Regularity Properties and Deformation of Wheeled Robots Trajectories

Quang-Cuong Pham and Yoshihiko Nakamura

***Abstract*— Our contribution in this article is twofold. First, we identify the regularity properties of the trajectories of planar wheeled mobile robots. By regularity properties of a trajectory we mean whether this trajectory, or a function computed from it, belongs to a certain class C^n (the class of functions that are differentiable n times with a continuous n^{th} derivative). We show that, under some generic assumptions about the rotation and steering velocities of the wheels, any non-degenerate wheeled robot belongs to one of the two following classes. Class I comprises those robots whose admissible trajectories in the plane are C^1 and piecewise C^2; and class II comprises those robots whose admissible trajectories are C^1, piecewise C^2 and, in addition, curvature-continuous. Second, based on this characterization, we derive new feedback control and gap filling algorithms for wheeled mobile robots using the recently-developed affine trajectory deformation framework.**

I. Introduction

The movements of any wheeled robot in the plane are restricted by *nonholonomic constraints*, which arise from the rolling-without-slipping constraint at work at the robot's wheels. A large body of work has been devoted to the planning and control of wheeled robots under such nonholonomic constraints, see e.g. [1] for a review. For instance, a number of articles built upon the pioneering results of Dubins [2] to construct optimal trajectories for car-like robots as a concatenation of line segments and circular arcs. However, as remarked by Boissonnat et al. [3] and Fraichard and Scheuer [4], the curvature of such paths is discontinuous at the transitions between line segments and circular arcs, or between circular arcs of different radii, such that a car-like robot that follows such paths would need to stop at the transition points in order to reorient its wheels (since the orientations – or steering angles – of the wheels are related to the path curvature). As a consequence, these authors developed methods to plan *curvature-continuous paths*, which can hence be followed without stopping by car-like robots.

We say that a trajectory (x, y) of a wheeled robot is *non-halting* if the linear velocity $\sqrt{\dot{x}^2 + \dot{y}^2}$ is always strictly positive. In particular, according to the previous paragraph, a non-halting trajectory for a car-like robot should not contain any point of discontinuous curvature. In section II, we study a more general question: for a *given* wheeled robot, what are the non-halting trajectories that it can execute in the plane? We characterize such *admissible* trajectories by their regularity properties, that is, by whether these trajectories, or some functions computed from them, belong to certain classes C^n (C^i is the class of functions that are differentiable i times with a continuous i^{th} derivative). Our work mainly builds upon the general classification of planar wheeled robots obtained by Campion et al. [5]. Note that the trajectories we are interested in are the (x, y) trajectories of a given fixed point on the robot's frame, and not the trajectories of the robot's full *posture* (x, y, θ). The former problem is sometimes referred to as the *point tracking problem*, as opposed to the *posture tracking problem* [6].

Next, based on that characterization of admissible trajectories, we derive in section III new feedback control and gap filling algorithms for wheeled robots using the framework of affine trajectory deformations [7]. This recently-developed framework allows making corrections to a previously planned trajectory in order to deal with unexpected disturbances (such as a change of the target position or the appearance of an unforeseen obstacle), without having to recompute the entire trajectory. In contrast with previous trajectory deformation methods [8], [9], affine-geometry-based algorithms are exact (given by closed-form formulas), can be executed in one step, and do not require any trajectory re-integration [7]. The new feedback control and gap filling algorithms we propose thus naturally inherit these computational advantages.

II. Regularity properties of wheeled robots trajectories

A. Model description, admissible commands

At the kinematic level, any non-degenerate wheeled robot whose wheels obey the rolling-without-slipping constraints can be modeled by [5]

$$\begin{cases} \dot{\boldsymbol{\xi}} &= \mathbf{B}(\boldsymbol{\xi}, \boldsymbol{\beta})\boldsymbol{\eta} \\ \dot{\boldsymbol{\beta}} &= \boldsymbol{\zeta} \end{cases}$$

where $\boldsymbol{\xi} = (x, y, \theta)^\top$ is the *posture* of the robot and $\boldsymbol{\beta} = (\beta_1 \dots \beta_h)^\top$ contains the steering angles of the *centered orientable conventional wheels* ($h = 0$ if there are no such wheels, see the top diagram in Fig. 1 for illustration). The control inputs, or commands, to the system are given by $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$.

Before proceeding further with the regularity constraints on the commands, let us define some notations.

Regularity classes: For convenience, we shall denote by D^n the class of functions that are C^{n-1} and piecewise C^n. For instance, a D^0 function is piecewise continuous (but not

This work was supported by the Nissan Global Foundation (Cognitive Science Field, 2009-2011), by "Grants-in-Aid for Scientific Research" for JSPS fellows, and by a JSPS postdoctoral fellowship.

Q.-C. Pham and Y. Nakamura are with the Department of Mechano-Informatics, University of Tokyo, Japan cuong.pham@normalesup.org, nakamura@ynl.t.u-tokyo.ac.jp

necessarily continuous), while a D^1 function is continuous and piecewise differentiable (see [7]). $\triangle$

The commands $\boldsymbol{\zeta} = (\zeta_1 \dots \zeta_n)$ correspond to the *steering velocities* of the robot's centered orientable conventional wheels. We make the assumption that the ζ_i's (or equivalently, the $\dot{\beta}_i$'s) are D^0, which means that they can contain discontinuities. Such discontinuities would correspond to e.g. hard turns of the steering wheel in a car. This assumption is often implicitly made in the literature: for instance, the derivative of the curvature of the trajectories that are constructed for car-like robots in [4] are *discontinuous* (see e.g. Fig. 2 of [4]), which corresponds to a discontinuous steering velocity of the wheels (from the relation $\kappa \propto \tan\beta$ in the car, where κ is the trajectory curvature).

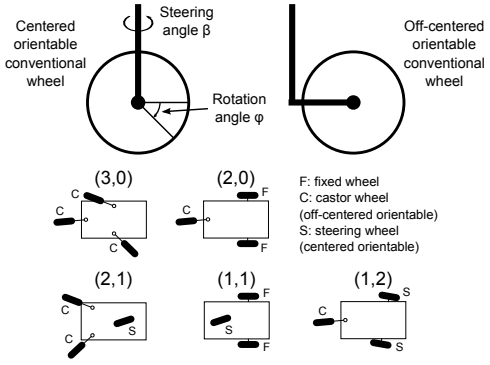

Fig. 1. Top: centered orientable and off-centered orientable conventional wheels. Bottom: examples of wheeled robots of each type (following [5]).

On the other hand, the commands $\boldsymbol{\eta}$ control the ranges of movement of the robot that can be made without steering any of its wheels (see section IV.B of [5] for details). In fact, one has (see equation 32 of [5])

$$\dot{\boldsymbol{\phi}} = \mathbf{E}(\boldsymbol{\beta}, \boldsymbol{\beta}_{oc})\boldsymbol{\eta}, \tag{1}$$

where $\boldsymbol{\beta}_{oc}$ contains the steering angles of the *off-centered orientable conventional wheels* (see the top diagram in Fig. 1 for illustration) and $\boldsymbol{\phi} = (\phi_1 \dots \phi_H)$ contains the *rotation angles* of *all the wheels* of the robot. Next, we make the assumption that the $\dot{\phi}_i$'s are D^1, meaning that they must be *continuous*. Note that this contrasts with the previously discussed case of the *steering angles*, where the $\dot{\beta}_i$'s can contain discontinuities. This assumption is also often made implicitly in the literature: for instance, when authors assume that the velocity of a car-like robot is continuous. Thus, from equation (1), one can deduce that the admissible commands η_i's are also D^1. Note however that the $\dot{\eta}_i$'s (or equivalently, the $\ddot{\phi}_i$'s) can contain discontinuities. Such discontinuities would correspond to hard presses on the throttle or on the brake pedal in a car.

In summary, the space of admissible commands $\boldsymbol{\eta}$ is D^1 and the space of admissible commands $\boldsymbol{\zeta}$ is D^0.

B. Admissible trajectories

As shown in [5], any planar wheeled mobile robot can be described by one out of the five sets of "forward" kinematic equations of Table I (see also Fig. 1, bottom panel), given a suitable choice of a reference point and of a basis attached to the robot frame.

For each type of robot, we now characterize the admissible trajectories *given the sets of admissible commands assumed in the previous section*. The reader is referred to Table I for the notations and equations.

1) Type (3,0): Consider $(\eta_1, \eta_2, \eta_3) \in D^1$. The third "forward" kinematic equation ($\dot{\theta} = \eta_3$) implies that $\theta \in D^2$. The first and the second forward equations then imply that $x \in D^2$ and $y \in D^2$.

Conversely, consider a trajectory $\mathcal{C} = (x, y) \in D^2$. One can choose an arbitrary function $\theta \in D^2$ and then compute $(\eta_1, \eta_2, \eta_3) \in D^1$ by the "reverse" equations.

In summary, a trajectory of a (3,0) wheeled robot is admissible if and only if it belongs to D^2.

2) Type (2,0): Consider $(\eta_1, \eta_2) \in D^1$. As previously, the forward equations imply that x and y belong to D^2.

Conversely, consider a trajectory $\mathcal{C} = (x, y) \in D^2$. Since the trajectory is non-halting ($\dot{x}^2 + \dot{y}^2 > 0$), one can compute θ by the first reverse equation $\theta = \arctan2(\dot{y}, \dot{x})$, where

$$\arctan2(b, a) = \begin{cases} \pi/2 & \text{if } a = 0 \text{ and } b \geq 0 \\ -\pi/2 & \text{if } a = 0 \text{ and } b < 0 \\ \arctan(b/a) & \text{if } a \neq 0 \end{cases}.$$

Remark that the so-calculated θ belongs to D^1, but not necessarily to D^2. Next, one can compute η_2 by the third reverse equation. For η_2 to be in D^1, one would need $\theta \in D^2$. As just remarked, the latter condition is *not automatically* guaranteed by $\mathcal{C} = (x, y) \in D^2$. On the other hand, demanding that $\mathcal{C} = (x, y) \in D^3$ would be unduly restrictive. Thus the condition $\theta = \arctan2(\dot{y}, \dot{x}) \in D^2$ must be specified as an independent supplementary condition. Note that the condition $\arctan2(\dot{y}, \dot{x}) \in D^2$ is equivalent to requiring the path curvature to be continuous [3], [4].

In summary, a trajectory $\mathcal{C}$ of a (2,0) robot is admissible if and only if it belongs to D^2, *and* if the function $\arctan2(\dot{y}, \dot{x})$ also belongs to D^2.

3) Type (2,1): Consider $(\eta_1, \eta_2) \in D^1$ and $\zeta \in D^0$. The third and fourth forward equations imply that θ and β belong respectively to D^2 and D^1. Next, the first and second forward equations imply that x and y belong to D^2.

Conversely, consider a trajectory $\mathcal{C} = (x, y) \in D^2$. One can choose an arbitrary function $\theta \in D^2$ and then compute successively $\beta \in D^1$, $(\eta_1, \eta_2) \in D^1$, and $\zeta \in D^0$ by the reverse equations.

In summary, as for (3,0) robots, a trajectory of a (2,1) robot is admissible if and only if it belongs to D^2.

4) Type (1,1): As previously, a necessary condition for the admissibility of a trajectory is that it belongs to D^2. Conversely, consider $\mathcal{C} = (x, y) \in D^2$. The first reverse equation allows to compute $\theta \in D^1$. Remark that, as for $(2, 0)$ robots, the so-calculated θ does not necessarily belong to D^2. Next, β can be computed from the second reverse equation. Remark that the derivative of θ is used in the computation of β, such that β belongs to D^0, but not necessarily to D^1. However, in order to compute next ζ, one needs $\beta \in D^1$, and consequently $\theta = \arctan2(\dot{y}, \dot{x}) \in D^2$.

TABLE I

FORWARD AND REVERSE KINEMATIC EQUATIONS FOR PLANAR WHEELED ROBOTS

Type	Examples	"Forward" kinematic equations (cf [5])	"Reverse" equations	Admissibility cond.
(3,0)	Omnidirectional robots (see Fig. 1, bottom panel)	$\dot{x} = \eta_1 \cos\theta - \eta_2 \sin\theta$ $\dot{y} = \eta_1 \sin\theta + \eta_2 \cos\theta$ $\dot{\theta} = \eta_3$	$\theta \in D^2$ (arbitrary) $\eta_1 = \dot{x}\cos\theta + \dot{y}\sin\theta$ $\eta_2 = -\dot{x}\sin\theta + \dot{y}\cos\theta$ $\eta_3 = \dot{\theta}$	$(x, y) \in D^2$
(2,0)	Two-wheel differential drive	$\dot{x} = -\eta_1 \sin\theta$ $\dot{y} = \eta_1 \cos\theta$ $\dot{\theta} = \eta_2$	$\theta = \arctan2(\dot{y}, \dot{x})$ $\eta_1 = \sqrt{\dot{x}^2 + \dot{y}^2}$ $\eta_2 = \dot{\theta}$	$(x, y) \in D^2$ $\arctan2(\dot{y}, \dot{x}) \in D^2$
(2,1)	Unicycle	$\dot{x} = -\eta_1 \sin(\theta + \beta)$ $\dot{y} = \eta_1 \cos(\theta + \beta)$ $\dot{\theta} = \eta_2$ $\dot{\beta} = \zeta_1$	$\theta \in D^2$ (arbitrary) $\beta = \arctan2(\dot{y}, \dot{x}) - \theta$ $\eta_1 = \sqrt{\dot{x}^2 + \dot{y}^2}$ $\eta_2 = \dot{\theta}$ $\zeta_1 = \dot{\beta}$	$(x, y) \in D^2$
(1,1)	Bicycle, car-like robot	$\dot{x} = -\eta_1 L \sin\theta \sin\beta$ $\dot{y} = \eta_1 L \cos\theta \sin\beta$ $\dot{\theta} = \eta_1 \cos\beta$ $\dot{\beta} = \zeta_1$	$\theta = \arctan2(\dot{y}, \dot{x})$ $\beta = \arctan2(\dot{y}/(L\cos\theta), \dot{\theta})$ $\eta_1 = \sqrt{\dot{x}^2 + \dot{y}^2}/(L\sin\beta)$ $\zeta_1 = \dot{\beta}$	$(x, y) \in D^2$ $\arctan2(\dot{y}, \dot{x}) \in D^2$
(1,2)	Kludge robot (cf [5])	$\dot{x} = -\eta_1(2L\cos\theta\sin\beta_1\sin\beta_2 + L\sin\theta\sin(\beta_1+\beta_2))$ $\dot{y} = -\eta_1(2L\sin\theta\sin\beta_1\sin\beta_2 - L\cos\theta\sin(\beta_1+\beta_2))$ $\dot{\theta} = \eta_1 \sin(\beta_2 - \beta_1)$ $\dot{\beta}_1 = \zeta_1$ $\dot{\beta}_2 = \zeta_2$	$\theta \in D^2$ (arbitrary) $\beta_1 = \arctan2(\dot{x}\cos\theta + \dot{y}\sin\theta, 2L\dot{\theta} - \dot{x}\sin\theta + \dot{y}\cos\theta)$ $\beta_2 = \arctan2(\dot{x}\cos\theta + \dot{y}\sin\theta, -2L\dot{\theta} - \dot{x}\sin\theta + \dot{y}\cos\theta)$ $\eta_1 = \dot{\theta}/\sin(\beta_2 - \beta_1)$ $\zeta_1 = \dot{\beta}_1$ $\zeta_2 = \dot{\beta}_2$	$(x, y) \in D^2$

In summary, as for (2,0) robots, a trajectory $\mathcal{C}$ of a (1,1) robot is admissible if and only if it belongs to D^2, *and* if the function $\arctan2(\dot{y}, \dot{x}) \in D^2$ also belongs to D^2.

5) Type (1,2): This type of robots can be treated in the same way as (3,0) and (2,1) robots. A trajectory of a (1,2) robot is admissible if and only if it belongs to D^2.

6) Summary: From the previous development, one can divide wheeled robots into two classes. Class I comprises robots of type (3,0), (2,1), and (1,2), or in other words, those whose *degree of maneuvrability* [5] equals 3. A trajectory for robots of this class is admissible if and only if it belongs to D^2.

Class II comprises robots of type (2,0) and (1,1), or in other words, those whose *degree of maneuvrability* equals 2. A trajectory $\mathcal{C} = (x, y)$ for robots of this class is admissible if and only if it belongs to D^2 *and* if the function $\arctan2(\dot{y}, \dot{x})$ also belongs to D^2 (i.e. the path is curvature-continuous). Note that the last condition is *not* required for robots of class I since, for these robots, θ is not directly related to (x, y) by $\theta = \arctan2(\dot{y}, \dot{x})$.

Important remark: From a computational viewpoint, if one obtains an admissible trajectory $\mathcal{C}'(t)_{t\in[0,T]}$ (for instance by deforming a given $\mathcal{C}(t)_{t\in[0,T]}$), the reverse equations allow to easily compute the commands that generate that trajectory by some differentiations and elementary operations. $\triangle$

Physical constraints: In this article, we have been only concerned with the differential constraints that stem from the nonholonomic nature of the wheeled robots. In practice, other constraints, such as upper limits on the absolute acceleration or on the trajectory curvature, further restrict the set of admissible trajectories. These constraints cannot be dealt with by studying the regularity properties of the trajectories alone: other specific methods are thus required at the trajectory planning stage in addition to the methods discussed here (see also the discussion of section III-B). $\triangle$

III. AFFINE DEFORMATION OF WHEELED ROBOTS TRAJECTORIES AND APPLICATIONS

A. Affine deformation framework

In [7] we developed a method based on affine transformations to correct the trajectories of nonholonomic mobile robots, such as planar wheeled robots or 3D underwater vehicles, while respecting their nonholonomic constraints. We now briefly recall some features of affine trajectory corrections for 2D mobile robots. For a general presentation, the reader is referred to [7].

Let $\mathbb{A}$ be the affine plane. An element $\mathbf{w} \in \mathbb{R}^2$ transforms a point $P \in \mathbb{A}$ into another point P' by $P' = P + \mathbf{w}$, which can also be noted $\overrightarrow{PP'} = \mathbf{w}$.

Given a point $O \in \mathbb{A}$ (the origin), an affine transformation $\mathcal{F}$ can be defined by a couple $(\mathbf{w}, \mathcal{M})$ where $\mathbf{w} \in \mathbb{R}^2$ and $\mathcal{M}$ is a nonsingular linear map $\mathbb{R}^2 \to \mathbb{R}^2$. The transformation $\mathcal{F}$ acts on $\mathbb{A}$ by

$$\forall P \in \mathbb{A} \quad \mathcal{F}(P) = O + \mathcal{M}(\overrightarrow{OP}) + \mathbf{w}.$$

If P_0 is a fixed-point of $\mathcal{F}$, then $\mathcal{F}$ can be written in the form

$$\forall P \in \mathbb{A} \quad \mathcal{F}(P) = P_0 + \mathcal{M}(\overrightarrow{P_0P}).$$

Note that the affine transformations of the plane form a Lie group, denoted GA_2, of dimension 6: 2 coordinates for the translation and 4 coordinates for the linear map.

Let $\mathcal{C}(t)_{t\in[0,T]}$ be a trajectory in the affine plane $\mathbb{A}$ and $\tau \in [0, T]$, a given time instant. We say that a transformation $\mathcal{F}$ occurring at τ *deforms* $\mathcal{C}(t)_{t\in[0,T]}$ into $\mathcal{C}'(t)_{t\in[0,T]}$ if

$$\begin{array}{ll} \forall t < \tau & \mathcal{C}'(t) = \mathcal{C}(t) \\ \forall t \geq \tau & \mathcal{C}'(t) = \mathcal{F}(\mathcal{C}(t)). \end{array} \quad (2)$$

Given an admissible trajectory $\mathcal{C}$ and a time instant τ, an affine transformation $\mathcal{F}$ is said to be admissible if $\mathcal{F}$ deforms $\mathcal{C}$ at time τ into an admissible trajectory.

B. Affine deformations for wheeled robots

We have seen in section II that, regarding the conditions of admissibility of trajectories, wheeled robots are divided into two classes: class I in which a trajectory is admissible if and only if it belongs to D^2, and class II in which a trajectory (x, y) is admissible if and only if it belongs to D^2 and if the function $\theta = \arctan2(\dot{y}, \dot{x})$ also belongs to D^2. Thus, as long as we are interested in admissible trajectories and in affine trajectory corrections, there is *no difference* between any robot of class I and the unicycle, nor is there any difference between any robot of class II and the bicycle. The only difference appears at the stage of calculating the commands corresponding to the deformed trajectory. However, for each type of robot, this can be done easily using the reverse equations of Table I.

We now briefly recall the results obtained in [7] regarding affine corrections for the unicycle (which thus also apply for robots of class I) and for the bicycle (which thus also apply for robots of class II).

Unicycle: Let $\mathcal{C}(t)_{t\in[0,T]}$ be an admissible trajectory for a unicycle in the plane. At each time instant τ where $\mathcal{C}(\tau)$ is not an inflection point, the admissible affine deformations form a Lie subgroup of dimension 2 of GA_2. Using an admissible affine transformation at τ, it is thus possible to correct the final position towards any desired position in the plane. It is also possible to correct the final orientation of the unicycle, or, using two successive deformations, avoid obstacles. $\triangle$

Bicycle: Let $\mathcal{C}(t)_{t\in[0,T]}$ be an admissible trajectory for a bicycle in the plane. At each time instant τ where $\mathcal{C}(\tau)$ is not an inflection point, the admissible affine deformations form a Lie subgroup of dimension 1 of GA_2. There are thus less "freedom" to make corrections. However, if the set of *tangents* to the initial trajectory is rich enough, it is possible to correct the final position or the final orientation of the bicycle towards a large range of values. $\triangle$

Physical constraints: As mentioned earlier, other constraints than the differential constraints we considered, such as upper limits on the absolute acceleration or on the trajectory curvature, could further restrict the set of admissible trajectories and hence the set of admissible affine deformations. This can be dealt with by observing that the changes in acceleration or curvature from the original trajectory can be computed from the affine transformation at hand (see also [10]). The integration of such constraints into the current framework represents an important task (see e.g. [4], [11]). $\triangle$

C. Position and orientation correction for the bicycle using three *affine deformations*

In addition to the results just recalled, we now show that, for robots of class II, composing *three* successive deformations allows reaching any desired final position P_d and orientation, as long as the initially planned trajectory is not a straight line, as follows

1) select three (non-inflection) time instants τ_1, τ_2, and τ_3 with $\tau_1 < \tau_2 < \tau_3$, such that the velocity vectors $\mathbf{v}(\tau_1)$, $\mathbf{v}(\tau_2)$ and $\mathbf{v}(\tau_3)$ are pairwise non-collinear. Such three time instants exist since $\mathcal{C}$ is not a straight line;
2) apply a first deformation on $\mathcal{C}$ at τ_3 to obtain $\mathcal{C}'$, with $\mathcal{C}'(T) = \mathcal{C}(T) + \alpha_3\mathbf{v}(\tau_3)$, where α_3 is a coefficient to be tuned later;
3) decompose $\overrightarrow{\mathcal{C}'(T)P_d}$ in the basis $\{\mathbf{v}(\tau_1), \mathbf{v}(\tau_2)\}$ as $\overrightarrow{\mathcal{C}'(T)P_d} = \alpha_1\mathbf{v}(\tau_1) + \alpha_2\mathbf{v}(\tau_2)$;
4) apply a second deformation on $\mathcal{C}'$ at τ_2 to obtain $\mathcal{C}''$, with $\mathcal{C}''(T) = \mathcal{C}'(T) + \alpha_2\mathbf{v}(\tau_2)$;
5) apply a third deformation on $\mathcal{C}''$ at τ_1 to obtain $\mathcal{C}'''$, with $\mathcal{C}'''(T) = \mathcal{C}''(T) + \alpha_1\mathbf{v}(\tau_1)$. By construction $\mathcal{C}'''(T) = \mathcal{C}'(T) + \alpha_2\mathbf{v}(\tau_2) + \alpha_1\mathbf{v}(\tau_1) = P_d$.

Remark that the final *orientation* of $\mathcal{C}'''$ depends on α_3 as shown in Fig. 2. The formula to algebraically compute α_3 as a function of the desired final orientation can be obtained in a similar way as in [7].

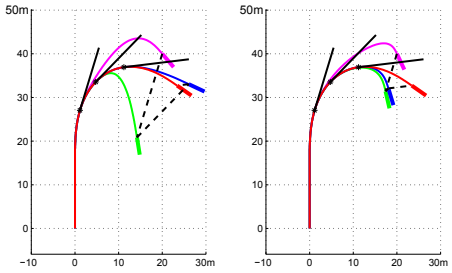

Fig. 2. Position and orientation correction using three successive affine deformations. The original trajectory ($\mathcal{C}$) is in red. The resulting trajectories after the first, second and third deformations are respectively in blue ($\mathcal{C}'$), green ($\mathcal{C}''$) and magenta ($\mathcal{C}'''$). The left and right plots correspond to two different values of α_3. Remark that the trajectory $\mathcal{C}'''$ (magenta) ends at the same position ($P_d = (20, 40)$) in the left and right plots, but that its final orientation differs significantly between the two plots. By varying α_3, it is thus possible to cover a large range of possible desired final orientations while keeping the desired final position fixed.

Finally, remark that one can also set the final linear speed to arbitrary values while keeping the final position and orientation unchanged by using the extension technique of section III-E.

D. Feedback control

So far, we have been focusing on perturbations affecting the state of the target (position and/or orientation) or the environment (unexpected appearance of obstacles). Here we show, through a simplified feedback control algorithm, how affine corrections can also be used to deal with perturbations affecting the robot's own state.

Consider the example of the bicycle or car-like robot (robots of class II), of equation

$$\left\{\begin{array}{rcl} \dot{x} &=& v\cos(\theta) \\ \dot{y} &=& v\sin(\theta) \\ \dot{\theta} &=& \frac{v\tan(\beta)}{L} \\ \dot{\beta} &=& \zeta \end{array}\right. , \qquad (3)$$

which can be put easily in the form of a robot of type (1,1). For convenience, we assume here that the control inputs are $(a, \zeta) \in D^0$ (instead of $v \propto \eta_1 \in D^1$ and $\zeta \in D^0$, which is equivalent).

Assume that a trajectory has been initially planned (black trajectory in Fig. 3A), in terms of the time series of the control inputs $(a_{\text{plan}}(t)_{t\in[0,T]}, \zeta_{\text{plan}}(t)_{t\in[0,T]})$. Assume now that these control inputs are *corrupted* by random perturbations

$$\forall t \in [0,T] \quad \left\{ \begin{array}{rcl} a(t) & = & a_{\text{plan}}(t) + \xi_1(t) \\ \zeta(t) & = & \zeta_{\text{plan}}(t) + \xi_2(t) \end{array} \right. ,$$

where ξ_1 and ξ_2 two piecewise constant random functions. The red trajectories in Fig. 3A represent several trajectories of the robot corresponding to different realizations of the perturbations ξ_1 and ξ_2. One can notice that the perturbations make the final positions of the red trajectories deviate randomly from the target (denoted by the magenta dot). This can also be noted from the variability profile (red curve in Fig. 3B), which is nonzero at the end of the movement.

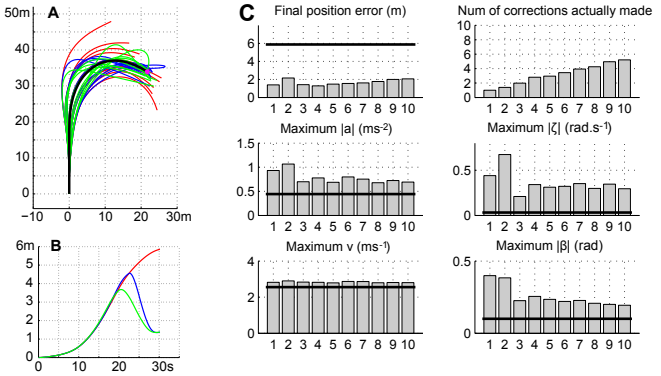

Fig. 3. Feedback control using affine corrections for a car-like robot. **A**: uncorrected sample trajectories (red curves), corrected using at most one correction (blue curves, $S = 1$ in text) or at most five corrections (green, $S = 5$ in text). The initially planned trajectory is in black. **B**: variability profiles computed across 2000 realizations of the random perturbations ξ_1 and ξ_2. **C**: statistics of the feedback control algorithm across 2000 realizations of the random perturbations ξ_1 and ξ_2. The X-axis represents the maximum number of corrections allowed S. The horizontal lines report the values corresponding to the uncorrected trajectories ($S = 0$).

We propose the following feedback control algorithm inspired from [12], [13]. The algorithm maintains at every step two time series $(a_{\text{cur}}(t)_{t\in[0,T]}, \zeta_{\text{cur}}(t)_{t\in[0,T]})$ termed "currently planned control inputs". These time series are initialized at the values of $(a_{\text{plan}}(t)_{t\in[0,T]}, \zeta_{\text{plan}}(t)_{t\in[0,T]})$. The movement time T is divided in $S + 1$ equal parts. At each time instant $t_i = iT/(S + 1)$, $i = 1 \dots S$, the robot is given the possibility to make a correction as follows

1) compute the final position of the robot, had the control inputs $(a_{\text{cur}}(t)_{t\in[t_i,T]}, \zeta_{\text{cur}}(t)_{t\in[t_i,T]})$ been applied starting at the current state $\bar{\mathcal{C}}(t_i)$ and until the end of the movement. Denote this final simulated position $(x_{\text{sim}}, y_{\text{sim}})$;
2) compute appropriate trajectory deformations with $\tau > t_i$ to correct the final position from $(x_{\text{sim}}, y_{\text{sim}})$ towards $(x_{\text{target}}, y_{\text{target}})$. This gives rise to new time series of control inputs, denoted $(a_{\text{new}}(t)_{t\in[t_i,T]}, \zeta_{\text{new}}(t)_{t\in[t_i,T]})$;
3) if the new control inputs are acceptable (i.e. do not imply too large accelerations or too sharp turns), set $a_{\text{cur}}(t)_{t\in[t_i,T]} \leftarrow a_{\text{new}}(t)_{t\in[t_i,T]}$ and $\zeta_{\text{cur}}(t)_{t\in[t_i,T]} \leftarrow \zeta_{\text{new}}(t)_{t\in[t_i,T]}$. Otherwise, keep the current values of a_{cur} and ζ_{cur}.

Figure 3A shows the results of the feedback control algorithm for $S = 1$ (blue curves) and $S = 5$ (green curve). Note that the blue and green curves are driven by the same realizations of the perturbations as the red curves (uncorrected trajectories). However, the blue and green curves end up much closer to the target position. Figure 3B confirms this observation: the final variabilities of the corrected trajectories (blue and green profiles) at T are much lower ($\sim$1.3m) than that of the uncorrected trajectories ($\sim$6m).

One could ask: why make multiple corrections (green) while making one unique correction (blue) yields approximately the same final average error? Figure 3C shows that $S = 1$ is associated with larger values of a, ζ and β than $S = 5$. This is because when the robot is allowed to make multiple corrections, the changes to a and ζ are *distributed* instead of being concentrated in one single large correction near the end of the trajectory. Figure 3B confirms this observation: the green variability profile ($S = 5$) starts decreasing before the blue variability profile ($S = 1$). Note however that choosing $S > 5$ does not improve the algorithm.

Finally, note that this algorithm is not a trajectory-tracking algorithm but rather a simplified implementation of "optimal feedback control" [12], [13].

E. Gap filling for sampling-based kinodynamic planners

Gap reduction techniques are a core component of any sampling-based kinodynamic planner [14]. As an example, consider the approach proposed in [15], which consists of growing two rapidly-exploring random tree (RRT) rooted respectively at the initial state and at the target state – a solution trajectory is obtained when these two trees intersect. When nonholonomic constraints are present, exact intersections of the trees occur with probability zero, such that one usually assumes intersection when the trees are within a nonzero distance of each other, yielding thereby a *gap* in the solution trajectory. As the performance of the planner critically depends on the permitted gap size (the larger the permitted gap size, the quicker the growing trees find an "intersection", but also the more difficult filling the gaps), efficient gap reduction techniques have been shown to dramatically improve the performance of the planner [14].

We now show how affine corrections can be used to fill trajectory gaps. Consider two trajectories $\mathcal{C}_1(t)_{t\in[0,T_1]}$ and $\mathcal{C}_2(t)_{t\in[0,T_2]}$ of a car-like robot (respectively in red and cyan in Fig. 4) separated by a gap. We first "prepare" the two trajectories as follows

1) grow a first stub with time duration Δ_a at the end of $\mathcal{C}_1$. Using the time interval $[T_1, T_1 + \Delta_a]$, bring the steering angle β_1 to 0 by "counter-steering" (i.e.

turning the steering wheel back to the straight-ahead position);

2) grow a second stub with time duration Δ_b at the end of the extended $\mathcal{C}_1(t)$. During this time interval, the steering angle β_1 is kept to 0, resulting in a straight segment. One can easily verify that the (doubly) extended trajectory $\mathcal{C}_1(t)_{t\in[0,T_1+\Delta_a+\Delta_b]}$ is admissible. The two stubs are shown by dashed red lines in Fig. 4;
3) similarly, grow two other stubs at the *beginning* of $\mathcal{C}_2$ (shown by dashed cyan lines in Fig. 4).

After this "preparation", we have two trajectories which respectively ends and begins by straight segments. The lengths of the added stubs depend on the Δ's and can be made relatively short if the β's are small and large braking and counter-steering rates are permitted. We can now use the position and orientation algorithms given in the previous sections to bring the end of the extended $\mathcal{C}_1$ towards the beginning of the extended $\mathcal{C}_2$. Fig. 4 shows an example of such correction using three successive affine deformations (cf. section III-B). The admissibility conditions are verified by observing that

- since affine transformations preserve collinearity [7], the corrected extended trajectory $\mathcal{C}_1'''$ (magenta) also ends by a straight segment. When this straight segment connects with the straight segment at the beginning of the extended $\mathcal{C}_2$, the continuity of β is guaranteed;
- regarding the continuity of v, one can use the straight parts around the connection point to modulate the speed profile to make it continuous *without altering the geometric path*: see the yellow lines in the plots of a and v in Fig. 4.

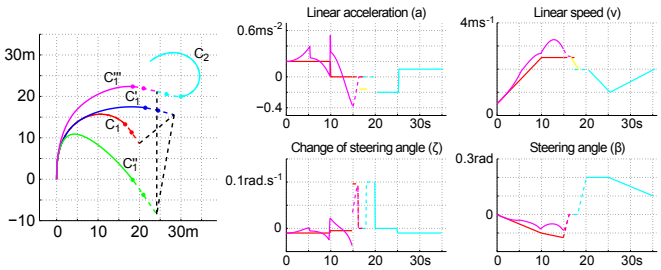

Fig. 4. Filling trajectory gaps for a car-like robot. Left: geometric paths. The original trajectories to be connected are shown in plain red line ($\mathcal{C}_1$) and plain cyan line ($\mathcal{C}_2$). These trajectories are first "prepared" by growing stubs at their extremities (red and cyan dashed lines). The extended $\mathcal{C}_1$ is then corrected into $\mathcal{C}_1'''$ (magenta) by three successive affine deformations (the blue and green lines represent the intermediate trajectories $\mathcal{C}_1'$ and $\mathcal{C}_1''$). Note that $\mathcal{C}_1'''$ smoothly connects with $\mathcal{C}_2$. Right: profiles of the other variables. The yellow lines in the plots of a and v show the modifications that make v continuous without changing the geometric paths.

IV. Conclusion

Because of the nonholonomic constraints induced by the rolling-without-slipping constraint at work at its wheels, a wheeled robot cannot execute every trajectory in the plane. In this article, we have identified the conditions for a trajectory to be an admissible trajectory for a given wheeled robot, based on some generic and usually made assumptions about the rotation and steering velocities of the wheels. For robots of class I (whose *degree of maneuvrability* is 3, e.g. the unicycle), such trajectories are C^1 and piecewise C^2. For robots of class II (whose *degree of maneuvrability* is 2, e.g. the bicycle or car-like robots), such trajectories are C^1, piecewise C^2 *and* with continuous curvature. Based on these results and on the affine trajectory deformation framework, we have presented fast, exact, feedback control and gap filling algorithms for wheeled robots. Our current research focuses on testing these algorithms on actual wheeled robotic platforms, which will probably require the joint integration of planning and control aspects, as in e.g. [16].

Acknowledgments

The authors would like to thank Prof. G. Campion and Mr. S. Ok for their highly valuable suggestions and comments.

References

[1] J.-P. Laumond, *Robot Motion Planning and Control*. New York: Springer-Verlag, 1998.
[2] L. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
[3] J. Boissonnat, A. Cerezo, and J. Leblond, "A note on shortest paths in the plane subject to a constraint on the derivative of the curvature," *Rapports de recherche-INRIA*, 1994.
[4] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.
[5] G. Campion, G. Bastin, and B. D'Andrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, 1996.
[6] B. d'Andrea Novel, G. Campion, and G. Bastin, "Control of nonholonomic wheeled mobile robots by state feedback linearization," *The International journal of robotics research*, vol. 14, no. 6, pp. 543–559, 1995.
[7] Q.-C. Pham, "Fast trajectory correction for nonholonomic mobile robots using affine transformations," in *Robotics: Science and Systems*, 2011.
[8] F. Lamiraux, D. Bonnafous, and O. Lefebvre, "Reactive path deformation for nonholonomic mobile robots," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 967–977, 2004.
[9] K. Seiler, S. Singh, and H. Durrant-Whyte, "Using Lie group symmetries for fast corrective motion planning," in *Algorithmic Foundations of Robotics IX*, 2010.
[10] D. Bennequin, R. Fuchs, A. Berthoz, and T. Flash, "Movement timing and invariance arise from several geometries," *PLoS Comput Biol*, vol. 5, p. e1000426, Jul 2009.
[11] M. Hillion and F. Lamiraux, "Taking into account velocity and acceleration bounds in nonholonomic trajectory deformation," in *IEEE International Conference on Robotics and Automation*, pp. 3080–3085, IEEE, 2007.
[12] E. Todorov and M. Jordan, "Optimal feedback control as a theory of motor coordination.," *Nat Neurosci*, vol. 5, pp. 1226–35, Nov. 2002.
[13] Q.-C. Pham and H. Hicheur, "On the open-loop and feedback processes that underlie the formation of trajectories during visual and nonvisual locomotion in humans.," *J Neurophysiol*, vol. 102, no. 5, pp. 2800–2815, 2009.
[14] P. Cheng, E. Frazzoli, and S. LaValle, "Improving the performance of sampling-based motion planning with symmetry-based gap reduction," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 488–494, 2008.
[15] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.
[16] K. Pathak and S. K. Agrawal, "An integrated path-planning and control approach for nonholonomic unicycles using switched local potentials," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1201–1208, 2005.