The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

A Minimalist Feedback Control for Path Tracking in Cartesian Space

Antonio Sgorbissa and Renato Zaccaria

Abstract— **The article proposes a new feedback control model that allows to track a generic curve in the Cartesian Space expressed through its implicit equation, and has minimal requirements in terms of measurement and computation capabilities. The model measures only the distance between the vehicle and the path, whereas it ignores the vehicle's orientation. In spite of this, it allows to regulate to zero both the distance to the path and the difference between the vehicle's orientation and the tangent to the curve, and it is asymptotically stable.**

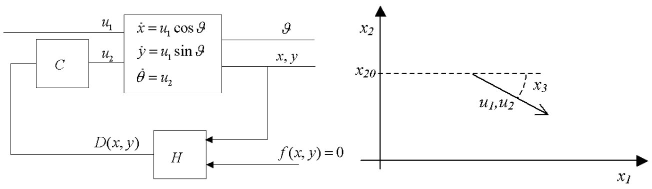

Fig. 1. Left: block diagram of the system; right: Tracking a straight path.

I. Introduction

The article proposes a new feedback control model that allows path tracking for unicycle vehicles. The problem has a particular relevance in mobile robotics [4][5][6], since path tracking is a fundamental prerequisite for navigation both when the path has been planned a priori, and when it is generated on line in order to avoid unforeseen obstacles.

Given a curve in the Cartesian Space, the control model proposed here allows to regulate to zero *a) the distance to the path* as well as *b) the difference between the vehicle's orientation and the tangent to the curve*, and it is proven to be *asymptotically stable*. In addition, when compared to other models in literature, it is "minimalist" in two senses:

- only the distance error D between the vehicle and the path is measured and fed to the controller, whereas most approaches require to measure both the lateral distance from the path *and* the difference between the desired and the actual orientation (Figure 1 on the left);
- the path to follow is expressed through the implicit equation of the curve in the form $f(x, y) = 0$.

About the first point, remark that the distance and the orientation errors cannot be reliably estimated through the integration of state equations: it is assumed here that odometric reconstruction is available, but it is affected by systematic and non systematic errors that make the resulting estimate unreliable, even after a short path (which is the case especially for cheapest robots for domestic use, e.g., cleaning robots). Under these circumstances, external sources of information are usually required for the robot to localize itself with respect to the path; unfortunately, orientation is usually more affected by errors and more difficult to be measured with exteroceptor sensors than position or distance.

An example is wall following: the actual distance from the wall can be sensed through proximity sensors (i.e., sonar, laser scanners, etc.) by considering raw measurements alone, whereas measuring orientation requires more complex computations (compasses are not sufficiently accurate). Another example is GPS-based or IMU-based outdoor navigation: the heading information returned by the GPS is accurate only when velocity is high, whereas Inertial Measurement Units have the problem of temporal drift, and therefore they are not a reliable source of information in the long term. Notice that – with a differentially driven unicycle vehicle – even a small error in wheels encoders can produce a significant error in orientation when moving at low speed.

The second point is important in that it allows to specify paths in a very simple and intuitive way. Most approaches in literature deal with "trajectory tracking" instead of path tracking, by assuming *a point* that moves on the curve with an assigned law of motion, and by dealing with the problem of regulating the distance between the vehicle and the moving point [4][2][8][3][7]. These approaches have been demonstrated to be Lyapunov-stable, and they automatically control the translational speed of the vehicle (in addition to the rotational speed) depending on the path's curvature. However, they require additional computations to determine the law of motion of the reference point. Path-tracking approaches usually require more complex computations as well, e.g., in order to update – every control cycle – the configuration of a Frenet frame which moves on the path and is used to compute feedback errors.

The approach proposed here aims at reducing the computational load towards the end of being implementable even on low-cost robots: every control cycle the distance error D can either be directly measured through distance sensors (if available), or it can be computed starting from the measured robot's position (x, y) by simply evaluating an analytical function $D(x, y)$ and its partial derivatives, whose expression depends on $f(x, y)$ (this latter case is considered in Figure 1 on the left and in the rest of the paper). Finally, the translational speed u_1 is considered a free variable: asymptotic stability is guaranteed by controlling only the rotational speed. If required, the model allows to increase/reduce the translational speed subject to actuator saturation constraints.

Section II describes general ideas, and introduces the

A. Sgorbissa and R. Zaccaria are with DIST, University of Genova, Via Opera Pia 13, 16145, Genova, Italy. Email:{sgorbiss,renato}@dist.unige.it.

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

case of a linear path that lies along the X-axis; Section III consider a generic straight line; Section IV considers a circular path; finally, Section V consider the case of a generic curve expressed through an implicit equation in the form $f(x, y) = 0$. A formal stability analysis is performed for all cases. Experiments with a real robot are described in Section VI; Conclusions follow.

II. Straight line (case 1)

A state vector $\mathbf{x}$ is defined as $\mathbf{x} \equiv [\ x \quad y \quad \vartheta\ ]^T$ where x, y, θ correspond, as usual, to the vehicle's position and orientation with respect to a Cartesian frame. The state equations that describe the unicycle kinematics are:

$$\begin{aligned} \dot{x_1} &= u_1 \cos x_3 \\ \dot{x_2} &= u_1 \sin x_3 \\ \dot{x_3} &= u_2 \end{aligned} \tag{1}$$

where inputs u_1 and u_2 correspond – respectively – to the translational and the rotational speed.

It is initially assumed that the path to follow is a line which corresponds to the $X-$axis of the reference system (Figure 1 on the right): in this case, $x_2 = y$ corresponds to the distance to the line, and $x_3 = \theta$ is the difference between the orientation of the line (which is zero) and the orientation of the robot. $x_1 = x$ increases as the robot moves along the line. It can be demonstrated that, to achieve asymptotic stability, it is sufficient to set control inputs as follows:

$$\begin{aligned} u_1 &= U_1(t) \\ u_2 &= K(-x_2 - \dot{x_2}) \end{aligned} \tag{2}$$

The underlying idea is simple. If the robot moves with a translational speed u_1, the component of u_1 along x_2 is $\dot{x_2} = u_1 \sin x_3$. This is called *approaching velocity*: in Figure 1 on the right, $\dot{x_2}$ is negative since x_3 increases counter clockwise. The *approaching velocity* $\dot{x_2}$ can be increased (up to a maximum value u_1) or decreased (down to a minimum value $-u_1$) by controlling x_3, which – on its turn – requires to operate on the rotational speed u_2. In particular (2) sets the rotational speed u_2 as proportional to the difference between a *reference approaching velocity* $-x_2$ and the the real *approaching velocity* $\dot{x_2}$. The *reference approaching velocity* has the following properties:

- when the robot lies on the path, $-x_2 = 0$;
- when the robot is "above" the path, $-x_2 < 0$ (the robot needs to head downward);
- when the robot is "below" the path, $-x_2 > 0$ (the robot needs to head upward).

The system which results from (1) and (2) is regular and time-invariant: the dynamics of the system depend only on $U_1(t)$ and on initial conditions. $U_1(t)$ can have a generic profile, given that it satisfies kinematics and dynamics constraints. In the following it is assumed that $U_1(t) = U_1$ is constant.

Since x_1 is never in equilibrium (the robot is required to move along the line), the system which is checked for stability comprises only the second and third rows in (1).

$$\begin{aligned} \dot{x_2} &= U_1 \sin x_3 \\ \dot{x_3} &= K(-x_2 - U_1 \sin x_3) \end{aligned} \tag{3}$$

Equilibrium points are given by the solutions of:

$$\begin{aligned} 0 &= U_1 \sin x_3 \\ 0 &= K(-x_2 - U_1 \sin x_3) \end{aligned} \tag{4}$$

Equilibrium points correspond to the set $\{(x_2 = 0, x_3 = k\pi) | k \in Z\}$, i.e., when the distance from the line is null and the robot is oriented along the line. It is necessary to demonstrate that points in the set $\{(x_2 = 0, x_3 = 2k\pi) | k \in Z\}$ are stable equilibrium points, whereas $\{(x_2 = 0, x_3 = (2k+1)\pi) | k \in Z\}$ are not (the robot is moving along the line but in the wrong direction). By considering – for the moment – the only point $(x_2 = 0, x_3 = 0)$, a C^1 Lyapunov function $V = V(x_2, x_3)$ can be defined:

$$V = \frac{K x_2^2}{2} + U_1(1 - \cos x_3) \tag{5}$$

which is locally positive definite for $U_1 > 0$ (i.e., the vehicle is moving forward) and $V(0, 0) = 0$. Its derivative with respect to time:

$$\begin{aligned} \dot{V} &= K x_2 U_1 \sin x_3 + U_1 \sin x_3 K(-x_2 - U_1 \sin x_3) \\ &= -K U_1^2 \sin^2 x_3 \end{aligned} \tag{6}$$

is negative semidefinite, since it does not depend on x_2 and hence $\dot{V}(x_2, x_3) = 0$ for all points in the set $\{(x_2 = x_{20}, x_3 = 0) | x_{20} \in \Re\}$, i.e., corresponding to the x_2-axis. This is sufficient to state that the equilibrium point $(x_2 = 0, x_3 = 0)$ is locally stable. To prove asymptotic stability, it is possible to refer to the LaSalle's principle:

- by considering the level curves of $V(x_2, x_3)$ in Figure 2 a closed region Ω_1 exists which contains the origin, such that $V(x_2, x_3) < l$ for $\{(x_2, x_3) \in \Omega_1\}$;
- $\dot{V}(x_2, x_3) \leq 0$ in Ω_1 since it is negative semidefinite.

The subregion E of Ω_1 for which $\dot{V}(x_2, x_3) = 0$, corresponds to the intersection between Ω_1 and the x_2-axis. It is easy to see that the maximum invariant set M in E contains only the origin: in fact, by considering a generic point $(x_2 = x_{20}, x_3 = 0)$ in E, the second row in (3) becomes:

$$\dot{x_3} = K(-x_{20} - U_1 \sin 0) \tag{7}$$

i.e., it drives the system outside E unless $x_{20} = 0$. This is sufficient to state that the origin is asymptotically stable.

An analysis in the plane of phases is shown in Figure 3: the horizontal axis corresponds to x_3 (i.e., the vehicle's orientation), whereas the vertical axis corresponds to x_2 (i.e., the distance to the path). Figure 3 is obtained by computing, for each given (x_2, x_3), the arctangent of:

$$\frac{\dot{x_3}}{\dot{x_2}} = \frac{K(-x_2 - U_1 \sin x_3)}{U_1 \sin x_3} \tag{8}$$

By inspecting the resulting vector field as well as trajectories in the plane of phases, it is evident that equilibrium

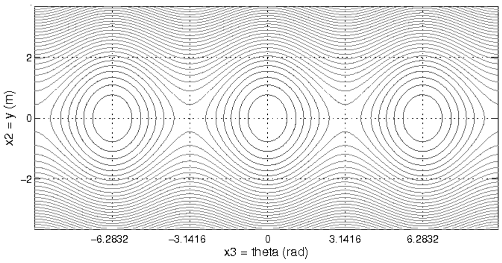

Fig. 2. The level curves of $V(x_2, x_3)$ with $U_1 = 1$, $K = 1$.

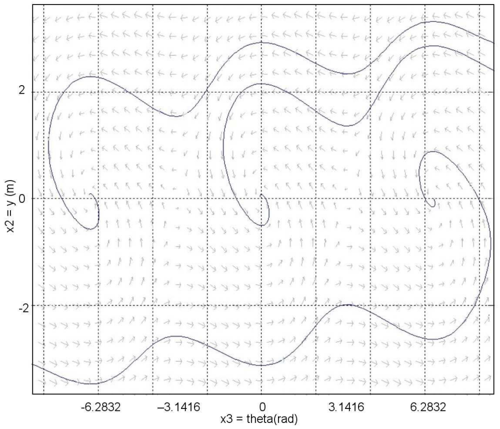

Fig. 3. Analysis in the plane of phases with $U_1 = 1$, $K = 1$.

points in the set $\{(x_2 = 0, x_3 = 2k\pi) | k \in Z\}$ are asymptotically stable (only the subset $\{(0, -2\pi), (0, 0), (0, 2\pi)\}$ is shown). A similar analysis by increasing the proportional gain has been performed as well (not shown here): with $K >> 1$ the trajectories in the plane of phases tends to equilibrium more directly, without performing the spirals in Figure 3. However, when implementing the control law on a real vehicle, increasing K can lead to high-frequency oscillations due to finite sampling time. Therefore, a compromise must be found.

The instability of points in the set $\{(x_2 = 0, x_3 = (2k + 1)\pi) | k \in Z\}$, which guarantees that the vehicle cannot head towards the wrong direction, can be verified in Figure 3 and, more clearly, in Figure 4 (only the subset $\{(0, -\pi), (0, \pi)\}$ is shown). A detailed analysis is not shown for sake of brevity.

Unfortunately, the control law in (2) has a problem: when the distance to the curve (and hence the *reference approaching velocity*) is greater than the maximum *approaching velocity* (whose absolute value is U_1), the *reference approaching velocity* cannot be achieved by controlling x_3. The obvious result is that $u_2 \neq 0$ even when the vehicle is heading perpendicularly to the line (i.e., approaching at the maximum achievable speed); the distance x_2 still converges to zero, but the vehicle possibly turns on itself and finally converges to an equilibrium point $\{(x_2 = 0, x_3 = 2k\pi) | k \in Z, k \neq 0\}$. This effect is evident when considering the trajectories in Figure 3. Consider for example the trajectory than ends up in the equilibrium point $(0, 2\pi)$: if the initial distance from the path is sufficiently high, the robot's orientation starts from a value lower than -2π. The robot performs two complete loops on itself before reaching equilibrium. In general, depending on the initial value of x_2, the number of loops that the robot performs before reaching equilibrium can become arbitrarily high.

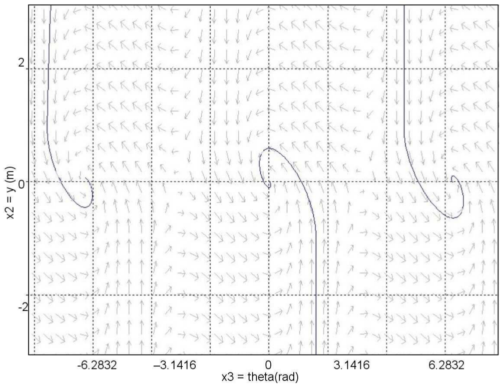

Fig. 4. Analysis in the plane of phases with $U_1 = 1$, $K = 1$ and saturation.

To avoid this, it is sufficient to define the following C^0 function $f_{sat}(x)$, which saturates the values of x:

$$\begin{array}{ll} f_{sat}(x) = -U_1 & x \leq -U_1 \\ f_{sat}(x) = x & -U_1 < x < U_1 \\ f_{sat}(x) = U_1 & U_1 \leq x \end{array} \tag{9}$$

and to redefine the state equations as follows:

$$\begin{array}{l} \dot{x_2} = U_1 \sin x_3 \\ \dot{x_3} = K(-f_{sat}(x_2) - U_1 \sin x_3) \end{array} \tag{10}$$

The analysis in the plane of phases is shown in Figure 4. Asymptotically stable and unstable equlibrium points are the same as previously; however, saturation now guarantees that the robot's orientation x_3 varies within a range of values which depends on the initial conditions but is always within $-\pi$ and π.

III. STRAIGHT LINE (CASE 2)

These results can be extended to a generic straight line, defined through its implicit equation:

$$f(x_1, x_2) = ax_1 + bx_2 + c = 0 \tag{11}$$

with partial derivatives:

$$\begin{array}{l} f_{x_1} = \frac{\partial f(x_1, x_2)}{\partial x_1} = a \\ f_{x_2} = \frac{\partial f(x_1, x_2)}{\partial x_2} = b \end{array} \tag{12}$$

This case can be easily reduced to the previous one through an appropriate coordinate transform. It is considered separately only because it provides theoretical bases to better understand more complex cases described in the following.

A distance function $D(x_1, x_2)$ is required, which can be derived from the standard formula to compute the distance of a point from a straight line:

$$D(x_1, x_2) = \frac{ax_1 + bx_2 + c}{\sqrt{a^2 + b^2}} \tag{13}$$

The distance is taken as a signed value, which allows to distinguish states on the right half-plane with respect to states on the left half-plane, and to consequently assign a direction to the line: that is, $f(x_1, x_2)$ and $-f(x_1, x_2)$ produce a motion along the same line, but towards opposite directions. The partial derivatives of $D(x_1, x_2)$ are:

$$\begin{array}{l} D_{x_1} = \frac{\partial D(x_1,x_2)}{\partial x_1} = \frac{a}{\sqrt{a^2+b^2}} \\ D_{x_2} = \frac{\partial D(x_1,x_2)}{\partial x_2} = \frac{b}{\sqrt{a^2+b^2}} \end{array} \tag{14}$$

In order to guarantee stability, it is sufficient to set:

$$\begin{array}{l} u_1 = U_1 \\ u_2 = K(-D(x_1, x_2) - \frac{d}{dt}D(x_1, x_2)) \end{array} \tag{15}$$

In the following, the dependence of D and ∇D on (x_1, x_2) is not explicited. The whole system can be written as:

$$\begin{array}{l} \dot{x_1} = U_1 \cos x_3 \\ \dot{x_2} = U_1 \sin x_3 \\ \dot{x_3} = K(-D - \frac{d}{dt}D) \end{array} \tag{16}$$

Remark that x_1 and x_2 are never in equilibrium, and x_3 tends to a constant, non-zero value. Therefore, a different system is considered for stability, which is obtained through the following variable substitution:

$$\begin{array}{ll} x_4 = D & \dot{x}_4 = \frac{d}{dt}D \\ x_5 = x_3 - \tan^{-1}\frac{-f_{x_1}}{f_{x_2}} & \dot{x}_5 = \dot{x_3} - \frac{d}{dt}\tan^{-1}\frac{-f_{x_1}}{f_{x_2}} \end{array} \tag{17}$$

That is, x_4 tends to zero as the distance from the line tends to zero, whereas x_5 tends to zero as x_3 tends to the orientation of the line $\tan^{-1}\frac{-f_{x_1}}{f_{x_2}}$. The state equations of the new system can be written as:

$$\begin{array}{l} \dot{x_4} = U_1 D_{x_1} \cos x_3 + U_1 D_{x_2} \sin x_3 \\ \dot{x_5} = K(-x_4 - U_1 D_{x_1} \cos x_3 - U_1 D_{x_2} \sin x_3) \end{array} \tag{18}$$

In (18) there is still a dependence on x_3. To eliminate such dependence consider that that the inverse tangent is an odd function, and hence it holds:

$$\begin{array}{l} \cos x_3 = \cos\left(x_5 - \tan^{-1}\frac{f_{x_1}}{f_{x_2}}\right) = \\ = \cos x_5 \cos\left(\tan^{-1}\frac{f_{x_1}}{f_{x_2}}\right) + \sin x_5 \sin\left(\tan^{-1}\frac{f_{x_1}}{f_{x_2}}\right) = \\ = \cos x_5 \frac{f_{x_2}}{\sqrt{f_{x_1}^2+f_{x_2}^2}} + \sin x_5 \frac{f_{x_1}}{\sqrt{f_{x_1}^2+f_{x_2}^2}} \end{array} \tag{19}$$

Analogously, it holds:

$$\begin{array}{l} \sin x_3 = \sin\left(x_5 - \tan^{-1}\frac{f_{x_1}}{f_{x_2}}\right) = \\ = \sin x_5 \cos\left(\tan^{-1}\frac{f_{x_1}}{f_{x_2}}\right) - \cos x_5 \sin\left(\tan^{-1}\frac{f_{x_1}}{f_{x_2}}\right) = \\ = \sin x_5 \frac{f_{x_2}}{\sqrt{f_{x_1}^2+f_{x_2}^2}} - \cos x_5 \frac{f_{x_1}}{\sqrt{f_{x_1}^2+f_{x_2}^2}} \end{array} \tag{20}$$

which yields (by substituting the values of f_{x_1}, f_{x_2}, D_{x_1}, D_{x_2} defined in (12) and (14)):

$$\begin{array}{ll} \dot{x_4} & = \cos x_5 \frac{U_1 D_{x_1} f_{x_2}}{\sqrt{f_{x_1}^2+f_{x_2}^2}} + \sin x_5 \frac{U_1 D_{x_1} f_{x_1}}{\sqrt{f_{x_1}^2+f_{x_2}^2}} + \\ & + \sin x_5 \frac{U_1 D_{x_2} f_{x_2}}{\sqrt{f_{x_1}^2+f_{x_2}^2}} - \cos x_5 \frac{U_1 D_{x_2} f_{x_1}}{\sqrt{f_{x_1}^2+f_{x_2}^2}} = \\ & = U_1 \sin x_5 \end{array} \tag{21}$$

The system can be finally rewritten as

$$\begin{array}{l} \dot{x_4} = U_1 \sin x_5 \\ \dot{x_5} = K(-x_4 - U_1 \sin x_5) \end{array} \tag{22}$$

Equation (22) have exactly the same structure as (3), and hence it has the same stable/instable equilibrium points $\{(x_4 = 0, x_5 = k\pi) | k \in Z\}$.

IV. Circular path

By assuming – without loosing generality – a circumference with center in $(0, 0)$, the implicit equation of the curve can be written as:

$$f(x_1, x_2) = x_1^2 + x_2^2 - R^2 = 0 \tag{23}$$

whose partial derivatives are:

$$\begin{array}{l} f_{x_1} = \frac{\partial f(x_1,x_2)}{\partial x_1} = 2x_1 \\ f_{x_2} = \frac{\partial f(x_1,x_2)}{\partial x_2} = 2x_2 \end{array} \tag{24}$$

where R is the radius of the circumference. The distance to the curve can be written as:

$$D(x_1, x_2) = \sqrt{(x_1^2 + x_2^2)} - R \tag{25}$$

whose partial derivatives are:

$$\begin{array}{l} D_{x_1} = \frac{\partial D(x_1,x_2)}{\partial x_1} = \frac{x_1}{\sqrt{x_1^2+x_2^2}} \\ D_{x_2} = \frac{\partial D(x_1,x_2)}{\partial x_2} = \frac{x_2}{\sqrt{x_1^2+x_2^2}} \end{array} \tag{26}$$

which are defined everywhere with the exception of $(x_1 = 0, x_2 = 0)$. It can be demonstrated that, in order to guarantee stability, it is sufficient to set control inputs as:

$$\begin{array}{l} u_1 = U_1 \\ u_2 = K(-D - \frac{dD}{dt}) + \frac{d}{dt}\tan^{-1}\left(\frac{-f_{x_1}}{f_{x_2}}\right) \end{array} \tag{27}$$

The whole system turns out to be:

$$\begin{array}{l} \dot{x_1} = U_1 \cos x_3 \\ \dot{x_2} = U_1 \sin x_3 \\ \dot{x_3} = K(-D - \frac{d}{dt}D) + \frac{d}{dt}\tan^{-1}\left(\frac{-f_{x_1}}{f_{x_2}}\right) \end{array} \tag{28}$$

Remark that the $\frac{d}{dt}\tan^{-1}\left(\frac{-f_{x_1}}{f_{x_2}}\right)$ term in (27) and (28) is necessary since – when the distance to the curve is zero and hence the proportional term is null – the rotational speed of the robot must now be non-zero (otherwise the resulting path tends to be a linear approximation of the desired one). Similarly to what has been done for the straight line, two new state variables x_4 and x_5 are introduced, describing – respectively – the distance to the curve and the difference between the heading and the tangent to the curve:

$$\begin{array}{ll} x_4 = D & \dot{x}_4 = \frac{d}{dt}D \\ x_5 = x_3 - \tan^{-1}\frac{-f_{x_1}}{f_{x_2}} & \dot{x}_5 = \dot{x_3} - \frac{d}{dt}\tan^{-1}\frac{-f_{x_1}}{f_{x_2}} \end{array} \tag{29}$$

The state equations are re-written as:

$$\begin{aligned}\dot{x_4} &= U_1 D_{x_1}\cos x_3 + U_1 D_{x_2}\sin x_3\\ \dot{x_5} &= K(-x_4 - U_1 D_{x_1}\cos x_3 - U_1 D_{x_2}\sin x_3)\end{aligned} \tag{30}$$

The system in (29) and (30) has exactly the same structure as previously (the "straight line" path in (17) and (18)). One could object that the system depends on D_{x_1} and D_{x_2}, which – differently from the previous case – now are not constant (26). To eliminate such dependence consider again (19) and (20): symbolic computations return exactly the same result of (21) and, finally, the system in (22), for which stability and instability have been already demonstrated.

V. Generic curve

It would be of great value to extend the same results to a generic curve in the plane, expressed through its implicit equation $f(x_1, x_2) = 0$. However, since it is not trivial to express the distance from a generic curve in closed form, a different approach must be pursued. Even if it does not represent the Euclidean distance from the curve, a possibility is to use the function itself $f(x_1, x_2)$ as the distance function $D(x_1, x_2)$, i.e.:

$$D(x_1, x_2) = f(x_1, x_2) \tag{31}$$

When defined in this way, $D(x_1, x_2)$ still has some good properties which make it appropriate for our purpose:

- $D(x_1, x_2)$ is a scalar field.
- $D(x_1, x_2) = 0$ when (x_1, x_2) lies on the curve; $D(x_1, x_2)$ locally increases/decreases monotonically depending on which side of the plane (x_1, x_2) is located with respect to the curve.

From (31) it derives:

$$\begin{aligned}D_{x_1} &= \tfrac{\partial D(x_1,x_2)}{\partial x_1} = f_{x1}\\ D_{x_2} &= \tfrac{\partial D(x_1,x_2)}{\partial x_2} = f_{x2}\end{aligned} \tag{32}$$

In order to guarantee stability, it is possible to set:

$$\begin{aligned}u_1 &= U_1\\ u_2 &= K(-\|\nabla f\| D - \tfrac{d}{dt}D) + \tfrac{d}{dt}\tan^{-1}\left(\tfrac{-f_{x_1}}{f_{x_2}}\right)\end{aligned} \tag{33}$$

by omitting to write – once again – the dependence of D and ∇f on (x_1, x_2). As it will be shown in the following, the term $\|\nabla f\|$ is now necessary to guarantee stability. As in the case of a circumference, the rotational speed is given by a term proportional to the distance plus a term which depends on the curvature of the isocline in the current position.

The whole system can be expressed as:

$$\begin{aligned}\dot{x_1} &= U_1\cos x_3\\ \dot{x_2} &= U_1\sin x_3\\ \dot{x_3} &= K(-\|\nabla f\| D - \tfrac{d}{dt}D) + \tfrac{d}{dt}\tan^{-1}\left(\tfrac{-f_{x_1}}{f_{x_2}}\right)\end{aligned} \tag{34}$$

Two new state variables (x_4, x_5) are introduced as in (29), with exactly the same meaning: when x_4 tends to zero, D tends asymptotically to zero; when x_5 tends to zero, the orientation of the robot's tends asymptotically to the tangent to the path. The resulting state equations are:

$$\begin{aligned}\dot{x_4} &= U_1 f_{x_1}\cos x_3 + U_1 f_{x_2}\sin x_3\\ \dot{x_5} &= K(-\|\nabla f\| x_4 - U_1 f_{x_1}\cos x_3 - U_1 f_{x_2}\sin x_3)\end{aligned} \tag{35}$$

In (35) there is still dependence on x_3, f_{x_1} and f_{x_2}, the latter depending – on their turn – on x_1 and x_2. Equations (19) and (20) allow to eliminate this dependence. However, it now derives a different expression for $\dot{x_4}$:

$$\begin{aligned}\dot{x_4} \quad &= \cos x_5 \tfrac{U_1 f_{x_1} f_{x_2}}{\sqrt{f_{x_1}^2 + f_{x_2}^2}} + \sin x_5 \tfrac{U_1 f_{x_1} f_{x_1}}{\sqrt{f_{x_1}^2 + f_{x_2}^2}} +\\ &+ \sin x_5 \tfrac{U_1 f_{x_2} f_{x_2}}{\sqrt{f_{x_1}^2 + f_{x_2}^2}} - \cos x_5 \tfrac{U_1 f_{x_2} f_{x_1}}{\sqrt{f_{x_1}^2 + f_{x_2}^2}} =\\ &= \|\nabla f\| U_1 \sin x_5\end{aligned} \tag{36}$$

The system can be written as:

$$\begin{aligned}\dot{x_4} &= \|\nabla f\|\, U_1 \sin x_5\\ \dot{x_5} &= K(-\|\nabla f\| x_4 - \|\nabla f\|\, U_1 \sin x_5)\end{aligned} \tag{37}$$

The system no more depends on x_3. One could object that there is still a dependence on (x_1, x_2) through $\|\nabla f\|$; however, when analysing the system in the plane of phases, it is necessary to compute the ratio:

$$\frac{\dot{x}_5}{\dot{x}_4} = \frac{K(-\|\nabla f\| x_4 - \|\nabla f\|\, U_1 \sin x_5)}{\|\nabla f\|\, U_1 \sin x_5} \tag{38}$$

By simplifying the $\|\nabla f\|$ term both in the numerator and in the denominator, (38) yields the same results as (8), which have been already depicted in Figure 3; that is, the behaviour of the system in (37) is identical to the previous cases, when analyzed in the plane of phases. Obviously, the meaning of x_4 is different, since now x_4 represents the value of $f(x_1, x_2)$ instead of the Euclidean distance to the curve. However, this is sufficient to claim that the system is still guaranteed to reach an Equilibrium state in which the robot's position lies on the path (i.e., $f(x_1, x_2) = 0$), and the robot's heading is tangent to the path itself.

Finally, it is convenient – as in previous cases – to saturate the *reference approaching velocity* to avoid that x_5 converges to an equilibrium point different from 0.

VI. Experiments

The model has been extensively tested both in simulation (the MatLab Simulink environment) and on real robots: among the others, the indoor robot Staffetta and the outdoor robot ANSER, which – in 2006/2007 – has been performing surveillance patrols at the Villanova D'Albenga Airport.

Simulates experiments are not shown here for sake of brevity. Figures 5 and 6 show five different experiments with the indoor robot Staffetta, moving at the constant translational speed $1m/sec$, along three different types of paths: a straight line (Section III), a circle (Section IV), and a sinusoidal profile defined through its implicit equation (Section V). All Figures show two different plots: on the left, the path followed by the robot in the Cartesian Space (the scale along the two Cartesian axes is not the same); on the right, the error, measured as the distance to the path.

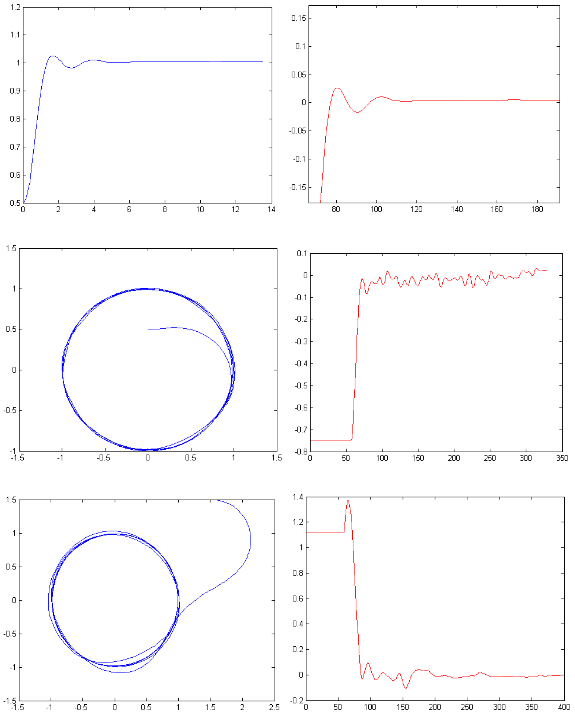

Fig. 5. Top: Straight line. Middle: Circle, robot starting inside. Bottom: Circle, robot starting outside. Left: robot's path; Right: error plot.

Figure 5 on the top shows convergence to a straight line. The robot starts $50cm$ away from the path, and quickly coverges to the line: the error at steady state is below $1cm$ (Figure 5 on the top zooms on a part of the error plot). Before approaching the line, it is possible to observe that the robot's trajectory tends to follow a damped sinusoidal profile around the reference path: this was expected from the analysis in Figure 3, since all trajectories in the plane of phases tend to follow a spiral around the equilibrium point before reaching it. This effect can be reduced by increasing the proportional gain K: however, this can lead to high-frequency oscillations and, finally, to instability.

Figure 5 in the middle and Figure 5 on the bottom show convergence to a circle with $1m$ radius, starting from two different positions inside and outside the circle. As expected, the error at steady state is bigger in this case, even if it is always below $5cm$. This is not a surprise, by considering that the control loop for path tracking is closed every $50ms$, which is quite a long sampling time compared with the translational speed u_1: in fact, in $50ms$ the robot moves of about $5cm$.

Figure 6 on the top and Figure 6 on the bottom show convergence to a sinusoidal path. The error at steady state is much lower in Figure 6 on the top, i.e., $< 2cm$, than in Figure 6 on the bottom, i.e., about $10cm$. However, notice that the two profiles have a different amplitude and frequency: approximately, $0.6m$ amplitude and $10m$ period in the former case, $2m$ amplitude and $5m$ period in the second case. The second profile turns out to be quite challenging for a robot that moves at $1m/sec$, and the measured distance error is consequently affected.

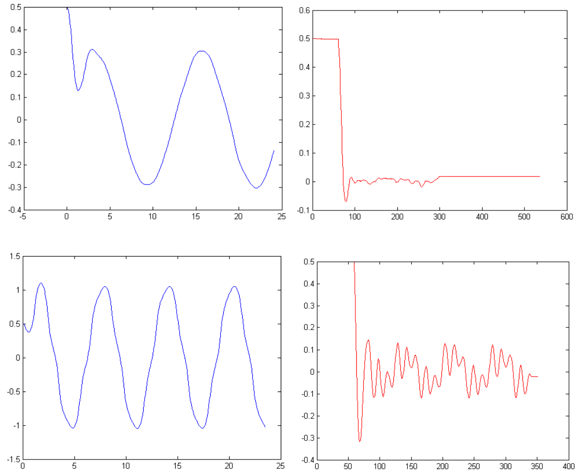

Fig. 6. Top: Sine, $\approx 0.6m$ amplitude, $\approx 10m$ period. Bottom: Sine, $\approx 2m$ amplitude, $\approx 5m$ period. Left: robot's path; Right: error plot.

VII. Conclusions

The article describes a novel control law for path tracking that, in spite of its simplicity, is demonstrated to be asymptotically stable and to perform well in realistic conditions. In particular, the model proves to guarantee smooth path following even in presence of big errors in the vehicle orientations. Finally, even if not illustrated in this article, the model has been recently adapted to consider the case of a vehicle that moves among obstacles with arbitrary shape, which position is not know a priori, and possibly changes in time. See [9] for a reference.

References

[1] S. M. Lavalle, Planning Algorithms, Cambridge Univ Press, 2006

[2] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino. Closed Loop Steering of Unicyle-Like Vehicles via Lyapunov Techniques. IEEE Robotics and Automation Magazine, 1995

[3] G. Indiveri and A. Nüchter and K. Lingemann. High Speed Differential Drive Mobile Robot Path Following Control With Bounded Wheel Speed Commands. 2007 IEEE International Conference on Robotics and Automation Roma, Italy, 10-14 April 2007

[4] C. Samson and K. Ait-Abderrahim, Mobile Robot Control Part 1: Feedback Control of A Non-Holonomic Mobile Robots, Technical Report No. 1281, INRIA, Sophia-Antipolis, France, June 1991

[5] C. Canudas de Wit, H. Khennoul, C. Samson, and O. J. Sordalen, Nonlinear control design for mobile robots, in Recent Trends in Mobile Robots, ser. Robotics and Automated Systems, Y. F. Zheng, Ed. World Scientific, 1993, ch. 5, pp. 121156

[6] Z. P. Jiang and H. Nijmeijer, A recursive Technique for Tracking Control of Nonholonomic Systems in Chained Form, IEEE Trans. on Robotics and Automation, Vol 44, No 2, 1999, pp. 265-279

[7] M. K. Bugeja and S. G. Fabri Dual Adaptive Control for Trajectory Tracking of Mobile Robots, 2007 IEEE International Conference on Robotics and Automation Roma, Italy, 10-14 April 2007

[8] L. Lapierre, R. Zapata and P. Lepinay, Simultaneous Path Following and Obstacle Avoidance Control of a Unicycle-type Robot, 2007 IEEE International Conference on Robotics and Automation Roma, Italy, 10-14 April 2007

[9] A. Sgorbissa, A. Villa, A. Vargiu, R. Zaccaria, A Lyapunov-stable, sensor-based model for real-time trajectory-tracking among unknown obstacles, 2009 IEEE/RSJ International Conference on Intelligent RObots and Systems, October 11 – 15, 2009, St. Louis, MO, USA