# Control of Mobile Platforms Using a Virtual Vehicle Approach

M. Egerstedt, X. Hu, and A. Stotsky

***Abstract*—Two model independent solutions to the problem of controlling wheel-based mobile platforms are proposed. These two algorithms are based on a so called virtual vehicle approach, where the motion of the reference point on the desired trajectory is governed by a differential equation containing error feedback. This, combined with the fact that the proven stable control algorithms are basically proportional regulators with arbitrary positive gains, make the solutions robust with respect to errors and disturbances, as demonstrated by the experimental results.**

***Index Terms*—Control of mobile platforms, nonlinear feedback control, trajectory tracking.**

## I. Introduction

In this note, the problem of controlling wheel-based mobile platforms is studied. Many industrial applications need problems like this to be solved in order to have good and robust path tracking algorithms for different types of automated tasks. Naturally, this has been a well studied topic [3], [4], [6]–[8], [10], [12]–[14], and quite a few methods have been proposed to solve the problem, for example the curvature steering method [5], [7], or the flatness approach [8]. However, many of these methods either use open loop control, which is quite sensitive to measurement errors and disturbances, or are highly model dependent, making the controllers very complex and hard to implement in practice, since exact modeling of the platform is typically not an easy task.

Our solution to this problem consists of two slightly different control strategies for tracking a reference trajectory based on position and orientation error feedback. These strategies are largely model independent because they provide only the rotational and translational velocity control. In other words, they are higher level controls. Naturally, for platforms that do not have direct control over the velocities, one needs to design the actuator control so that these velocity controls are realized. The implementation could be just a static mapping, as in the car case, or a dynamic regulator. The first of the two strategies is designed in such a way that it only requires control in the lateral direction, i.e., rotational control, while keeping the longitudinal velocity at a constant value. This strategy is developed for mobile platforms that do not support fine and accurate translational velocity control, such as fairly cheap RC cars. As a price one has to pay for using only one controlled input, the algorithm only works locally. Therefore the second of the two strategies requires both rotational and translational control, and will be shown to be globally stable.

The two proposed algorithms are furthermore both based on a so-called *virtual vehicle* approach, where the motion of the reference point (the virtual vehicle) on the planned trajectory is governed by a differential equation containing error feedback. It can be viewed as a combination of the conventional trajectory tracking, where the reference trajectory is parameterized in time, and a dynamic path following approach [14], where the criterion is to stay close to the geometric path, but not necessarily close to an a priori specified point at a given time. The main idea behind our approach can be seen in Fig. 1, and the reason for calling the reference point, together with the associated differential equation, a virtual vehicle is that the reference point is moving on the path that we want the platform to follow. At the same time it has its own dynamics for describing the motion, and one of the advantages with our approach is that it is quite robust with respect to measurement errors and external disturbances. This is due to the fact that the motion of the virtual vehicle is governed by tracking error feedback, and basically only proportional controls are used. If both the tracking errors and disturbances are within certain bounds, the reference point moves along the reference trajectory while the robot follows it within the prespecified look-ahead distance. Otherwise, the reference point slows down and waits for the robot.

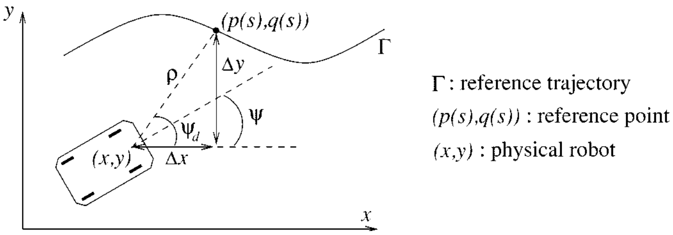


Fig. 1. The general idea behind the virtual vehicle approach.

In this note, we study the performance of the two proposed path following algorithms analytically and show how they are implemented on two different platforms respectively: a RC car and a Nomadic 200 mobile platform. In order to compare the two algorithms, the stability analysis will be performed on kinematic models of the two platforms, which are in essence the same.

This note is organized as follows. In Section II, we present our two control algorithms, followed by the stability analysis in Section III. In Section IV, the first controller is implemented on a small RC car, and the second algorithm is implemented on the Nomad 200 in order to stress the fact that our proposed solutions really are model independent. These experiments also show that our proposed solutions do not only work in theory, but in practice as well.

## II. Control Algorithms

### A. Problem Formulation

Our main task is to find a lateral control $\delta_f(t)$ and longitudinal control $v(t)$ that make a robot follow a smooth reference path, parameterized by a virtual vehicle $s(t)$, moving on the path. The path is given by

$$\begin{aligned} x_d &= p(s) \\ y_d &= q(s), \quad (0 \le s \le s_f) \end{aligned} \tag{1}$$

where we assume that ${p'}^2(s) + {q'}^2(s) \neq 0 \; \forall s \in [0, s_f]$, and the subscript $d$ stands for desired position. Furthermore, our control objectives are

$$\limsup_{t \to \infty} \rho(t) \le d_\rho \tag{2}$$

$$\limsup_{t \to \infty} |\psi - \psi_d| \le d_\psi \tag{3}$$

where $\rho(t) = \sqrt{\Delta x^2 + \Delta y^2}$, and $\Delta x = x_d - x, \Delta y = y_d - y$. In (3) is the yaw angle (orientation of the vehicle), $\psi_d = \text{atan2}(\Delta y, \Delta x)$ is the desired orientation, and $(x, y)$ is a reference point on the robot, for

Manuscript received November 20, 1998; revised August 23, 1999 and August 14, 2000. Recommended by Associate Editor G. Bastin. This work was sponsored in part by the Swedish Foundation for Strategic Research through its Centre for Autonomous Systems at KTH, and in part by the Swedish Research council for Engineering Sciences.

The authors are with the Optimization and Systems Theory, Royal Institute of Technology, SE-100 44 Stockholm, Sweden (e-mail: magnuse@math.kth.se; hu@math.kth.se; stotsky@math.kth.se).

Publisher Item Identifier S 0018-9286(01)10351-X.

example the center of gravity. Furthermore, $d_\psi > 0$ is a small number that, among other things, depends on the maximum curvature of the reference path, and $d_\rho$ is the look-ahead distance that will be more carefully defined later.

### B. Control Algorithm 1

In this section, we present the first of the two control algorithms. We assume that the longitudinal velocity $v = \sqrt{\dot{x}^2 + \dot{y}^2}$ is constant, and we thus only control the lateral velocity. This control algorithm is designed for platforms which do not have good actuators, such as our fairly cheap radio-controlled car. Therefore, our purpose here is to keep the longitudinal and lateral control as simple as possible. In order to realize (2), we introduce the parameter $\gamma$, and require [11]

$$\dot{\rho} - \dot{d}_\rho = -\gamma(\rho - d_\rho) \tag{4}$$

which, after differentiating $\rho$, gives

$$\frac{1}{\rho}(\Delta x(\dot{x}_d - \dot{x}) + \Delta y(\dot{y}_d - \dot{y})) = -\gamma(\rho - d_\rho) + \dot{d}_\rho. \tag{5}$$

Now, since $\dot{x}_d = p'(s)\dot{s}, \dot{y}_d = q'(s)\dot{s}$, we have

$$\dot{s} = (\Delta x p'(s) + \Delta y q'(s))^{-1} \times (\Delta x \dot{x} + \Delta y \dot{y} - \gamma\rho(\rho - d_\rho) + \rho\dot{d}_\rho). \tag{6}$$

If we let $\psi_v$ denote the angle between the velocity vector and the $x$-axis (for most practical systems, $\psi_v$ is almost identical to $\psi$), then

$$\Delta x \dot{x} + \Delta y \dot{y} - \gamma\rho(\rho - d_\rho) + \rho\dot{d}_\rho = \rho(v\cos(\psi_d - \psi_v) - \gamma(\rho - d_\rho) + \dot{d}_\rho).$$

Now, let $\gamma$ in (4) be given by $\gamma = \alpha v \cos(\psi_d - \psi)$, with $\alpha > 0$, and $d_\rho = 1/\alpha$, and let $\theta_r$ denote the orientation angle of the tangent to the reference curve at $s$. Then (6) can be rewritten as

$$\dot{s} = \frac{1}{\sqrt{p'^2 + q'^2}\cos(\psi_d - \theta_r)}(2v\cos(\psi_d - \psi_v) - \gamma\rho). \tag{7}$$

It can be seen from (4) that $\rho(t) \to d_\rho$ if $\cos(\psi_d - \psi_v)$ is kept positive and $\cos(\psi_d - \theta_r)$ stays nonzero, and thus the first control objective (2) would be realized if this were to hold. However, $\cos(\psi_d - \theta_r)$ will be zero only if $(\Delta x, \Delta y)^T$ is normal to the curve at $(p(s), q(s))$. In order to avoid this singularity, we need to make the following assumption about the reference curve.

*Assumption 2.1:* The bound on the curvatures of the reference curve is sufficiently small.

A typical example here is a straight line. However, this assumption alone may not be enough. In fact, the lateral control has to be designed such that the following holds.

*Assumption 2.2:*

$$2v\cos(\psi_d - \psi_v) - \gamma\rho \geq \epsilon > 0$$

for all $t \geq 0$.

*Remark:* It is easy to see that if the curvature of the reference curve is sufficiently small, then if initially $\Delta x p'(s) + \Delta y q'(s) > 0$, it will stay positive, provided that Assumption 2.2 is also satisfied (think of a straight line).

The satisfaction of Assumption 2.2 depends both on the initial condition and on the lateral control. In fact, it is easy to see that if Assumption 2.2 is satisfied initially, and if, for some $\delta > 0$, the lateral control keeps $|\psi_d(t) - \psi_v(t)| \leq \pi/2 - \delta$ for all $t \geq 0$, then Assumption 2.2 is always satisfied. This also implies that the second control objective (3) would be fulfilled.

For a car-like robot, we propose the following proportional, lateral control:

$$\delta_f = -k(\psi - \psi_d), \quad k > 0 \tag{8}$$

and we summarize the first control algorithm as follows.

*Algorithm 2.1:*

$$\begin{cases} \dot{s} = \frac{1}{\sqrt{p'^2+q'^2}\cos(\psi_d - \theta_r)}(2v\cos(\psi_d - \psi_v) - \gamma\rho)) \\ \delta_f = -k(\psi - \psi_d) \\ v = \text{const}, \end{cases} \tag{9}$$

where $\Delta_f$ and $v$ control the lateral and longitudinal velocities respectively.

We thus propose a simple control algorithm for steering mobile platforms, such as car-like robots, that is given by (7) and (8). Later we will use a kinematic model of the car to show that Assumption 2.2 is met by using the lateral control (8), provided that the reference curve satisfies Assumption 2.1.

Since Assumptions 2.1 and 2.2 can be satisfied initially only for a subset of the initial conditions, this algorithm is not valid globally. On the other hand, if one is able to use finer longitudinal control, such singularities can be avoided, as will be shown in the next subsection where the second of our control algorithms is presented.

### C. Control Algorithm 2

For this algorithm we assume that the actuators are accurate enough so that fine longitudinal velocity control is available. In contrast to the previous algorithm, this new algorithm will furthermore be defined globally. We should first point out that in [6] and [12], tracking controls are designed so that the tracking errors tend to zero globally. However, the controls there are complex and highly model dependent, while our aim is to produce controls that are quite intuitive and model independent.

From (1), we, as already mentioned, have $\dot{x}_d = p'(s)\dot{s}, \dot{y}_d = q'(s)\dot{s}$, or $\dot{x}_d p'(s) + \dot{y}_d q'(s) = (p'^2(s) + q'^2(s))\dot{s}$, which implies that if the robot would track the path perfectly, i.e., $\dot{x} = \dot{x}_d$ and $\dot{y} = \dot{y}_d$, we would have

$$\dot{s} = \frac{p'(s)}{p'^2(s) + q'^2(s)}\dot{x} + \frac{q'(s)}{p'^2(s) + q'^2(s)}\dot{y}. \tag{10}$$

The reason for expressing $\dot{s}$ in this way is that $p'^2(s) + q'^2(s)$ is always nonzero by our assumption about the smoothness of the reference path.

If we now denote $v = \sqrt{\dot{x}^2 + \dot{y}^2}$, and assume that $\dot{s} \geq 0$, then

$$\dot{s} = v/\sqrt{p'^2(s) + q'^2(s)}.$$

On the other hand, these expressions do not contain any position error feedback, which is important for robustness. We thus add error feedback to $\dot{s}$, and propose the dynamics for the reference point as follows:

$$\dot{s} = \frac{ce^{-\alpha\rho}v_0}{\sqrt{p'^2(s) + q'^2(s)}} \tag{11}$$

where $\alpha$ and $c$ are positive numbers that are to be determined later, and with the appropriate choice of $\alpha$ and $c$, $v_0$ will be the desired speed at which one wants the vehicle to track the path.

Since $\psi_d$ becomes undefined when $\rho = 0$, in order to avoid this singularity, we define

$$\tilde{\psi}_d = \begin{cases} \psi_d & \text{if } \rho > \epsilon \\ \frac{\psi_d(-2\rho^3+3\epsilon\rho^2)+\theta_r(-2(\epsilon-\rho)^3+3\epsilon(\epsilon-\rho)^2)}{\epsilon^3} & \text{if } \rho \leq \epsilon \end{cases}$$

where $\epsilon$ can be taken as $d_\rho$, i.e., the desired tracking bound. It is easy to see that $\tilde{\psi}_d$ is also well defined at $\rho = 0$ since $\lim_{\rho\to 0} \psi_d(-2\rho^3 + 3\epsilon\rho^2) = 0$.

Consequently, we can define the translational and rotational control as follows.

*Algorithm 2.2:*

$$\begin{cases} \dot{s} = \frac{ce^{-\alpha\rho}v_0}{\sqrt{p'^2(s)+q'^2(s)}} \\ \delta_f = k\Delta\tilde{\psi} + \dot{\tilde{\psi}}_d \\ v = \gamma(\Delta x\cos(\psi) + \Delta y\sin(\psi)), \end{cases} \tag{12}$$

where both $\gamma$ and $k$ are positive, and $\Delta\tilde{\psi} = \tilde{\psi}_d - \psi$.

*Remark:* At $\rho = \epsilon$ it holds that

$$\tilde{\psi}_d|_{\rho=\epsilon} = \psi_d \quad \text{and} \quad \dot{\tilde{\psi}}_d|_{\rho=\epsilon} = \dot{\psi}_d$$

and therefore the rotational control is continuous everywhere. Obviously, the translational control is also defined and continuous everywhere. If $\rho \neq 0$, then the translational control can be also expressed as

$$v = \gamma\rho\cos(\Delta\psi)$$

where $\Delta\psi = \psi_d - \psi$.

It is obvious that Algorithm 2.2 just steers the robot toward the reference point on the desired path, with a speed proportional to the distance tracking error.[1]

## III. Stability Analysis

Even though our control algorithms are largely model independent, the stability analysis still has to be done with respect to mathematical models of the platforms. We apply the first control algorithm to a RC car platform, where fine control of the velocity is extremely difficult to achieve. We then apply our second algorithm to a Nomad platform, where one can apply fine control to both the longitudinal and the lateral velocities.

### A. Stability Analysis of a Car-Like Platform

Consider a simple kinematic model for a car-like platform, where we let $(x, y)$ denote the middle point of the rear axle, and let $\delta$ denote the steering angle. Then, it is well known (see [13], for example) that the model can be written as

$$\begin{aligned} \dot{x} &= v\cos(\psi) \\ \dot{y} &= v\sin(\psi) \\ \dot{\psi} &= \frac{v}{l}\tan(\delta) \end{aligned} \tag{13}$$

where $l$ is the length between the front and the rear axles. In this model, $\psi_v = \psi$.

If we plug in the lateral control law (8), we have

$$\frac{v}{l}\tan(\delta) = -k(\psi - \psi_d).$$

So, $\delta = -\tan^{-1}((lk/v)(\psi - \psi_d))$, where $k$ should be chosen to reflect the steering range accessible by the robot.

Now, we define the errors

$$\Delta\rho = \rho - d_\rho, \quad \Delta\psi = \psi - \psi_d, \quad \Delta_d = \psi_d - \theta_r$$

which gives us the following error dynamics:

$$\begin{aligned} \dot{\Delta}\rho &= -\alpha\cos(\Delta\psi)\Delta\rho \\ \dot{\Delta}\psi &= -k\Delta\psi + \frac{v}{\Delta\rho + d_\rho}(\sin(\Delta\psi) \\ &\quad + \tan(\Delta_d)\cos(\Delta\psi)(1 - \alpha\Delta\rho)) \\ \dot{\Delta}_d &= -\frac{v}{\Delta\rho + d_\rho}(\sin(\Delta\psi) \\ &\quad + \tan(\Delta_d)\cos(\Delta\psi)(1 - \alpha\Delta\rho)) + \kappa(s)w \end{aligned} \tag{14}$$

where $\kappa(s) = (q''p' - p''q'/(p'^2 + q'^2))^{3/2}$ is the curvature of the path, and $w = (v/\cos(\Delta_d))\cos(\Delta\psi)(1 - \alpha\Delta\rho)$. It is obvious that the error dynamics (14) is not globally stable, since $|\Delta_d| = (\pi/2)$ or $\rho = 0$ (where $\psi_d$ is not well defined) are singular points.

However, if we consider $w$ as an input or perturbation to the system defined in (14), then we can show that the dynamics is locally exponentially stable if the input is set to zero. It is easy to compute the linearized matrix $A$ of (14) at $(0, 0, 0)$ as

$$A = \begin{pmatrix} -\alpha & 0 & 0 \\ 0 & -k + \frac{v}{d_\rho} & \frac{v}{d_\rho} \\ 0 & -\frac{v}{d_\rho} & -\frac{v}{d_\rho} \end{pmatrix}$$

and straightforward calculations show that $A$ is a stable matrix for all $v > 0$ and $k > 0$. Therefore, (14) is a locally exponentially stable system driven by the input $\kappa w$. It is well known that for such a system, if both the initial condition $(\Delta\rho(0), \Delta\psi(0), \Delta_d(0))$ and $|\kappa w|$ are small enough, then $(\Delta\rho(t), \Delta\psi(t), \Delta_d(t))$ will remain small. This also implies that the singular points will not be reached. In order to keep $|\kappa w|$ small, it suffices to assume that the upper bound for $\kappa(s)$ is small enough.

*Remark:* It is easy to see that with this model, Assumption 2.2 is satisfied. We note that $2v\cos(\psi_d - \psi_v) - \gamma\rho = v\cos(\Delta\psi)(2 - (\rho/d_\rho))$, and $\rho(t)$ tends to $d_\rho$ monotonically if $|\Delta\psi|$ is bounded from $\pi/2$. Therefore, if initially, $|\Delta\psi(0)|$ is small enough and $(2 - (\rho(0)/d_\rho)) > 0$, then Assumption 2.2 is satisfied.

### B. Stability Analysis of an Omni-Directional Platform

*1) Vehicle Model:* The Nomad 200, depicted in Fig. 3, is a synchro-drive tri-wheeler, where the two slave wheels are mechanically turned in the same direction as the actively controlled master wheel. If we do not want to model side slips, due to the intended low speed application that the second control algorithm was designed for[2], or the mechanical delays between the wheels, we get the model

$$\begin{aligned} \dot{x} &= v\cos(\psi) \\ \dot{y} &= v\sin(\psi) \\ \dot{\psi} &= \delta_f. \end{aligned} \tag{15}$$

[1] In practice, the range of the accessible speed is always limited. One can replace the speed control $v = \gamma\rho\cos(\Delta\psi)$ by, for example, $v = \gamma\text{atan}(\rho)\cos(\Delta\psi)$ in order to take care of this saturation problem.

[2] The algorithm was designed with an intelligent service agent application in mind [2], where low speeds are typically enough.

It should be pointed out that for this model, the lateral control, $\delta_f$, is already the angular velocity of the vehicle and we assume the actuators do not have any saturation bound.

*2) Stability Analysis:* Let us consider the error dynamics:

$$\begin{aligned}\dot{\Delta} x &= p'(s)\dot{s} - \gamma(\Delta x\cos(\psi) + \Delta y\sin(\psi))\cos(\psi)\\ \dot{\Delta} y &= q'(s)\dot{s} - \gamma(\Delta x\cos(\psi) + \Delta y\sin(\psi))\sin(\psi)\\ \dot{\Delta}\tilde{\psi} &= -k\Delta\tilde{\psi}\end{aligned} \tag{16}$$

where $\dot{s}$ is defined in (11). Since this control algorithm is globally defined, in contrast to Algorithm 2.1, we do not need to include $\Delta_d$ in the error dynamics.

Suppose that the desired speed of the robot, $v_0$, is greater than zero, and that $s_f = \infty$. (In practice, this means that the desired path should be long enough).

From (16), it directly follows that:

$$\Delta\tilde{\psi} = \Delta\tilde{\psi}(0)e^{-kt}. \tag{17}$$

Furthermore, since $\rho = \sqrt{\Delta x^2 + \Delta y^2}$, then if it holds that $\rho \neq 0$

$$\begin{aligned}\dot{\rho} &= \frac{1}{\rho}(\Delta x\dot{\Delta} x + \Delta y\dot{\Delta} y)\\ &= \frac{1}{\rho}\left[-\gamma(\Delta x\cos(\psi) + \Delta y\sin(\psi))^2 + \frac{\Delta x p' + \Delta y q'}{\sqrt{p'^2 + q'^2}}\dot{s}\right] \qquad (18)\\ &= -\gamma\cos^2(\Delta\psi)\rho + ce^{-\alpha\rho}v_0\cos(\psi_d - \theta_r). \qquad (19)\end{aligned}$$

Now, suppose that at $t = \bar{t}$, $\rho(\bar{t}) = 0$. Then using L'Hospital's rule we have

$$\lim_{t\to\bar{t}}\frac{\Delta x}{\sqrt{\Delta x^2 + \Delta y^2}} = \lim_{t\to\bar{t}}\frac{1}{\sqrt{1 + (\frac{\Delta y}{\Delta x})^2}} = \frac{p'(s(\bar{t}))}{\sqrt{p'^2 + q'^2}}.$$

Similarly, we have

$$\lim_{t\to\bar{t}}\frac{\Delta y}{\sqrt{\Delta x^2 + \Delta y^2}} = \frac{q'(s(\bar{t}))}{\sqrt{p'^2 + q'^2}}.$$

*Remark:* This also implies that even though $\psi_d$ is not defined at $\rho(\bar{t}) = 0, \lim_{t\to\bar{t}}\psi_d$ exists and is equal to $\theta_r$.

Since $\dot{s}(\bar{t}) = (cv_0/\sqrt{p'^2 + q'^2})$, from (18) we have

$$\dot{\rho}|_{\rho(t)\to 0} = cv_0.$$

On the other hand, at $t = \bar{t}$, since $v(\bar{t}) = 0$, we have

$$\frac{\Delta\rho}{\Delta t} = \sqrt{\frac{(p(s(\bar{t})) - p(s(\bar{t} + \Delta t)))^2 + (q(s(\bar{t})) - q(s(\bar{t} + \Delta t)))^2}{\Delta t^2}}.$$

So, by the chain rule we have

$$\dot{\rho}|_{\rho=0} = cv_0$$

and thus (18) is well defined. Since $cv_0 > 0$, this implies that $\rho = 0$ can happen only as an initial condition (or perturbed to there by disturbances, for example).

Now, let us denote

$$a(t) = -\gamma\cos^2(\Delta\psi(t)).$$

We should notice that $\Delta\psi(t) = \Delta\tilde{\psi}(t)$ only if $\rho(t) \geq \epsilon$. Now also let $\Phi(t, s)$ denote

$$\exp\left(\int_s^t a(\tau)\,d\tau\right).$$

We can always formally express

$$\rho(t) = \Phi(t, 0)\rho(0) + \int_0^t \Phi(t, s)cv_0\cos(\psi_d(s) - \theta_r(s))\,ds.$$

Due to the facts that $\lim_{\rho(t)\to 0}\psi_d$ exists and $\psi_d$ can be zero only at $t = 0$, the above expression is valid for all initial conditions and $0 \leq t < T$, where $T$ is either $\infty$ or a possible finite escape time. If $\rho(0) > \epsilon$, then

$$\begin{aligned}|\Phi(t, s)| &= \exp\left(\int_s^t a(\tau)d\tau\right)\\ &= \exp\left(-\int_s^t \gamma[1 - \sin^2(\Delta\tilde{\psi})]d\tau\right)\\ &\leq \exp\left(-\gamma\int_s^t (1 - \Delta\tilde{\psi}^2)d\tau\right)\\ &= \exp\left(-\gamma(t - s) - \gamma\frac{\Delta\psi^2(0)}{2k}[e^{-2ks} - e^{-2kt}]\right)\\ &\leq e^{\gamma(\Delta\psi^2(0)/k - (t-s))}, \quad \forall T_\epsilon \geq t \geq s \geq 0.\end{aligned} \tag{20}$$

where $T_\epsilon$ is such that $\rho(T_\epsilon) = \epsilon$ and

$$\begin{aligned}|\Phi(t, 0)|\rho(0) &+ \int_0^t |\Phi(t, s)|cv_0\cos(\psi_d - \theta_r)\,ds\\ &\leq e^{\gamma(\Delta\psi^2(0)/k - t)}\rho(0) + \frac{v_0}{\gamma}ce^{\gamma\Delta\psi^2(0)/k}.\end{aligned}$$

Thus

$$\rho(t) \leq \max\left\{\epsilon, e^{\gamma(\Delta\psi^2(0)/k - t)}\rho(0) + \frac{v_0}{\gamma}ce^{\gamma\Delta\psi^2(0)/k}\right\} \quad \forall t \geq 0.$$

It is easy to see that after the transient decays exponentially, the tracking error, $\rho$, can meet any *a priori* given bound by tuning $\epsilon$, $\gamma$ and $k$. Now, if $\rho(t)$ is small enough then $\psi_d \approx \theta_r$ (since $\lim_{\rho\to 0}\psi_d = \theta_r$) thus $\Delta\psi \approx \Delta\tilde{\psi} \to 0$. Then, inspecting (19) gives us that $\rho(t)$ should be close to the steady-state solution of

$$\dot{\rho} = -\gamma\rho + ce^{-\alpha\rho}v_0$$

which is defined by

$$\gamma\rho = ce^{-\alpha\rho}v_0$$

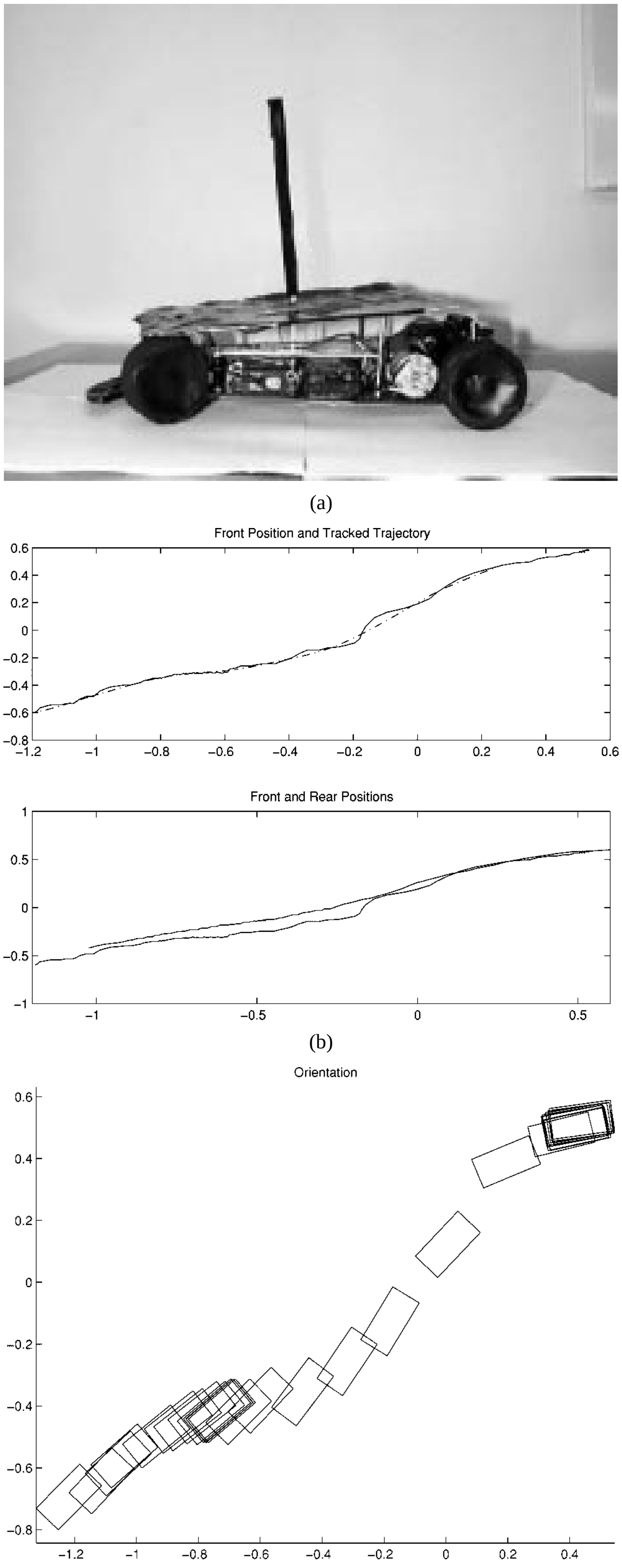
where the choice of $c = e^{\alpha v_0/\gamma}$ gives that $\gamma\rho = v_0$. Thus, one can interpret $v_0$ as the desired speed, since, by our design, the actual speed is given by $v = \gamma\rho\cos(\Delta\psi)$.

## IV. Implementation

### A. Implementation of the First Control Algorithm

We chose to implement the first control algorithm on a small, radio controlled car, where we simply connect the transmitter to a computer.

However, our car system is based on a fairly cheap toy car with coarse A/D and D/A conversions as well as a dead zone in the servo system. Therefore the steering is far from precise, making questions concerning robustness very important, so what works in simulation may not work at all here. However, as seen in Fig. 2, the experiments



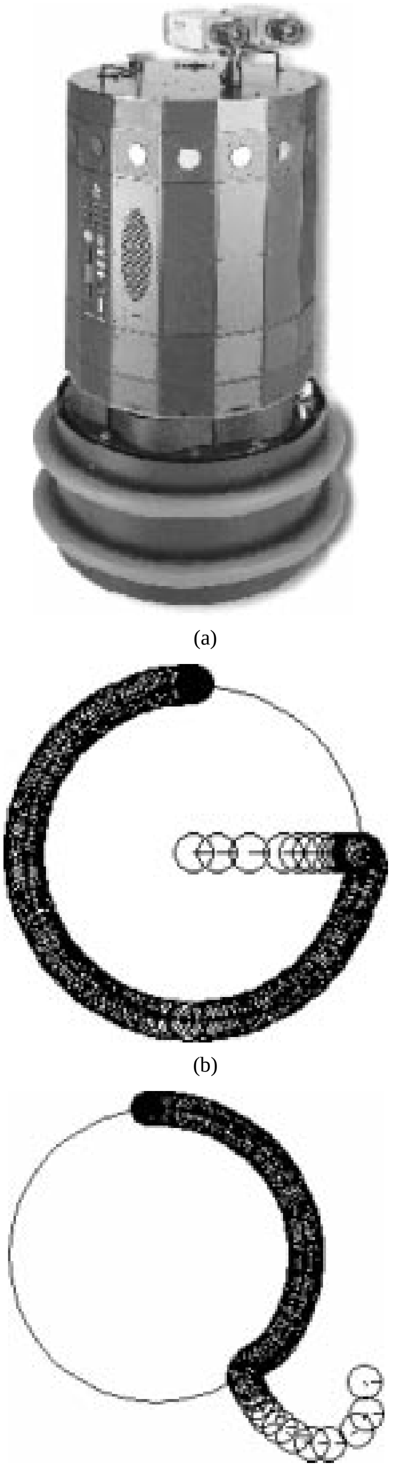
(a)

(b)

(c)

Fig. 2. In (a), the radio controlled car used for trying out the first control algorithm is depicted. In (b), the tracked trajectory (dotted) and the front point (solid) on the actual car can be seen, as well as the front and the rear points plotted together. In (c), the orientation of the car can be seen.

turned out to be satisfying, supporting the practical feasibility of our first control algorithm.



(a)

(b)

(c)

Fig. 3. The Nomad 200 [(a)] together with test runs with two different initial positions [(b) and (c)] are depicted in order to stress the global convergence of the second control algorithm. In this example, a circular path was tracked.

### B. Implementation of the Second Algorithm on a Nomad 200 Mobile Platform

In order to stress the fact that our proposed control algorithms really are model independent, we chose to implement the second algorithm on a Nomad 200 mobile platform. The dynamics of this platform differ quite a lot from nonholonomic RC cars, but the approach still works well, which can be seen in Fig. 3, where we have plotted different test runs on the `Nserver`, the Nomad simulator. These test runs, where the robot tracks a circle from different initial positions and configurations, clearly indicate that our proposed second control algorithm works globally in a stable and robust way. It should be emphasized that if we were to use the first, local algorithm on these test runs, it would fail since the initial positions and orientations would make $\Delta x p'(s) + \Delta y q'(s) = 0$, and thus $\dot{s}$ would not be defined anymore if the first algorithm were to be used.

## V. Conclusion

In this note, two intuitive, model independent path following control strategies are proposed, and the stability analysis is done with respect to two different platforms. What is new here is that by combining the conventional trajectory tracking approach and the more recent geometric path following approach, we can design a virtual vehicle that moves on the reference path and is regulated in a closed-loop fashion by exploiting the position error. In the first algorithm, the velocity is kept constant, while the other, global method depends on the possibility of fine velocity control.

Implementing these ideas on actual robots gives us some experimental data that show that our controllers work in practice as well as in theory, which is what we were aiming for, since our main design strategy was to keep the control algorithms model independent and as simple as possible.

## References

[1] J. Ackermann, *Robust Control*. London, U.K.: Springer-Verlag, 1993.

[2] M. Andersson, A. Orebäck, M. Lindström, and H. I. Christensen, *Intelligent Sensor Based Robotics*. Heidelberg, Germany: Springer-Verlag, 1999. Ch. ISR: An Intelligent Service Robot, Lecture Notes in Artificial Intelligence.

[3] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory*, R. W. Brockett, R. W. Millmann, and R. W. Sussman, Eds. Boston, MA: Birkhauser, 1983, pp. 181–191.

[4] G. Campion, G. Bastin, and B. D'Andréa-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 47–62, Feb. 1996.

[5] C. Canudas de Wit, A. D. NDoudi-Likoho, and A. Micaelli, "Feedback control for a train-like vehicle," *Proc. 1994 IEEE Int. Conf. Robotics Automation*, 1994.

[6] C. Canudas de Wit, B. Siciliano, and G. Bastin, *Theory of Robot Control*. New York: Springer-Verlag, 1996.

[7] C. Canudas de Wit, "Trends in mobile robot and vehicle control," in *Control Problems in Robotics*, B. Siciliano and K. P. Valavanis, Eds. London, U.K.: Springer-Verlag, 1998, pp. 151–176. Lecture Notes in Control and Information Sciences 230.

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

[9] E. Freund and R. Mayr, "Nonlinear path control in automated vehicle guidance," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 49–60 , Feb. 1997.

[10] J. Guldner and V. Utkin, "Stabilization of nonholonomic mobile robots using Lyapunov functions for navigation and sliding mode control," in *Proc. 33rd Conf. Decision Control*, FL, Dec. 1994, pp. 2967–2972.

[11] S. V. Gusev and I. A. Makarov, "Stabilization of program motion of transport robot with track laying chassis," presented at the Proceedings of LSU, vol. 1, 1989.

[12] Z. Jiang and H. Nijmeijer, "Tracking control of mobile robots: A case study in backstepping," *Automatica*, vol. 33, no. 7, pp. 1393–1399.

[13] R. Murray and S. Sastry, "Nonholonomic motion planning: Steering using sinusoids," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 700–716, May 1993.

[14] N. Sarkar, X. Yun, and V. Kumar, "Dynamic path following: A new control algorithm for mobile robots," presented at the Proc. 32nd Conference Decision Control, San Antonio, TX, Dec. 1993.

[15] A. Stotsky, X. Hu, and M. Egerstedt, "Sliding mode control of a car-like mobile robot using single-track dynamic model," presented at the Proc. IFAC'99:14th World Congress, Beijing, China, July 1999.

# A Polynomial Approach to Nonlinear System Controllability

Yufan Zheng, Jan C. Willems, and Cishen Zhang

***Abstract*—This note uses a polynomial approach to present a necessary and sufficient condition for local controllability of single-input–single-output (SISO) nonlinear systems. The condition is presented in terms of common factors of a noncommutative polynomial expression. This result exposes controllability properties of a nonlinear system in the input–output framework, and gives a computable procedure for examining nonlinear system controllability using computer algebra.**

***Index Terms*—Common factor, controllability, differential fields, noncommutative ring, nonlinear systems.**

## I. Introduction

Controllability is one of the central notions of modern control theory. The results on controllability of linear systems have been seminal in the development of the field, and the literature on controllability of nonlinear systems is vast. See, for example, [16], [1], [7], [8], [11], [15], and [19]. Traditionally, controllability is defined for linear state space systems and refers to the possibility of transferring a system from any initial to any terminal state. For nonlinear state-space systems the notion of controllability or strong accessibility refers to the case where the control can act on the system state, but may be insufficient to transfer it to a specified terminal state. Often, nonlinear system controllability is defined in terms of system state equation and tested by means of Lie distributions or their dual form.

The notion of controllability is recently extended to systems in more general framework. For linear systems, controllability is viewed in [18] in terms of system trajectories which may not necessarily be the system state. A system is defined to be controllable if one can switch from any feasible past trajectory in the system behavior to any feasible future trajectory, after some time delay. It is observed that the lack of behavioral controllability implies the existence of an autonomous system 'output', which is a nontrivial function of the system variables. It turns out that a linear time-invariant input–output system is controllable if and only if it does not have autonomous variables in its behavior and if and only if the polynomial matrices that specify the system behavior are left coprime.

The notion of autonomous variables is also used to describe controllability of nonlinear systems [1], [5], [7], [19]. In [7], local controllability of nonlinear state space systems is described in terms of the absence of local first integrals which are autonomous variables of the system state. In [1] and [19], controllability of nonlinear state-space systems is described by the absence of autonomous variables in terms of differential one-forms. Moreover, the need for a controllability concept for nonlinear input/output systems is discussed in [13], where a

Manuscript received April 20, 2000; revised February 1, 2001. Recommended by Associate Editor H. Huijberts. This work was supported by the Australia Research Council and the Natural Science Foundation of China.

Y. Zheng is with the Department of Electrical and Electronic Engineering, the University of Melbourne, Parkville, Vic. 3010, Australia, and also with the Institute of Systems Science, East China Normal University, Shanghai 200062, China (e-mail: y.zheng@ee.mu.oz.au).

J. C. Willems is with the Institute of Mathematics and Computing Science, University of Groningen, 9700 AV Groningen, The Netherlands (e-mail: J.C.Willems@math.rug.nl).

C. Zhang is with the Department of Electrical and Electronic Engineering, the University of Melbourne, Parkville, Vic. 3010, Australia (e-mail: c.zhang@ee.mu.oz.au).

Publisher Item Identifier S 0018-9286(01)10352-1.