[15] J. Angeles, G. Yang, and I. M. Chen, "Singularity analysis of three-legged, six-DOF platform manipulators with RRRS legs," in *Proc. 2001 IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*, July 2001, pp. 32–36.
[16] F. C. Park and J. W. Kim, "Singularity analysis of closed kinematic chains," *ASME J. Mechan. Des.*, vol. 121, no. 2, pp. 32–38, Mar. 1999.
[17] G. Yang, I. M. Chen, W. Lin, and J. Angeles, "Singularity analysis of three-legged parallel robots based on passive joint velocities," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 413–422, Aug. 2001.
[18] C. M. Gosselin, "Determination of the workspace of 6-DOF parallel manipulators," *ASME J. Mechan. Des.*, vol. 112, pp. 331–336, Sept. 1990.
[19] I. A. Bonev and J. Ryu, "A new approach to orientation workspace analysis of 6-DOF parallel manipulators," *Mechanism and Machine Theory*, vol. 36, no. 1, pp. 15–28, 2001.
[20] E. F. Fichter, "A Stewart platform-based manipulator: General theory and practical construction," *Int. J. Robot. Res.*, vol. 5, no. 2, pp. 157–182, 1986.
[21] O. Masory and J. Wang, "Workspace evaluation of Stewart platforms," in *Proc. ASME 22nd Biennial Mechanisms Conf.*, vol. 45, 1992, pp. 337–346.
[22] T. Arai, T. Tanikawa, J.-P. Merlet, and T. Sendai, "Development of a new parallel manipulator with fixed linear actuator," in *ASME Japan/USA Symp. Flexible Automation*, 1996, pp. 145–149.
[23] E. Ottaviano and M. Ceccarelli, "Optimal design of CaPaMan (Cassino Parallel Manipulator) with a specified orientation workspace," *Robotica*, vol. 20, pp. 159–166, 2002.
[24] J.-P. Merlet, "Détermination de l'espace de travail d'un robot parallèle pour une orientation constante," *Mechanism and Machine Theory*, vol. 29, no. 8, pp. 1099–1113, 1994.
[25] R. P. Podhorodeski and K. H. Pittens, "A class of parallel manipulators based on kinematically simple branches," *ASME J. Mechan. Des.*, vol. 116, pp. 908–914, 1994.
[26] R. Brockett, "Robotic manipulators and the product of exponential formula," in *Int. Symp. Math. Theory of Network and Systems*, Israel, 1983, pp. 120–129.
[27] F. C. Park, "Computational aspect of manipulators via product of exponential formula for robot kinematics," *IEEE Trans. Automat. Contr.*, vol. 39, no. 9, pp. 643–647, 1994.
[28] R. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.
[29] C. M. Gosselin and J. Sefrioui, "Polynomial solutions for the direct kinematic problem of planar three-degree-of-freedom parallel manipulators," in *Proc. 5th Int. Conf. Advanced Robotics*, Pisa, Italy, June 19–22, 1991, pp. 1124–1129.
[30] L.-W. Tsai, *Robot Analysis*. New York: Wiley, 1999.
[31] G. Yang, W. H. Chen, and I. M. Chen, "A geometrical method for the singularity analysis of 3-RRR planar parallel robots with different actuation schemes," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2002, pp. 2055–2060.
[32] J. S. Rao and R. V. Dukkipati, *Mechanism and Machine Theory*. New Delhi, India: Wiley Eastern Ltd., 1989.

Smooth Motion Generation for Unicycle Mobile Robots Via Dynamic Path Inversion

Corrado Guarino Lo Bianco, Aurelio Piazzi, and Massimo Romano

***Abstract*—A new motion-generation approach is proposed for wheeled mobile robots described by the unicycle kinematic model. This approach permits the generation of smooth continuous-acceleration controls using a dynamic path-inversion procedure that exploits the concept of G^3-paths, i.e., Cartesian paths with third-order geometric continuity (both the curvature function and its derivative, with respect to the arc length, are continuous). The exposed steering method is well suited to be adopted for the robot's iterative steering within a supervisory control architecture for sensor-based autonomous navigation. A worked example illustrates the approach.**

***Index Terms*—Dynamic path inversion, geometric continuity, smooth motion generation, unicycle mobile robots.**

I. Introduction

The work of Dubins [1] and Reeds and Shepp [2] devoted to the planning with minimal length paths are well-known fundamental results in the field of nonholonomic motion generation of wheeled mobile robots and car-like vehicles (see the survey [3] by Laumond *et al.*). A known limit of the Dubins–Reeds–Shepp approach is that the resulting paths are composed by line segments and circular arcs and do not have overall continuous curvature. Therefore, swift high-performance maneuvers cannot be achieved. At the end of the 1980s, many authors [4]–[6] proposed several curve primitives to overtake this limit: in all of the cases, continuous-curvature paths, usually denoted as G^2-paths, were generated. Subsequently, many authors have worked on planning schemes ensuring continuous-curvature paths [7], [8]. A recent approach to continuous-curvature motion generation was proposed in [9] where continuous-curvature paths were obtained by using a new primitive, the η-spline or quintic G^2-spline. This primitive was adopted to achieve a straightforward approach for the iterative steering of vision-based autonomous vehicles.

In this paper, the discussion is focused on the generation of a smooth motion control for unicycle mobile robots (UMRs), i.e., wheeled mobile robots whose kinematics is described by the unicycle model. In more detail, it will be demonstrated that, if such robots are driven with smooth linear and angular velocity command signals, i.e., signals which are continuous with their derivatives, they generate G^3-paths, i.e., paths for which both the curvature function and its derivative with respect to the arc length are continuous. It will be further demonstrated that, conversely, the UMRs can be driven along any G^3-path by means of appropriately devised C^1 command signals. A possible control strategy is then proposed. Its main features are the following.

1) A dynamic path inversion algorithm is designed to synthesize the UMR inputs. This algorithm requires planning with G^3-paths [10] using interpolating conditions at the path end-points.

Manuscript received July 9, 2003; revised December 19, 2003. This paper was recommended for publication by Associate Editor K. Lynch and Editor H. Arai upon evaluation of the reviewers' comments. This work was supported in part by MIUR Scientific Research Funds under the framework of a COFIN 2002 project. This paper was presented in part at the 2002 IEEE Intelligent Vehicles Symposium, Versailles, France, June 18–20, 2002.

The authors are with the Dipartimento di Ingegneria dell'Informazione, Università di Parma, I-43100 Parma, Italy (e-mail: guarino@ce.unipr.it; piazzi@ce.unipr.it; romano@ce.unipr.it).

Digital Object Identifier 10.1109/TRO.2004.832827

2) The UMR inputs, i.e., the linear and angular robot velocities, are globally generated as C^1 functions (velocities and accelerations are continuous time functions).

The proposed inversion-based procedure, which does not require any integration because it relies on the differential flatness of the unicycle model [11], will permit the real-time generation of continuous-acceleration feedforward inputs.

Due to the inevitable mismatch between the ideal unicycle model and the real one, the UMR tends to drift from the planned paths. These path errors can be reduced with a timely updating of the path replanning in accordance with an iterative steering technique [12]. As is known, this technique can be interpreted as a discrete-time feedback which is very useful in a real robot navigation scenario [13].

This paper is organized as follows. Section II introduces notation and preliminaries on first-, second-, and third-order geometric continuity of curves and paths. Section III poses a reachability problem in an extended state space of the unicycle model and derives a first result (*Proposition 1*). The dynamic path-inversion-based procedure is detailed in Section IV. A worked example illustrates the approach in Section V. The concluding remarks of Section VI end the paper.

II. Notation and Preliminaries on G^3-Paths

The Euclidean norm of a vector $\mathbf{p}$ is denoted by $\|\mathbf{p}\|$. Let C^i indicate the set of functions that are continuous until the ith derivative and let C_p indicate the class of piecewise continuous functions. A curve on the $\{x, y\}$ plane can be described by means of the following parametrization $\mathbf{p}(u)$:

$$\begin{aligned}\mathbf{p} : [u_0, u_1] &\to \mathbb{R}^2 \\ u &\to [\alpha(u)\beta(u)]^T \end{aligned} \tag{1}$$

where $[u_0, u_1]$ is a real closed interval. The "path" associated with the curve $\mathbf{p}(u)$ is $\mathbf{p}([u_0, u_1])$, i.e., the image of $[u_0, u_1]$ under the vectorial function $\mathbf{p}(u)$.

Definition 1: A curve $\mathbf{p}(u)$ is regular if $\dot{\mathbf{p}}(u) \in C_p([u_0, u_1])$ and $\dot{\mathbf{p}}(u) \neq \mathbf{0}\ \forall u \in [u_0, u_1]$.

The curve length measured along $\mathbf{p}(u)$ is denoted by s; it can be expressed as a function f of u as

$$\begin{aligned} f : [u_0, u_1] &\to [0, f(u_1)] \\ u &\to s = \int_{u_0}^{u} \|\dot{\mathbf{p}}(\xi)\|\, d\xi. \end{aligned} \tag{2}$$

Evidently, given a regular curve $\mathbf{p}(u)$, the length function $f(\cdot)$ is continuous over $[u_0, u_1]$ and bijective; hence, its inverse is continuous too and it will be denoted as

$$\begin{aligned} f^{-1} : [0, f(u_1)] &\to [u_0, u_1] \\ s &\to u = f^{-1}(s). \end{aligned} \tag{3}$$

Associated with every point of a regular curve $\mathbf{p}(u)$, there is the orthonormal moving reference frame $\{\boldsymbol{\tau}(u), \boldsymbol{\nu}(u)\}$ which is congruent with the axes of the $\{x, y\}$ plane and where $\boldsymbol{\tau}(u) = \dot{\mathbf{p}}(u)/\|\dot{\mathbf{p}}(u)\|$ denotes the unit tangent vector to the curve $\mathbf{p}(u)$. Define $\arg\{\boldsymbol{\tau}(\cdot)\}$ as the angle θ between $\boldsymbol{\tau}$ and the x axis. The angle θ is counterclockwise positive.

Definition 2 (G^1-curves): A parametric curve $\mathbf{p}(u)$ has first-order geometric continuity, and we say $\mathbf{p}(u)$ is a G^1-curve, if $\mathbf{p}(u)$ is regular and its unit tangent vector is a continuous function along the curve, i.e., $\boldsymbol{\tau}(\cdot) \in C^0([u_0, u_1])$.

For any regular curve such that $\ddot{\mathbf{p}}(u) \in C_p([u_0, u_1])$, the scalar curvature can be defined according to the Frenet formula $\dot{\boldsymbol{\tau}}(u) = \kappa_c(u)\boldsymbol{\nu}(u)$ (see, for example, [14, p. 109]). This defines the curvature function with respect to the parameter u as follows:

$$\begin{aligned} \kappa_c : [u_0, u_1] &\to \mathbb{R} \\ u &\to \kappa_c(u). \end{aligned} \tag{4}$$

According to the theory of planar curves, an explicit expression of $\kappa_c(u)$ is $\kappa_c(u) = [\dot{\alpha}(u)\ddot{\beta}(u) - \ddot{\alpha}(u)\dot{\beta}(u)]/(\dot{\alpha}^2(u) + \dot{\beta}^2(u))^{3/2}$. The scalar curvature can also be expressed as a function of the curve length s. In the following, such a function will be indicated as

$$\begin{aligned} \kappa : [0, f(u_1)] &\to \mathbb{R} \\ s &\to \kappa(s) \end{aligned} \tag{5}$$

and the bijectivity of function $f(u)$ makes it possible to write

$$\kappa(s) = \kappa_c\left(f^{-1}(s)\right). \tag{6}$$

Evidently, by virtue of relation (6), the curvature function $\kappa_c(u)$ is continuous if and only if function $\kappa(s)$ is continuous, i.e., $\kappa_c(\cdot) \in C^0([u_0, u_1]) \Leftrightarrow \kappa(\cdot) \in C^0([0, f(u_1)])$.

Definition 3 (G^2-curves): A parametric curve $\mathbf{p}(u)$ has second-order geometric continuity, and we say $\mathbf{p}(u)$ is a G^2-curve, if $\mathbf{p}(u)$ is a G^1-curve, $\ddot{\mathbf{p}}(\cdot) \in C_p([u_0, u_1])$ and its scalar curvature is continuous along the curve, i.e., $\kappa_c(\cdot) \in C^0([u_0, u_1])$ or $\kappa(\cdot) \in C^0([0, f(u_1)])$.

G^1- and G^2-curves were originally introduced by Barsky and Beatty [15] in a computer graphics context. The main results of this paper require the introduction of curves with third-order geometric continuity. This can be done according to the following definition.

Definition 4 (G^3-curves): A parametric curve $\mathbf{p}(u)$ has third-order geometric continuity, and we say that $\mathbf{p}(u)$ is a G^3-curve, if $\mathbf{p}(u)$ is a G^2-curve, $\dddot{\mathbf{p}}(\cdot) \in C_p([u_0, u_1])$, and the derivative of the scalar curvature, with respect to the arc length s, is continuous along the curve, i.e., $\dot{\kappa}(\cdot) \in C^0([0, f(u_1)])$.

Definition 5 (G^1-, G^2-, and G^3-paths): A path of a Cartesian space, i.e., a set of points of this space, is a G^i-path ($i = 1, 2, 3$) if there exists a parametric G^i-curve whose image is the given path.

Remark 1: The definitions provided herein for the geometric continuity are not the most general ones and are restricted to planar curves. A recent survey devoted to the kth-order geometric continuity of curves and surfaces in a general context can be found in [16].

III. The Problem and a First Result

Consider a wheeled mobile robot governed by the nonholonomic unicycle model

$$\dot{x}(t) = v(t)\cos\theta(t) \tag{7}$$

$$\dot{y}(t) = v(t)\sin\theta(t) \tag{8}$$

$$\dot{\theta}(t) = \omega(t) \tag{9}$$

where x and y indicate the robot position with respect to a stationary frame, θ is its heading angle, and v and ω are its linear and angular velocities to be considered as the control inputs of the robot. In order to achieve high-motion performances, these inputs $v(t)$ and $\omega(t)$ will be synthesized as C^1-functions, i.e., linear and angular accelerations will be continuous signals.

From a mathematical standpoint, the state of model (7)–(9), at time t, is given by $\{x(t), y(t), \theta(t)\}$. In the following, it is convenient to use an *extended state*, comprising the inputs and their derivatives, defined as

$$\{x(t), y(t), \theta(t), v(t), \dot{v}(t), \omega(t), \dot{\omega}(t)\}.$$

Then, the considered motion generation problem can be stated as a reachability problem in the extended state space.

The Problem: Given any assigned traveling time $t_f > 0$, find control inputs $v(\cdot), \omega(\cdot) \in C^1([0, t_f])$ such that the mobile robot starting from an arbitrary initial extended state

$$\begin{aligned}\mathbf{p}_A &= [x_A \ y_A]^T = [x(0) \ y(0)]^T \\ \theta_A &= \theta(0) \\ v_A &= v(0) \\ \dot{v}_A &= \dot{v}(0) \\ \omega_A &= \omega(0) \\ \dot{\omega}_A &= \dot{\omega}(0)\end{aligned}$$

reaches the arbitrary final extended state

$$\begin{aligned}\mathbf{p}_B &= [x_B \ y_B]^T = [x(t_f) \ y(t_f)]^T \\ \theta_B &= \theta(t_f) \\ v_B &= v(t_f) \\ \dot{v}_B &= \dot{v}(t_f) \\ \omega_B &= \omega(t_f) \\ \dot{\omega}_B &= \dot{\omega}(t_f).\end{aligned}$$

A solution to the introduced problem will be provided by the path dynamic inversion procedure described in Section IV and the overall motion strategy can be then based on an iterative steering [12] issued by a supervisory control system [17]. The real-time knowledge of the robot position is used by the supervisory system to steer the UMR from the current extended state to a future extended state in an iterative fashion. In such a way, for the UMR, swift high-performance motion is possible while intelligent or elaborate behaviors are performed.

The following proposition that is the first contribution of the paper is essential to understand how to plan a desired path connecting $\mathbf{p}_A$ with $\mathbf{p}_B$.

Proposition 1: Assign any $t_f > 0$. If a Cartesian path is generated by model (7)–(9) with inputs $v(t)$, $\omega(t) \in C^1([0, t_f])$, and $v(t) \neq 0 \ \forall t \in [0, t_f]$, then it is a G^3-path. Conversely, given any G^3-path, there exist inputs $v(t), \omega(t) \in C^1([0, t_f])$ with $v(t) \neq 0 \ \forall t \in [0, t_f]$ and initial conditions such that the path generated by model (7)–(9) coincides with the given G^3-path.

Proof: Let us demonstrate the first part of the proposition. Consider a Cartesian planar $\{x, y\}$-path generated by the model (7)–(9) by means of two command signals $v(t)$ and $\omega(t)$ continuous with their derivatives. The generated path can be found by explicitly solving (7)–(9) for any t belonging to a given time interval $[0, t_f]$. Let us indicate by $[x(t), y(t)]^T$ the generated trajectory. A known result of the planar curve theory makes it possible to express the curve unit tangent vector as

$$\boldsymbol{\tau}(t) = \frac{[\dot{x}(t) \ \dot{y}(t)]^T}{\sqrt{\dot{x}^2(t) + \dot{y}^2(t)}} = [\cos\theta(t) \ \sin\theta(t)]^T, \quad \text{if } v(t) > 0$$

$$\boldsymbol{\tau}(t) = \frac{[\dot{x}(t) \ \dot{y}(t)]^T}{\sqrt{\dot{x}^2(t) + \dot{y}^2(t)}} = -[\cos\theta(t) \ \sin\theta(t)]^T, \quad \text{if } v(t) < 0.$$

Taking into account model (7)–(9) and the continuity of ω, it is possible to deduce that θ and, consequently, the unit tangent vector $\boldsymbol{\tau}$, are continuous.

The scalar curvature for a planar path is defined as

$$\kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{(\dot{x}^2(t) + \dot{y}^2(t))^{\frac{3}{2}}}. \tag{10}$$

Explicit expressions for $\ddot{x}$ and $\ddot{y}$ can be derived from (7)–(9) as

$$\ddot{x}(t) = \dot{v}(t)\cos\theta(t) - v(t)\omega(t)\sin\theta(t) \tag{11}$$

$$\ddot{y}(t) = \dot{v}(t)\sin\theta(t) + v(t)\omega(t)\cos\theta(t). \tag{12}$$

By using (11)–(12) and the model (7)–(9), (10) becomes

$$\kappa(t) = \frac{\omega(t)}{v(t)}, \quad \text{if } v(t) > 0 \tag{13}$$

$$\kappa(t) = -\frac{\omega(t)}{v(t)}, \quad \text{if } v(t) < 0. \tag{14}$$

Hence, the continuity on v and ω infers the continuity of the curvature κ.

Finally, the curvature derivative with respect to curve length s is expressed as

$$\frac{d\kappa(t)}{ds} = \frac{\frac{d\kappa(t)}{dt}}{\frac{ds(t)}{dt}} = \frac{\dot{\omega}(t)v(t) - \omega(t)\dot{v}(t)}{v^2(t)}\frac{1}{v(t)}.$$

As a consequence, it is possible to assert that, according to the continuity of v, $\dot{v}$, ω, and $\dot{\omega}$, the curvature derivative $d\kappa/ds$ is continuous also. This demonstrates that the path generated by model (7)–(9) is a G^3-path.

The second part of the proposition will be demonstrated constructively. Suppose that we want to assign a path in the $\{x, y\}$ plane by means of a G^3-curve parametrized by u, i.e., $\mathbf{p}(u) := [\alpha(u) \ \beta(u)]^T$ with $u \in [u_0, u_1]$.

Let us assume as initial conditions $x(0) = \alpha(u_0)$, $y(0) = \beta(u_0)$, and $\theta(0) = \arg\{\boldsymbol{\tau}(u_0)\}$. Freely assign $v(t) \in C^1([0, t_f])$ with $t \in [0, t_f]$ such that $v(t) > 0$, and the equation

$$\int_0^{t_f} v(\xi)d\xi = f(u_1) \tag{15}$$

is satisfied.

Finally, assign $\omega(t)$ according to the following equation:

$$\omega(t) := v(t)\kappa(s)\big|_{s=\int_0^t v(\xi)d\xi}. \tag{16}$$

Owing to the G^3 continuity of the planned curve, we can claim the continuity of $\kappa(s)$ so that, since $v(t) \in C^1([0, t_f])$, it is possible to conclude that $\omega(t) \in C^0([0, t_f])$. Moreover, its first derivative can be expressed as

$$\dot{\omega}(t) = v^2(t)\left.\frac{d\kappa}{ds}\right|_{s=\int_0^t v(\xi)d\xi} + \dot{v}(t)\kappa(s)\big|_{s=\int_0^t v(\xi)d\xi}. \tag{17}$$

All of the functions appearing in (17) are continuous. This proves that, as required, $\omega(t) \in C^1([0, t_f])$.

To conclude the demonstration of the second part of the proposition, it is necessary to prove that the following time functions:

$$\alpha(u)\big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)} \tag{18}$$

$$\beta(u)\big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)} \tag{19}$$

$$\arg\{\tau(u)\}\big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)} \tag{20}$$

are a solution of model (7)–(9). First of all, it is immediately possible to verify that (18)–(20) satisfy the given initial conditions. By differentiating with respect to the time both (18) and (19), we can write

$$\begin{aligned}&\frac{d}{dt}\alpha(u)\big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)} \\ &= \left.\frac{\dot{\alpha}(u)}{\|\dot{\mathbf{p}}(u)\|}\right|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)} v(t) \\ &= v(t)\cos\left[\arg\left(\frac{\dot{\mathbf{p}}(u)}{\|\dot{\mathbf{p}}(u)\|}\right)\right]\Bigg|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)} \\ &= v(t)\cos\left[\arg\left(\boldsymbol{\tau}(u)\right)\right]\big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)}\end{aligned} \tag{21}$$

$$\begin{aligned}\frac{d}{dt}\beta(u)\big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)} &= \left.\frac{\dot\beta(u)}{\|\dot{\mathbf{p}}(u)\|}\right|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)} v(t)\\ &= v(t)\,\sin\left[\arg\left(\frac{\dot{\mathbf{p}}(u)}{\|\dot{\mathbf{p}}(u)\|}\right)\right]\Bigg|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)}\\ &= v(t)\sin\left[\arg\left(\boldsymbol{\tau}(u)\right)\right]\big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)}. \end{aligned} \tag{22}$$

Equations (21) and (22) exactly match the first two equations of model (7)–(9).

The time derivative of (20) can be evaluated bearing in mind (23), shown at the bottom of the page. Thus

$$\begin{aligned}\frac{d}{dt}\arg\left[\boldsymbol{\tau}(u)\right]\big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)} &= v(t)\left.\frac{\dot\alpha(u)\ddot\beta(u)-\ddot\alpha(u)\dot\beta(u)}{(\alpha^2(u)+\beta^2(u))^{\frac{3}{2}}}\right|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)}\\ &= v(t)\,\kappa_c(u)\big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)}. \end{aligned} \tag{24}$$

Definition 6 makes it possible to rewrite (16) as

$$\omega(t) := v(t)\kappa_c(u)\big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)} \tag{25}$$

so we can conclude that

$$\frac{d}{dt}\arg\left[\boldsymbol{\tau}(u)\right]\big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)} = \omega(t) \tag{26}$$

i.e., (26) matches the third equation of model (7)–(9).

We can conclude that the trajectory (18)–(20) is generated by model (7)–(9) owing to the chosen input functions $v(t)$ and $\omega(t)$. Hence, the path generated by (7)–(9) coincides with the planned curve $\mathbf{p}(u)$. ■

Remark 2: In order to generate G^3-paths with model (7)–(9), it is not strictly necessary to guarantee that $v(t) \neq 0$. On the other hand, if such a condition is not satisfied, there exist degenerate situations where the required geometric continuity is lost. For example, assign two command signals $v(t), \omega(t) \in C^1([0,t_f])$ such that there exists $t^* \in [0,t_f]$ where $v(t^*) = 0$ and $\omega(t^*) \neq 0$. It is immediately possible to verify that, according to (10)–(12), we have

$$\kappa(t^*) = \lim_{t\to t^*}\frac{v^2(t)\omega(t)}{[v^2(t)]^{\frac{3}{2}}} = \infty \tag{27}$$

hence, the curvature κ is not defined at t^*. This implies that, according to *Definitions 3–5*, the corresponding path is neither a G^2-path nor a G^3-path.

IV. Path-Inversion Algorithm

In this section, the proposed motion generation problem is solved by means of a dynamic path-inversion procedure. The aim is to synthesize a feedforward control such that, for any given interval $[0,t_f]$, the UMR starting at time 0 from the extended state

$$A = \{\mathbf{p}_A, \theta_A, v_A, \dot v_A, \omega_A, \dot\omega_A\}$$

will reach, at time t_f, the extended state

$$B = \{\mathbf{p}_B, \theta_B, v_B, \dot v_B, \omega_B, \dot\omega_B\}$$

while the following geometric and kinematics requirements are ensured:

I) The robot Cartesian path connecting $\mathbf{p}_A$ with $\mathbf{p}_B$ is a G^3-path.
II) $v(t) \neq 0\ \forall t \in (0,t_f)$.

The above conditions I and II impose some formal restrictions on both the initial and final extended states. In order to satisfy condition I, all of the following statements must be simultaneously verified.

1) If $v_A = 0$, then $\omega_A = 0$.
2) If $(v_A = 0) \wedge (\dot v_A = 0)$, then $\omega_A = 0$, $\dot\omega_A = 0$.
3) If $v_B = 0$, then $\omega_B = 0$.
4) If $(v_B = 0) \wedge (\dot v_B = 0)$, then $\omega_B = 0$, $\dot\omega_B = 0$.

On the other hand, condition II is satisfied only if all of the following statements are true.

5) If $v_A > 0$, then $v_B \geq 0$.
6) If $(v_A > 0) \wedge (v_B = 0)$, then $\dot v_B \leq 0$.
7) If $v_A < 0$, then $v_B \leq 0$.
8) If $(v_A < 0) \wedge (v_B = 0)$, then $\dot v_B \geq 0$.
9) If $(v_A = 0) \wedge (\dot v_A > 0)$, then $v_B \geq 0$.
10) If $(v_A = 0) \wedge (\dot v_A > 0) \wedge (v_B = 0)$, then $\dot v_B \leq 0$.
11) If $(v_A = 0) \wedge (\dot v_A < 0)$, then $v_B \leq 0$.
12) If $(v_A = 0) \wedge (\dot v_A < 0) \wedge (v_B = 0)$, then $\dot v_B \geq 0$.

In light of *Proposition 1*, it is sensible to impose that the path connecting $\mathbf{p}_A$ with $\mathbf{p}_B$ is a G^3-path even for the cases where $v(0) = 0$ and/or $v(t_f) = 0$. Consequently, it is necessary to enforce statements 1)–4). Indeed, if any of the statements from 1) to 4) are violated, the corresponding robot motion path cannot be a G^3-path

Analogously, focusing on statements 5)–12), if any of these are violated then, considering that $v(t)$ must be synthesized as a C^1-function, condition II cannot be satisfied. For example, assume that statement 6) is not true because $v_A > 0$, $v_B = 0$, and $\dot v_B > 0$. Therefore, it is easy to show that there does not exist a function $v(\cdot) \in C^1([0,t_f])$ satisfying $v(0) = v_A$, $v(t_f) = v_B$, $\dot v(t_f) = \dot v_B$, and $v(t) > 0\ \forall t \in (0,t_f)$.

Condition II simply means that we can have either $v(t) > 0$ or $v(t) < 0$ for all of the time instants t belonging to the open interval $(0,t_f)$. This should not be considered a loss of generality. Indeed, if the robot motion direction needs to be inverted as, for example, when there exists $t^* \in (0,t_f)$, such that $v(t) > 0\ \forall t \in (0,t^*)$ and $v(t) < 0\ \forall t \in (t^*,t_f)$, then the supervisor can split the motion generation into two parts, each of them characterized by a well-defined sign of the velocity inside the pertinent time interval. In the following, we denote as a forward movement (FM) the robot motion for which $v(t) > 0\ \forall t \in (0,t_f)$. Conversely, a backward movement (BM) is characterized by $v(t) < 0\ \forall t \in (0,t_f)$.

The overall procedure can be described with four steps. First, the supervisor has to decide for a FM or BM to reach the final extended state B. This choice mainly depends on the given interpolating data. The same data are used in the second step to plan the desired G^3-path. Then, the linear velocity command signal $v(t)$ is synthesized and, finally, the angular velocity command $\omega(t)$ is designed by exploiting the constructive proof provided for *Proposition 1*.

$$\arg\left[\boldsymbol{\tau}(u)\right]\big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)} = \begin{cases} \arctan\left[\frac{\dot\beta(u)}{\dot\alpha(u)}\right]\Big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)}, & \text{if } \dot\alpha(u) > 0\\ \pi + \arctan\left[\frac{\dot\beta(u)}{\dot\alpha(u)}\right]\Big|_{u=f^{-1}\left(\int_0^t v(\xi)d\xi\right)}, & \text{if } \dot\alpha(u) < 0\\ \frac{\pi}{2}, & \text{if } \dot\alpha(u) = 0 \text{ and } \dot\beta(u) > 0\\ -\frac{\pi}{2}, & \text{if } \dot\alpha(u) = 0 \text{ and } \dot\beta(u) < 0 \end{cases} \tag{23}$$

TABLE I
SELECTION CRITERIA FOR THE MOTION DIRECTION

$(v_A > 0) \wedge (v_B > 0)$ $(v_A > 0) \wedge (v_B = 0) \wedge (\dot{v}_B \leq 0)$ $(v_A = 0) \wedge (v_B > 0) \wedge (\dot{v}_A \geq 0)$ $(v_A = 0) \wedge (v_B = 0) \wedge (\dot{v}_A \geq 0) \wedge (\dot{v}_B < 0)$ $(v_A = 0) \wedge (v_B = 0) \wedge (\dot{v}_A > 0) \wedge (\dot{v}_B \leq 0)$	FM
$(v_A < 0) \wedge (v_B < 0)$ $(v_A < 0) \wedge (v_B = 0) \wedge (\dot{v}_B \geq 0)$ $(v_A = 0) \wedge (v_B < 0) \wedge (\dot{v}_A \leq 0)$ $(v_A = 0) \wedge (v_B = 0) \wedge (\dot{v}_A \leq 0) \wedge (\dot{v}_B > 0)$ $(v_A = 0) \wedge (v_B = 0) \wedge (\dot{v}_A < 0) \wedge (\dot{v}_B \geq 0)$	BM
$(v_A = v_B = 0) \wedge (\dot{v}_A = \dot{v}_B = 0)$	FM or BM

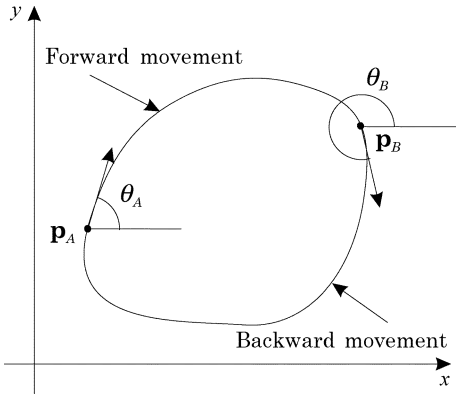

Fig. 1. If $v_A = v_B = 0$ and $\dot{v}_A = \dot{v}_B = 0$, then B can be reached by means of an FM or a BM.

Step 1: The motion direction is decided on the basis of the interpolating data v_A, $\dot{v}_A$, v_B, and $\dot{v}_B$. Table I can be used to select the motion direction (FM or BM). Note that, in the last case of Table I ($v_A = v_B = 0$ and $\dot{v}_A = \dot{v}_B = 0$), the supervisor can arbitrarily choose an FM or a BM (see Fig. 1).

Step 2: Determine a G^3-path connecting $\mathbf{p}_A$ with $\mathbf{p}_B$. This corresponds to finding a G^3-curve $\mathbf{p}(u)$, denoted according to (1), that satisfies the interpolation data deduced from the extended states A and B.

Two cases can be distinguished:

1) a general case: $v_A \neq 0$ and $v_B \neq 0$;
2) a critical case: $v_A = 0$ and/or $v_B = 0$.

For both cases, the curve $\mathbf{p}(u)$ must satisfy the following interpolating conditions:

$$\mathbf{p}(u_0) = \mathbf{p}_A, \quad \mathbf{p}(u_1) = \mathbf{p}_B \tag{28}$$

$$\boldsymbol{\tau}(u_0) = \begin{cases} [\cos\theta_A \ \sin\theta_A]^T, & \text{if FM} \\ [-\cos\theta_A \ -\sin\theta_A]^T, & \text{if BM} \end{cases} \tag{29}$$

$$\boldsymbol{\tau}(u_1) = \begin{cases} [\cos\theta_B \ \sin\theta_B]^T, & \text{if FM} \\ [-\cos\theta_B \ -\sin\theta_B]^T, & \text{if BM.} \end{cases} \tag{30}$$

From Section II, let us recall that $\kappa(s)$ and $\dot{\kappa}(s)$ denote the curvature and its derivative with respect to the arc length s [in particular, see (6)]. Then define

$$\kappa_A := \kappa(0), \quad \dot{\kappa}_A = \dot{\kappa}(0),$$
$$\kappa_B := \kappa\left(f(u_1)\right), \quad \dot{\kappa}_B := \dot{\kappa}\left(f(u_1)\right).$$

For the general case, the curvatures and their derivatives at the path end-points must be determined according to the formulas

$$\kappa_A = \begin{cases} \frac{\omega_A}{v_A}, & \text{if FM} \\ -\frac{\omega_A}{v_A}, & \text{if BM} \end{cases} \tag{31}$$

$$\kappa_B = \begin{cases} \frac{\omega_B}{v_B}, & \text{if FM} \\ -\frac{\omega_B}{v_B}, & \text{if BM} \end{cases} \tag{32}$$

$$\dot{\kappa}_A = \frac{\dot{\omega}_A v_A - \omega_A \dot{v}_A}{v_A^3} \tag{33}$$

$$\dot{\kappa}_B = \frac{\dot{\omega}_B v_B - \omega_B \dot{v}_B}{v_B^3}. \tag{34}$$

Now, consider the critical case. If $v_A = 0$, $\dot{v}_A = 0$, $\omega_A = 0$, and $\dot{\omega}_A = 0$, then the supervisor can freely assign any desired κ_A and $\dot{\kappa}_A$. Analogously, if $v_B = 0$, $\dot{v}_B = 0$, $\omega_B = 0$, and $\dot{\omega}_B = 0$, then κ_B and $\dot{\kappa}_B$ can be arbitrarily chosen.

On the other hand, if $v_A = 0$, $\omega_A = 0$, and $\dot{v}_A \neq 0$, then

$$\kappa_A = \begin{cases} \frac{\dot{\omega}_A}{\dot{v}_A}, & \text{if FM } (\dot{v}_A > 0) \\ -\frac{\dot{\omega}_A}{\dot{v}_A}, & \text{if BM } (\dot{v}_A < 0) \end{cases} \tag{35}$$

and $\dot{\kappa}_A$ can be freely assigned. Analogously, if $v_B = 0$, $\omega_B = 0$, and $\dot{v}_B \neq 0$, then

$$\kappa_B = \begin{cases} \frac{\dot{\omega}_B}{\dot{v}_B}, & \text{if FM } (\dot{v}_B > 0) \\ -\frac{\dot{\omega}_B}{\dot{v}_B}, & \text{if BM } (\dot{v}_B < 0) \end{cases} \tag{36}$$

and $\dot{\kappa}_B$ is arbitrarily chosen.

The interpolating conditions on curve $\mathbf{p}(u)$ are then completed by imposing [see (6)]

$$\kappa_c(u_0) = \kappa_A \tag{37}$$

$$\kappa_c(u_1) = \kappa_B \tag{38}$$

$$\dot{\kappa}_c(u_0) = \dot{\kappa}_A \left\|\dot{\mathbf{p}}(u_0)\right\| \tag{39}$$

$$\dot{\kappa}_c(u_1) = \dot{\kappa}_B \left\|\dot{\mathbf{p}}(u_1)\right\|. \tag{40}$$

An actual G^3-curve $\mathbf{p}(u)$ can be obtained by means of the closed-form expressions set out in [10]. In that work, a new curve primitive, named G^3-spline, was proposed. Such a primitive makes it possible to satisfy any set of interpolating conditions (28)–(30) or (37)–(40) and, at the same time, to finely shape the resulting curve by means of some freely assignable parameters.

Step 3: Choose $v(\cdot) \in C^1([0, t_f])$ with $v(t) \neq 0 \ \forall t \in (0, t_f)$, such that

$$v(0) = v_A \tag{41}$$

$$v(t_f) = v_B \tag{42}$$

$$\dot{v}(0) = \dot{v}_A \tag{43}$$

$$\dot{v}(t_f) = \dot{v}_B \tag{44}$$

$$\int_0^{t_f} v(\xi) d\xi = \begin{cases} f(u_1), & \text{if FM} \\ -f(u_1), & \text{if BM} \end{cases} \tag{45}$$

(we recall that $f(u_1)$ is the total arc length of the planned G^3-path connecting $\mathbf{p}_A$ with $\mathbf{p}_B$). Choosing $v(t)$ according to the interpolating conditions, (41)–(45) can be accomplished according to various schemes.

A viable velocity planning has been recently proposed in [18]. The command signal $v(\cdot) \in C^1([0, t_f])$ is generated with five properly joined spline curves ($i = 1, 2, \ldots, 5$):

$$v_i(t) = a_{1i} + 2a_{2i}t + 3a_{3i}t^2, \quad t \in [0, h_i] \tag{46}$$

with $\sum_{i=1}^{5} h_i = t_f$. In [18], it has been demonstrated that the generated velocity function, in the absence of velocity and acceleration upper bounds, is C^1 and strictly positive for any $t \in (0, t_f)$ and, moreover, satisfies with certainty constraints (41)–(45) for any set of interpolating conditions. When kinematics bounds have to be considered, as is certainly the case in the practical applications, it is still possible to use this velocity planning by applying a time-scaling procedure on t_f.

Step 4: The angular velocity function $\omega(\cdot) \in C^1([0,t_f])$ is defined according to

$$\omega(t) := v(t)\kappa(s)\big|_{s=\int_0^t v(\xi)d\xi} \quad \forall t \in [0,t_f] \text{ if FM} \tag{47}$$

$$\omega(t) := -v(t)\kappa(s)\big|_{s=-\int_0^t v(\xi)d\xi} \quad \forall t \in [0,t_f] \text{ if BM} \tag{48}$$

where $\kappa(s)$ is the curvature expressed as a function of the arc length s [see (6)].

The following result highlights the role of the inversion algorithm in steering the UMR.

Proposition 2: Let us consider any traveling time $t_f > 0$ and any extended states A and B satisfying assumptions 1)–12). Then, the control inputs $v(\cdot)$, $\omega(\cdot) \in C^1([0,t_f])$, synthesized by the proposed procedure, steer the UMR from the extended state A, at time 0, to the extended state B, at time t_f, in such a way that the generated motion path exactly matches the G^3-path planned at step 2 of the procedure.

Proof: This proof mainly relies on the "sufficiency" proof of *Proposition 1*.

First note that, as required, both $v(t)$ and $\omega(t)$ belong to $C^1([0,t_f])$. By definition, velocity $v(t)$ is $C^1([0,t_f])$ when planned according to the method proposed in [18]. It has already been demonstrated that an angular command signal defined according to (16) belongs to $C^1([0,t_f])$ when the planned curve is G^3. Thus, in the case of FM, the command signal (47) belongs to $C^1([0,t_f])$. Using similar reasonings in the case of BM, the command (48) belongs to $C^1([0,t_f])$.

As a second step, it is necessary to prove that the path generated by the robot exactly matches the planned G^3-path. In the case of FM, in the proof of *Proposition 1*, it has been demonstrated that the trajectory (18)–(20) is generated by model (7)–(9) with inputs (46) and (47) and initial state $[x_A\ y_A\ \theta_A]^T$. Thus, the path generated by (7)–(9) coincides with the planned curve $\mathbf{p}(u)$ so that, at time t_f, the robot state exactly coincides with the desired final state $[x_B\ y_B\ \theta_B]^T$. It is relevant to observe that the demonstration does not degenerate even in the case of initial and final velocities equal to zero.

In the case of BM, it is immediately possible to verify that $\theta(\cdot) = \arg\{\boldsymbol{\tau}(\cdot)\}$ so that it is necessary to demonstrate that the following time functions:

$$\alpha(u)\big|_{u=f^{-1}\left(-\int_0^t v(\xi)d\xi\right)} \tag{49}$$

$$\beta(u)\big|_{u=f^{-1}\left(-\int_0^t v(\xi)d\xi\right)} \tag{50}$$

$$\arg\{-\tau(u)\}\big|_{u=f^{-1}\left(-\int_0^t v(\xi)d\xi\right)} \tag{51}$$

satisfy model (7)–(9) with command inputs (46) and (48) and initial state $[x_A\ y_A\ \theta_A]^T$. The demonstration is analogous to that already seen for the FM and is omitted for conciseness.

To conclude the proof, it is necessary to demonstrate that the boundary conditions are satisfied not only for the robot state but also for the whole "extended state." The linear velocity profile is planned by imposing (41)–(44) so that the boundary conditions are automatically satisfied for both $v(t)$ and $\dot{v}(t)$. The demonstration for $\omega(t)$ and $\dot{\omega}(t)$ requires one to consider several cases depending on the initial and final linear velocities $v(0)$ and $v(t_f)$. In the following, only the FM case will be analyzed.

First consider the nondegenerate case $v(0) \neq 0$. Taking into account the command signal (47) and the first equation of (31), we obtain

$$\omega(0) = v(0)\kappa(0) = v_A\kappa_A = v_A\frac{\omega_A}{v_A} = \omega_A.$$

The angular acceleration $\dot{\omega}(0)$ can be obtained by deriving (47) [see also (17)]. The initial angular acceleration is obtained with the help of (31) and (33) as follows:

$$\begin{aligned}\dot{\omega}(0) &= v^2(0)\dot{\kappa}(0) + \dot{v}(0)\kappa(0)\\ &= v_A^2\dot{\kappa}_A + \dot{v}_A\kappa_A\\ &= v_A^2\frac{\dot{\omega}_A v_A - \omega_A\dot{v}_A}{v_A^3} + \dot{v}_A\frac{\omega_A}{v_A}\\ &= \dot{\omega}_A.\end{aligned}$$

Analogously, for $t = t_f$ and $v(t_f) \neq 0$, (32) makes it possible to write

$$\omega(t_f) = v(t_f)\kappa(t_f) = v(t_f)\kappa\left(f(u_1)\right) = v_B\kappa_B = v_B\frac{\omega_B}{v_B} = \omega_B.$$

Moreover, considering also (34), we obtain

$$\begin{aligned}\dot{\omega}(t_f) &= v^2(t_f)\dot{\kappa}(t_f) + \dot{v}(t_f)\kappa(t_f)\\ &= v^2(t_f)\dot{\kappa}\left(f(u_1)\right) + \dot{v}(t_f)\kappa\left(f(u_1)\right)\\ &= v_B^2\dot{\kappa}_B + \dot{v}_B\kappa_B\\ &= v_B^2\frac{\dot{\omega}_B v_B - \omega_B\dot{v}_B}{v_B^3} + \dot{v}_B\frac{\omega_A}{v_B}\\ &= \dot{\omega}_B.\end{aligned}$$

This demonstrates that, in case of nondegenerate situations, the initial and final interpolating conditions are exactly matched.

Now consider the critical situation where $v_A = 0$ and $\dot{v}_A = 0$. For any finite value of κ_A and $\dot{\kappa}_A$, we correctly obtain [see statement 2) of Section IV]

$$\begin{aligned}\omega(0) &= v(0)\kappa(0) = v_A\kappa_A = 0\\ \dot{\omega}(0) &= v^2(0)\dot{\kappa}(0) + \dot{v}(0)\kappa(0) = v_A^2\dot{\kappa}_A + \dot{v}_A\kappa_A = 0.\end{aligned}$$

Analogously, when $v_B = 0$ and $\dot{v}_B = 0$, we correctly obtain [see statement 4) of Section IV]

$$\begin{aligned}\omega(t_f) &= v(t_f)\kappa(t_F) = v(t_f)\kappa\left(f(u_1)\right) = v_B\kappa_B = 0\\ \dot{\omega}(t_f) &= v^2(t_f)\dot{\kappa}(t_f) + \dot{v}(t_f)\kappa(t_f)\\ &= v^2(t_f)\dot{\kappa}\left(f(u_1)\right) + \dot{v}(t_f)\kappa\left(f(u_1)\right)\\ &= v_A^2\dot{\kappa}_A + \dot{v}_A\kappa_A\\ &= 0.\end{aligned}$$

To conclude, consider the last critical case $v_A = 0$ and $\dot{v}_A \neq 0$. For the FM, taking into account (35), the initial angular velocity is correctly [see statement 1) of Section IV]

$$\omega(0) = v(0)\kappa(0) = v_A\kappa_A = v_A\frac{\dot{\omega}_A}{\dot{v}_A} = 0$$

while the angular acceleration, for any arbitrarily assigned $\dot{\kappa}_A$, coincides with the assigned $\dot{\omega}_A$ as follows:

$$\dot{\omega}(0) = v^2(0)\dot{\kappa}(0) + \dot{v}(0)\kappa(0) = v_A^2\dot{\kappa}_A + \dot{v}_A\kappa_A = \dot{v}_A\frac{\dot{\omega}_A}{\dot{v}_A} = \dot{\omega}_A.$$

Further considering the FM, and taking into account (36), the final angular velocity is correctly [see statement 3) of Section IV]

$$\omega(t_f) = v(t_f)\kappa(t_f) = v_B\kappa_B = v_B\frac{\dot{\omega}_B}{\dot{v}_B} = 0$$

while the angular acceleration, for any arbitrarily assigned $\dot{\kappa}_B$, coincides with the assigned $\dot{\omega}_B$ as follows:

$$\begin{aligned}\dot{\omega}(t_f) &= v^2(t_f)\dot{\kappa}(t_f) + \dot{v}(t_f)\kappa(t_f)\\ &= v^2(t_f)\dot{\kappa}\left(f(u_1)\right) + \dot{v}(t_f)\kappa\left(f(u_1)\right)\\ &= v_B^2\dot{\kappa}_B + \dot{v}_B\kappa_B\\ &= \dot{v}_B\frac{\dot{\omega}_B}{\dot{v}_B}\\ &= \dot{\omega}_B.\end{aligned}$$

The same reasonings are applicable to the BM case.

Thus, it is possible to conclude that the proposed control law makes it possible to steer the UMR from any feasible extended state A to any feasible extended state B while the resulting motion path coincides with the planned G^3-path. ■

Obviously, if the extended state A, used to replan the trajectory, coincides with the current extended state of the system, the C^1 continuity of the command signals is guaranteed also at the updating time. Thus, the composite command signals are globally C^1 and the piecewise path resulting from this iterative steering approach is globally a G^3-path.

V. Motion Generation Example

Consider that, at time 0, the extended state of the UMR is $A = \{[2\ 1]^T, \pi/4, 0, 0, 0, 0\}$. The desired future extended state at the chosen time $t_f = 4$ s is $B = \{[4\ 3]^T, -\pi/6, 0.5, 0, -0.5, 0.05\}$. In A and B, the Cartesian coordinates are expressed in meters, the angles in radians, the (angular) velocities in (radians/s) m/s, and the (angular) accelerations in (radians/s^2) m/s^2. These extended states satisfy all the assumptions 1)–12) (see Section IV) so that the steps of the inversion algorithm can be directly applied.

Step 1: According to Table I, we plan an FM. Indeed, the logical statement $(v_A = 0) \wedge (v_B > 0) \wedge (\dot{v}_A \geq 0)$ is true.

Step 2: The supervisor has to choose a G^3-curve $\mathbf{p}(u)$ satisfying interpolating conditions that depend on the extended states A and B. First, let us assume $u_0 = 0$ and $u_1 = 1$ for simplicity. Therefore, from (28)–(30), we have

$$\begin{aligned}\mathbf{p}(0) &= \mathbf{p}_A = \begin{bmatrix}2\\1\end{bmatrix}\\ \mathbf{p}(1) &= \mathbf{p}_B = \begin{bmatrix}4\\3\end{bmatrix}\\ \boldsymbol{\tau}(0) &= \begin{bmatrix}\frac{\sqrt{2}}{2}\\ \frac{\sqrt{2}}{2}\end{bmatrix}\\ \boldsymbol{\tau}(1) &= \begin{bmatrix}\frac{\sqrt{3}}{2}\\ -\frac{1}{2}\end{bmatrix}\end{aligned}$$

Because the extended state A is a critical case with $v_A = 0$, $\dot{v}_A = 0$, $\omega_A = 0$, and $\dot{\omega}_A = 0$, we can arbitrarily choose κ_A and $\dot{\kappa}_A$, for example, $\kappa_A = 1$ and $\dot{\kappa}_A = 0$. Consequently, from (37) and (39), we obtain

$$\kappa_c(0) = 1, \quad \dot{\kappa}_c(0) = 0.$$

On the other hand, from (32) and (34), we compute

$$\kappa_B = -1, \quad \dot{\kappa}_B = 0.2,$$

and, eventually, from (38) and (40)

$$\kappa_c(1) = -1, \quad \dot{\kappa}_c(1) = 0.2\left\|\dot{\mathbf{p}}(1)\right\|.$$

TABLE II
Coefficients of the Polynomial G^3-Curve

i	0	1	2	3	4	5	6	7
x_i	2.00	2.33	-3.85	0.00	4.75	11.37	-20.61	8.00
y_i	1.00	2.33	3.85	0.00	-15.04	18.79	-10.07	2.13

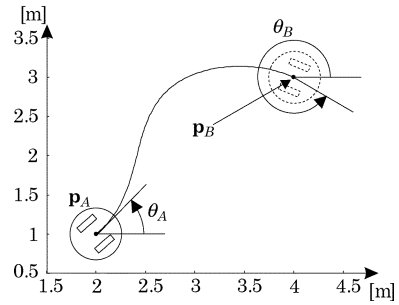

Fig. 2. G^3-path planning example.

TABLE III
Coefficients of the Polynomial Velocity Function $v(t)$ and Corresponding Traveling Times

i	a_{1i}	a_{2i}	a_{3i}	h_i
1	0	0	0.2719	0.8034
2	0.5264	0.6552	-0.1848	0.8000
3	1.2198	0.2116	-0.1134	0.7966
4	1.3412	-0.0593	-0.1820	0.8000
5	0.8969	-0.4961	0.2067	0.8000

The above interpolating conditions are then applied to a seventh-order polynomial curve $\mathbf{p}(u)$ where

$$\begin{aligned}\alpha(u) :=\ & x_0 + x_1u + x_2u^2 + x_3u^3 + x_4u^4\\ &+ x_5u^5 + x_6u^6 + x_7u^7\\ \beta(u) :=\ & y_0 + y_1u + y_2u^2 + y_3u^3 + y_4u^4\\ &+ y_5u^5 + y_6u^6 + y_7u^7.\end{aligned}$$

The coefficients x_i and y_i, listed in Table II, are deduced following the approach proposed in [10]. The resulting G^3-path connecting $\mathbf{p}_A$ with $\mathbf{p}_B$ is plotted in Fig. 2. The total path length is $f(u_1) = 3.3856$ m.

Step 3: The command signal $v(\cdot) \in C^1([0,4])$ is made of five properly joined polynomial curves (46), according to the approach proposed in [18]. The splines coefficients are shown in Table III together with the traveling time of each single curve. The overall velocity function is C^1 and positive for any $t \in (0,4)$. Moreover, it satisfies the boundary conditions (41)–(45).

Step 4: Taking into account both (47) and (6), the command angular velocity $\omega(\cdot) \in C^1([0,4])$ can be numerically computed as

$$\omega(t) = v(t)\kappa_c\left(f^{-1}(s)\right)\Big|_{s=\int_0^t v(\xi)d\xi}. \tag{52}$$

The control inputs $v(t)$ and $\omega(t)$ are plotted in Fig. 3.

It is worth noting that the supervisor can decide to do a path replanning before the running commands (46) and (52) are completed. There may be a variety of reasons to perform an early replanning, for example,

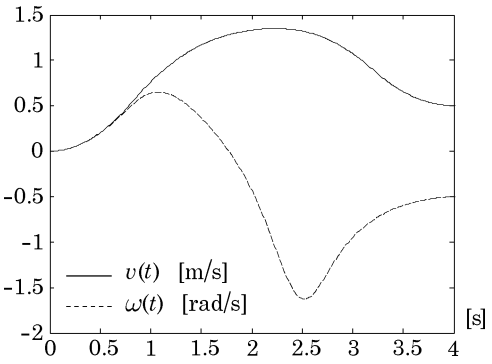

Fig. 3. Command signals $v(t)$ and $\omega(t)$.

to adjust for a sudden obstacle interfering with the robot's motion or to correct for an increasing path following error due to the mismodeling determined by (7)–(9) with respect to the robot actual behavior.

VI. Conclusion

In this paper, a new steering method that guarantees continuous-acceleration control inputs has been proposed for unicycle wheeled mobile robots. This approach relies on a path planning with third-order geometric continuity and on a dynamic path-inversion algorithm.

The exposed method is well suited to be implemented using an iterative steering strategy issued by a supervisory control system to perform sensor-based autonomous navigation. The design of such a supervisor is indeed a promising possible direction for future research work.

References

[1] L. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, pp. 497–517, 1957.

[2] J. Reeds and R. Shepp, "Optimal paths for a car that goes both forward and backward," *Pacific J. Math.*, vol. 145, no. 2, pp. 367–393, 1990.

[3] J. Laumond, S. Sekhavat, and F. Lamiraux, "Guidelines in nonholonomic motion planning for mobile robots," in *Robot Motion Planning and Control*, J.-P. Laumond, Ed. Berlin, Germany: Springer, 1998, pp. 1–53.

[4] W. Nelson, "Continuous-curvature paths for autonomous vehicles," in *Proc. IEEE Conf. Robotics and Automation*, vol. 3, May 1989, pp. 1260–1264.

[5] Y. Kanayama and B. Hartman, "Smooth local path planning for autonomous vehicles," in *Proc. IEEE Int. Conf. Robotics and Automation, ICRA89*, vol. 3, Scottsdale, AZ, May 1989, pp. 1265–1270.

[6] K. Komoriya and K. Tanie, "Trajectory design and control of a wheeled-type mobile robot using b-spline curve," in *Proc. IEEE/RSJ Int. Workshop Intelligent Robots and Systems, IROS89*, Tsukuba, Japan, Sept. 1989, pp. 398–405.

[7] H. Delingette, M. Hébert, and K. Ikeuchi, "Trajectory generation with curvature constraint based on energy minimization," in *Proc. IEEE-RSJ Int. Conf. Intelligent Robots and Systems*, Osaka, Japan, Nov. 1991, pp. 206–211.

[8] S. Fleury, P. Souères, J.-P. Laumond, and R. Chatila, "Primitives for smoothing paths of mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Atlanta, GA, Sept. 1993, pp. 832–839.

[9] 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. Transport. Syst.*, vol. 3, pp. 27–36, Mar. 2002.

[10] A. Piazzi, M. Romano, and C. Guarino Lo Bianco, "G^3-splines for the path planning of wheeled mobile robots," in *Proc. 2003 Eur. Control Conf. ECC 2003*, Cambridge, U.K., Sept. 2003.

[11] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: introductory theory and examples," *Int. J. Contr.*, vol. 61, no. 6, pp. 1327–1361, 1995.

[12] P. Lucibello and G. Oriolo, "Stabilization via iterative state steering with application to chained-form systems," in *Proc. 35th IEEE Conf. Decision and Control*, vol. 3, Kobe, Japan, Dec. 1996, pp. 2614–2619.

[13] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot Motion Planning and Control*, J.-P. Laumond, Ed. Berlin, Germany: Springer, 1998, pp. 171–253.

[14] C.-C. Hsiung, *A First Course in Differential Geometry*. Cambrige, MA: International Press, 1997.

[15] B. A. Barsky and J. C. Beatty, "Local control of bias and tension in beta-spline," *Computer Graph.*, vol. 17, no. 3, pp. 193–218, 1983.

[16] J. Peters, "Geometric continuity," in *Handbook of Computer Aided Geometric Design*, G. Farin, J. Hoschek, and M.-S. Kim, Eds. Amsterdam, The Netherlands: North-Holland, 2002, pp. 193–229.

[17] L. de Souza and M. Veloso, "AI planning in supervisory control systems," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, vol. 4, Oct. 1996, pp. 3153–3158.

[18] C. Guarino Lo Bianco, A. Piazzi, and M. Romano, "Velocity planning for autonomous vehicles," in *Proc. IEEE Intelligent Vehicles Symp. IV2004*, Parma, Italy, June 14–17, 2004, pp. 413–418.

A Timing Model for Vision-Based Control of Industrial Robot Manipulators

Yanfei Liu, Adam W. Hoover, and Ian D. Walker

***Abstract*—Visual sensing for robotics has been around for decades, but our understanding of a timing model remains crude. By timing model, we refer to the delays (processing lag and motion lag) between "reality" (when a part is sensed), through data processing (the processing of image data to determine part position and orientation), through control (the computation and initiation of robot motion), through "arrival" (when the robot reaches the commanded goal). In this study, we introduce a timing model where sensing and control operate asynchronously. We apply this model to a robotic workcell consisting of a Stäubli RX-130 industrial robot manipulator, a network of six cameras for sensing, and an off-the-shelf Adept MV-19 controller. We present experiments to demonstrate how the model can be applied.**

***Index Terms*—Timing model, visual servoing, workcell.**

I. Introduction

Fig. 1 shows the classic structure for a visual servoing system [1]. In this structure, a camera is used in the feedback loop. It provides feedback on the *actual* position of something being controlled, for example, a robot. This structure can be applied to a variety of systems, including eye-in-hand systems, part-in-hand systems, and mobile robot systems.

In an eye-in-hand system [2]–[6], the camera is mounted on the end-effector of a robot and the control is adjusted to obtain the desired appearance of an object or feature in the camera. Gangloff [2] developed a visual servoing system for a six–degree–of–freedom (DOF)

Manuscript received July 3, 2003; revised January 22, 2004. This paper was recommended by Associate Editor Y.-H. Liu and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported by the South Carolina Commission on Higher Education and the U.S. Office of Naval Research. This paper was presented in part at the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, October 2003.

The authors are with the Electrical and Computer Engineering Department, Clemson University, Clemson, SC 29634-0915 USA (e-mail: lyanfei@clemson.edu; ahoover@clemson.edu; iwalker@clemson.edu).

Digital Object Identifier 10.1109/TRO.2004.829460