Path Following Using Dynamic Transverse Feedback Linearization for Car-Like Robots

Adeel Akhtar, Christopher Nielsen, and Steven L. Waslander, *Member, IEEE*

***Abstract*—This paper presents an approach for designing path-following controllers for the kinematic model of car-like mobile robots using transverse feedback linearization with dynamic extension. This approach is applicable to a large class of paths and its effectiveness is experimentally demonstrated on a Chameleon R100 Ackermann steering robot. Transverse feedback linearization makes the desired path attractive and invariant, while the dynamic extension allows the closed-loop system to achieve the desired motion along the path.**

***Index Terms*—Control design, motion control, mobile robots, nonlinear control systems, nonlinear dynamical systems, nonlinear systems, robot control.**

I. Introduction

THE problem of generating accurate motion along a given path for a control system can be broadly classified as either a path-following problem or a trajectory-tracking problem [1]. In a path-following problem, unlike trajectory tracking, the main task of the controller is to follow a path with no *a priori* time parameterization associated with the motion along the path.

The extra degree of freedom in path following of assigning the timing law associated with path traversal allows a significant improvement in the achievable performance for nonminimum phase systems [1], [2]. Another key advantage of adopting the path-following approach is that the path can be made an invariant set for the closed-loop system. In the context of mobile robotics, this means that once the mobile robot is on the path, with appropriate orientation, it never leaves the path. On the other hand, since a tracking controller tracks a specific system trajectory, if the robot is initialized on the path but its position does not coincide with the reference position, the robot may leave the path before asymptotically approaching the reference point on the path again [3]. In this paper, we design a path-following controller for the kinematic model of a car-like robot [4]. This model approximates the mobility of a car and is relevant in automated driving applications. Moreover, the car-like robot is the simplest nonholonomic vehicle that displays the general characteristics and the difficult maneuverability of higher dimensional systems, e.g., of a car towing trailers [4]. Accurate movement along a path is desirable for car-like robots when they operate in tight spatial conditions, like indoor robots moving in a room with obstacles. In these cases, path following can be used in conjunction with path planning to achieve collision-free motion.

Trajectory tracking and internal stability of the car-like robot were analyzed in [5]. The performance of the controller therein was tested both in simulation and on an experimental testbed. Path-following controllers were proposed for the car-like robot in [6]–[8]. The approach in [6] is similar to the one followed in this paper. The key difference is that we do not fix the translational velocity of the car, and consequently, the path can be rendered invariant while having variable dynamics along the path. In [9] and [10], a similar problem is solved in the presence of phase constraints and limited control resources. The car-like robot is treated as a single input system, and the translational velocity is a given, sign-definite, possibly time-varying, function. In [11], it was shown that transverse feedback linearization can be used to design path-following controllers for the car-like robot using only the steering input. In this paper, we provide explicit expressions for feedback control laws that achieve path following while allowing the motion along the path to change.[1]

Feedback linearization controllers are criticized because they only work "perfectly" in simulation, i.e., in the absence of disturbances and parameter uncertainty. The authors of [13] highlight that dynamically extended feedback linearized controllers can involve high-order derivative terms that can be sensitive to sensor noise and modeling uncertainty making them difficult to implement experimentally. Furthermore, the sensors used to estimate the states of car-like robots are relatively inaccurate with lower update rates. These practical constraints make the implementation of the proposed controller challenging. Experimental implementation of a path-following controller using sliding-mode control was presented in [8]. A reference-tracking and set-point regulation dynamic feedback linearization controller was presented in [14].

A large class of nonlinear systems fall in the category of differentially flat systems [15]. Finding a flat output is, in general, difficult and involves finding a function that satisfies the conditions given in [16]. The search for a flat output can be simplified by noting that they often have strong geometric interpretations [17]. In [18], a flatness-based approach is used to derive open-loop control laws for a kinematic car-like robot that are combined with interpolation using G^2-splines. In this paper, we choose a virtual output because it has very

Manuscript received May 26, 2014; revised October 30, 2014; accepted January 18, 2015. Date of publication February 16, 2015; date of current version April 2, 2015. This paper was recommended for publication by Associate Editor L. Pallottino and Editor T. Murphey upon evaluation of the reviewers' comments. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

A. Akhtar and S. L. Waslander are with the Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: a5akhtar@uwaterloo.ca; stevenw@uwaterloo.ca).

C. Nielsen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: cnielsen@uwaterloo.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2015.2395711

[1] A preliminary version of this paper, without Sections IV–VI, was presented in [12].

strong physical meaning for the path-following problem and, subsequently, show that it is a flat output. We use dynamic extension [19] of the original system to achieve the desired relative degree of the closed-loop system. Every system that is feedback linearizable via dynamic extension is differentially flat [20]. While we consider a kinematic model, the proposed controller can be extended to dynamic models using integrator backstepping [21].

A. Contributions

The main contributions of this paper are as follows: 1) an approach to designing path-following controllers for car-like vehicles that is physically intuitive and mathematically proven to achieve invariance of the path while traversing the path with desired dynamics; 2) a method for approximating arbitrary smooth parameterized paths as the zero level set of a function; and 3) experimental results that demonstrate accurate path following and feasibility of the proposed approach.

B. Notation

Let $\mathrm{col}(x_1, \ldots, x_n) = [\, x_1 \ \cdots \ x_n \,]^\top \in \mathbb{R}^n$, where $^\top$ denotes transpose. We denote the Euclidean inner product by $\langle x,\ y\rangle$ and the associated Euclidean norm by $\|x\|$. We let I_n represent the $n \times n$ identity matrix and $0_{m\times n}$ represent the $m \times n$ zero matrix. Given a set $A \subset \mathbb{R}^n$, the point-to-set distance to A is denoted $\|\cdot\|_A$. Given a function $f : A \to B$, we let $\mathrm{Im}(f)$ or $f(A)$ denote its image. A continuous function $\alpha : [0,\infty) \to [0,\infty)$ is said to belong to class-$\mathcal{K}_\infty$ if $\alpha(0) = 0$ and it is strictly increasing [22]. Given a C^1 mapping $\phi : \mathbb{R}^n \to \mathbb{R}^m$, let $d\phi_x$ be its Jacobian evaluated at $x \in \mathbb{R}^n$. If $f,\ g : \mathbb{R}^n \to \mathbb{R}^n$ are smooth vector fields, we use the following standard notation for iterated Lie derivatives $L_f^0\phi := \phi$, $L_f^k\phi := L_f(L_f^{k-1}\phi) = \langle dL_f^{k-1}\phi_x, f(x)\rangle$, $L_gL_f\phi := L_g(L_f\phi) = \langle dL_f\phi_x, g(x)\rangle$.

II. Problem Formulation

Consider the kinematic model[2] of a car-like robot with rear traction

$$\dot{x} = \begin{bmatrix} \cos x_3 & 0 \\ \sin x_3 & 0 \\ \frac{1}{\ell}\tan x_4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}$$

where $x \in \mathbb{R}^4$ is the state, the input $v \in \mathbb{R}$ is the translational speed, and $\omega \in \mathbb{R}$ is the angular velocity of the steering angle x_4. We impose a steering angle constraint

$$(\forall\, t \geq 0)\ -\frac{\pi}{2} < -\overline{x}_4 \leq x_4(t) \leq \overline{x}_4 < \frac{\pi}{2} \tag{2}$$

where $\overline{x}_4 > 0$ is given. We take the car's position in the plane as the output of (1)

$$y = h(x) = \mathrm{col}(x_1, x_2). \tag{3}$$

[2]We do not model friction because it has a negligible effect on vehicle performance for indoor scenarios with velocity and heading rate control inputs.

Suppose we are given a curve $\mathcal{C}$ in the output space $\mathbb{R}^2$ of (1) as a regular parameterized curve

$$\sigma : \mathbb{D} \to \mathbb{R}^2, \qquad \lambda \mapsto \mathrm{col}(\sigma_1(\lambda), \sigma_2(\lambda)) \tag{4}$$

where $\sigma \in C^r$ with $r \geq 3$ and $\mathcal{C} = \mathrm{Im}(\sigma)$. Since σ is regular, without loss of generality, we assume it is unit-speed parameterized, i.e., $\|\sigma'(\cdot)\| \equiv 1$. Under this assumption, the curve σ is parameterized by its arc length. For closed curves with finite length L, this means that $\mathbb{D} = \mathbb{R} \bmod L$ and σ is L-periodic, i.e., for any $\lambda \in \mathbb{D}$, $\sigma(\lambda + L) = \sigma(L)$. When the curve is not closed, $\mathbb{D} = \mathbb{R}$. We impose geometric restrictions on the class of curves considered [23].

Assumption 1 (submanifold): The curve $\mathcal{C}$ is a 1-D embedded submanifold of $\mathbb{R}^2$.

Assumption 1 imposes that the path has no self-intersections, no "corners," and does not approach itself asymptotically.

Assumption 2 (implicit representation): The curve $\mathcal{C} \subset \mathbb{R}^2$ has implicit representation $\mathcal{C} = \{y \in W : s(y) = 0\}$ where $s : W \subseteq \mathbb{R}^2 \to \mathbb{R}$, is a smooth function such that $ds_y \neq 0$ on $\mathcal{C}$ and W is an open set. Moreover, there exist two class-$\mathcal{K}_\infty$ functions $\alpha, \beta : [0,\infty) \to [0,\infty)$ such that

$$(\forall y \in W)\ \ \alpha(\|y\|_{\mathcal{C}}) \leq \|s(y)\| \leq \beta(\|y\|_{\mathcal{C}}). \tag{5}$$

Assumption 2 asks that the entire path be represented as the zero level set of the function s. This is always possible, locally, if Assumption 1 holds. The second part of Assumption 2 ensures that, when $\mathcal{C}$ is not bounded, $s(y) \to 0$ if and only if $y \to \mathcal{C}$.

Since $dh_x = I_2$ for the output (3), the map $h : \mathbb{R}^4 \to \mathbb{R}^2$ is transversal [24] to $\mathcal{C}$ and, therefore, if Assumption 1 holds, the *lift* of $\mathcal{C}$ to $\mathbb{R}^4$

$$\Gamma := (s \circ h)^{-1}(0) = \left\{x \in \mathbb{R}^4 : s(h(x)) = 0\right\} \tag{6}$$

is a 3-D submanifold.

Assumption 3 (curvature constraint [18], [25]): Given a steering angle constraint (2), the curvature $\kappa(\lambda)$ of (4) satisfies

$$(\forall\, \lambda \in \mathbb{D})\ \ \kappa(\lambda) < \frac{1}{\ell}\tan(\overline{x}_4). \tag{7}$$

Assumption 3 ensures that the path is feasible, in light of the steering angle constraint, for the car-like vehicle.

Problem 1: Given a curve $\mathcal{C}$ satisfying Assumptions 1–3 find, if possible, a smooth control law for (1), (3) of the form

$$\begin{aligned} \dot{\zeta} &= a(x,\zeta) + b(x,\zeta)u \\ \begin{bmatrix} v \\ \omega \end{bmatrix} &= c(x,\zeta) + d(x,\zeta)u \end{aligned} \tag{8}$$

with $\zeta \in \mathbb{R}^q$, $u = (u_1, u_2) \in \mathbb{R}^2$ such that for some open set of initial conditions $U \times V \subset \mathbb{R}^4 \times \mathbb{R}^q$ with $\mathcal{C} \subset h(U)$

PF1 The solution $(x(t), \zeta(t))$ of the closed-system (1), (8) exists for all $t \geq 0$ and $\|h(x(t))\|_{\mathcal{C}} \to 0$ as $t \to \infty$.

PF2 The curve $\mathcal{C}$ is output invariant independent of the desired motion along the path, i.e., if properly initialized, then $\|h(x(t))\|_{\mathcal{C}} = 0$ for all $t \geq 0$.

PF3 The system asymptotically tracks a given motion profile $\sigma(\lambda^{\mathrm{ref}}(t))$, where $\lambda^{\mathrm{ref}} : \mathbb{R} \to \mathbb{D}$ is smooth and $\dot{\lambda}^{\mathrm{ref}}(t)$ is uniformly bounded away from zero.

III. Differentially Flat Path-Following Outputs

The path-following manifold, denoted $\Gamma^\star$, associated with the curve $\mathcal{C}$ is the maximal controlled invariant subset of the lift (6). Physically, it consists of all those motions of the car-like robot (1) for which the output signal (3) can be made to remain on the curve $\mathcal{C}$ by suitable choice of control signal [3]. The path-following manifold is the key object that allows one to treat the path-following problem as a set stabilization problem. If the path-following manifold can be made attractive and controlled invariant for the closed-loop system, then **PF1** and **PF2** are satisfied.

When we apply the above definition to the car-like robot or, more generally, to any drift-less system, it is immediate that $\Gamma^\star = \Gamma$. This is because one can trivially make the entire set Γ controlled invariant by setting $v = 0$. This characterization of $\Gamma^\star$ cannot be used to solve Problem 1 because path invariance is not achieved *independently* of the motion along the path. Stabilizing Γ with $v|_\Gamma = 0$ ensures path invariance (**PF2**) but fixes the motion along the path.

On the other hand, when $v = \mathrm{v} \neq 0$ is a fixed constant, the path-following manifold can be characterized [11] using the steering input ω. Physically, this means that the car-like robot can be made to follow $\mathcal{C}$ solely using its steering input. The main deficiency with the solution presented in [11] is that **PF3** cannot be satisfied since v is fixed. To overcome this difficulty, a time-scaled transformation was applied in [9] and [10], which made it possible to ensure path invariance for variable speed $v(t)$.

To overcome this problem, let $v = \mathrm{v} + \zeta_1$, where ζ_1 is the first state of our dynamic controller and $\mathrm{v} \neq 0$ is constant. We take the simplest possible structure for the control law (8), and let $\dot{\zeta}_1 = \zeta_2$. In order to finish defining the control law, we let $\dot{\zeta}_2 = u_1$, where u_1 is a new auxiliary input. To simplify notation, henceforth, we do not distinguish between physical states of the system (x_1, x_2, x_3, x_4) and states of the controller (ζ_1, ζ_2). Let $x_5 := \zeta_1, x_6 := \zeta_2$. Therefore, the system we study has the form

$$\begin{aligned}\dot{x} =\ & f(x) + g_1(x)u_1 + g_2(x)u_2 \\ =\ & \begin{bmatrix} (\mathrm{v}+x_5)\cos x_3 \\ (\mathrm{v}+x_5)\sin x_3 \\ \frac{(\mathrm{v}+x_5)}{\ell}\tan x_4 \\ 0 \\ x_6 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u_2 \end{aligned} \tag{9}$$

and the set Γ in (6) is embedded in the extended state space $\mathbb{R}^6$. The dynamic extension allows us to enforce **PF2** independently of the function $\lambda^{\mathrm{ref}}(t)$ from **PF3**. Similar ideas have been applied to a tower crane model in [26].

A. Feedback Linearization

We treat the path-following problem as a set stabilization problem, and we follow the general approach of [3] and [23]. In order to satisfy **PF1** and **PF2**, we first stabilize the path-following manifold $\Gamma^\star$. Once the path manifold has been stabilized, we use the remaining freedom in the control law to impose desired dynamics on the path and satisfy **PF3**.

Let $\mathcal{N}(\mathcal{C}) \subset \mathbb{R}^2$ denote a neighborhood of the curve $\mathcal{C}$. The neighborhood $\mathcal{N}(\mathcal{C})$ has the property that if $y \in \mathcal{N}(\mathcal{C})$, then there exists a unique $y^\star \in \mathcal{C}$ such that $\|y\|_\mathcal{C} = \|y - y^\star\|$. This allows us to define the function

$$\begin{aligned} &\varpi :\ \mathcal{N}(\mathcal{C}) \to \mathbb{D} \\ &y \mapsto \arg\inf_{\lambda \in \mathbb{D}} \|y - \sigma(\lambda)\|. \end{aligned} \tag{10}$$

This function is as smooth as σ is which, by assumption, is at least C^3. Using (10), define the "path-following output"

$$\hat{y} = \begin{bmatrix} \pi(x) \\ \alpha(x) \end{bmatrix} = \begin{bmatrix} \varpi \circ h(x) \\ s \circ h(x) \end{bmatrix}. \tag{11}$$

Let $\Gamma_+ := \Gamma \cap \{x \in \mathbb{R}^6 : x_5 + \mathrm{v} > 0\}$, $\Gamma_- := \Gamma \cap \{x \in \mathbb{R}^6 : x_5 + \mathrm{v} < 0\}$. The next lemma shows that the output (11) yields a well-defined relative degree on $\Gamma_+ \coprod \Gamma_-$, where $\coprod$ denotes disjoint union.

Lemma III.1: The dynamic extension of the car-like robot (9) with output (11) yields a well-defined vector relative degree of $\{3, 3\}$ at each point on $\Gamma_+ \coprod \Gamma_-$.

Proof: Let $x^\star \in \Gamma_+ \coprod \Gamma_-$ be arbitrary. By definition of Γ, the output $h(x^\star)$ is on the path $\mathcal{C}$. Let $\lambda^\star \in \mathbb{D}$ be such that $h(x^\star) = \sigma(\lambda^\star)$. By the definition of vector relative degree, we must show that $L_{g_1} L_f^i \pi(x) = L_{g_2} L_f^i \pi(x) = L_{g_1} L_f^i \alpha(x) = L_{g_2} L_f^i \alpha(x) = 0$ for $i \in \{0, 1\}$ in a neighborhood of $x^\star$ and that the decoupling matrix

$$D(x) = \begin{bmatrix} L_{g_1} L_f^2 \pi(x) & L_{g_2} L_f^2 \pi(x) \\ L_{g_1} L_f^2 \alpha(x) & L_{g_2} L_f^2 \alpha(x) \end{bmatrix} \tag{12}$$

is nonsingular at $x = x^\star$. Since

$$\frac{\partial \pi(x)}{\partial x_i} = \frac{\partial \alpha(x)}{\partial x_i} \equiv 0$$

for $i \in \{3, 4, 5, 6\}$, it is easy to check that $L_{g_j} L_f^i \pi(x) = L_{g_j} L_f^i \alpha(x) = 0$ for $i \in \{0, 1\}$, $j \in \{1, 2\}$.

To show that the decoupling matrix (12) is nonsingular at $x = x^\star$, we first find that

$$\det(D(x)) = \frac{(\mathrm{v} + x_5)^2}{\ell \cos^2 x_4} \left(\sigma_1'(\lambda^\star)\partial_{x_2}\alpha - \sigma_2'(\lambda^\star)\partial_{x_1}\alpha\right).$$

The only way for this determinant to vanish is if either 1) $\mathrm{v} = -x_5$ or 2) $\sigma_1'(\lambda^\star)\partial_{x_2}\alpha - \sigma_2'(\lambda^\star)\partial_{x_1}\alpha = 0$. Condition 1 does not occur for $x \in \Gamma_+ \coprod \Gamma_-$. We now argue that condition 2 never occurs on the path because the vectors $\mathrm{col}(\partial_{x_1}\alpha, \partial_{x_2}\alpha)$ and $\sigma'(\lambda^\star)$ are orthogonal.

The chain rule and the form of the output map (3) yield $\mathrm{col}(\partial_{x_1}\alpha(x^\star), \partial_{x_2}\alpha(x^\star)) = ds_{h(x^\star)}^\top$. By Assumption 2, the differential $ds_y \neq 0$ for $y \in \mathcal{C}$. Thus, the vector $ds_{h(x^\star)}^\top$ is a nonzero gradient vector and is orthogonal to the path at $h(x^\star)$. On the other hand, the vector $\sigma'(\lambda^\star)$ is nonzero because σ is regular and also tangent to the curve. Hence $\langle ds_{h(x^\star)}^\top,\ \sigma'(\lambda^\star)\rangle = 0$. If we rotate the vector $ds_{h(x^\star)}^\top$ by $\pi/2$ rad, then the rotated vector

and $\sigma'(\lambda^\star)$ are linearly dependent. Let $R_{\frac{\pi}{2}}$ be a rotation of the plane by $\pi/2$. Then

$$R_{\frac{\pi}{2}} ds^\top_{h(x^\star)} = k(\sigma(\lambda^\star))\sigma'(\lambda^\star)$$

for some smooth, scalar-valued, nonzero function $k : \mathbb{R}^2 \to \mathbb{R}$. The function k is never equal to zero because the vector $ds^\top_{h(x^\star)}$ is never zero.

Returning to the expression for $\det(D(x))$, we have that

$$\begin{aligned}\sigma_1'(\lambda^\star)\partial_{x_2}\alpha - \sigma_2'(\lambda^\star)\partial_{x_1}\alpha &= \left\langle R_{\frac{\pi}{2}} ds^\top_{h(x^\star)},\ \sigma'(\lambda^\star)\right\rangle \\ &= k(\sigma(\lambda^\star))\langle\sigma'(\lambda^\star),\ \sigma'(\lambda^\star)\rangle \\ &= k(\sigma(\lambda^\star))\|\sigma'(\lambda^\star)\|^2 \\ &= k(\sigma(\lambda^\star)).\end{aligned}$$

■

Let $\Gamma^\star := \left\{x \in \mathbb{R}^6 : \alpha(x) = L_f\alpha(x) = L_f^2\alpha(x) = 0\right\}$. Define $\Gamma^\star_+ := \Gamma^\star \cap \Gamma_+$ and $\Gamma^\star_- := \Gamma^\star \cap \Gamma_-$. The next result defines a diffeomorphism valid in a neighborhood of $\Gamma^\star_+$. The equivalent result for $\Gamma^\star_-$ is omitted to avoid repetition.

Corollary III.2: Let $x^\star \in \Gamma^\star_+$. There exists a neighborhood $U_+ \subset \mathbb{R}^6$ containing $\Gamma^\star_+$ such that $T : U_+ \to T(U_+)$

$$\begin{bmatrix}\eta_1\\ \eta_2\\ \eta_3\\ \xi_1\\ \xi_2\\ \xi_3\end{bmatrix} = T(x) = \begin{bmatrix}\pi(x)\\ L_f\pi(x)\\ L_f^2\pi(x)\\ \alpha(x)\\ L_f\alpha(x)\\ L_f^2\alpha(x)\end{bmatrix} \tag{13}$$

is a diffeomorphism onto its image.

Proof: In order to show that (13) is a diffeomorphism in a neighborhood of $\Gamma^\star_+$, we appeal to the generalized inverse function theorem [24, p. 56]. We must show that 1) for all $x \in \Gamma^\star_+$, dT_x is an isomorphism, and 2) $T|_{\Gamma^\star_+} : \Gamma^\star_+ \to T(\Gamma^\star_+)$ is a diffeomorphism. An immediate consequence of Lemma III.1 and [19, Lemma 5.2.1] is that the first condition holds. To show that the second condition holds, we explicitly construct the inverse of T restricted to $\Gamma^\star_+$. On $\Gamma^\star_+$, $\xi_1(x) = \xi_2(x) = \xi_3(x) = 0$, and simple calculations show that the inverse of T restricted to $\Gamma^\star_+$ is[3]

$$\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\end{bmatrix} = T|^{-1}_{\Gamma^\star_+}(\eta, 0) = \begin{bmatrix}\sigma_1(\eta_1)\\ \sigma_2(\eta_1)\\ \varphi(\eta_1)\\ \arctan(\ell\kappa(\eta_1))\\ \eta_2 - \mathrm{v}\\ \eta_3\end{bmatrix}$$

where $\varphi : \mathbb{D} \to \mathbb{R} \bmod 2\pi$ is the map that associates to each $\eta_1 \in \mathbb{D}$, the angle of the tangent vector $\sigma'(\eta_1)$ to $\mathcal{C}$ at $\sigma(\eta_1)$, and $\kappa : \mathbb{D} \to \mathbb{R}$ is the signed curvature. The inverse is clearly smooth, which shows that $T|_{\Gamma^\star_+}$ is a diffeomorphism onto its image. ■

This coordinate transformation of Corollary III.2 is physically meaningful for path-following applications. When $\xi = 0$, the system is restricted to evolve on the path-following manifold $\Gamma^\star$. We call the ξ-subsystem the transversal subsystem and the states ξ the transversal states. On the path-following manifold, the motion of the car-like robot on the path is governed by the η-dynamics. We call the η-subsystem the tangential subsystem and states η the tangential states. When the robot is on the path-following manifold, i.e., $\xi = 0$, then η_1 determines the position of the robot on the path, η_2 represents velocity of the robot along the path, and η_3 represents acceleration of the robot along the path.

We apply the regular feedback transformation

$$\begin{bmatrix}u_1\\ u_2\end{bmatrix} := D^{-1}(x)\left(\begin{bmatrix}-L_f^3\pi\\ -L_f^3\alpha\end{bmatrix} + \begin{bmatrix}v^{\parallel}\\ v^{\pitchfork}\end{bmatrix}\right) \tag{14}$$

where $(v^{\parallel}, v^{\pitchfork}) \in \mathbb{R}\times\mathbb{R}$ are auxiliary control inputs. By Lemma III.1, this controller is well defined in the neighborhood of $\Gamma^\star_+$ from Corollary III.2. Thus, in a neighborhood of $\Gamma^\star_+$ [respectively, $\Gamma^\star_-$], the closed-loop system becomes

$$\begin{aligned}\dot\eta_1 &= \eta_2, & \dot\xi_1 &= \xi_2\\ \dot\eta_2 &= \eta_3, & \dot\xi_2 &= \xi_3\\ \dot\eta_3 &= v^{\parallel}, & \dot\xi_3 &= v^{\pitchfork}\end{aligned} \tag{15}$$

We refer to the control input $v^{\pitchfork}$ as the transversal input and $v^{\parallel}$ as the tangential input. The control law (14) has decoupled the transversal and tangential subsystems, which makes designing $(v^{\parallel}, v^{\pitchfork})$ to solve Problem 1 particularly easy. In summary, we have shown that the extended car-like robot is differentially equivalent to a controllable linear time-invariant system in a neighborhood of each connected component, $\Gamma^\star_+$ and $\Gamma^\star_-$, of the path-following manifold. Another way to state this is to say that the output (11) is a flat output for the car-like robot (1) [27], [28].

B. Transversal and Tangential Control Design

The objective of the transversal controller is to force the system to converge to the path. For that, we to stabilize the origin of the transversal subsystem. The simplest choice for the transversal input is

$$v^{\pitchfork}(\xi) = k_1\xi_1 + k_2\xi_2 + k_3\xi_3 \tag{16}$$

with $k_i < 0$, $i \in \{1, 2, 3\}$, chosen so that the polynomial $s^3 - k_3s^2 - k_2s - k_1$ is Hurwitz. This controller exponentially stabilizes $\xi = 0$ and hence, under (5) in Assumption 2, makes the path-following manifold attractive. These gains can be chosen using, for instance, pole-placement or quadratic optimization (LQR).

Assumption 4 (desired motion on path): The desired motion on $\mathcal{C}$ given by a smooth function $\eta^{\mathrm{ref}}(t) := (\lambda^{\mathrm{ref}}(t), \dot\lambda^{\mathrm{ref}}(t), \ddot\lambda^{\mathrm{ref}}(t))$, $t \geq 0$ with $|\dot\lambda^{\mathrm{ref}}|$ uniformly bounded away from zero.

[3]The inverse is obtained under the assumption that the curve is arc-length parameterized.

Assumption 4 ensures that $x_5 + \mathrm{v} \neq 0$ for the desired motion. When $x_5 + \mathrm{v} = 0$, the robot has no translational velocity, the decoupling matrix loses rank, and the control law (14) is not well defined. Given a desired motion that satisfies Assumption 4, let

$$\begin{aligned} v^{\parallel}(\eta) = \; & k_4(\eta_1 - \eta_1^{\mathrm{ref}}(t)) + k_5(\eta_2 - \dot{\eta}_1^{\mathrm{ref}}(t)) \\ & + k_6(\eta_3 - \ddot{\eta}_1^{\mathrm{ref}}(t)) + \dddot{\eta}_1^{\mathrm{ref}}(t) \end{aligned} \tag{17}$$

with gains $k_i < 0$, $i \in \{4, 5, 6\}$, chosen so that the polynomial $s^3 - k_6 s^2 - k_5 s - k_4$ is Hurwitz. The numerical value of these gains can be chosen similarly to the transversal gains. Typically, one seeks that the closed-loop transversal dynamics converge to zero faster than the closed-loop tangential dynamics.

Proposition III.3: The control law (14), (16), (17) solves Problem 1.

Proof: Assume that[4] $\dot{\eta}_1^{\mathrm{ref}} > 0$ and let $x(0) \in U_+$, where $U_+ \subseteq \mathbb{R}^6$ is defined in Corollary III.2. By Lemma III.1, and by shrinking U_+ if necessary, the control law (14), (16), (17) is well defined in U_+. The transversal controller (16) exponentially stabilizes $\xi = 0$ and hence, by Assumption 2, $x \to \Gamma_+^\star \subset U_+$ and **PF1** holds. Since $\xi = 0$ is an equilibrium of the closed-loop transversal subsystem, if $x(0) \in \Gamma_+^\star \subset U_+$, $h(x(t))$ remains on the path for all future time. Therefore, (16) achieves **PF1**, **PF2** in Problem 1.

Define errors coordinates $e_\eta := \eta - \eta^{\mathrm{ref}}$. It is straightforward to show that, for the tangential controller (17), $e_\eta \to 0$. Once again, by shrinking U_+ if necessary, it is possible to ensure that $\eta_2(t) \neq 0$ during the transient phase in which $\eta_2 \to \dot{\eta}_1$. This shows that the closed-loop tangential dynamics satisfy **PF3**, and hence, Problem 1 is solved. ∎

Remark III.4: When the desired motion corresponds to velocity tracking, then $\eta^{\mathrm{ref}}(t) = (0, \eta_2^{\mathrm{ref}}(t), \dot{\eta}_2^{\mathrm{ref}}(t))$. In this case, we select $k_4 = 0$. Similar comments apply to acceleration tracking.

Remark III.5: Proposition III.3 shows that the region of attraction[5] is an open subset of U_+ [respectively, U_-]. Since (13) relies (10), the image of this set under (3) must be a neighborhood of $\mathcal{C}$ in which the closest point on the path is well defined. This is a necessary, far from sufficient, property of the region of attraction.

IV. Curve Representation

The control design technique discussed in this paper relies on having both a parametric representation [see (4)] and an implicit representation (see Assumption 2) of the path $\mathcal{C}$. Although such curve representation pairs are well known for many commonly used paths such as circles and Cassini ovals, not all cases can be addressed this way.

Given an arbitrary curve $\mathcal{C}$ in $\mathbb{R}^2$ with a C^r, $r \geq 3$, regular parameterization (4), we provide the following procedure for finding its implicit representation. First, we approximate the parametric representation as a rational parametric curve using the Weierstrass approximation theorem. Second, relying on elimination theory [30], we represent the image of the rational approximation as an implicit function.

A. Polynomial Approximation

Given $\sigma(\lambda) = \mathrm{col}(\sigma_1(\lambda), \sigma_2(\lambda))$, we generate polynomial approximations $p_j(\lambda)$ to the functions $\sigma_j : \mathbb{D} \to \mathbb{R}$, $j \in \{1, 2\}$. We start by sampling the domain $\mathbb{D}$. Let $\{\lambda_1, \lambda_2, \ldots, \lambda_{q+1}\}$ be $q+1$ points in $\mathbb{D}$ with $\lambda_i < \lambda_{i+1}$, $i \in \{1, \ldots, q\}$. Let $I := [\lambda_1, \lambda_{q+1}] \subset \mathbb{R}$. The associated points on $\mathcal{C}$ are given by $\sigma(\lambda_i) = \mathrm{col}(\sigma_1(\lambda_i), \sigma_2(\lambda_i))$, $i \in \{1, \ldots, q+1\}$. If we seek a single polynomial $p_j(\lambda)$ of fixed order N that approximates $\sigma_j(\lambda)$ at the sample points, then we can simply solve two least-squares optimization problems for $p_j(\lambda) = \sum_{i=0}^{N} a_i^j \lambda^i$, $j \in \{1, 2\}$ to find the coefficients $a_i^j \in \mathbb{R}$. The Weierstrass approximation theorem [31] ensures that for any $\epsilon > 0$, there exists N sufficiently large such that

$$\max_{\lambda \in I} \|\sigma_j(\lambda) - p_j(\lambda)\| < \epsilon. \tag{18}$$

Furthermore, at the sample points $\sigma(\lambda_i)$, the above polynomial approximation is optimal in the least-squares sense.

A drawback of the above approach is that, for a given set $\{\lambda_1, \lambda_2, \ldots, \lambda_{q+1}\}$ and a given $\epsilon > 0$, the order N of the polynomial required to ensure (18) holds is *a priori* unknown. In such cases, we propose an algorithm to recursively compute polynomials p_j that satisfy (18).

Our algorithm is based on the constructive proof of Weierstrass approximation theorem presented in [31]. In that proof, given a uniformly continuous function $f : [0, 1] \to \mathbb{R}$, one constructs a Bernstein polynomial

$$B_n^f(\theta) := \sum_{k=0}^{n} c_k h_n^k(\theta), \tag{19}$$

where $c_k := f(k/n)$ and $h_n^k(\theta) := \binom{n}{k} \theta^k (1-\theta)^{n-k}$, which is shown to converge uniformly to f as the order of B_n^f gets sufficiently large. In our application, on the interval $I = [\lambda_1, \lambda_{q+1}]$, define the function $\tau : [0, 1] \to I$ as

$$\tau(\theta) = \lambda_1 + \theta(\lambda_{q+1} - \lambda_1).$$

This function is a homeomorphism between $[0, 1]$ and I. We use it to define

$$f_j(\theta) := \sigma_j \circ \tau(\theta), \qquad j \in \{1, 2\}. \tag{20}$$

Using the Weierstrass approximation theorem and Bernstein polynomials for the function (20), we have that for any $\epsilon > 0$, there exists an integer $N > 0$ such that

$$(\forall n \geq N) \max_{\theta \in [0,1]} \|f_j(\theta) - B_n^{f_j}(\theta)\| < \epsilon. \tag{21}$$

These arguments lead to the following conclusion.

Lemma IV.1: There exists a positive finite integer N such that (18) holds with

$$p_j(\lambda) = B_n^{f_j}(\tau^{-1}(\lambda)), \qquad n \geq N, \qquad j \in \{1, 2\}.$$

Algorithm 1, given below, shows how to use Lemma IV.1 to find the polynomials p_j.

[4] If $\dot{\eta}_1^{\mathrm{ref}} < 0$, then the proof is the same, *mutatis mutandis*, using $U_- \subseteq \mathbb{R}^6$.

[5] See [29] for further discussion on estimating the region of attraction.

Algorithm 1: Curve approximation

input : $\sigma_j : \mathbb{D} \to \mathbb{R}$
$\epsilon > 0$
$N = 1$; (start with the smallest possible order)
$I = [\lambda_1, \lambda_{q+1}]$
output : $p_j(\lambda)$
compute: $f_j = \sigma_j \circ \tau(\theta)$
while $error > \epsilon$ **do**
  for $k = 0 : N$ **do**
    compute: $c_k = f_j(k/N)$
    compute: $h_N^k(\theta) = \binom{N}{k}\theta^k(1-\theta)^{N-k}$
    compute: $p_j(\lambda) = p_j(\lambda) + c_k h_N^k(\tau^{-1}(\lambda))$
  end
  calculate error: $\max_{\lambda \in I} \|\sigma_j(\lambda) - p_j(\lambda)\|$
  $N = N + 1$
end

B. Sylvester Matrix Elimination Method

We now apply elimination theory to form an implicit representation of the planar curves obtained in Section IV-A. There are multiple ways to accomplish this, but we present Sylvester's method [32].

Sylvester's dialytic expansion [32] computes the resultant of a given polynomial system by constructing a matrix that is rank deficient whenever the polynomial system has a solution. Consider two polynomials constructed using the ideas of Section IV-A $p_1(\lambda) = \sum_{i=0}^{n} a_i \lambda^i$, $p_2(\lambda) = \sum_{i=1}^{m} b_i \lambda^i$. Let $y_1 = p_1(\lambda)$ and $y_2 = p_2(\lambda)$ and rewrite the polynomials as

$$P_1(\lambda) : = a_n \lambda^n + \cdots + a_1 \lambda + (a_0 - y_1) = 0$$
$$P_2(\lambda) : = b_m \lambda^m + \cdots + b_1 \lambda + (b_0 - y_2) = 0. \quad (22)$$

The key insight in [33] is that, by viewing the terms $(a_0 - y_1)$ and $(b_0 - y_2)$ in (22) as constant terms, the associated resultant expresses the relationship, which must exists among the coefficients in order for there to exist λ that simultaneously satisfies both equations (22). In other words, the resultant itself is the implicit form of the parametric curve.

Let $\mathrm{Syl}(P_1, P_2, \lambda)$ denote the $(n+m) \times (n+m)$ Sylvester matrix of P_1 and P_2 with respect to λ. Then, the resultant of P_1 and P_2 with respect to λ is denoted by $\mathrm{Res}(P_1, P_2, \lambda)$ and is given by $\mathrm{Res}(P_1, P_2, \lambda) = \det(\mathrm{Syl}(P_1, P_2, \lambda))$.

In summary, we use the following two-step approach to implicitize a curve $\mathcal{C} \subset \mathbb{R}^2$ with regular parameterization (4).

1) Approximate the function $\sigma_j : \mathbb{D} \to \mathbb{R}$, $j \in \{1, 2\}$ as a univariate polynomial $p_j(\lambda)$ over a compact interval. Do this using least-squares optimization or Algorithm 1.
2) Form the Sylvester matrix $\mathrm{Syl}(P_1, P_2, \lambda)$ using (22); then, set $s(y) = \mathrm{Res}(P_1, P_2, \lambda) = \det(\mathrm{Syl}(P_1, P_2, \lambda))$ as the zero-level set representation of the approximation of $\mathcal{C}$ obtained above.

V. Implementation Issues

In order to implement the proposed controller, the coordinate transformation (13), the feedback (14) with $D(x)$ defined in (12), and transversal and tangential controllers (16), (17) must be computed. In this section, we address two main issues that complicate implementing the above. The first is that the parameterization of $\mathcal{C}$ may not be unit-speed. The second is that the computation of the tangential states (η_1, η_2, η_3) involves computing the projection (10) and its derivatives. In general, the function (10) does not have a closed form, which makes these calculations nonobvious.

(a)

(b)

Fig. 1. Chameleon R100 robot and the experimental setup are shown. The position and orientation of the robot is measured by the IPS. (a) Chameleon R100 robot. (b) Experimental setup.

Following the discussion in Section IV, we assume that we are given $\mathcal{C}$, a not-necessarily unit-speed parameterization $\widetilde{\sigma} : \mathbb{R} \to \mathbb{R}^2$, and a function $s : \mathbb{R}^2 \to \mathbb{R}$ such that Assumption 2 holds. Note that having an expression for $s(y)$ makes the computation of the transversal states straightforward using symbolic algebra software.

Let $\mathcal{N}(\mathcal{C}) \subseteq \mathbb{R}^2$ be a neighborhood of $\mathcal{C}$. Now, introduce a projection operator, defined in $\mathcal{N}(\mathcal{C})$, which is the same as (10) except it uses a nonunit speed parameterization

$$\lambda^\star = \widetilde{\varpi}(y) = \arg \inf_{\lambda \in \mathbb{R}} \|y - \widetilde{\sigma}(\lambda)\|. \quad (23)$$

The value $\lambda^\star$ can be effectively numerically computed using line search algorithms. For closed curves, this calculation is straightforward because the line search is over a compact interval of $\mathbb{R}$. For nonclosed curves, heuristic methods must be employed to compute the infimum. To calculate the first tangential state, we find the arc length

$$\eta_1 = g(\lambda^\star) := \int_0^{\lambda^\star} \left\| \frac{d\widetilde{\sigma}}{d\lambda} \right\| du \quad (24)$$

so that $\eta_1 = g \circ \widetilde{\varpi} \circ h(x)$. To calculate η_2, we note

$$\eta_2 = \frac{\partial(g \circ \widetilde{\varpi} \circ h)}{\partial x}\frac{dx}{dt} = \left(\frac{\partial g}{\partial \lambda}\right)\Big|_{\lambda=\lambda^\star} \left(\frac{\partial \widetilde{\varpi}}{\partial y}\right)\Big|_{y=h(x)} \begin{bmatrix} (\mathrm{v}+x_5)\cos(x_3) \\ (\mathrm{v}+x_5)\sin(x_3) \end{bmatrix}.$$

Simple geometric arguments, similar to those used in the proof of Lemma III.1, show that $\frac{\partial \widetilde{\varpi}}{\partial y}\Big|_y$ is given by

$$\frac{\partial \widetilde{\varpi}}{\partial y} = \frac{(\widetilde{\sigma}'(\lambda^\star))^\top}{\|\widetilde{\sigma}'(\lambda^\star)\|^2}. \tag{25}$$

Differentiating (24), one obtains

$$\frac{\partial g}{\partial \lambda}\Big|_{\lambda=\lambda^\star} = \|\widetilde{\sigma}'(\lambda^\star)\| \tag{26}$$

and therefore

$$\eta_2 = \frac{(\widetilde{\sigma}'(\lambda^\star))^\top}{\|\widetilde{\sigma}'(\lambda^\star)\|} \begin{bmatrix} (\mathrm{v}+x_5)\cos(x_3) \\ (\mathrm{v}+x_5)\sin(x_3) \end{bmatrix}. \tag{27}$$

To simplify notation, let

$$\Delta(x) := \frac{(\widetilde{\sigma}'(\lambda^\star))^\top}{\|\widetilde{\sigma}'(\lambda^\star)\|}, \qquad \Omega := \begin{bmatrix} (\mathrm{v}+x_5)\cos(x_3) \\ (\mathrm{v}+x_5)\sin(x_3) \end{bmatrix}.$$

To find η_3, we differentiate (27) and get $\eta_3 = \dot{\Delta}\Omega + \Delta\dot{\Omega}$. The term $\dot{\Omega}$ is easy to compute using the system dynamics (9). The term $\dot{\Delta} = \Delta'\dot{\lambda}$ can be found by noting that

$$\Delta' := \frac{\partial \Delta}{\partial \lambda} = \frac{(\widetilde{\sigma}'')^\top \|\widetilde{\sigma}'\|^2 - (\widetilde{\sigma}')^\top \sum_{i=1}^2 \widetilde{\sigma}_i' \widetilde{\sigma}_i}{\|\widetilde{\sigma}'\|^3} \tag{28}$$

and, using (24) and the chain rule,

$$\dot{\lambda} = \frac{1}{\|\widetilde{\sigma}'\|^2}\eta_2. \tag{29}$$

This shows that the tangential state η_3 can be computed effectively using (25), (28), (29), Ω and $\dot{\Omega}$.

Finally, in order to implement the feedback transformation (14), we must find expressions for $L_f^3\pi$ and the first row of the decoupling matrix (12). The decoupling matrix is straightforward but tedious to compute, and calculations give

$$L_f^3\pi = \frac{\eta_2}{\|\widetilde{\sigma}'\|}\left(\Delta'\dot{\Omega} + \Delta'\Omega + \frac{d\Delta'}{dt}\Omega + \frac{\eta_2 \sum_{i=1}^2 \widetilde{\sigma}_i'\widetilde{\sigma}_i}{\|\widetilde{\sigma}'\|^3}\Delta'\Omega\right) q + \dot{\Delta}\dot{\Omega} + \Delta\Omega_2 \tag{30}$$

where

$$\Omega_2 := \frac{(\mathrm{v}+x_5)}{\ell}\tan(x_4)\begin{bmatrix} (1-2x_6)\sin(x_3) \\ (1+2x_6)\cos(x_3) \end{bmatrix} - \frac{(\mathrm{v}+x_5)^3}{\ell^2}\tan^2(x_4)\begin{bmatrix} \cos(x_3) \\ \sin(x_3) \end{bmatrix}.$$

Implementation of controller and the regular feedback (14) is summarized by Algorithm 2.

Algorithm 2: Control algorithm

input : $\widetilde{\sigma}(\lambda) : \mathbb{R} \to \mathbb{R}^2$ (nonunit speed)
$s : W \subseteq \mathbb{R}^2 \to \mathbb{R}$
System model (9)
Current state $x \in \mathbb{R}^6$
output : (u_1, u_2)
for each **do**
Using (23) numerically compute $\lambda^\star$.
Compute $\widetilde{\sigma}'(\lambda^\star)$, $\widetilde{\sigma}''(\lambda^\star)$, $\widetilde{\sigma}'''(\lambda^\star)$, $\|\widetilde{\sigma}'(\lambda^\star)\|$.
Numerically compute η_1 using (24).
Compute η_2 using (27).
Compute η_3 using (25), (28), (29), Ω and $\dot{\Omega}$.
Compute $L_f^3\pi$ using expression (30).
Compute ξ_1, ξ_2, ξ_3, $L_f^3\alpha$.
Compute (u_1, u_2) using (14), (16), (17).
end

TABLE I
CONTROLLER GAINS USED IN SECTION VI

Description	Symbols	Values
Transversal gains (16)	$\{k_1, k_2, k_3\}$	$\{-46.3, -38.7, -10.8\}$
Tangential gains (17)	$\{k_4, k_5, k_6\}$	$\{0, -1.3, -2.3\}$

VI. EXPERIMENTAL VERIFICATION

A. Experimental Platform and Setup

The Chameleon R100 built by Clearpath Robotics Inc. [see Fig. 1(a)] is a low-cost car-like robot for testing control and estimation algorithms.

A DC motor is attached to the rear axle of the robot. A servo motor is used to control the steering angle of the robot. The maximum steering angle is approximately $\overline{x}_4 = 0.4712$ rad (27°). This means that Assumption 3 is satisfied, given that $\ell = 22.9$ cm in this case, if the maximum curvature of the path is 2.22 m^{-1}. The wheels of the robot provide sufficient friction with the ground to make the rolling without slipping assumption implicitly made in (1) hold. However, the steering linkage to front wheels permits up to ± 7 degrees of error. This error source is not captured by the mathematical model (1) used for control design. The chassis of the robot measures $30 \times 22 \times 20$ cm (l/w/h) and is controlled from an Intel Atom Notebook. Onboard electronics provide low-level commands to the motors, while the proposed control algorithm is implemented on the notebook, hereafter called the control computer, running the Robot Operating System in Linux.

To implement Algorithm 2, all of the robot's states are needed. To this end, an indoor positioning system (IPS) is employed using the NaturalPoint OptiTrack local positioning system. The IPS uses 16 near-infrared (IR) cameras. IR reflectors are attached to the robot's chassis to make the position (x_1, x_2) and orientation x_3 available for feedback, via the IPS, over WiFi. The control computer uses multithreaded publish/subscribe model to read the position and orientation of the robot at 100 Hz from the IPS.

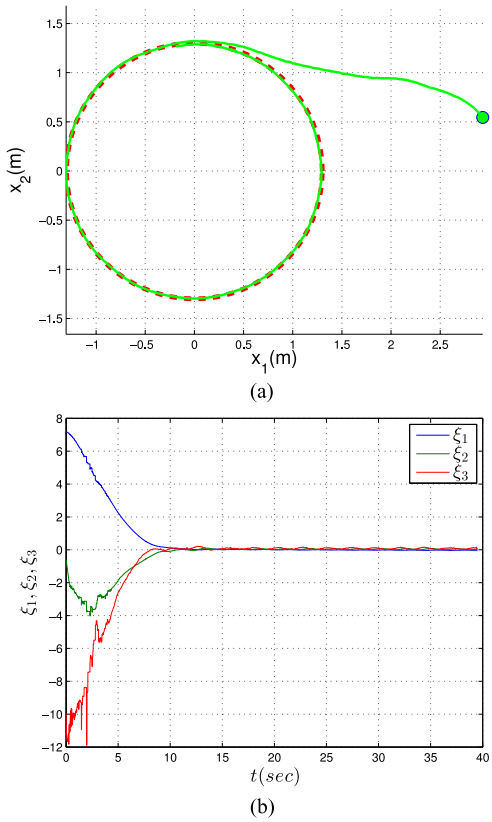

Fig. 2. Chameleon R100 robot following the circular curve $\sigma : [0, 2.6\pi) \to \mathbb{R}^2, \lambda \mapsto \mathrm{col}(1.3\sin(\lambda/1.3), 1.3\cos(\lambda/1.3))$. (a) Chameleon R100 following the circular path. (b) Convergence of ξ_1, ξ_2, ξ_3 states.

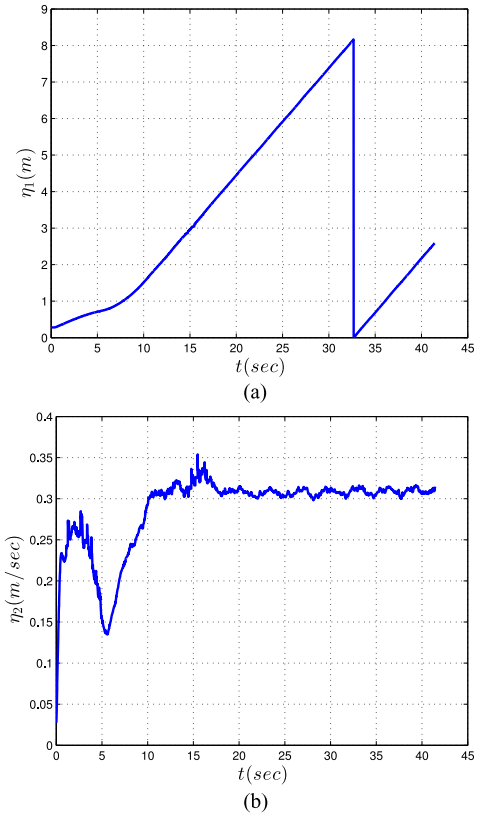

Fig. 3. Velocity and position of the Chameleon R100 robot, while following the circular curve $\sigma : [0, 2.6\pi) \to \mathbb{R}^2, \lambda \mapsto \mathrm{col}(1.3\sin(\lambda/1.3), 1.3\cos(\lambda/1.3))$. (a) Position η_1 of the robot along the path. (b) Chameleon R100 maintaining a desired speed of 0.3 m/s along the path.

In many car-like robot platforms, the steering angle can be directly measured using a potentiometer or an absolute optical encoder; however, the Chameleon R100 lacks this feature. Since the steering angle cannot be measured by the IPS, a standard extended Kalman filter (EKF) is used to obtain estimate $(x_1, x_2, x_3, x_4, x_5, x_6)$ from the measurements (x_1, x_2, x_3) and the control inputs (u_1, u_2). The control inputs of the Chameleon are its steering angle and translational speed. However, the control inputs of (1) are the rate of change of the steering angle and translational speed. The steering control input can be computed from the rate of change of steering angle by integration.

B. Experimental Results

In the first experiment, the Chameleon R100 robot is asked to follow a circular path of radius $r = 1.3$ meters, while maintaining a constant speed of $\eta_2^{\mathrm{ref}}(t) = 0.3$ m/s along the path, i.e., $\eta^{\mathrm{ref}}(t) = (0, 0.3, 0)$ which clearly satisfies Assumption 4.

In the experiment, pole placement was used to select the gains so that the control signals did not saturate. The closed-loop transversal dynamics were designed to converge to zero faster than the closed-loop tangential error dynamics to promote convergence to the path over progress along the path. In the following experiments, the desired pole locations for the transversal states ξ were chosen as $-3.9, -3.6, -3.3$. The desired pole location for the tangential error states $e_\eta = (\eta_2 - \eta_2^{\mathrm{ref}}, \eta_3 - \dot{\eta}_2^{\mathrm{ref}})$ are chosen to be $-1.2, -1.1$. The controller gains computed for the desired pole locations are shown in Table I.

The robot's initial position is indicated by a solid green dot in Fig. 2(a). The desired circle is represented by a dotted line in the figure.

The position of the robot along the path is given by the transformed state $\eta_1 \in [0, 2.6\pi)$. In this example, the path is closed and has arc length 2.6π; therefore, $\mathbb{D} = [0, 2\pi r) = [0, 2.6\pi)$ and η_1 remains bounded between 0 to $2\pi r$, as shown in Fig. 3(a). The tangential state η_2 is shown in Fig. 3(b). A fixed tangential speed of 0.3 m/s was chosen due to the limited capabilities of the vehicle and the limited test area available inside the IPS capture region. Simulation examples with variable speed profiles can be found in [12].

In the second experiment, the Chameleon R100 robot is made to follow a nonclosed sinusoidal path. Fig. 4(a) shows that the robot first converges to the desired path and follows it. Due to limited lab space, the robot is asked to follow only a small portion of the sinusoidal path. All the transversal states (the ξ states) converge to zero [see Fig. 4(b)]. As the robot follows the sinusoid path, a desired speed of 0.3 m/s is achieved as shown in Fig. 5.

In the third experiment, the repeatability of the proposed controller is tested on a circular path of radius 1.3 m. The experiment is repeated six times, and the convergence of the path-following error is analyzed. In each test, the robot converges to the desired

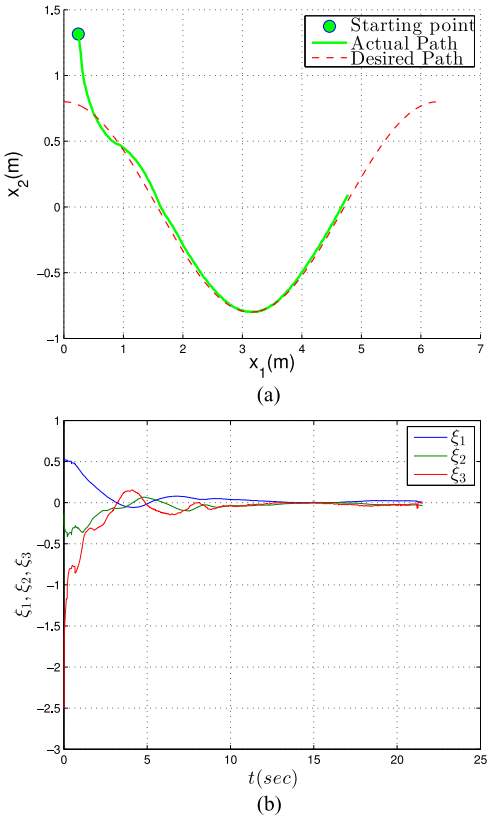

Fig. 4. Chameleon R100 robot following the nonclosed, nonunit speed, sinusoidal path $\tilde{\sigma}(\lambda) = \mathrm{col}(\lambda, 0.8\cos(\lambda))$. (a) Chameleon R100 following the sinusoidal path. (b) Convergence of ξ_1, ξ_2, ξ_3 states.

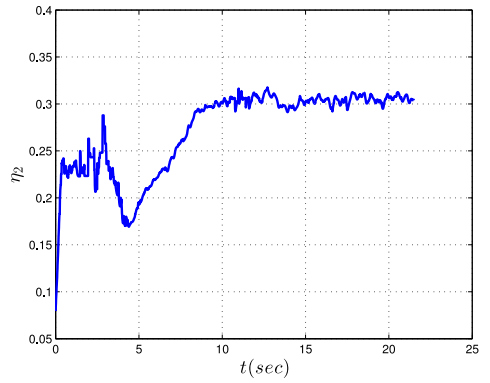

Fig. 5. Chameleon R100 maintaining a desired speed of 0.3 m/s along the sinusoidal path.

path starting from an initial point away from the path as shown in Fig. 6(a). The path-following error $e_{\mathrm{PF}} := \sqrt{x_1^2 + x_2^2} - 1.3$ is shown in Fig. 6(b). The initial pose (position and orientation) and steady-state path-following error $|e_{\mathrm{PF}}^{\mathrm{ss}}| := \lim_{t\to\infty} \sup |e_{\mathrm{PF}}|$ of the robot in each run is presented in Table II. Fig. 7 gives a zoomed-in view of the path-following error. We see that the path-following error in each run remains within ± 0.015 m. We conclude that path-following controller gives fairly accurate and reliable results as the mean path-following error of the six runs is 1.0689 cm with a standard deviation of 0.154 cm.

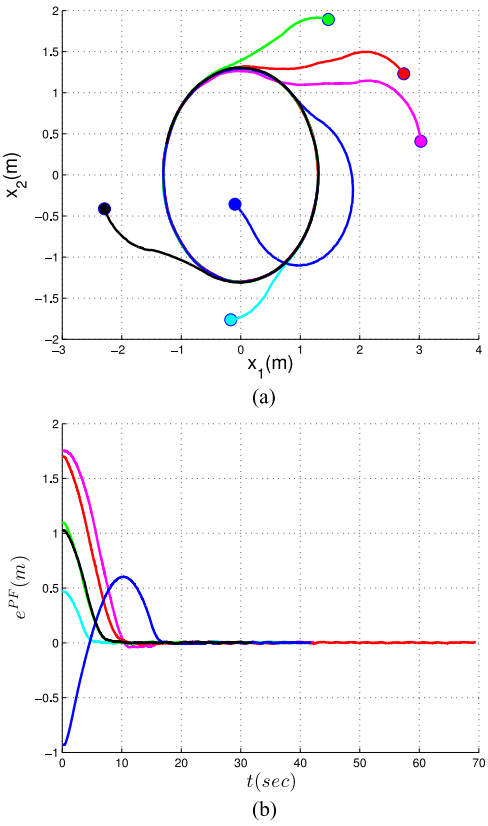

Fig. 6. Multiple experiments following circular path $\sigma : [0, 2.6\pi) \to \mathbb{R}^2$, $\lambda \mapsto \mathrm{col}(1.3\sin(\lambda/1.3), 1.3\cos(\lambda/1.3))$. (a) Convergence of the robot's position to the circle. (b) Path-following error.

TABLE II
STEADY-STATE PATH-FOLLOWING ERROR

Test	$(x_1(0), x_2(0))$	$x_3(0)$	$\lvert e_{\mathrm{PF}}^{\mathrm{ss}} \rvert$
1	$(3.0267, 0.4083)$	1.8153	1.0580
2	$(-0.1675, -1.7628)$	0.1440	1.3766
3	$(2.7383, 1.2309)$	2.3205	0.9556
4	$(1.4719, 1.8907)$	2.9793	1.0089
5	$(-0.0971, -0.3565)$	-0.6987	1.0148
6	$(-2.2894, -0.4131)$	-1.0454	0.9992

The initial conditions $(x_1(0), x_2(0))$ and $x_3(0)$ are given in meters and radians, respectively. The path-following error is given in centimeters.

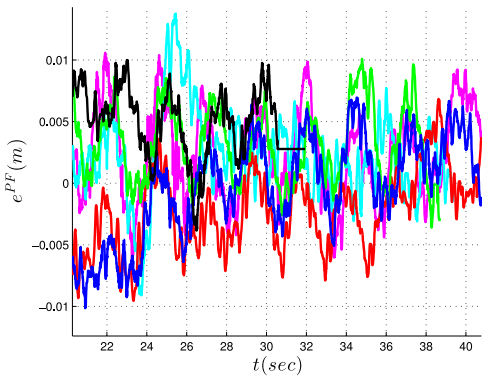

Fig. 7. Magnified view of the path-following error after the convergence of the robot to the desired path.

We observed that the closed-loop performance is very sensitive to IPS calibration errors. A small misalignment between the center of IR reflectors and center of the rear axle, i.e., (x_1, x_2), is reflected in the path-following error. Moreover, we observed that the error is reduced by a few centimeters if an EKF is used, as described above, on all six states of the system. An adaptive path-following controller may perform better in the face of calibration errors.

VII. Conclusion and Future research

In this paper, a path-following controller has been designed for the kinematic model of a car-like mobile robot using transverse feedback linearization with dynamic extension for a large class of paths. The control method is experimentally demonstrated on a Chameleon R100 Ackermann steering robot. It has been shown that the path-following controller forces the robot to converge and then follow the desired path with very small error. Future research includes precise characterizations of the region of attraction of the proposed controllers and the use of adaptive nonlinear PI path-following controllers based on the notion of immersion and invariance [34] to reduce sensitivity to sensor calibration errors.

References

[1] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotović, "Performance limitations in reference tracking and path following for nonlinear systems," *Automatica*, vol. 44, no. 3, pp. 598–610, 2008.

[2] D. E. Miller and R. H. Middleton, "On limitations to the achievable path tracking performance for linear multivariable plants," *IEEE Trans. Autom. Control*, vol. 53, no. 11, pp. 2586–2601, Dec. 2008.

[3] C. Nielsen, C. Fulford, and M. Maggiore, "Path following using transverse feedback linearization: Application to a Maglev positioning system," *Automatica*, vol. 46, pp. 585–590, Mar. 2010.

[4] A. D. Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot, " in, *Robot Motion Planning and Control*. New York, NY, USA: Springer, 1998.

[5] D. Wang and G. Xu, "Full-state tracking and internal dynamics of nonholonomic wheeled mobile robots," *IEEE/ASME Trans. Mechatronics*, vol. 8, no. 2, pp. 203–214, Jun. 2003.

[6] L. Consolini, A. Piazzi, and M. Tosques, "Path following of car-like vehicles using dynamic inversion," *Int. J. Control*, pp. 1724–1738, 2003.

[7] G. Indiveri and M. L. Corradini, "Switching linear path following for bounded curvature car-like vehicles," in *Proc. 5th IFAC/EURON Symp. Intell. Auton. Vehicles*, Jul. 2004.

[8] O. H. Dağci, U. Y. Ogras, and U. Özguner, "Path following controller design using sliding mode control theory," in *Proc. 2003 Am. Control Conf.*, Jun. 2003, vol. 1, pp. 903–908.

[9] A. Pesterev and L. Rapoport, "Construction of invariant ellipsoids in the stabilization problem for a wheeled robot following a curvilinear path," *Autom. Remote Control*, vol. 70, no. 2, pp. 219–232, 2009.

[10] R. Gilimyanov, A. Pesterev, and L. Rapoport, "Motion control for a wheeled robot following a curvilinear path," *J. Comput Syst. Sci. Int.*, vol. 47, no. 6, pp. 987–994, 2008.

[11] C. Nielsen and M. Maggiore, "Maneuver regulation via transverse feedback linearization: Theory and examples," in *Proc. Symp. Nonlinear Control Syst.*, Sep. 2004.

[12] A. Akhtar and C. Nielsen, "Path following for a car-like robot using transverse feedback linearization and tangential dynamic extension," in *Proc. 50th IEEE Conf. Decision Control Eur. Control Conf.*, Dec. 2011, pp. 7949–7979.

[13] D. Lee, H. Jin Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *Int. J. Control, Autom. Syst.*, vol. 7, no. 3, pp. 419–428, 2009.

[14] G. Oriolo, A. De Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation, and experimental validation," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 6, pp. 835–852, Nov. 2002.

[15] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "On differentially flat nonlinear systems," in *Proc. IFAC-Symp. NOLCOS'92 Bordeaux*, 1992, pp. 408–412.

[16] P. Rouchon, M. Fliess, J. Lévine, and P. Martin,"Flatness, motion planning and trailer systems," in *Proc. Conf. Decision Control*, 1993, pp. 2700–2705.

[17] O. J. Sørdalen, "Conversion of the kinematics of a car with n trailers into a chained form," in *Proc. Int. Conf. Robot. Autom.*, 1993, pp. 382–387.

[18] A. Broggi, M. Bertozzi, A. Fascioli, C. G. L. Bianco, and A. Piazzi, "The ARGO autonomous vehicles vision and control systems," *Int. J. Intell. Control Syst.*, vol. 3, no. 4, pp. 409–441, 1999.

[19] A. Isidori, *Nonlinear Control Systems*. Secaucus, NJ, USA: Springer-Verlag, 1995.

[20] R. M. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," in *Proc. 1995 ASME Int. Cong. Exp.*, 1995.

[21] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos, *Nonlinear and Adaptive Control Design*, 1st ed. New York, NY, USA: Wiley, 1995.

[22] H. K. Khalil, *Nonlinear systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

[23] A. Hladio, C. Nielsen, and D.Wang, "Path following for a class of mechanical systems," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 6, pp. 2380–2390, Nov. 2013.

[24] V. Guillemin and A. Pollack, *Differential Topology*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1974.

[25] A. Piazzi, C. Guarino Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic G^2-splines for the iterative steering of vision-based autonomous vehicles," *IEEE Trans. Intell. Trans. Syst.*, vol. 3, no. 1, pp. 27–36, Mar. 2002.

[26] M. Böck and A. Kugi, "Real-time nonlinear model predictive path-following control of a laboratory tower crane," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 4, pp. 1461–1473, Jul. 2014.

[27] M. van Nieuwstadt, M. Rathinam, and R. M. Murray, "Differential flatness and absolute equivalence of nonlinear control systems," *SIAM J. Control Optim.*, vol. 36, pp. 1225–1239, Jul. 1998.

[28] P. Martin and P. Rouchon, "Feedback linearization and driftless systems," *Math. Control, Signals, Syst.*, vol. 7, pp. 235–254, 1994.

[29] A. Pesterev and L. Rapoport, "Canonical representation of the path following problem for wheeled robots," *Autom. Remote Control*, vol. 74, no. 5, pp. 785–801, 2013.

[30] J. Rowe, "A new method of finding the equation of a rational planer curve from its parametric equaions," *Bull. Amer. Math Soc.*, vol. 22, pp. 338–340, 1916.

[31] K. R. Davidson and A. P. Donsig, *Real Analysis with Real Applications*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2001.

[32] J. J. Sylvester, "On a theory of the syzygetic relations of two rational integral functions, comprising an application to the theory of sturm's functions, and that of the greatest algebraical common measure," *Philosoph. Trans. Royal Soc. London*, vol. 143, pp. 407–548, 1853.

[33] T. W. Sederberg, D. C. Anderson, and R. N. Goldman, "Implicit representation of parametric curves and surfaces," *Comput. Vision, Graphics, Image Process.*, vol. 28, no. 1, pp. 72–84, 1984.

[34] A. Astolfi, D. Karagiannis, and R. Ortega, *Nonlinear and Adaptive Control with Applications*. New York, NY, USA: Springer, 2008.

Adeel Akhtar received the B.E. degree from National University of Science and Technology, Islamabad, Pakistan, and the M.A.Sc. degree from University of Waterloo, Waterloo, ON, Canada, in 2011, where he is currently working toward the Ph.D. degree with the Department of Mechanical and Mechatronics Engineering.

In 2011 he joined Waterloo Autonomous Vehicles Laboratory (WAVELab). His research interests include nonlinear control and estimation of autonomous vehicles.

Christopher Nielsen received the B.A.Sc. degree in electrical engineering from University of Waterloo, Waterloo, ON, Canada, in 2002, and the M.A.Sc. and Ph.D. degrees from University of Toronto, Toronto, ON, Canada, in 2004 and 2008, respectively.

In 2008 he joined the Department of Electrical and Computer Engineering, University of Waterloo, where he is currently an Associate Professor. His research interests include geometric control theory, motion control, and output feedback.

Steven L. Waslander (M'05) received the B.Sc.E. degree from Queen's University, Kingston, ON, Canada, in 1998, and the M.S. and Ph.D. degrees in aeronautics and astronautics degree from Stanford University, Stanford, CA, USA, in 2002 and 2007, respectively.

He was a Control Systems Analyst with Pratt & Whitney Canada from 1998 to 2001. In 2008 he joined the Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON, Canada, as an Assistant Professor. He is the Director of the Waterloo Autonomous Vehicles Laboratory (WAVELab). His research interests are in the areas of multiagent control and coordination, air traffic management, and autonomous mobile robotics, specifically quadrotor helicopters.