the system (22) is an infinite-dimensional system .

2.2.3 Dynamics of Linear Curvature Curves

In this section, we look at a special case: the ground planar curve Γ is a *linear curvature curve* (defined below). Its image dynamics ξ can then be reduced to a three-dimensional system, which turns out to be controllable (as will soon be shown in the following sections).

Definition 1 *We say that a planar curve has* **linear curvature** *if the derivative of its curvature $k(s)$ with respect to its arc-length parameter s is a non-zero constant, i.e. $k'(s) \equiv c \neq 0$. If $k'(s) \equiv 0$, the curve is a* **constant curvature** *curve.*

Note that, according to this definition, both straight lines and circles are constant curvature curves, but not linear curvature curves. Constant curvature curves can be regarded as degenerate cases of linear curvature curves. For linear curvature curves, we have

Lemma 2 *For a ground curve Γ of linear curvature, i.e. $k'(s) \equiv c \neq 0$, for any $i \geq 4$, ξ_i can be expressed as a function of only ξ_1, ξ_2, and ξ_3.*

Proof Consider the ground curve $\Gamma = (\gamma_x(y,t), y, \gamma_z(y,t))^T$ where $\gamma_z(y,t)$ is given in (8). For the arc-length parameter s and the curvature k, the following relationships hold

$$s'(y) = \sqrt{\left(\frac{\partial \gamma_x}{\partial y}\right)^2 + 1 + \left(\frac{\partial \gamma_z}{\partial y}\right)^2} \tag{23}$$

$$k(y) = \frac{\|\Gamma'(y) \times \Gamma''(y)\|_2}{s'(y)^3} = \frac{a\frac{\partial^2 \gamma_x}{\partial y^2}}{\left(\sqrt{a^2 + (\frac{\partial \gamma_x}{\partial y})^2}\right)^3} \tag{24}$$

where a is defined as $a \equiv \sqrt{1 + \cot^2 \phi} = (\sin \phi)^{-1}$. Thus the derivative of the curvature k with respect to the arc-length parameter s is given by

$$k'(s) = \frac{k'(y)}{s'(y)} = a\frac{\frac{\partial^3 \gamma_x}{\partial y^3}(a^2 + (\frac{\partial \gamma_x}{\partial y})^2) - 3\frac{\partial \gamma_x}{\partial y}\left(\frac{\partial^2 \gamma_x}{\partial y^2}\right)^2}{\left(a^2 + (\frac{\partial \gamma_x}{\partial y})^2\right)^3} \equiv c. \tag{25}$$

Using the definition of ξ_i, from (25) ξ_4 can be expressed by

$$\xi_4 = \frac{c(a^2 + \xi_2^2)^3/a + 3\xi_2\xi_3^2}{a^2 + \xi_2^2}. \tag{26}$$

Therefore, ξ_4 is a function of only ξ_1, ξ_2, and ξ_3. According to the definition of ξ_i, it follows that, for all $i > 4$, ξ_i are functions of only ξ_1, ξ_2,and ξ_3 as well. □

Using Lemma 2, for a ground linear curvature curve Γ, the dynamics of its orthographic projection image $\tilde{\Gamma}$, *i.e.* system (22) for ξ, can then be simplified to be the following three-dimensional system ξ^3:

$$\begin{pmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \end{pmatrix} = -\begin{pmatrix} \xi_2\xi_1 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_3\xi_1 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_4\xi_1 \sin\phi + 3\xi_2\xi_3 \sin\phi \end{pmatrix}\omega + \begin{pmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \end{pmatrix} v \tag{27}$$

where ξ_4 is given by (26).

Combining Lemma 1 and Lemma 2, we have the following remark

Remark 1 *For a ground curve of linear curvature, the dynamics of ζ for the perspective projection image of the curve are completely determined by three independent states $\zeta_1, \zeta_2, \zeta_3$, or equivalently, for i ≥ 4, ζ_i is a function of only ζ_1, ζ_2, and ζ_3. The two systems $\zeta^3 = (\zeta_1, \zeta_2, \zeta_3)^T$ and $\xi^3 = (\xi_1, \xi_2, \xi_3)^T$ are equivalent and related by equation (18). This implies, for instance, that these two systems have the same controllability.*

Comments In the case that Γ is a constant curvature curve, *i.e.* $k'(s) \equiv 0$, one can show that ξ_3 is actually a function of only ξ_1, ξ_2, so for all $\xi_i, i > 3$ are functions of only ξ_1, ξ_2. There are then only two independent states ξ_1, ξ_2 for the dynamics of system ξ.

Linear curvature is an *intrinsic property* (which is preserved under Euclidean motions *i.e.* $SE(2)$) of planar curves. Thus, the expression (26) always holds under robot's planar motions. However, some other seemingly natural and simple assumptions that people are prone to take for the ground curve (so as to simplify the problem) might fail to be preserved under the robot's motions. For example, if, in order to simplify (22), one assumes $\xi_i = 0$ for $i \geq 4$, *i.e.* $\gamma_x(y,t)$ is of the form

$$\gamma_x(y,t) = \xi_1(y_0,t) + \xi_2(y_0,t)(y-y_0) + \frac{1}{2}\xi_3(y_0,t)(y-y_0)^2 \tag{28}$$

This property is not preserved under rotations. More generally, it is actually not an intrinsic property for a planar curve that its Taylor series expansion has a finite number of terms. Therefore, one cannot simplify system (22) to a finite-dimensional system by assuming that the curve's Taylor series expansion is finite (which might be the case only at special positions).[5]

[5] Essentially, it only "simplifies" the initial conditions of the system (22), not the system dimensions.

3 Controllability Issues

"The paradox is now fully established that the utmost abstractions are the true weapons with which to control our thought of concrete fact."
— A. N. Whitehead, *Science and the Modern World, 1925*

We are interested in being able to control the shape of the image curves. From the above discussions, this problem is equivalent to the problem of controlling system ξ (22) in the unicycle case. For linear curvature curves, the infinite-dimensional system ξ is reduced to the three-dimensional system ξ^3 (27). In this section, we look at controllability of such systems. If the systems characterizing the curve Γ are controllable, we essentially means that given our control inputs we can steer the mobile base in order to achieve desired position and shape of the curve in the image plane. Controllability of system (27) is directly checked in Section 3.1; controllability of system (22) is obtained through studying the controllability for arbitrary ground-based mobile robots in Section 3.2.

Note that ξ and ζ are still functions of y (or Y). They need to be evaluated at a fixed y (or Y). Since the ground curve Γ is analytic, it does not matter at which specific y to evaluate them (as long as the relation between ξ and ζ are well-defined according to Lemma 1).[6] However, evaluating ξ or ζ at some special y might simplify the formulation of some control tasks.

For example, suppose a mobile robot is to track the given ground curve Γ. According to Figure 5, let A' be the orthographic projection image of the point A where the wheel of the mobile robot touches the ground. Obviously, the coordinates of A' are given by $(0, -d\cos\phi, 1)^T$. When the mobile robot is perfectly tracking the given curve Γ, *i.e.* the wheel keeps touching the curve, the orthographic projection image $\tilde{\Gamma} = (\gamma_x(y,t), y, 1)^T$ of the curve Γ should satisfy

$$\gamma_x(y,t)|_{y=-d\cos\phi} \equiv 0. \tag{29}$$

Furthermore, the tangent to the curve Γ at $y = -d\cos\phi$ should be in the same direction of

[6] For analytic curves, there is a one-to-one correspondence between the two sets of coefficients of the Taylor series expanded at two different points.

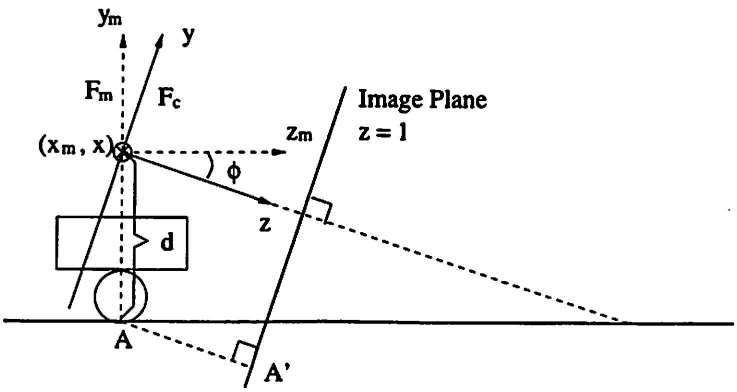

Figure 5: The orthographic projection image of the point A where the wheel touches the ground.

the mobile robot. This requires

$$\frac{\partial \gamma_x(y,t)}{\partial y}|_{y=-d\cos\phi} \equiv 0. \tag{30}$$

Thus, if ξ is evaluated at $y = -d\cos\phi$, the task of tracking Γ becomes a control problem of steering both ξ_1 and ξ_2 to 0 for the system (22). *For these reasons, from now on, we always evaluate ξ (or ζ) at $y = -d\cos\phi$ unless explicitly stated.*

3.1 Controllability in the Linear Curvature Curve Case

If the given ground curve Γ is a linear curvature curve, the dynamics of its image is given by (27).

Theorem 1 *Consider the system (27)*

$$\dot{\xi}^3 = f_1\omega + f_2 v \tag{31}$$

where the vector fields (f_1, f_2) are

$$f_1 = -\begin{pmatrix} \xi_1\xi_2\sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_1\xi_3\sin\phi + \xi_2^2\sin\phi + \frac{1}{\sin\phi} \\ \xi_1\xi_4\sin\phi + 3\xi_2\xi_3\sin\phi \end{pmatrix} \quad f_2 = \begin{pmatrix} \xi_2\sin\phi \\ \xi_3\sin\phi \\ \xi_4\sin\phi \end{pmatrix} \tag{32}$$

and $\xi_4 = \frac{c(a^2+\xi_2^2)^3/a+3\xi_2\xi_3^2}{a^2+\xi_2^2}$. If $\phi \neq 0$, and $y = -d\cos\phi$, then the distribution $\Delta_{\mathcal{L}}$ spanned by the Lie algebra $\mathcal{L}(f_1, f_2)$ generated by (f_1, f_2) is of rank 3 when $c \neq 0$, and is of rank 2 when $c = 0$.

Proof Directly calculate the Lie bracket $[f_1, f_2]$

$$[f_1, f_2] = (-1, 0, 0)^T. \tag{33}$$

The determinant of matrix $(f_1, f_2, [f_1, f_2])$ is

$$det(f_1, f_2, [f_1, f_2]) = -c(a^2 + \xi_2^2)^3/a^3. \tag{34}$$

Therefore, the distribution $\Delta_{\mathcal{L}}$ spanned by $\mathcal{L}(f_1, f_2)$ is of rank 3 if $c \neq 0$, and of rank 2 if $c = 0$. □

Comments Since $\Delta_{\mathcal{L}}$ is of full rank at all points, it is involutive as a distribution. Chow's Theorem [13] states that the reachable space of system (27) for ξ^3 is of 3 dimensions when $c \neq 0$, and 2 when $c = 0$. This makes sense since, when $c = 0$, *i.e.* the case of constant curvature curves, there are only two independent parameters, ξ_1 and ξ_2, needed to describe the image curves, the reachable space of such system has to be of at most 2 dimensions.

In the next section, we study the general case and show that, for an *arbitrary* analytic ground curve under the motion of an *arbitrary* mobile robot, the dimension of the reachable space of the system ξ (and ζ) is at most 3. This highlights the importance of Theorem 1 from two aspects: the controllability is at most 3 for controlling the shape of the image of an arbitrary curve, which means linear curvature curves already capture all the features $(\xi_1, \xi_2, \xi_3)^T$ that may be totally controlled; on the other hand, any other nonholonomic mobile robot cannot do essentially "better" in controlling the shape of the image curve than the unicycle.

3.2 General Case

In this section, we study how the image of an arbitrary (analytic) ground curve changes under the motion of an arbitrary ground-based mobile robot. Since the dynamics of the mobile robot are now assumed to be general, one can no longer get an explicit expression of the dynamics of the system ξ as we did in the unicycle case. However, the set of all possible motions of any ground-based mobile robot, regardless of the dynamics, turns out to be in a same space: *i.e.* the group of planar rigid body motions $SE(2)$. Denote this group by $\mathcal{G}$.

There exists an explicit representation for $\mathcal{G}$ in the camera coordinate frame F_c. Choose $n = (0, \cos\phi, -\sin\phi)^T$, which, in the camera coordinate frame F_c, is an orthogonormal vector to the ground, and choose vectors e_1, e_2 such that (n, e_1, e_2) forms an orthonormal basis for the camera coordinate frame. Thus, in the camera coordinate frame, the matrix M which describes the rotation of the mobile robot about the y_m-axis is given by

$$M = (n, e_1, e_2) \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} n^T \\ e_1^T \\ e_2^T \end{pmatrix} \tag{35}$$

where θ is the rotation angle. Each motion $g \in \mathcal{G}$ (including rotation and translation) then can be represented by three parameters $(\theta, \alpha_1, \alpha_2)$ as the following transformation

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} = g \circ \begin{pmatrix} x \\ y \\ z \end{pmatrix} = M \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \alpha_1 e_1 + \alpha_2 e_2. \tag{36}$$

Consider a planar analytic curve $\Gamma = (\gamma_x(y), y, \gamma_z(y))^T$ in the camera coordinate frame F_c where $\gamma_x(y)$ is expressed by Taylor series expansion at $y = -d\cos\phi$ as:

$$\gamma_x(y) = \sum_{i=1}^{\infty} \frac{\xi_i(-d\cos\phi)}{(i-1)!}(y + d\cos\phi)^{i-1}. \tag{37}$$

Then after motion g, according to equation (36), the new parameterization $\hat{\gamma}_x, \hat{y}$ of the curve can be written as

$$\hat{\gamma}_x = h_1(y, \theta, \alpha_1, \alpha_2) \tag{38}$$

$$\hat{y} = h_2(y, \theta, \alpha_1, \alpha_2) \tag{39}$$

for some smooth real functions h_1 and h_2 which satisfy $h_1(y, 0, 0, 0) = \gamma_x(y)$ and $h_2(y, 0, 0, 0) = y$. By Assumption 2, $h_2(y, \theta, \alpha_1, \alpha_2)$ is invertible as a function of y. Thus,

$$\hat{\gamma}_x = h_1(h_2^{-1}(\hat{y}, \theta, \alpha_1, \alpha_2), \theta, \alpha_1, \alpha_2) \tag{40}$$

$$= h(\hat{y}, \theta, \alpha_1, \alpha_2) \tag{41}$$

for some smooth real function h which satisfies $h(y, 0, 0, 0) = \gamma_x$. Now we can expand $\hat{\gamma}_x$ at $\hat{y} = -d\cos\phi$ the same as (37) and get

$$\hat{\gamma}_x(\hat{y}) = \sum_{i=1}^{\infty} \frac{\hat{\xi}_i(-d\cos\phi)}{(i-1)!}(\hat{y} + d\cos\phi)^{i-1}. \tag{42}$$

Clearly, each coefficient $\hat{\xi}_i(-d\cos\phi)$ is a function of $(\theta, \alpha_1, \alpha_2)$. Therefore ξ may be parameterized by θ, α_1, and α_2 and the reachable space (or the configuration space) of ξ is of at most three-dimensional. We have proved the following theorem:

Theorem 2 *Consider an arbitrary ground-based mobile robot and an arbitrary ground analytic curve $\Gamma = (\gamma_x(y), y, \gamma_z(y))^T$ with γ_z given by (8). ξ is defined as the vector of the coefficients of the Taylor series of $\gamma_x(y)$ expanded at $y = -d\cos\phi$,[7] Then (i) the (locally) reachable space of ξ under the motion of the mobile robot has at most 3 dimensions; (ii) if under the motion of the mobile robot, ξ (and ζ) is also a dynamic system, the rank of the distribution spanned by the Lie algebra generated by the vector fields associated to such a system is at most 3.*

Combining this theorem with the previous results about the unicycle and the linear curvature curves, we have the following corollaries:

Corollary 1 *Consider a linear curvature curve (i.e. $k'(s) = c \neq 0$) and an arbitrary maximally nonholonomic ground-based mobile robot, the reachable space of ξ^3 (and ζ^3) under the motion of the robot is (locally) of dimension 3. Furthermore, if under the motion of this mobile robot, ξ^3 (and ζ^3) is itself a dynamic system, then the rank of the distribution spanned by the Lie algebra generated by the vector fields associated to such a system is exactly 3.*

This corollary is true because, for a special nonholonomic mobile robot: the unicycle, according to Theorem 1, the local reachable space has exactly 3 dimensions in the case of linear curvature curves. For two maximally nonholonomic ground-based mobile robots, their motion spaces are the same as $SE(2)$. Thus they have the same ability in controlling the shape of the image curve.

In the case of the unicycle, as already derived in section 2.2.2, ξ is a dynamic system given by (22), therefore

[7] This definition of ξ turns out to be exactly the same as we defined in (11) and (12).

Corollary 2 *The rank of the distribution spanned by the Lie algebra generated by the vector fields associated to the system (22) is at most 3. And in the case of linear curvature curves, the rank is exactly 3 (as previously declared in Theorem 1).*

Comments In the case of constant curvature curves, there are only two independent parameters ξ_1 and ξ_2 needed to decide the image curve. Obviously, all the above corollaries still hold by changing 3 to 2.

4 Control Design in the Image Plane

"It is with logic that one proves,
it is with intuition that one invents."
— Henri Poincare

In this section, we study the design of control laws based on controlling the shape of the image curve in the image plane so as to facilitate successful navigation of the ground-based mobile robot.

4.1 Controlling the Shape of Image Curves

According to the controllability results presented in the previous section, one can only control up to three parameters $(\xi_1, \xi_2, \xi_3)^T$ of the image of a given ground curve. This means the shape of the image curve can only be controlled up to the linear curvature features of the given curve. In this section, we study how to obtain control laws so as to arbitrarily control the image of a linear curvature curve, as well as propose how to control the image of an general curve.

For a unicycle mobile robot, the dynamics of the image of a linear curvature ground curve is given by system (27). According to Theorem 1, this two-input three-dimensional system is controllable (*i.e.* has one degree of nonholonomy) for $c \neq 0$. Thus, using the algorithm given in Murray and Sastry [13] [14], system (27) can be transformed to the canonical *chained-form.*

The resulting change of coordinates is

$$\left\{ \begin{array}{rcl} x_1 & = & \xi_2 \\ x_2 & = & -\frac{a^3\xi_3}{c(a^2+\xi_2^2)^3} \\ x_3 & = & (\xi_1 - \frac{a\xi_2\xi_3}{c(a^2+\xi_2^2)^2}) \\ \omega & = & \frac{-ca(a^2+\xi_2^2)^3+3a^2\xi_2\xi_3^2}{c(a^2+\xi_2^2)^4}u_1 - \frac{\xi_3}{a}u_2 \\ v & = & \frac{-ca(a^2+\xi_2^2)^3+3a^2\xi_2\xi_3(a^2+\xi_2^2+\xi_3)}{c(a^2+\xi_2^2)^4}u_1 - \frac{a^2+\xi_2^2+\xi_3}{a}u_2 \end{array} \right. \qquad (43)$$

where $a = (\sin\phi)^{-1}$. Then, the transformed system has the chained-form:

$$\left\{\begin{array}{rcl} \dot{x}_1 & = & u_1 \\ \dot{x}_2 & = & u_2 \\ \dot{x}_3 & = & x_2 u_1 \end{array}\right. \tag{44}$$

For the chained-form system (44), using *piecewise smooth sinusoidal inputs* [14], one can arbitrarily steer the system from one point to another in R^3. Therefore, one can arbitrarily control the shape of the image of a linear curvature curve.

As for controlling the image of an arbitrarily given ground (analytic) curve, the best we can do is to approximate this curve locally by a linear curvature curve (if $k''(s) \approx 0$) and then, the controls for controlling the image of this approximating linear curvature curve can approximately control the image of the original curve freely up to its first three parameters $(\xi_1, \xi_2, \xi_3)^T$ in a local range.

Note that when $c = 0$, *i.e.* the curve is of constant curvature, the above transformation is not well-defined. This is because the system ξ now only has two independent states ξ_1 and ξ_2. It is much easier to steer such a two-input two-state system than the above chained-form system.

Remark 2 *Using Lemma 1, the dynamic system ζ^3 of the perspective projection image of a linear curvature curve can be also transformed to the chained-form.*

4.2 Tracking Ground Curves

4.2.1 Tracking Analytic Curves

In this section, we formulate the problem of mobile robot tracking a ground curve as a problem of controlling the shape of its image with the dynamics described by (22). We design a *state feedback* control law for this system such that the mobile robot (unicycle) asymptotically tracks the given curve.

First, let us study the *necessary and sufficient conditions* for perfect tracking of a given curve. As already explained in the beginning of Section 3, when the mobile robot is perfectly tracking the given curve

$$\xi_1 = \gamma_x(y,t)|_{y=-d\cos\phi} \equiv 0 \tag{45}$$

$$\xi_2 = \frac{\partial\gamma_x(y,t)}{\partial y}|_{y=-d\cos\phi} \equiv 0. \tag{46}$$

From (27) when $\xi_1 = \xi_2 \equiv 0$, we have

$$\dot{\xi}_2 = -\xi_3 v \sin\phi + \omega/\sin\phi \equiv 0. \tag{47}$$

This gives the *perfect tracking angular velocity*

$$\omega = \xi_3 \sin^2\phi v. \tag{48}$$

It is already known that system (22) is a nonholonomic system. According to Brockett [1], there do not exist smooth state feedback control laws which asymptotically stabilize a *point* of a nonholonomic system. In the following theorem, we give a piecewise smooth state feedback control law which stabilizes the system ξ around the *subset* $\{\xi \in R^\infty : \xi_1 = \xi_2 = 0\}$, *i.e.* the mobile asymptotically tracks the given curve, and as we will soon see, under certain conditions, this control law may become a smooth one.

Theorem 3 (Tracking Control Laws) *Consider closing the loop of system ξ (22) with control (ω, v) given by*

$$\begin{aligned} \omega &= \xi_3\sin^2\phi v_0 + \sin^2\phi\xi_1 v_0 + K_\omega\xi_2 \\ v &= v_0 + \sin^2\phi\xi_1(\xi_1+\xi_3)v_0 - K_v\xi_2 sign(\xi_1+\xi_3) \end{aligned} \tag{49}$$

where K_ω, K_v are strictly positive constants. The closed-loop system asymptotically converges to the subset

$$\{\xi \in R^\infty : \xi_1 = \xi_2 = 0\} \tag{50}$$

for initial conditions with ξ_1 and ξ_2 are small enough. Once on the subset, the mobile robot has the given linear velocity v_0 and the perfect tracking angular velocity $\omega_0 = \xi_3\sin^2\phi v_0$.

Proof Consider the "partial" Lyapunov function $V = \xi_1^2 + \xi_2^2$. Through direct calculation, we get

$$\begin{aligned} \dot{V} = & - & K_\omega / \sin\phi \xi_2^2 - K_\omega \sin\phi \xi_2^4 \\ & - & \xi_2^2 \sin\phi \left[(K_\omega \xi_1 + \xi_2 \sin^2 \phi v_0) + K_v sign(\xi_1 + \xi_3) \right] (\xi_1 + \xi_3). \end{aligned} \tag{51}$$

There exists $\epsilon > 0$ such that, when $|\xi_1| < \epsilon$ and $|\xi_2| < \epsilon$,

$$|K_\omega \xi_1 + \xi_2 \sin^2 \phi v_0)| < K_v. \tag{52}$$

Thus, in the set $W_\epsilon \equiv \{\xi \in R^\infty : \xi_1^2 + \xi_2^2 < \epsilon^2\}$

$$\dot{V} \leq -K_\omega / \sin\phi \xi_2^2 \leq 0. \tag{53}$$

For any initial value in W_ϵ, $V(t)$ is a non-increasing function and is bounded from below, therefore it has a limit as $t \to \infty$. Let

$$c_0 = \lim_{t \to \infty} V(t). \tag{54}$$

We claim that c_0 has to be zero. We prove this by contradiction. Suppose that $c_0 > 0$ for a trajectory $\xi(t)$ starting from some initial value $\xi(0)$ in W_ϵ. Let

$$\Omega_{c_0} \equiv \{(\xi_1, \xi_2)^T \in R^2 : \xi_1^2 + \xi_2^2 = c_0\}. \tag{55}$$

Consider the projection π from R^∞ to R^2:

$$\pi : \xi \to \xi^2. \tag{56}$$

Then $\pi(\xi(t))$ is bounded and has a non-empty limit set L in Ω_{c_0}. Thus, $\pi^{-1}(L)$ is the largest set to which the trajectory $\xi(t)$ must converge in W_ϵ. From the inequality (53), for any $(\xi_1, \xi_2)^T$ in L, we have $\xi_2 = 0$. Thus, in $\pi^{-1}(L)$, $\xi_2 \equiv 0$ so that $\dot{\xi}_2 \equiv 0$. For the closed-loop system, when $\xi_2 = 0$, we have $\dot{\xi}_2 = -\sin\phi \xi_1 v_0$. This implies that $\xi_1 = 0$ for $\xi \in \pi^{-1}(L)$. This is a contradiction to relationship $\xi_1^2 + \xi_2^2 = c_0 > 0$ for $\xi \in \pi^{-1}(L)$ since $L \subseteq \Omega_{c_0}$. □

Comments Although Theorem 3 only guarantees local stability of the control law (49), combining it with the results obtained in Section 4.1, we can track a given linear curvature curve from an arbitrary initial position by a two-step control scheme: first, using sinusoidal inputs, steer the system to a position with ξ_1, ξ_2 close to zero; then, switch to the tracking control law (49). In fact, it can be shown by simulation that, with appropriately choosing K_v and K_ω, the tracking control law (49) itself has a very large stable region. Thus, in most cases, the first step is not quite necessary.

4.2.2 Tracking C^1-Smooth Piecewise Analytic Curves

Although Theorem 3 only concerns about analytic (C^ω) curves, it actually can be generalized to *C^1-smooth piecewise analytic curves*[8].

Corollary 3 *Consider an arbitrary C^1-smooth piecewise analytic (ground) curve. If the maximum curvature $|k|_{max}$ exists for the whole curve, then, when $K_\omega > 0$ and $K_v \geq 0$ the feedback control law given by (49) guarantees that the mobile robot locally asymptotically tracks the given curve.*

Proof Consider the same Lyapunov function as the one chosen in Theorem 3. Then, $\dot{V}$ is still given by (51). Since $|k|_{max}$ exists, from (24), $|\xi_3|$ is bounded. Then, according to (51), if $K_\omega > 0$ and $K_v \geq 0$, there exist $\epsilon > 0$, which is independent of ξ_3, such that when $\xi \in W_\epsilon = \{\xi_1^2 + \xi_2^2 < \epsilon^2\}$

$$\dot{V} \leq -K_\omega / \sin\phi \xi_2^2 \leq 0. \tag{57}$$

The rest of the proof follows that of Theorem 3. □

Comments It is very important to note that, in the proof of the Theorem 3, the choice of ϵ is independent of ξ_3. For a C^1-smooth piecewise analytic curve, ξ_1 and ξ_2 are always continuous. Only ξ_3 may not change continuously. But the proof shows that the $\dot{V}$ does not depend on ξ_3 when ξ_1 and ξ_2 are small enough. Therefore the (C^1-smooth) switching between different analytic pieces of the curve does not affect the convergence of the system. In the Corollary 3, since ξ_3 is bounded, K_v can then be set to 0 and the inequality (53) for $\dot{V}$ still holds locally. In the case of $K_v = 0$, the control (49) becomes a smooth one.

[8] "C^1-smooth" means that the tangent vector along the whole curve is continuous.

Remark 3 *Using Lemma 1, the control (49) can be converted to a stabilizing tracking control law for ζ of the perspective projection image.*

4.2.3 Tracking Arbitrary Curves

Corollary 3 suggests that, for tracking an arbitrary continuous (C^0-*smooth*) ground curve (not necessarily analytic), one may approximate it by a C^1-smooth piecewise analytic curve, a *virtual curve*, and then track this approximating virtual curve by using the control law (49). However, since the virtual curve cannot be "seen" in the image, how could one get the estimates of ξ for the "image" of the virtual curve so as to get the feedback controls v and ω subsequently? It turns out that, the virtual ξ is exactly the solution of the differential equation of the closed-loop system (22) with v and w given by (49). The initial conditions for solving such differential equation can be obtained from when designing the virtual curve.

Now, the control becomes an open-loop scheme, and in order to track this virtual curve, one has to solve the differential equation (22) (with v and ω given by (49)) in advance and then get the desired controls v and ω. It is computationally expensive to approximate a given curve by an arbitrary analytic curve in which case, principally, we have to solve the infinite-dimensional differential equation (22).

However, as argued in Section 2.2.2, a special class of analytic curves, the linear curvature curves, can reduce the infinite-dimensional system (22) to a three-dimensional system (27), and the three states ξ^3 of the system (27) also have captured all the controllable features of the system ξ, according to Theorem 2. Therefore, it is much more computationally economical to approximate the given curve by a C^1-smooth piecewise linear curvature curve and, then, solve the three-dimensional differential equation (27) to get the appropriate controls v and ω.

Few applications do require tracking of arbitrary (analytic) curves. The target curves usually can be modeled as piecewise linear curvature curves. For instance, in the case of vehicle control, in the United States, most highways are designed to be of piecewise constant curvature, and in Europe, of piecewise linear curvature. Therefore, piecewise linear curvature curves are simple as well as good models for most tracking tasks.

Comments Strictly speaking, when approximating a given curve by a piecewise polynomial curve, for example by using splines [6], in order to get the estimate of ξ for the evolution of the approximating virtual curve, one has to solve the infinite-dimensional differential equation (22). What the "polynomial" property really simplifies is just the initial conditions of the differential equation but not the dimension of the problem, as already argued in Section 2.2.3.

Example (Mobile Robot Tracking Corridors) For illustration, let us consider a simple example: the mobile robot is supposed to track a piecewise linear curve consisting of intersection of l_1 and l_2 (as a reasonable model for corridors inside a building), as shown in Figure 6. A natural and simple way to smoothly connect them together is to use a piece of arc AB which is tangential to both of the straight lines (at points A and B respectively). From point A, the mobile robot switches to track the virtual curve, arc AB until it smoothly steers into the next piece, *i.e.* the line l_2. The $\xi^3(t)$ for tracking this virtual arc AB is then given by the solution of the closed-loop system of (27) with $c = k'(s) \equiv 0$ and the initial conditions at point A: $\xi^3(0) = (0, 0, -a^2/r)^T$.

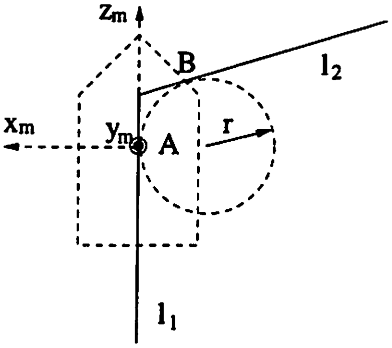

Figure 6: Using arcs to connect curves which are piecewise straight lines.

Since the approximating virtual curve is to be as close to the original curve as possible, the radius r of the arc AB should be as small as possible. But, in real applications, the radius r is limited by the maximal curvature that the mobile robot can track ($r = 1/|k|$). Thus, one needs to consider this extra constraint when designing the virtual curves. The following result tells us a way to decide the maximal curvature $|k|_{max}$ that the mobile robot can track:

Fact 1 *Consider the unicycle mobile robot. If its linear velocity v and angular velocity ω satisfy $|v| \geq c_1$ and $\omega^2 + v^2 \leq c_2^2$, then the maximal curvature that it can track is*

$$|k| \leq \sqrt{\left(\frac{c_2}{c_1}\right)^2 - 1}. \tag{58}$$

Consider now that the image curve obtained is not even continuous, *i.e.* the robot "sees" several chunks of the image of the real curve that it is supposed to track. Basically, there are two different approaches that one might take in order to track such a curve: first, one may use some estimation schemes and based on the estimated features of the real curve to apply the feedback control law (as studied by Frezza and Picci [6]); second, one may just smoothly connect these chunks of the image curve by straight lines, arcs or linear curvature curves and then apply the virtual tracking scheme as given in the above to track the approximating virtual curves.

4.3 Simulation Results of Tracking Ground Curves

In this section, we show simulation results of the mobile robot tracking some specific ground curves with using the control schemes designed in previous sections. We assume that all the image features ξ are already available. In next section, we discuss how to actually estimate ξ from the real (probably noisy) images. For all the following simulations, we choose the camera tilt angle $\phi = \pi/3$, and the control parameters $K_\omega = 1, K_v = 0.5$, and $v_0 = 1$. The reference coordinate frame F_f is chosen such that the initial position of the mobile robot is $z_{f0} = 0, x_{f0} = 0$ and $\theta_0 = 0$.

4.4 Tracking a Circle

A circle is a constant curvature curve, *i.e.* $c = k'(s) \equiv 0$. For the simulation results shown in Figure 7, the initial position of the nominal circle given in the image plane is $\xi_{10} = 1$, $\xi_{20} = 1$, and $\xi_{30} = 1$.

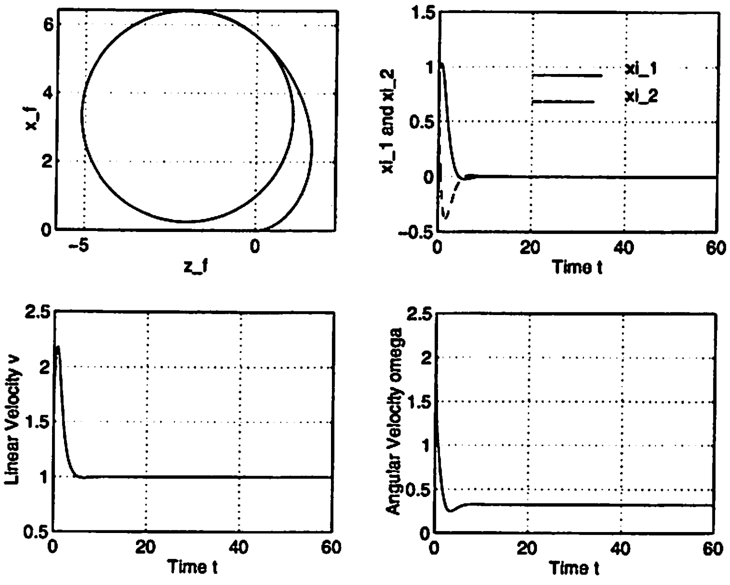

Figure 7: Simulation results for tracking a circle. Subplot 1: the trajectory of the mobile robot in the reference coordinate frame; subplot 2: the image curve parameters ξ_1 and ξ_2; subplot 3 and 4: the control inputs v and ω.

4.5 Tracking a Linear Curvature Curve

For the simulation results given in Figure 8, the nominal trajectory is chosen to be a linear curvature curve with the constant curvature varying rate $c = k'(s) \equiv -0.05$. Its initial position given in the image plane is $\xi_{10} = 0.1$, $\xi_{20} = 0.1$, and $\xi_{30} = 2$.

4.6 Tracking Piecewise Straight-Line Curves

Consider now the example discussed in Section 4.2.3: the mobile robot is to track a piecewise linear curve consisting of intersection of l_1 and l_2 as shown in the Figure 9. We compare the simulation results of two schemes: 1. Using only the feedback tracking control law; 2. Using a pre-designed approximating virtual curve (an arc in this case) around the "broken" point. From Figure 9, it is obvious that, by using the pre-designed virtual curve, the over-shoot can be avoided. But the computation is more intensive: one needs to design the virtual curve and calculate the desired control inputs for tracking it.

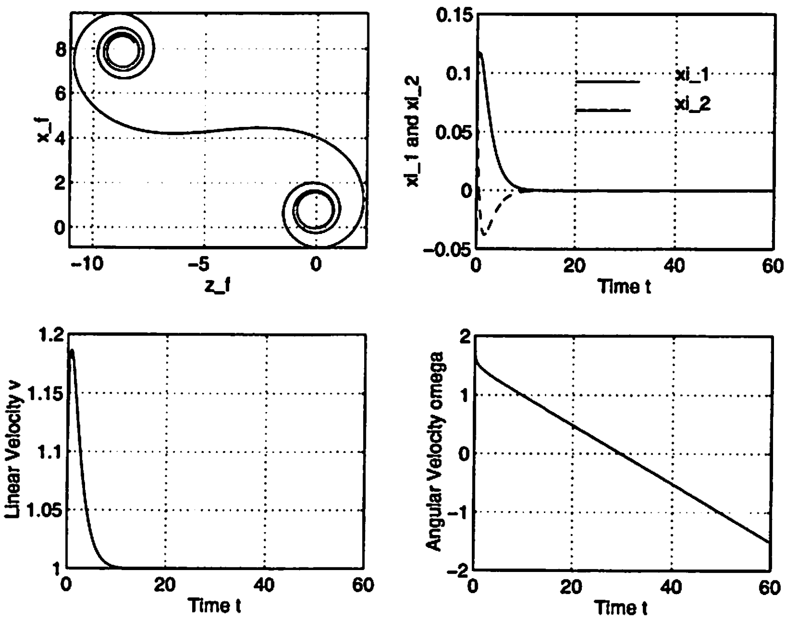

Figure 8: Simulation results for tracking a linear curvature curve ($c = k'(s) = -0.05$). Subplot 1: the trajectory of the mobile robot in the reference coordinate frame; subplot 2: the image curve parameters ξ_1 and ξ_2; subplot 3 and 4: the control inputs v and ω.

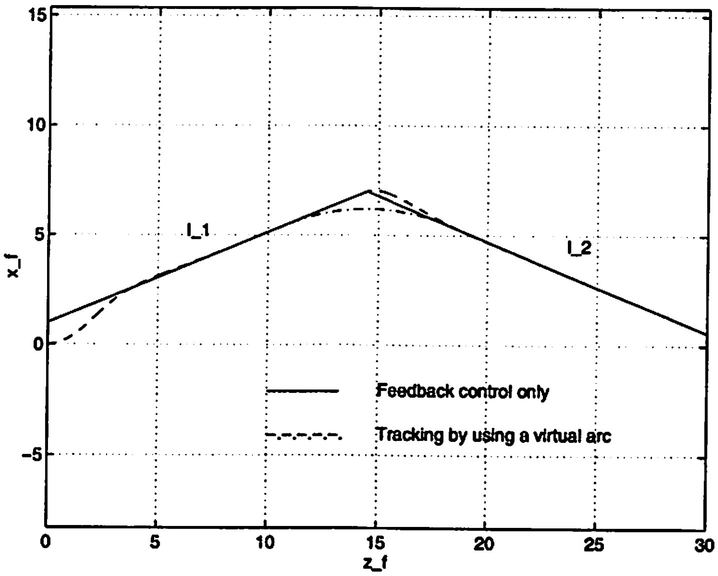

Figure 9: Comparison between two schemes for tracking a piecewise straight-line curve: 1. Using only the feedback tracking control law; 2. Using a pre-designed approximating virtual curve (an arc in this case) around the "broken" point.

5 Observability Issues and Estimation of Image Quantities

> *"Upon the exactness with which we pursue phenomenon into the infinitely small, does our knowledge of their causal connections essentially depend."*
>
> — G. R. B. Riemann, *On the Hypotheses Which Lie at the Foundations of Geometry, 1854*

As we have discussed in Section 2.2.1, ξ are features of the orthographic projection image $\hat{\Gamma}$of the ground curve Γ, and are not yet the real image (which, by convention, means the perspective projection image Λ) quantities ζ. However, ξ and ζ are algebraically related by Lemma 1. In principle, one can obtain ξ from the directly-measurable ζ.

In order to apply the tracking control law (49), one need to know the values of ξ_1, ξ_2, and ξ_3, *i.e.* ζ_1, ζ_2 and ζ_3. Suppose, at each instant t, the camera provides N measurements of the image curve Λ:

$$\{(\lambda_X(Y_k, t), Y_k)\} \qquad k = 1, \ldots, N \tag{59}$$

where $\{Y_1, Y_2, \ldots, Y_N\}$ are fixed distances from the origin. If the distances between Y_k are small enough, one can estimate the values of $\zeta_1(Y_k), \zeta_2(Y_k)$, and $\zeta_3(Y_k)$ simply by:

$$\begin{array}{rcl} \hat{\zeta}_1(Y_k) & = & \lambda_X(Y_k, t) \\ \hat{\zeta}_2(Y_k) & = & \frac{\lambda_X(Y_{k+1}, t) - \lambda_X(Y_k, t)}{Y_{k+1} - Y_k} \\ \hat{\zeta}_3(Y_k) & = & (\frac{\lambda_X(Y_{k+2}, t) - \lambda_X(Y_{k+1}, t)}{Y_{k+2} - Y_{k+1}} - \frac{\lambda_X(Y_{k+1}, t) - \lambda_X(Y_k, t)}{Y_{k+1} - Y_k})/(Y_{k+1} - Y_k) \end{array} \qquad k = 1, \ldots, N-2. \tag{60}$$

However, in practice, the measurements $\{(\lambda_X(Y_k, t), Y_k)\}$ are noisy and the estimates (60) for ζ^3 become very inaccurate, especially for the higher order terms ζ_2 and ζ_3. It is thus appealing to estimate ζ^3 or ξ^3 by only using the measurements $\{(\lambda_X(Y_k, t), Y_k)\}$ but not their differences.

5.1 Sensor Models and Observability Issues

5.1.1 General Analytic Curves

The curve dynamics are already given by (22). If we only use the measurement $\zeta_1 = \lambda_X(Y, t)$ as the output of the vision sensor, then we have the following sensor model:

$$\begin{aligned} \begin{pmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \vdots \\ \dot{\xi}_i \\ \vdots \end{pmatrix} &= -\begin{pmatrix} \xi_1\xi_2 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_1\xi_3 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_1\xi_4 \sin\phi + 3\xi_2\xi_3 \sin\phi \\ \vdots \\ \xi_1\xi_{i+1} \sin\phi + g_i(\xi_1, \dots, \xi_i) \\ \vdots \end{pmatrix} \omega + \begin{pmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \\ \vdots \\ \xi_{i+1} \sin\phi \\ \vdots \end{pmatrix} v \\ h(\xi) &= \zeta_1 = \frac{\sin\phi}{d + y\cos\phi}\xi_1 \end{aligned} \tag{61}$$

where $h(\xi)$ is the measurable output.

Theorem 4 (Observability of the Camera System) *Consider the system given by (61). Let*

$$f_1 = -\begin{pmatrix} \xi_1\xi_2 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_1\xi_3 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_1\xi_4 \sin\phi + 3\xi_2\xi_3 \sin\phi \\ \vdots \\ \xi_1\xi_{i+1} \sin\phi + g_i(\xi_1, \dots, \xi_i) \\ \vdots \end{pmatrix} \qquad f_2 = \begin{pmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \\ \vdots \\ \xi_{i+1} \sin\phi \\ \vdots \end{pmatrix}. \tag{62}$$

If $\phi \neq 0$, then the annihilator Q of the smallest codistribution Ω invariant under f_1, f_2 and which contains $dh(\xi)$ is empty.

Proof Through direct calculations, the k-th order Lie derivative of the covector field $dh(\xi)$ along the vector field f_2 is

$$L_{f_2}^k dh(\xi) = \frac{\sin^{k+1}\phi}{d + y\cos\phi} d\xi_{k+1} \qquad k = 0, 1, 2, \dots, \infty. \tag{63}$$

Thus, Ω contains all $d\xi_i$, $i \in \mathcal{N}$ and therefore Q is an empty distribution. □

Comments According to the Theorem 1.9.8 in Isidori [10], Theorem 4 guarantees that the system (61) is observable. Or, in other words, the (locally) *maximal output zeroing manifold* of the system (61) does not exist, according to the Proposition 10.16 in Sastry [18]. Since this system is observable, ideally, one then can estimate the $\hat{\xi}$ from the output $h(\xi)$. However, the observer construction may be difficult.

5.1.2 Linear Curvature Curves

The sensor model (61) is an infinite-dimensional system. In order to build an applicable estimator for ξ^3 (so as to apply the tracking control law (49)), one has to assume some regularity on the given curve Γ so that the sensor model becomes a finite-dimensional system. In other words, one has to approximate Γ by simpler curve models which have finite-dimensional dynamics.

In Frezza and Picci [6], the models are chosen as *third-order B-splines*. However, as we have pointed out in Section 2.2.3, the polynomial form is not an intrinsic property of a curve and it cannot be preserved under the motion of the mobile robot. Furthermore, simple curves like a circle cannot be expressed by third-order B-splines. We thus propose to use (piecewise) linear curvature curves as the models. The reasons for this are obvious from the discussions in previous sections: the dynamics of a linear curvature curve are a three-dimensional system (27); such a system has very nice control properties; and piecewise linear curvature curves are also natural models for highways in the case of using vision for autonomous vehicle control. However, a most important reason for using linear curvature curves is that, according to Theorem 3, one actually only needs the estimation of three image quantities, *i.e.* ξ_1, ξ_2 and ξ_3 to be able to tracking any analytic curve. All the "higher order terms" $\xi_i \, i \geq 4$ are not necessary.

For a linear curvature curve, since we do not have a priori knowledge about the constant curvature varying rate $c = k'(s)$, we also need to estimate it. Let $\eta = c$ and we have the following sensor model for linear curvature curves:

$$\begin{aligned}\begin{pmatrix}\dot{\xi}_1\\ \dot{\xi}_2\\ \dot{\xi}_3\\ \dot{\eta}\end{pmatrix} &= -\begin{pmatrix}\xi_2\xi_1\sin\phi + d\cot\phi + \frac{y}{\sin\phi}\\ \xi_3\xi_1\sin\phi + \xi_2^2\sin\phi + \frac{1}{\sin\phi}\\ \xi_4\xi_1\sin\phi + 3\xi_2\xi_3\sin\phi\\ 0\end{pmatrix}\omega + \begin{pmatrix}\xi_2\sin\phi\\ \xi_3\sin\phi\\ \xi_4\sin\phi\\ 0\end{pmatrix}v \qquad (64)\\ h(\xi^3,\eta) &= \zeta_1 = \tfrac{\sin\phi}{d+y\cos\phi}\xi_1\end{aligned}$$

where $\xi_4 = \frac{\eta(a^2+\xi_2^2)^3/a+3\xi_2\xi_3^2}{a^2+\xi_2^2}$ and $h(\xi^3,\eta)$ is the measurable output.

Theorem 5 *Consider the system (64). Let*

$$f_1 = -\begin{pmatrix}\xi_2\xi_1\sin\phi + d\cot\phi + \frac{y}{\sin\phi}\\ \xi_3\xi_1\sin\phi + \xi_2^2\sin\phi + \frac{1}{\sin\phi}\\ \xi_4\xi_1\sin\phi + 3\xi_2\xi_3\sin\phi\\ 0\end{pmatrix} \qquad f_2 = \begin{pmatrix}\xi_2\sin\phi\\ \xi_3\sin\phi\\ \xi_4\sin\phi\\ 0\end{pmatrix}. \qquad (65)$$

If $\phi \neq 0$, then the smallest codistribution Ω invariant under f_1, f_2 and which contains $dh(\xi^3,\eta)$ is of constant rank 4.

Proof Through direct calculations, we have

$$L_{f_2}^k dh(\xi^3,\eta) = \frac{\sin^{k+1}\phi}{d+y\cos\phi}d\xi_{k+1} \qquad k = 0,1,2 \qquad (66)$$

and

$$L_{f_2}^3 dh(\xi^3,\eta) = \frac{(a^2+\xi_2^2)^2\sin^5\phi}{d+y\cos\phi}d\eta. \qquad (67)$$

Thus, Ω contains all $d\xi_1, d\xi_2, d\xi_3$, and $d\eta$ and it has constant rank 4. □

Therefore, the system (64) is observable according to the Theorem 1.9.8 in Isidori [10] or the Proposition 10.16 in Sastry [18].

5.2 Estimation of Image Quantities by Extended Kalman Filter

The sensor model (64) is a nonlinear observable system. The *extended Kalman filter* (EKF) is a widely used scheme to estimate the states of such systems. In the computer vision community, estimation schemes based on Kalman filter have been commonly used for dynamical estimation of motion [19] [21] or road curvature [4] [3], etc. Here, we use the EKF algorithm to estimate on-line the $\hat{\xi}_1, \hat{\xi}_2, \hat{\xi}_3$, and $\hat{\eta}$. Alternatives to the EKF, which are based on nonlinear filtering, are quite complicated and are rarely used.

5.2.1 Multiple-Measurement Sensor Model

In order to make the EKF converge faster, we need to use more than one measurements (in the sensor models (61) and (64)). From the N measurements

$$\{(\lambda_X(Y_k, t), Y_k)\} \qquad k = 1, \ldots, N \tag{68}$$

we have N outputs

$$h_k(\xi) = \zeta_1(Y_k) = \frac{\sin\phi}{d + y_k \cos\phi}\xi_1(y_k) \qquad k = 1, \ldots, N \tag{69}$$

where Y_k and y_k are related by (9) $Y_k = \frac{y_k \sin\phi}{d + y_k \cos\phi}$.

For linear curvature curves, all $\xi_1(y_k)$ are functions of only ξ^3 and η since all the Taylor series expansion coefficients ξ_i, $i \in \mathcal{N}$ are functions of only ξ^3 and η according to Lemma 2. Let

$$h(\xi^3, \eta, y) = \sum_{i=1}^{\infty} \frac{\xi_i}{(i-1)!}(y + d\cos\phi)^{i-1}. \tag{70}$$

$\xi_1(y_k)$ are then given by $\xi_1(y_k) = h(\xi^3, \eta, y_k)$.

The sensor model (64) can be modified as

$$\begin{aligned} \begin{pmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \dot{\eta} \end{pmatrix} &= -\begin{pmatrix} \xi_2\xi_1 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_3\xi_1 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_4\xi_1 \sin\phi + 3\xi_2\xi_3 \sin\phi \\ 0 \end{pmatrix} \omega + \begin{pmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \\ 0 \end{pmatrix} v \\ h_k(\xi^3, \eta) &= \zeta_1(Y_k) = \frac{\sin\phi}{d + y_k\cos\phi} h(\xi^3, \eta, y_k) \qquad k = 1, \ldots, N \end{aligned} \tag{71}$$

where $\xi_4 = \frac{\eta(a^2+\xi_2^2)^3/a + 3\xi_2\xi_3^2}{a^2+\xi_2^2}$, and h_k are the measurable outputs.

5.2.2 Noise Analysis

In order to track the variations in the rate of change of the curvature of a curve, we choose

$$\dot{\eta} = \mu_\eta \tag{72}$$

where μ_η is white noise of appropriate variance.[9]

The output measurements are inevitably noisy, and the actual ones are given by

$$h_k(\xi^3, \eta) \quad = \quad \zeta_1(Y_k) = \frac{\sin\phi}{d + y_k \cos\phi} h(\xi^3, \eta, y_k) + \mu_{h_k} \qquad k = 1, \ldots, N \tag{73}$$

where μ_{h_k} are appropriate noise models for the N outputs. Strictly speaking, μ_{h_k} are color noise processes since image quantization errors[10] are main sources for μ_{h_k} which generically produce color noises. The explicit forms for the output h_k are given by the Taylor series expansion (70). In real applications, one has to truncate it. This can be regarded as another color noise source for the output noises μ_{h_k}. However, in order to approximately estimate the states ξ^3 and η, we may simplify μ_{h_k} by white noise processes and then can actually build an EKF (Jazwinski [11], Mendel [12]) to get the estimates $\hat{\xi}^3$ and $\hat{\eta}$ for the states of the nonlinear stochastic model:

$$\begin{pmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \dot{\eta} \end{pmatrix} = - \begin{pmatrix} \xi_2\xi_1 \sin\phi + d\cot\phi + \frac{y}{\sin\phi} \\ \xi_3\xi_1 \sin\phi + \xi_2^2 \sin\phi + \frac{1}{\sin\phi} \\ \xi_4\xi_1 \sin\phi + 3\xi_2\xi_3 \sin\phi \\ 0 \end{pmatrix} \omega + \begin{pmatrix} \xi_2 \sin\phi \\ \xi_3 \sin\phi \\ \xi_4 \sin\phi \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \mu_\eta \tag{74}$$

$$h_k(\xi^3, \eta) \quad = \quad \zeta_1(Y_k) = \tfrac{\sin\phi}{d + y_k \cos\phi} h(\xi^3, \eta, y_k) + \mu_{h_k} \qquad k = 1, \ldots, N$$

where $\xi_4 = \frac{\eta(a^2+\xi_2^2)^3/a + 3\xi_2\xi_3^2}{a^2+\xi_2^2}$, and μ_η and μ_{h_k} are white noises with appropriate variances.

5.2.3 The Extended Kalman Filter Algorithm

In order to have a clearer description of the EKF algorithm for system (74), we change the notation:

$$x_1 \quad = \quad \xi_1,\ x_2 = \xi_2,\ x_3 = \xi_3,\ x_4 = \eta \tag{75}$$

[9] One may also model η as a second order random walk.

[10] Including the errors introduced by the image-processing algorithms used to process the original images.

$$x_5 = \frac{x_4(a^2+x_2^2)^3/a + 3x_2x_3^2}{a^2+x_2^2} \tag{76}$$

$$\mathbf{x} = (x_1, x_2, x_3, x_4)^T \tag{77}$$

$$\mathbf{u} = (\omega, v)^T \tag{78}$$

$$\mathbf{h} = (h_1, \ldots, h_N)^T \tag{79}$$

$$\mu_{\mathbf{h}} = (\mu_{h_1}, \ldots, \mu_{h_N})^T. \tag{80}$$

Thus, system (74) can be rewritten as

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) + G\mu_\eta \\ \mathbf{z} = \mathbf{h}(\mathbf{x}, \mathbf{u}) + \mu_{\mathbf{h}} \end{cases} \tag{81}$$

where $G = (0, 0, 0, 1)^T$ and

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} -(x_2x_1\sin\phi + d\cot\phi + \frac{y}{\sin\phi}) & x_2\sin\phi \\ -(x_3x_1\sin\phi + x_2^2\sin\phi + \frac{1}{\sin\phi}) & x_3\sin\phi \\ -(x_5x_1\sin\phi + 3x_2x_3\sin\phi) & x_5\sin\phi \\ 0 & 0 \end{pmatrix} \mathbf{u}. \tag{82}$$

Assume the noise variances $E\{\mu_\eta\mu_\eta'\} = \sigma$ and $E\{\mu_{\mathbf{h}}\mu_{\mathbf{h}}'\} = R$. Let $T = t_{k+1} - t_k$ be the discretization time-step. The resulting EKF estimator equations are summarized in the following:

System Linearization and Discretization:

$$\delta\mathbf{x}(k+1) = \Phi(k+1, k)\delta\mathbf{x}(k) + \Psi(k+1, k)\delta\mathbf{u}(k) + \mu_{\eta d}(k) \tag{83}$$

$$\delta\mathbf{z}(k+1) = H_{\mathbf{x}}(k+1)\delta\mathbf{x}(k+1) + H_{\mathbf{u}}(k+1)\delta u(k+1)) + \mu_{\mathbf{h}}(k+1) \tag{84}$$

$$\Phi(k+1, k) = \Phi(t_{k+1}, t_k) \approx e^{F_kT} \tag{85}$$

$$\Psi(k+1, k) = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)C_k\, d\tau \approx C_kT + F_kC_k\frac{T^2}{2} \tag{86}$$

and

$$F_k = \left.\frac{\partial\mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}(k), \mathbf{u}=\mathbf{u}(k)} \qquad C_k = \left.\frac{\partial\mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial\mathbf{u}}\right|_{\mathbf{x}=\mathbf{x}(k), \mathbf{u}=\mathbf{u}(k)} \tag{87}$$

$$H_{\mathbf{x}}(k) = \left.\frac{\partial\mathbf{h}(\mathbf{x}, \mathbf{u})}{\partial\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}(k), \mathbf{u}=\mathbf{u}(k)} \qquad H_{\mathbf{u}}(k) = \left.\frac{\partial\mathbf{h}(\mathbf{x}, \mathbf{u})}{\partial\mathbf{u}}\right|_{\mathbf{x}=\mathbf{x}(k), \mathbf{u}=\mathbf{u}(k)} \tag{88}$$

$$Q_d = E\{\mu_{\eta d}(k)\mu_{\eta d}'(k)\} \approx \sigma GG'T. \tag{89}$$

Extended Kalman Filter:

- **Prediction Equations:**

$$\begin{aligned}
\hat{\mathbf{x}}(k+1|k) &= \hat{\mathbf{x}}(k|k) + \Psi(k+1,k)\mathbf{u}(k); \;\; \hat{\mathbf{x}}(0|0) = \hat{\mathbf{x}}_0 && (90)\\
P(k+1|k) &= \Phi(k+1,k)P(k|k)\Phi'(k+1,k) + Q_d; \;\; P(0|0) = P_0 && (91)\\
H_{\mathbf{x}}(k+1) &= \left.\frac{\partial \mathbf{h}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k+1|k),\mathbf{u}=\mathbf{u}(k)} && (92)
\end{aligned}$$

where

$$\begin{aligned}
\Phi(k+1,k) &= e^{F_k T} && (93)\\
\Psi(k+1,k) &= C_k T + F_k C_k \frac{T^2}{2} && (94)\\
F_k &= \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k),\mathbf{u}=\mathbf{u}(k)} && (95)\\
C_k &= \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k),\mathbf{u}=\mathbf{u}(k)} && (96)
\end{aligned}$$

- **Estimation Equations:**

$$\begin{aligned}
\delta\hat{\mathbf{z}}(k+1) &= \mathbf{z}(k+1) - \mathbf{h}(\hat{\mathbf{x}}(k+1|k), \mathbf{u}(k+1)) && (97)\\
\hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) + K(k+1)\delta\hat{\mathbf{z}}(k+1) && (98)\\
P(k+1|k+1) &= [I - K(k+1)H_{\mathbf{x}}(k+1)]P(k+1|k) && (99)
\end{aligned}$$

- **Gain:**

$$K(k+1) = P(k+1|k)H'_{\mathbf{x}}(k+1)[H_{\mathbf{x}}(k+1)P(k+1|k)H'_{\mathbf{x}}(k+1) + R]^{-1}. \quad (100)$$

Comments The computation complexity of Kalman filter is $O(n^3)$ where n is the system dimension [12]. Although, in some sense, both linear curvature curves and third-order B-splines (Frezza and Picci [6]) are third-order approximations for general curves, the dimension of the Kalman filter for estimating the B-spline parameters is $N+2$ where N is the number of measurements. However, the kalman filter we propose here is only 4-dimensional. Since the number of measurements N is usually larger than 4, the scheme proposed above is less computationally expensive.

5.2.4 Simulation Results of the Extended Kalman Filter

For illustration, we here give some simulation results of using the EKF to estimate the image quantities ξ^3 and η (*i.e.* the states of the system (74)). The Matlab source code of the EKF is given in Appendix A.

We first show by a simple example that the EKF converges. The curve is simply chosen to be a constant curvature curve (a circle) *i.e.* $c = k'(s) \equiv 0$. The initial values chosen for the estimates are $\hat{\xi}^3(0) = (0,0,0)^T$ and $\hat{\eta}(0) = 0.1$, and for the nominal states $\xi^3(0) = (0.1, 1, 1)^T$. The number of output measurements N is 5. The feedback tracking control laws (49) now use the estimates $\hat{\xi}^3$ for v and ω. For clarity, we do not put the noises on. The EKF then serves as a nonlinear observer. The simulation results are shown in Figure 10.

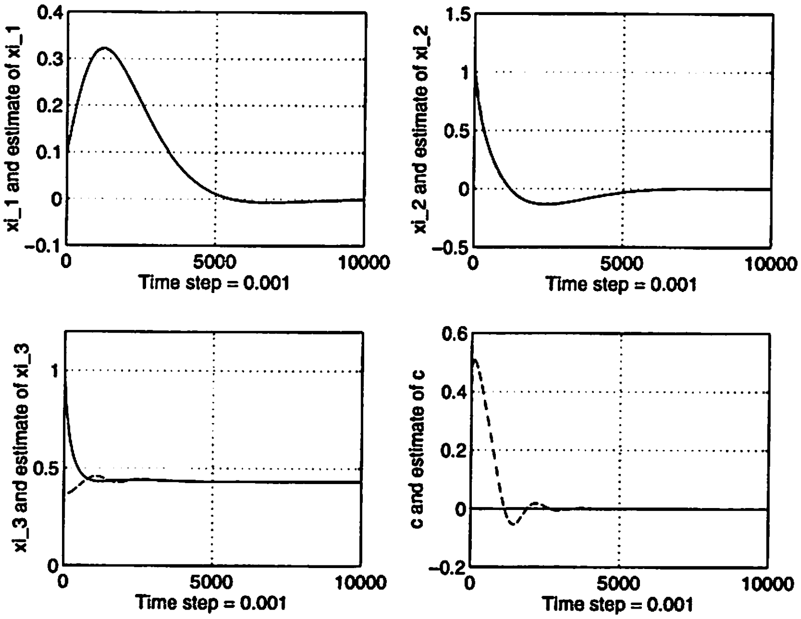

Figure 10: The simulation results of using the Extended Kalman Filter to estimate the image quantities ξ^3 and $\eta (= c = k'(s))$ with the number of output measurements $N = 5$: Solid curves are for true states; dashed curves are for estimates.

These results show that the estimates $\hat{\xi}^3$ and $\hat{\eta}$ converge to the nominal values ξ^3 and $\eta (= c)$. $\hat{\xi}_1$ and $\hat{\xi}_2$ converge especially quickly to ξ_1 and ξ_2 and their curves almost coincide. The results also show that the mobile robot eventually tracks the circle by using the estimates $\hat{\xi}^3$ for the tracking control laws (49) since both ξ_1 and ξ_2 eventually converge to zero.

As mentioned in Section 5.2.1, more output measurements might make the kalman filter converge faster. For comparison, we did the simulation for the case that there is only one output measurement, *i.e.* $N = 1$ (and all the other simulation conditions are still the same). The simulation results given in Figure 11 show that, compared to the simulation results for the $N = 5$ case, the convergence speed of the EKF is much slower.

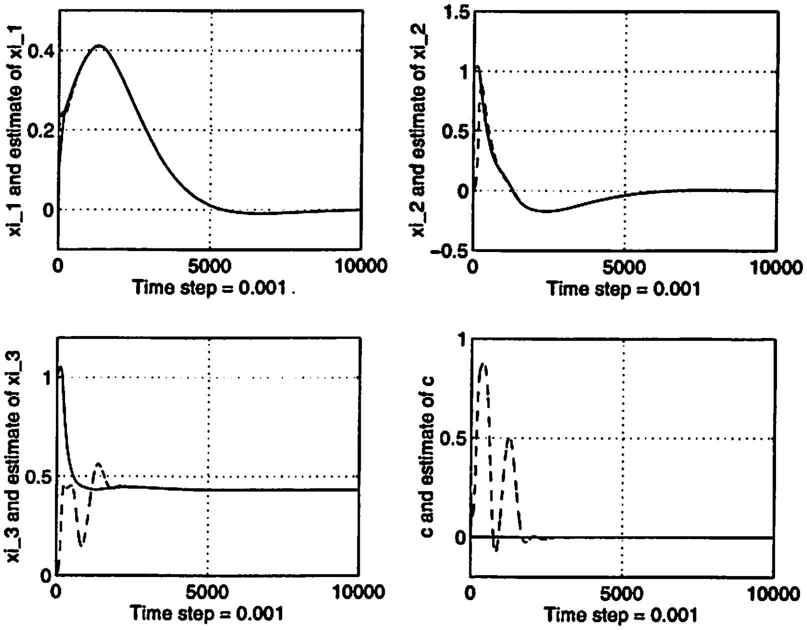

Figure 11: The simulation results of using the Extended Kalman Filter to estimate the image quantities ξ^3 and $\eta\,(= c = k'(s))$ with the number of output measurements $N = 1$: Solid curves are for true states; dashed curves are for estimates.

6 Simulation and Animation of the Vision Guided Navigation System

"Who changed the world more, Disney or Einstein?"

In the previous sections, we have developed control and estimation schemes for mobile robot navigation (tracking given curves) by using vision sensors. The image parameters needed for the tracking control schemes can be efficiently estimated from direct, probably noisy, measurements on the images. With combining the control and estimation schemes together, we thus obtain a complete *closed-loop vision-guided navigation system* which can be described in the following Figure 12.

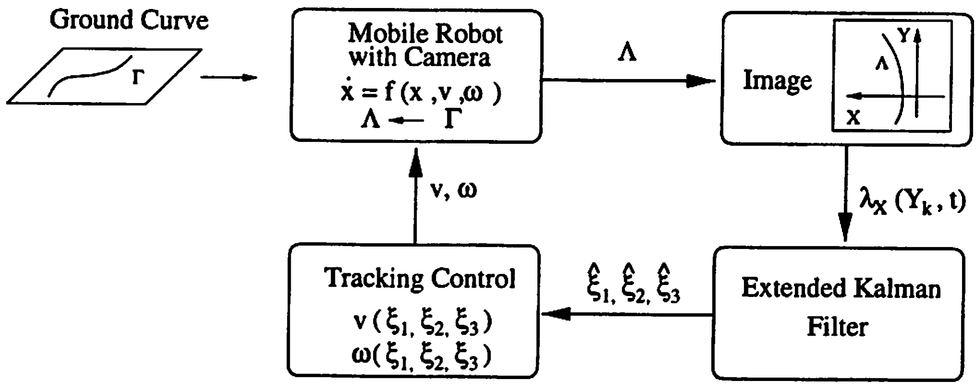

Figure 12: The closed-loop vision-guided navigation system for a ground-based mobile robot.

In order to know how this system works in real applications, we simulate it by using synthetic images of the ground curve. A *synthetic image* of a ground curve $\Gamma = (\gamma_x(y,t), y, \gamma_z(y,t))^T$ is to be a set of image points:

$$I = \{(\lambda_X(Y_i,t), Y_i)^T : (\lambda_X(Y_i,t), Y_i)^T = \pi \circ (\gamma_x(y_i,t), y_i, \gamma_z(y_i,t))^T,\ i = 1,2,\ldots,M\} \quad (101)$$

where π denotes the perspective projection map and the number of image points M maybe different for different time t. The output measurements from this synthetic image I are taken at N pre-fixed distances: $Y_1, \ldots, Y_N$. *Linear interpolation* is used to obtain approximate value of $\lambda_X(Y_k,t)$ if there is no point in I whose Y coordinate is Y_k.

Simulation results show that the control and the estimation schemes work well with each other in the closed-loop system. For illustration, Figure 13 gives out the simulation results for the simple case when Γ is a circle.

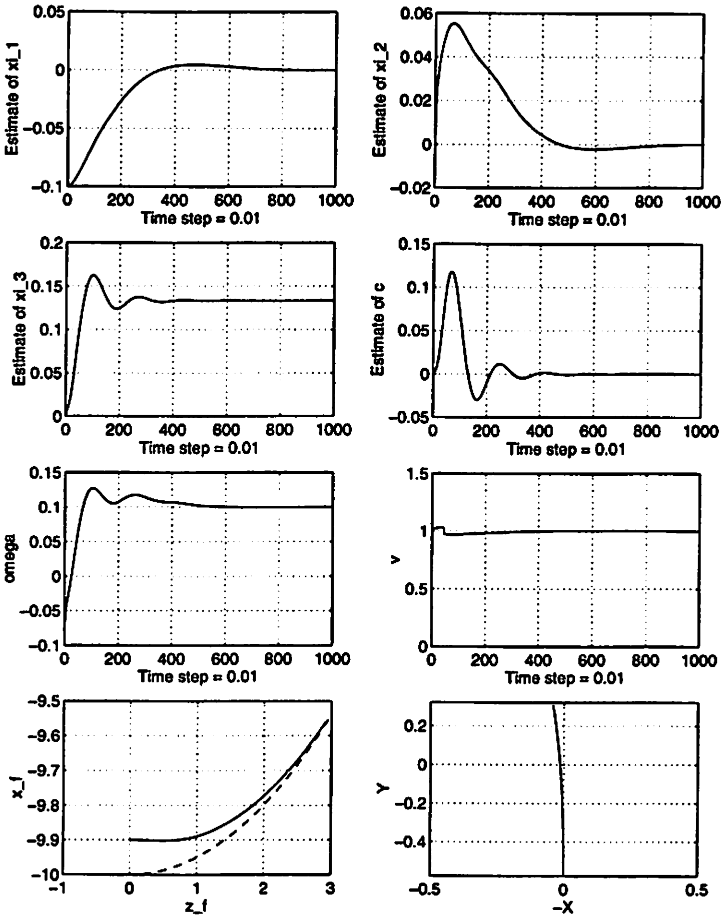

Figure 13: Simulation results for the closed-loop vision-guided navigation system for the case when the ground curve is a circle: In subplot 7, the solid curve is the actual mobile robot trajectory (in the space frame F_f) and the dashed one is the nominal circle; subplot 8 is the image of the circle viewed from the camera (at the last simulation step).

We can also make animations out of these synthetic images and simulation data. It is helpful for people to see how the entire closed-loop system works through different animated views. Figure 14 shows a synthetic image of a circular road viewed from the camera, which is from a single frame in the animation movie.[11]

[11] The Matlab code for generating this synthetic image is from Dr. Jana Kosecka.

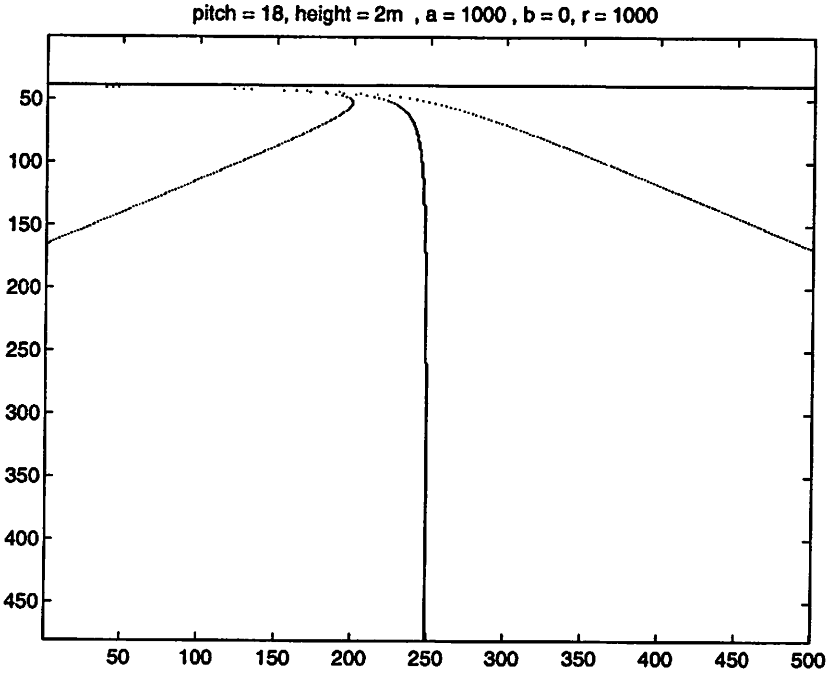

Figure 14: A synthetic image of a piece of circular road viewed from the camera.

7 Discussion and Future Work

"I will think about this tomorrow... After all, tomorrow is another day."
— Scarlett O'Hara, *Gone with the Wind*

In order to use the vision sensors inside the control servo loop, one first need to study the dynamics of the image. The dynamics of certain simple geometric primitives, like points, planes and circles, have been studied as well as exploited by people, Espiau [5] Pissard-Gibollet and Rives [15] *et al.* In this paper, we show that, for ground-based mobile robot, it is possible to study the dynamics of the image of a more general class of objects: analytic curves. Based on the understanding of image curve dynamics, we design control laws for tasks like controlling the shape of a image curve or tracking a given curve. Our study indicates that the shape of the image curve is controllable only up to its linear curvature terms (in the 2-dimensional case). However, Theorem 3 states that there exists a state feedback control law enabling the mobile robot to track arbitrary analytic curves. Such control laws are not necessarily unique. In real applications, other control laws may be designed and used to obtain better control performances.

Generally speaking, there are two basic ways to use information from vision sensors for control purposes: using vision sensors to provide environmental information for higher level decisions (so called open-loop planning); or using them directly in the feedback control loop as servoing sensors. As we show in Section 4.2.3 (Tracking Arbitrary Curves), the understanding of the image dynamics can also help to design appropriate open-loop control when the vision sensor does not provide enough information for applying feedback control.

In the cases that one have to approximate a general curve (which has infinite-dimensional dynamics) by simpler models, it is crucial to use models with properties which are invariant under the Euclidean motion (so-called intrinsic properties). We propose that linear curvature curves are very good candidates for being such models. In some sense, linear curvature curves are a third-order approximation for general curves, so are third-order B-splines used by Frezza and Picci [6]. However, the Extended Kalman Filters needed to estimate their parameters are 4-dimensional and $(N+2)$-dimensional respectively (where N is the number of output measurements). The computation intensities of the two schemes therefore are different.

Although the visual servoing for ground-based mobile robot navigation has been extensively studied, its applications in aerial robot navigation have not received much concern. In the aerial robot case, the motions are 3-dimensional rigid body motions $SE(3)$ instead of $SE(2)$ for ground-based mobile robots. Intrinsically, a mathematical formulation of this problem can be addressed as follows:

Consider $\Gamma(s)$ to be a curve to be tracked in R^3. The way to track this curve is through a projection to the camera plane

$$\pi : R^3 \rightarrow R^2. \tag{102}$$

π is either an orthographic or perspective projection. Further, $g(t) \in SE(3)$ represents the position and orientation of the camera respect to the spatial coordinate frame. Thus the curve $\Lambda(s,t)$ in the image plane is

$$\Lambda(s,t) = \begin{pmatrix} \lambda_1(s,t) \\ \lambda_2(s,t) \end{pmatrix} = \pi \circ g^{-1}(t) \circ \Gamma(s). \tag{103}$$

Given a mathematical model of the kinematics of the mobile robot

$$\dot{g} = f(g,u) \tag{104}$$

where $f(g,u) : SE(3) \times R^{n_i} \rightarrow T(SE(3))$ is a vector field on $SE(3)$, the kinematics can then be lifted to a dynamical system (the Riccati-type PDE)

$$\begin{pmatrix} \frac{\partial \lambda_1(s,t)}{\partial t} \\ \frac{\partial \lambda_2(s,t)}{\partial t} \end{pmatrix} \tag{105}$$

for the curve in the image plane. The results in this paper relate the controllability properties of (104) to the controllability properties of the new system (105) for the 2-dimensional case. A study of the more general 3-dimensional case is also necessary and important considering its potential applications in autonomous helicopter or aircraft navigation.

References

[1] R. W. Brockett, R. S. Millman, and H. J. Sussmann. *Differential Geometric Control Theory.* Boston: Birkhauser, 1983.

[2] E. D. Dickmanns and V. Graefe. Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1(4):241–261, 1988.

[3] E. D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1(4):223–240, 1988.

[4] E. D. Dickmanns and B. D. Mysliwetz. Recursive 3-d road and relative ego-state estimation. *IEEE Transactions on PAMI*, 14(2):199–213, February 1992.

[5] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313 - 326, June 1992.

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

[7] B. K. Ghosh and E. P. Loucks. A perspective theory for motion and shape estimation in machine vision. *SIAM Journal on Control and Optimization*, 33(5):1530–1559, Sept. 1995.

[8] John Hauser and Rick Hindman. Maneuver regulation from trajectory tracking: Feedback linearization systems. *A Postprint Volume from the 3rd IFAC Symposium Proceedins of IFAC Symposium on Nonlinear Control Systems Design*, 2:595–600, 1996.

[9] D. Heeger and A. Jepson. Subspace methods for recovering rigid motion. *International Journal of Computer Vision*, 7(2):95–117, Jan. 1992.

[10] Alberto Isidori. *Nonlinear Control Systems.* Communications and Control Engineering Series. Springer-Verlag, second edition, 1989.

[11] A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* NY: Academic Press, 1970.

[12] Jerry M. Mendel. *Lessons in Digital Estimation Theory.* Prentice-Hall Signal Processing Series. Prentice-Hall, first edition, 1987.

[13] Richard M. Murray, Zexiang Li, and Shankar S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC press Inc., 1994.

[14] Richard M. Murray and Shankar Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, May 1993.

[15] R. Pissard-Gibollet and P. Rives. Applying visual servoing techniques to control a mobile hand-eye system. In *Proceedings of the IEEE International Conference on Robotics and Automation, Nagoya, Japan*, volume 1, pages 166–171, May 1995.

[16] D. Raviv and M. Herman. A "non-reconstruction" approach for road following. In Proceedings of the SPIE, editor, *Intelligent Robots and Computer Vision*, volume 1608, pages 2–12, 1992.

[17] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control The Task Function Approach.* Oxford Engineering Science Series. Claderon Press, 1991.

[18] Shankar S. Sastry. *Nonlinear Systems: Analysis, Stability and Control.* Addison Wesley, to be published.

[19] S. Soatto, R. Frezza, and P. Perona. Motion estimation via dynamic vision. *IEEE Transactions on Automatic Control*, 41(3):393–413, March 1996.

[20] S. Soatto, R. Frezza, and P. Perona. Visual navigation by controlling apparent shape. *UC Berkeley AI/Robotics/Vision Seminar Notes*, October 1996.

[21] S. Soatto and P. Perona. Recursive estimation of camera motion from uncalibrated image sequences. In *Proceedins ICIP-94*, volume 3, pages 58–62, Nov. 1994.

[22] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2), November 1992.

[23] G. Walsh, D. Tilbury, S. Sastry, R. Murray, and J. P. Laumond. Stabilization of trajectories for systems with nonholonomic constraints. *IEEE Transactions on Automatic Control*, 39(1):216–222, January 1994.

A Matlab Source Code for the Extended Kalman Filter

The following source code is used for estimating the image quantities ξ_1, ξ_2, ξ_3, and η (the curvature varying rate c). The parameters shown are the ones used for the simulation of Section 5.2.4 Figure 10.

Main File: kalman.m

```
%Simulation Steps:
N    = 15000;
step = 0.001;
%System dimensions:
n    = 4;
%Number of outputs:
no   = 5;
%Measurements distances:
Dy   = 0.1;
%Variance of the disturbance on \eta:
sigmaeta = 1;

%Initialization of the estimate states:
Xf = zeros(n, N);
Pf = zeros(n, n*N);
Xi = [0; 0; 0; 0.1];
Pi = 10.*eye(n,n);
Xf(:, 1)   = Xi;
Pf(:, 1:n) = Pi;

%Initialization of the nominal states:
xi       = zeros(3, N);
xi0      = [0.1; 1; 1];
xi(:, 1) = xi0;

%Parameters for the system:
phi    = pi/3;
a      = 1/sin(phi);
C      = 0;
Kv     = 0.5;
Komega = 1;
v0     = 1;

%Simulation Loop:
for L  = 1 : N-1

omega0 = Xf(3,L)*sin(phi)^2*v0;
v      = v0 + sin(phi)^2*Xf(1,L)*(Xf(1,L) + Xf(3,L))*v0 -
         Kv*Xf(2,L)*sign(Xf(1,L)+Xf(3,L));
omega  = omega0 + sin(phi)^2*Xf(1,L)*v0 + Komega*Xf(2,L);
Ut = [omega; v];

%linearize and discretize the nominal system:
xi4   = (C*(a^2 + xi(2,L)^2)^3/a + 3*xi(2,L)*xi(3,L)^2)
        /(a^2 + xi(2,L)^2);
xi42  = 2*xi(2,L)*C*(a^2 + xi(2,L)^2)/a + 3*(a^2 - xi(2,L)^2)
        *xi(3,L)/(a^2 + xi(2,L)^2)^2;
xi43  = 6*xi(2,L)*xi(3,L)/(a^2 + xi(2,L)^2);
Fxi   = [-xi(2,L)*sin(phi)*omega, -xi(1,L)*sin(phi)*omega +
         sin(phi)*omega, 0;
         -xi(3,L)*sin(phi)*omega, -2*xi(2,L)*sin(phi)*omega,
         -xi(1,L)*sin(phi)*omega + sin(phi)*v;
         -xi4*sin(phi)*omega, -xi42*xi(1,L)*sin(phi)*omega -
         3*xi(3,L)*sin(phi)*omega + xi42*sin(phi)*v,
         -xi43*xi(1,L)*sin(phi)*omega - 3*xi(2,L)*sin(phi) +
         xi43*sin(phi)*v];
Fxit  = expm(step*Fxi);
Dxiu  = [-xi(1,L)*xi(2,L)*sin(phi), xi(2,L)*sin(phi);
         -(xi(1,L)*xi(3,L)*sin(phi) + xi(2,L)^2*sin(phi) +
         1/sin(phi)), xi(3,L)*sin(phi);
         -(xi4*xi(1,L)*sin(phi)+3*xi(2,L)*xi(3,L)*sin(phi)),
         xi4*sin(phi)];
Dxiut = Dxiu*step + Fxi*Dxiu*step^2/2;
xi(:,L+1) = xi(:,L) + Dxiut*Ut;

%linearize and discretize the kalman system:
xif4  = (Xf(4,L)*(a^2 + Xf(2,L)^2)^3/a + 3*Xf(2,L)*Xf(3,L)^2)
        /(a^2 + Xf(2,L)^2);
xif42 = 2*Xf(2,L)*Xf(4,L)*(a^2 + Xf(2,L)^2)/a + 3*(a^2 -
        Xf(2,L)^2)*Xf(3,L)/(a^2 + Xf(2,L)^2)^2;
xif43 = 6*Xf(2,L)*Xf(3,L)/(a^2 + Xf(2,L)^2);
xif44 = (a^2 + Xf(2,L)^2)^2/a;

Fxf = [-Xf(2,L)*sin(phi)*omega, -Xf(1,L)*sin(phi)*omega +
       sin(phi)*omega, 0, 0;
       -Xf(3,L)*sin(phi)*omega, -2*Xf(2,L)*sin(phi)*omega,
       -Xf(1,L)*sin(phi)*omega + sin(phi)*v, 0;
       -xif4*sin(phi)*omega, -xif42*Xf(1,L)*sin(phi)*omega -
       3*Xf(3,L)*sin(phi)*omega + xif42*sin(phi)*v,
       -xif43*Xf(1,L)*sin(phi)*omega - 3*Xf(2,L)*sin(phi) +
       xif43*sin(phi)*v, -xif44*Xf(1,L)*sin(phi)*omega +
       xif44*sin(phi)*v; 0, 0, 0, 0];
Ft  = expm(step*Fxf);
Gt  = [0; 0; 0; 1];
Du  = [-Xf(1,L)*Xf(2,L)*sin(phi), Xf(2,L)*sin(phi);
       -(Xf(1,L)*Xf(3,L)*sin(phi) + Xf(2,L)^2*sin(phi) +
       1/sin(phi)), Xf(3,L)*sin(phi);
       -(xif4*Xf(1,L)*sin(phi)+3*Xf(2,L)*Xf(3,L)*sin(phi)),
       xif4*sin(phi); 0, 0];
Dt = Du*step + Fxf*Du*step^2/2;

%Five output measurements:
Yt = [1,      1,          1,          1,          1;
      Dy,     2*Dy,       3*Dy,       4*Dy,       5*Dy;
      Dy^2/2, (2*Dy)^2/2, (3*Dy)^2/2, (4*Dy)^2/2, (5*Dy)^2/2;
      Dy^3/6, (2*Dy)^3/6, (3*Dy)^3/6, (4*Dy)^3/6, (5*Dy)^3/6]';
%One output measurement:
%Yt = [1, Dy, Dy^2/2, Dy^3/6];

Hx = [1, 0,     0,     0;
      0, 1,     0,     0;
      0, 0,     1,     0;
      0, xif42, xif43, xif44];
Ht = Yt*Hx;
Qt = sigmaeta;
%Output noise covariance matrix:
Rt = 1*eye(no,no);

%Extended kalman filter algorithm:
Xp = Xf(:,L) + Dt*Ut;
Pp = Ft*Pf(:, (n*(L-1)+1):(n*L))*Ft' + Gt*Qt*Gt';
```

```
HP = Ht*Pp;
Kg = HP'/(HP*Ht'+Rt);

xip4 = (Xp(4)*(a^2 + Xp(2)^2)^3/a + 3*Xp(2)*Xp(3)^2)/(a^2 +
        Xp(2)^2);
xi4  = (C*(a^2 + xi(2,L+1)^2)^3/a + 3*xi(2,L+1)*xi(3,L+1)^2)
        /(a^2 + xi(2,L+1)^2);
Zh   = Yt*[Xp(1);Xp(2);Xp(3);xip4];
Z    = Yt*[xi(1,L+1);xi(2,L+1);xi(3,L+1);xi4];

Xf(:, L+1) = Xp + Kg*(Z - Zh);
tmp        = Pp - Kg*HP;
Pt         = tmp-tmp*Ht'*Kg' + Kg*Rt*Kg';
tmp        = (triu(Pt) + triu(Pt'))/2.0;
Pf(:,(n*L+1):(n*(L+1))) = -diag(diag(Pt)) + tmp + tmp';

end

%Plot the outputs: nominal states v.s. estimate states:
figure(1);
clf;
w = 10000;
subplot(2,2,1), plot(1:w,xi(1,1:w), 'y-'), hold on,
plot(1:w,Xf(1,1:w),'g--'), ylabel('xi_1 and estimate of xi_1'),
xlabel('Time step = 0.001'), grid;
subplot(2,2,2), plot(1:w,xi(2,1:w), 'y-'), hold on,
plot(1:w,Xf(2,1:w),'g--'), ylabel('xi_2 and estimate of xi_2'),
xlabel('Time step = 0.001'), grid;
subplot(2,2,3), plot(1:w,xi(3,1:w), 'y-'), hold on,
plot(1:w,Xf(3,1:w),'g--'), ylabel('xi_3 and estimate of xi_3'),
xlabel('Time step = 0.001'), grid;
subplot(2,2,4), plot(1:w,zeros(1,w), 'y-'), hold on,
plot(1:w,Xf(4,1:w),'g--'), ylabel('c and estimate of c'),
xlabel('Time step = 0.001'), grid;

%%% End of file kalman.m
```