Proceedings of the 1999 IEEE
International Conference on Robotics & Automation
Detroit, Michigan • May 1999

A Generalization of Path Following for Mobile Robots

F. Díaz del Río, G. Jiménez, J. L. Sevillano, S. Vicente, A. Civit Balcells.

Facultad de Informática. Univ. Sevilla. Tf. (+34) 54552779. E-mail: fdiaz@icaro.fie.us.es
Address: Avda. Reina Mercedes s/n. 41012 Sevilla. SPAIN

Abstract

Several authors have proposed some methods for applying path following in specific cases to mobile robots ([3], [9], [11], etc.). When we try to extend the path following approach to the general problem several difficulties arise. In this paper we present a generalized technique to apply path following to a mobile robot with nonholonomic constraints. As an application example, we expose the case of mobile robots with a higher degree of maneuverability than the typical car-like robots. In particular we consider a robot that can turn around itself making a zero-radius turn; a case still not resolved as far as we know. Finally we propose a suitable control law for this example that ensures asymptotical convergence.

1 Introduction

The most extended systems in automatic control theory are servosystems. Here we track a mobile system at the time it moves; i.e. position, velocity or, in general, any magnitude in which we are interested, is the instantaneous reference that our system must follow. In figure *fig. 1a* we show this case for the state coordinates $q(t)$, the desired coordinates $q_{des}(t)$, and the error coordinates defined as $e_q(t)=q(t)-q_{des}(t)$. In servosystems this is the only possibility we have, because the reference trajectory is collected as we do the tracking.

On the other hand, in mobile robots it is usual that the trajectory is memorized or previously generated by a path generator module [fdiaz1]. For our purposes both cases are the same, and the term *memorized path* or merely *path* is used for both of them. A reference or desired path to be followed is described by a single parameter, namely r, and it can be expressed as a vector of state coordinates $q_{des}(r)$. Furthermore, we must emphasize the importance that convergence to a path acquires in mobile robots, as convergence to a fixed point q_o, can not be achieved through a smooth feedback stabilization control law (a direct result of Brockett's theorem [1]).

When we try to track a memorized path, the tracking methodology can be very different, as we know a priori the whole trajectory. Thus we can find several possibilities to do the tracking. Of course the classical servosystem tracking can be done just by identifying the parameter $r(t)$ associated to the path with time, that is $r(t)=t$. Another similar possibility called *trajectory tracking (TT)* is based in a more general assumption than simple servosystems (see *fig. 1b*): the parameter $r(t)$ is a generic function of time. Therefore the error coordinates are $e_q(t)=q(t)-q_{des}(r(t))$. Then we can go through the stored trajectory with the most appropriate scale for $r(t)$, for example $r=at$. Using an asymptotically stable control law (e.g. [8]), it is guaranteed that the system will converge to the desired trajectory in a deterministic time (except for the inherent perturbations that it may suffer).

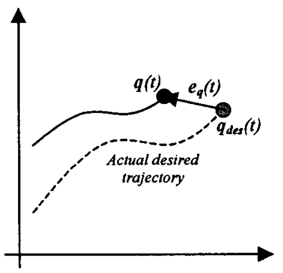

Fig. 1a: Tracking in a servosystem

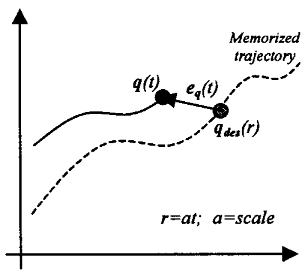

Fig. 1b: Trajectory Tracking

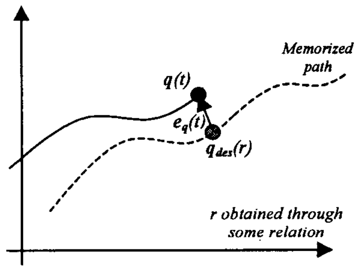

Fig. 1c: Path Following

Although TT is straightforward, it is not the only method (nor the most suitable) for following memorized paths. During the last years several alternatives have been proposed. The best established in literature and most suitable for many situations in which time is not a critical parameter (this is the case for most cases in industrial mobile robots) is *path following (PF)*. This is based in some relation between actual system's state $q(t)$ and the memorized path. This relation will give us the desired point $q_{des}(r)$ of memorized path to be tracked. The real system should try to follow this point instead of the one given by the other approach (see *fig. 1c*). The error coordinates are also $e_q(t)=q(t)-q_{des}(r)$. Using this approach, it is not guaranteed that the system will reach a point of the desired trajectory in a deterministic time.

The virtues of PF can be understood considering this example: if big perturbations force the system to be at rest, for TT the desired point will move unavoidably. This means that errors will grow up to some value that may introduce instability. On the other hand, if PF is used, the desired point will be the same in spite of these perturbations. This allows the system to overcome large perturbations avoiding possible unstable states. Moreover, the extraction of an asymptotically stable law using PF is not more difficult than using TT, as we can see in the

mentioned literature and also in the example of section IV. Thus interest in PF is growing rapidly.
There have been several trials to apply PF (we describe them below) in specific cases, but when we try to generalize PF several problems arise. In the next sections we will try to solve these problems, clarifying them with a complete example.

2 Definitions and Robot Model

Let's consider a general mobile robot as shown in *fig. 2* and let $q=(X,\ Y,\ \phi)^t$ be its state coordinates, which represent the cartesian position $(X,\ Y)\in R^2$ of a certain point P_O (typically the midpoint between the rear wheels) with respect to a fixed extrinsic coordinate system $\Re(O,\ \mathbf{i},\ \mathbf{j})$ and the orientation $\phi\in(-\pi,\pi]$ of the robot with respect to the X axis. We will choose $\mathbf{u}=(v,\ \omega)^t\in R^2$ as the pair of control variables for our system which represent the linear velocity of point P_O and the angular velocity of the robot (other pair of variables such as torques or voltages supplied to the motors, are analogue for the tracking study, as showed by [2]).
For these vector variables the state equation of the mobile robot are the well-known equations (that are non-linear in $\mathbf{q}$ and linear in $\mathbf{u}$):

$$\dot{\mathbf{q}} = \mathbf{B}(\mathbf{q})\mathbf{u} \ \ ; \ \mathbf{B} = \begin{bmatrix} \cos\phi & 0 \\ \operatorname{sen}\phi & 0 \\ 0 & 1 \end{bmatrix} ; \ \mathbf{u} = \begin{pmatrix} v \\ \omega \end{pmatrix} \ \ ; \ \ \mathbf{q} = \begin{pmatrix} X \\ Y \\ \phi \end{pmatrix} \qquad (1)$$

To study the tracking of a memorized reference path $q_{des}(r)=(X_{des}(r),\ Y_{des}(r),\ \phi_{des}(r))^t$ let us define another intrinsic coordinate system $\Im\{q_{des}(r(t)),\ \mathbf{t},\ \mathbf{n}\}$ linked to the path. $\mathbf{t}$ is the unitary vector tangent to the planar path in the desired point $q_{des}(r(t))$ and $\mathbf{n}$ the normal to it[1]. Let $(e_x,\ e_y)\in R^2$ be the position errors of point P_O relative to these axis and $e_\phi\in(-\pi,\pi]$ the robot orientation error, so $e_q(t)=(e_x,\ e_y,\ e_\phi)^t$ will be our relative error vector[2]. Let $u_{des}=(v_{des}(r),\omega_{des}(r))^t$ be the desired control state expressed as a function of the descriptor parameter r.
At this point it is important to define exactly which paths $q_{des}(r)=(X_{des}(r),\ Y_{des}(r),\ \phi_{des}(r))^t$ are valid. We can not choose an arbitrary function on r and assign it to the three desired state coordinates. These desired coordinates must have some properties to guarantee that the tracking is possible. First the domain of r must be infinite (for example the positive real line R^+) in order to guarantee that the reference trajectory does not end. This must be done to ensure the possibility of convergence, because, as it is well known for a mobile robot, Brockett's theorem [1] prevents feedback stabilization[3] of the robot to a fixed point. Second, and for the same reason, the reference trajectory can not contain singular points where the inputs are null, i.e. $\mathbf{u}_{des}=\mathbf{0}$. Finally, the path can be made by a robot like the studied one, and this implies that $(X_{des}(r),\ Y_{des}(r),\ \phi_{des}(r))$ must respect the nonholonomic constraints of our mobile robot.

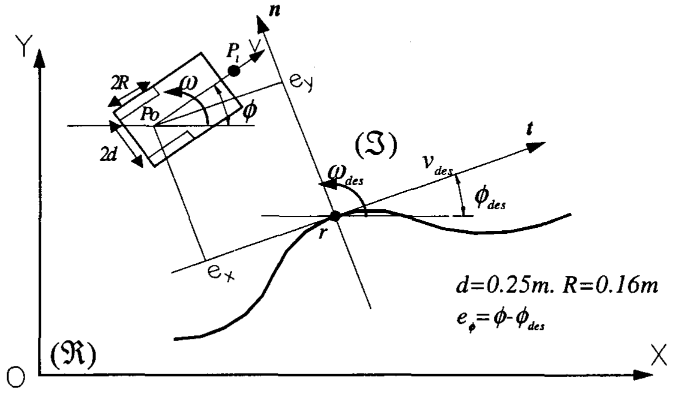

fig. 2: extrinsic and intrinsic robot coordinates.

Using the above relative variables and coordinates linked to the path, and by simple calculations, the following state equations can be easily found [6]:

$$\begin{pmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\phi \end{pmatrix} = \begin{pmatrix} -v_{des}(t) \\ 0 \\ -\omega_{des}(t) \end{pmatrix} + \begin{pmatrix} 0 & \omega_{des}(t) & 0 \\ -\omega_{des}(t) & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} e_x \\ e_y \\ e_\phi \end{pmatrix} + \begin{pmatrix} \cos(e_\phi) & 0 \\ \operatorname{sen}(e_\phi) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \qquad (2)$$

In its more general form if we define errors in a natural fashion, that is $\mathbf{e}_\mathbf{q} = \mathbf{R}(\mathbf{q}_{\mathbf{des}})(\mathbf{q} - \mathbf{q}_{\mathbf{des}})$, then the matrix form of the above equation can be expressed as:

$$\dot{\mathbf{e}}_\mathbf{q} = \mathbf{B}_{\mathbf{des}}(\mathbf{e}_\mathbf{q})\mathbf{u}_{\mathbf{des}}(t) + \mathbf{B}(\mathbf{e}_\mathbf{q})\mathbf{u} \qquad (3)$$

3 The General Method for Path Following

Previous studies. During the last decade there has been a great research effort to develope a tracking based in a PF. This has lead to several good approaches that have made emphasis in diverse aspects of PF according to the particular characteristics of the analyzed system or the reference paths to be tracked. The most important can be summarized in the following categories:

1. In [3] and [9] the desired point in the path is obtained through a normal projection along the vector that we have called $\mathbf{n}$. Therefore this projection chooses the point of the reference path that has null e_x coordinate (see *fig. 2*). They have to prohibit paths containing circles with small radius (we will call turns with null radius and infinite curvature "zero-radius turns") to ensure that the normal projection exists and is unique. A similar path following was used in Navlab [12]. As Navlab is a car-like robot, it cannot make zero-radius turns, so these paths were not considered.
2. In [11] the projection point chosen by the authors is the one that minimizes the euclidian distance between the real and the reference points P_l (see *fig. 2*). Using point P_l they avoid paths with curvature tending to infinite. But this strategy fails when the reference path is a turn around point P_o. In this case any actual configuration (having different orientations) whose point P_l is *on* the desired position for P_l will have zero distance. That is, the couple $(X_l,\ Y_l)$ does not

[1] The vectors $\mathbf{t}$ and $\mathbf{n}$ exists only when the linear velocity v_{des} at this point of the virtual robot that went through the reference path was not zero; if it were null, $\mathbf{t}$ can be chosen parallel to the virtual robot orientation, as the nonholonomic constraint requires the two vectors to be parallel.
[2] An analogous coordinate system was used in [Kanay90]. There the system was linked to the robot itself.
[3] We mean smooth control laws, so that we elude non-smooth laws for reasons of continuity on the control variables.

represent the whole state of a mobile robot, although these coordinates always change for every trajectory. This example shows us that the whole state of the robot must be considered to construct a generic tracking.

It is important to mention that previous studies have not given an exact definition or construction of PF as far as we know; they have specified a particular method, suitable for their requirements, that can be called PF. Conversely, as our goal in this paper is to construct a general approach to PF, and considering the experience extracted from previous works, we must have in mind the following two statements. First we must consider the whole robot's state (represented here by (X, Y, ϕ) for simplicity, and usually called the *robot posture*). Second, we must contemplate all the possible reference paths that can be followed by the robot, including zero-radius turns.

General path following characteristics. As a first step and in order to get a general construction of PF, we are going to extract the generic characteristics of *path following*. According to these studies and the intuitive behavior mentioned at the Introduction, the main characteristics that must rule PF and that differentiate it from TT, can be summarized as follows:

1. We only must consider the global shape of the path to do the following. The desired trajectory evolution (governed by r) must not play any role in the track as it does in TT.
2. In opposition to TT (where the desired posture is exactly determined by a rigid law like $r=r(t)$), in PF we must choose some relationship to determine the desired posture. We will call this relationship "projecting function" as it projects the actual posture to the reference path. We will denote it as $f_{proj}()=0$.
3. If the robot stops, the reference or goal point must also stops, as the parameter r does not grow by itself. The progress of r must not be independent (as in TT) but dependent on the real robot movement, that is $\dot{r}$ equation must be driftless.
4. The existence of the rigid law $r=r(t)$ in TT implies the reference evolution to be $q_{des}(r(t))$, and consequently "pulling" or "dragging" the robot to reach the reference. On the other hand, in PF the reference path can not "pull" (or "drag") the robot: the robot must move independently by some condition (of course, meanwhile a control law must ensure convergence to the path). We must impose a motion in the real system to guarantee it moves or progresses. We will call this condition "motion exigency" and we will denote it as $f_{motexig}(\boldsymbol{u})=0$.
5. A direct result of what is explained before is that there is no time exigency in the following. This means that we can not ensure that the robot will reach a reference point in a predictable period of time.

Path following construction. TT's construction is elementary. It requires only choosing the most suitable relation $r=r(t)$, for a deterministic tracking in the system. On the contrary PF construction is not so bare because it implies a special relationship between the actual point and the global path. Paradoxically, due to the more laborious character of PF, it permits more flexibility than TT.

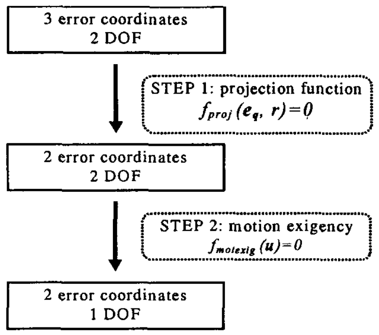

fig. 3. General path following construction scheme.

Some methods for PF construction have been developed in the references mentioned in this section, but they only apply to specific cases. Conversely, and based on the previous characteristics, we can straightforwardly find a PF construction based in two generic steps (see *fig. 3*). In this scheme we begin with a mobile robot that has three state coordinates (three error coordinates $\boldsymbol{e_q}$ expressed relative to the reference path) and two degrees of freedom $\boldsymbol{u}$ (DOF). Although we present the case for a mobile robot; the case for a generic nonholonomic system can be easily deduced.

First step: "projecting function". This is derived as a consequence of characteristics 1 and 2 of PF. Once a distance criterion is chosen, a "projecting function" $f_{proj}()=0$ relates real posture with global path. This gives us a projecting point on the desired path: it is the desired posture $q_{des}(r(t))$ at this instant of time. At the same time the projection is an holonomic constraint between error coordinates; so it supposes the elimination of one error coordinate. As the projection must be stated between error coordinates and the desired path, it depends on actual errors $\boldsymbol{e_q}$ and on the memorized path shape (in general, parameter r); that is $f_{proj}(\boldsymbol{e_q}, r)=0$. As we are talking about a geometric projection, vector $\boldsymbol{u}$ can not play a role in this step, because the real robot velocity does not influence on a geometric projection. Consequently the projection introduces the following coordinate transformation:

$$\{O, q_1, q_2, q_3\} \rightarrow \{q_{des}(r), r, e_1, e_2\} \;;\; e_p=(e_1, e_2)$$

Points $\{\boldsymbol{e_q}\}$ that obey $f_{proj}(\boldsymbol{e_q}, r)=0$ define a surface (two-dimensional in our case) where the robot is placed. Hence robot posture is now given by only two error coordinates $\boldsymbol{e_p}=(e_1, e_2)$ instead of the three $\boldsymbol{e_q}=(e_x, e_y, e_\phi)$ on a TT.

A classical example of projecting function is the normal projection described in [3] and [9], equivalent to making e_x null. That is, the first error coordinate e_x is eliminated and the robot posture is expressed by only two: $\boldsymbol{e_p}=(e_y, e_\phi)$ (error coordinates are called (y, θ) in these references). The two-dimensional surface is the e_y axis extended for all the possible robot orientations. This simple method for eliminating one of the error coordinates can not always be used, as we show in the example of the next section.

It is important to remark that parameter r is at this time the third state coordinate[4], and we should mention it to specify the

[4] In TT r gives us no state because r is determined only by time through the function $r=r(t)$.

whole robot posture, now given by (r, e_1, e_2). But r is not an *error coordinate*, that is, it does not play any role in the control or in the stabilization problem, that is centered only in making $e_p(t) \rightarrow 0$, regardless of r. At this step we can say that we have isolated parameter r from the path convergence problem, and that we have a system with two degrees of freedom and two state variables (forgetting r), where smooth stabilization is possible. This assertion does not contradict the above mentioned impossibility of feedback stabilization to a fixed posture in nonholonomic systems, because we only stabilize two coordinates with this PF convergence, neglecting the third coordinate r. In other words, we can stabilize $e_p(t) \rightarrow 0$ but not $e_q(t) \rightarrow 0$.

The addition of a projecting function $f_{proj}(e_q, r)=0$ gives us the way in which parameter r varies. Differentiation of this function lead us to[5]:

$$\frac{\partial f_{proj}}{\partial \mathbf{e}_q}\mathbf{e}_q + \frac{\partial f_{proj}}{\partial r}\dot{r} = 0$$

Now state equation (3) can be substituted, and using the "chain-law" $\dot{f} = f'\dot{r}$, we solve for $\dot{r}$ and have finally:

$$\Rightarrow \dot{r} = -\frac{\dfrac{\partial f_{proj}}{\partial \mathbf{e}_q}\mathbf{B}(\mathbf{e}_q, r)\mathbf{u}}{\dfrac{\partial f_{proj}}{\partial r} + \dfrac{\partial f_{proj}}{\partial \mathbf{e}_q}\mathbf{B}_{des}(\mathbf{e}_q)\mathbf{u}_{des}(r)} \qquad (4)$$

Now we have solved the problem of finding a closed expression for the variation of parameter r in an elegant way. This is clearly a PF, because variation of r does not depend implicitly on time. In the equation (4) there may be some operation restrictions; for example if denominator is null, variation of r is undefined. This case must be analyzed for each application and we will study it in depth for the example of section IV.

The optimum projection depends on the mobile robot structure and even on the application, but we can state some general conditions for a "good" projecting function to be coherent:

1. A projecting point can always be found for any system's state q and for any valid reference path, that is: $\forall (X,Y,\phi) \in \Re^2 \times (-\pi, \pi], \exists r_o \in \Re / f_{proy}(q-q_{des}(r_o), r_o)=0$
2. Uniqueness of the point on the path $q_{des}(r)$ must be ensured (at least locally[6]).
3. If the actual robot state is the same as a posture of the path: $q(t)=q_{des}(r_o)$, then the projected r must be r_o. That is f_{proy} has a zero for $e_q=0$: $f_{proy}(0, r)=0 \ \forall r$.
4. It would be desirable that the analytic equations could have a closed form, to help the finding of a control law whose stability is analytically demonstrable.

Second step: "motion exigency". Finally, as we described in PF characteristic 4, we need to imposed a "motion exigency" $f_{motexig}(\boldsymbol{u})=0$ to guarantee that the robot moves. Although the form of this function depends on the application, we must fulfill the following conditions:

1. No solution at the origin $\boldsymbol{u}=0$ so $\mathbf{u}$ is never null;
2. It is desirable that $f_{motexig}(\boldsymbol{u})=0$ is an even function on its components[7]. This ensures that the robot will approach the path through the most suitable inputs (negative or positive), mindless of the direction it must take on the path (increasing or decreasing r). This is particularly important when errors are big.
3. To help the control law to converge to the path, it would be desirable that components of $\boldsymbol{u}$ behave symmetrically, that is the total motion $|\boldsymbol{u}|$ should split identically between v and ω.

In the current mobile robot literature most motion exigencies (not called with this term) are applied to car-like robots, so it is usual to have $v=cte$, which is intuitive for cars. For robots with higher maneuverability others authors have preferred the exigency given by $|F_{cent}|=cte$, that is $|\omega|\,|v|=cte$, to avoid slippage. As the last one has an indetermination for null ω or v, and our (2,0) robot does not have motion restrictions, we will use the adequate "motion exigency" given by:

$$f_{motexig}(u)=0 \ \Leftrightarrow \ \sum_{i=1}^{n} b_i^2 u_i^2 = K_{mov}^2 \ \ ; b_i \in \Re \qquad (5)$$

where K_{mov} is the whole motion applied to the system and b_i is the scale factor for each input.

4 An Example Application: The Case Of A (2,0) Mobile Robot.

One of the most extended mobile robot configurations is that with degree of maneuverability 2 and 0 steering wheels (a (2,0)-robot according to the definition of [2]). The typical topology of these (2,0)-robot include two driver motors at each rear wheel, that can turn independently forward or backward. Furthermore it is one of the robots in which trajectories are more complicated, as it can not have complete maneuverability (that is, it is not omnidirectional) but it can make zero-radius turns. So it is a very interesting problem to apply our path following construction to these robots. As our group has been interested during the last years in the improvement of electrical wheelchairs ([4][5][7]) that incorporate this configuration, we have studied the complications that this topology introduces.

First step: "projecting function". Maneuverability in these robots is very high, and they have no additional movement restrictions (except for the inherent nonholonomic constraint). Thus we can choose the projecting point as that in the path that is nearest to the robot, i.e. the one which distance is minimal. As the three error coordinates must play a role in the distance, a "good" election for the distance d_q can be:

$$d_q^2(r,t) = \sum_{i=1}^{m} K_i^2 e_i^2(r,t)$$

where K_i are the scale factors between the different errors to guarantee dimensional homogeneity. In the case of our robot this leads to:

[5] Here derivation respect to a vector holds for a summation.

[6] Global uniqueness is impossible for a mobile robot that is fully controllable, since it can reach the same posture as many times as it wishes.

[7] We are assuming here that the robot motors have not a special structure or the application does not need to march in a unique direction. A car would be for example this case, because its rear direction is limited.

$$d_q^2(r,t) = e_x^2 + e_y^2 + K_\phi^2 e_\phi^2 \tag{6}$$

To choose the minimal distance point we "freeze" actual robot posture (that is ***u=0***) and "move" along the desired path (that is we vary *r*) looking for the point with a local minimum:

$$\left.\frac{\partial d_q^2}{\partial r}\right|_{u=0} = 0 \Rightarrow e_x(t) v_{des}(r) = -K_\phi^2 e_\phi(t) \omega_{des}(r) \tag{7}$$

where we omit simple calculations (using (2)) and use the "chain-law" $\dot{f} = f' \dot{r}$. Here (′) holds for differentiation respect to *r* and (˙) respect to *t*. So (7) is our $f_{proy}(e_q(t),r)=0$, where $v_{des}(r)$, $\omega_{des}(r)$ are the input control profiles of desired path. This projection is fully intuitive, because it always chooses the nearest point to the actual robot posture.

As we described in section III, differentiation of this projecting function gives us the new equation for the variation of parameter *r*:

$$\dot{r} = \frac{v v_{des}(r) \cos e_\phi + K_\phi^2 \omega \omega_{des}(r)}{v_{des}^2(r) - \omega_{des}(r) v_{des}(r) e_y + K_\phi^2 \omega_{des}^2(r) - K_\phi^2 e_\phi \omega'_{des}(r) - e_x v'_{des}(r)} \tag{8}$$

As this projection has been obtained through a generic PF construction, we can get some of the classical projections as particular cases. For example the normal projection used in [3][9], is found by doing $K_\phi=0$ (you should remind that normal projection to a plane curve coincides with minimal Euclidean distance, given by the distance d_q when $K_\phi=0$ [10]). In effect the nullity of K_ϕ and the use of the same parameter of these authors (characterized by $v_{des}(r)=1$), leads to:

$$v_{des}(r) = 1 \Rightarrow \dot{r} = \frac{v(t)\cos e_\phi}{1-\omega_{des}(r) e_y} = \frac{v(t)\cos e_\phi}{1-K_{des}(r) e_y}$$

Once we have chosen this projection we should confirm if it verifies the conditions of the above section to be considered a "good" projecting function. All of these conditions except number 2 are straightforwardly satisfied. Uniqueness is equivalent in our case to the non-nullity of denominator of equation (4) [6]. A deep study is made in [6] and its final result is that local uniqueness is reached under certain non-severe conditions[8]. These conditions are two bounds for K_ϕ and curvature derivative of the desired path $K'_{des}(r)$, that can be easily satisfied if errors are not unbearable and desired paths are not abrupt. In opposition to these non-severe conditions, normal projection must avoid paths containing circles with small radius to ensure that it exists and is unique, that is, it is far more restrictive for the feasible reference paths.

Second step: "motion exigency". In the case of the (2, 0)-robot, we have only two degrees of freedom; so a very suitable motion exigency is the one mentioned before:

$$v^2(t) + b_\phi^2 \omega^2(t) = K_{mov}^2 > 0 \tag{9}$$

where K_{mov} is the whole velocity applied to the system and b_ϕ is the scale for the angular speed, that give us how much the system can turn.

In conclusion we have reduced the system just to two state variables (not explicitly defined but obtained through the application of a "projecting" constraint given by equation (7) to the three errors e_q), and one degree of freedom (resulting from the use of (9) to vector input ***u***). In these variables we have condensed what we need to converge to a generic path through a path following.

Control law. Although control law selection is far from our objectives, it is convenient to show the behavior of our system and the way to get to an asymptotically stable control law. We will use Lyapunov's second method method with the quadratic error function as the Lyapunov function (which matches with the semidistance):

$$V = \frac{1}{2} d_q^2(r,t) = \frac{1}{2}\left(e_x^2 + e_y^2 + K_\phi^2 e_\phi^2\right) \tag{10}$$

Differentiating with respect to time and using state equations (2) we have:

$$\dot{V} = v\left(e_x \cos(e_\phi) + e_y \operatorname{sen}(e_\phi)\right) + K_\phi^2 e_\phi \omega$$

Now we impose (as our control law) this derivative $\dot{V}$ to be negative semi-definite[9] to ensure convergence, having:

$$\begin{aligned}\dot{V} &= \left(e_x \cos(e_\phi) + e_y \operatorname{sen}(e_\phi)\right)v + \left(K_\phi e_\phi\right)K_\phi \omega = \\ &= -\left[\tau_{xy}^{-1}\left(e_x \cos(e_\phi) + e_y \operatorname{sen}(e_\phi)\right)^2 + \tau_\phi^{-1} K_\phi^2 e_\phi^2\right]\end{aligned} \tag{11}$$

Equation (11) represents a line in the plane of the normalized control inputs (v, $K_\phi\omega$). If we choose, for convenience, $b_\phi=K_\phi$ in the motion exigency (9), this equation will represent a circle. The intersection of circle (9) and line (11) will give us the requested values for v and ω (if it does not exist, circle's nearest point to the line is chosen). Asymptotic convergence of the proposed law is demonstrated in [6] (it can be obtained, as usual for these kind of laws in mobile robots, through Barbalat's lemma [9]). The intersection of circle and line (that is, the control law) reduces to a first order differential equation when $e_\phi \to 0$ or $e_x \to 0$. In these cases, the parameters τ_{xy}, τ_ϕ play the role of *time constant.*

Path following evaluation. Even when asymptotic convergence is ensured, simulation is always a good way to verify and observe the control behavior. The proposed reference paths where a smooth PF control law must be evaluated have to be valid (see section 2). Hence a car-like robot or our (2,0) robot can not go through a piecewise path including curvature discontinuities (e.g. a straight line plus a circle); nevertheless the pieces should be linked by the path planning to ensure curvature continuity at least (for example through the addition of clothoids or similar curves). As we have shown analytically the generality of our projection function, then every path complying with the curvature continuity is identical for evaluating our PF. We have selected two very interesting examples: 1) approaching to a straight line, where typical convergence is showed, 2) converging to a zero-radius turn, where our PF shows its generality. The values for constant parameters have been chosen to ensure a smooth convergence, as $\tau_{xy}=0.5s$, $\tau_\phi=0.5s$. The whole motion is $K_{mov}=50cm/s$, $K_{mot}=0.5\ m/s$. The constants $b_\phi=K_\phi$ are $0.25m$.

[8] Mainly the non-severe condition is due to nonholomic nature of the system.

[9] Note that $\dot{V}$ can never be negative definite because the existence of the nonholonomic constraint, see [Díaz97a] for a demonstration.

In the first case we have selected big initial errors to prove the good convergence of the method in presence of extreme conditions (see *fig. 4*) and to compare it with TT. In PF the error e_x must always be zero according to the projecting function (7) and the solid line for the error e_y gives us the real robot trajectory. We have used for TT the control law of [8] tuning its constants so it behaves in a similar fashion for small errors ($K_x=20s^{-1}$; $Ky=6.0cm^{-1}$; $K_\theta=3.2cm^{-1}$) (its real trajectory is the dashed line). Note that in PF the robot begins the tracking in reverse direction in the first transient, in order to reduce the errors faster, and parameter r decreases too. In the end this will imply that PF method will reach less distance in the desired path. This case will never happen in a pure TT, where $r=t$, and the reference robots "pulls" the real one and it will advance the same as the reference trajectory. This is a well-known advantage of PF frequently commented in the literature [11][3], that reduces oscillations in the end. Moreover input commands in PF are limited by our motion exigency while in TT they are not. If they were in TT, its response would be even poorer.

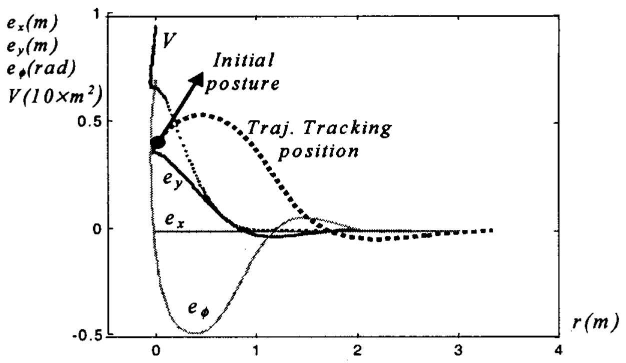

fig. 4: PF to a line (X axis) under big initial errors.

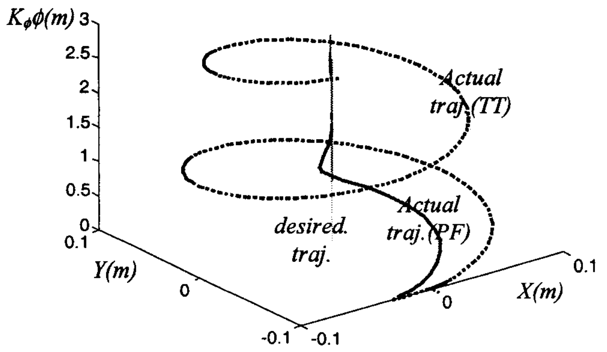

fig. 5: PF to a zero-radius turn($e_x(0)$=-0.03m, $e_y(0)$=-0.1m).

In the second example the desired trajectory is the vertical axis, because reference robot turns around its point P_o (see *fig. 5*). The real robot movement would be given by the projection on plane *XY*. As in the previous case, although initial errors were big, PF chooses the nearest point on the desired trajectory, i.e. that with zero e_ϕ (according to projecting function (7)). Note that in PF input controls v, ω are split in the most convenient form to get a fast convergence. In TT convergence is slower because constants were tuned for the tracking of a line. If constants were tuned for this last case then convergence to a line would be slower.

5 Conclusions.

We present a technique to construct path following in mobile robots (that has been shown by several studies to be more advantageous than trajectory tracking). It consists of two steps: choosing a "projecting function" to relate actual posture to desired path as a function of errors and intrinsic descriptor parameter, and imposing a "motion exigency" to ensure robot advances on the path. These steps have been obtained based on the general path following characteristics that we have previously extracted. We also stated the conditions that both steps must satisfy to be coherent. Thereafter we corroborate our proposition with a (2,0)-robot (according to the definition of [2]) that can make zero-radius turns, a case that has never been solved, as far as we know, using path following. Finally we present simulation results under big initial errors to exhibit the good and fast convergence of our path following approach.

Acknowledgements

This work has been supported by IASS agreement "Segundo Convenio de Colab. IASS-Univ. de Sevilla" and by CICYT project TER96-2056-C02-01. Authors will also thank Prof. C. Samson (INRIA, France) for his kind mailing of research reports and the reviewers for their worthwhile comments.

References.

[1] R.W. Brockett. "Asymptotic Stability and Feedback Stabilization". in Differential Geom. Contr. Theory, Birkhauser, pp. 181-208, 1983.

[2] G. Campion, G. Bastin, d'Andrea-Novel. "Structural Propierties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots". In IEEE Trans. Rob. And Autom., V. 12, No 1, Feb. 1996.

[3] C. Canudas de Wit, H. Khennouf, C. Samson and O.J. Sordalen. "Nonlinear Control design for Mobile Robots". In "Recent Trends in Mobile Robots". Edited by Yuan F. Zheng. Ed. World Scientific Series in Robotics and Automated Systems. 1993.

[4] A. Civit-Balcells, F. Díaz del Río, J. Sevillano, G. Jiménez "SIRIUS: A Low Cost High Performance Computerized Wheelchair". Proc. of the Int. Workshop on Medical Robots, pp. 23-30. Vienna. October 1996.

[5] A. Civit-Balcells, F. Díaz del Río, G. Jiménez, J.L. Sevillano. "A Proposal For A Low Cost Advanced Wheelchair Architecture". The 4th European Conference for the Advanc. Techn. AAATE Conference 1997. ISBN 4274901831C3047 (Ohmsha). ISBN 9051993617 (IOS Press). Oct 1997. Thessaloniki. Greece.

[6] F. Díaz del Río. "Analysis and Evaluation of Mobile Robot Control : Application to Electric Wheelchairs"(In Spanish). Ph. D. Thesis. University of Seville, (Spain), 1997.

[7] F. Díaz del Río, A. Civit, G. Jiménez, J.L. Sevillano. "Path Tracking In The SIRIUS Wheelchair". Same as [Civit97].

[8] Y. Kanayama et al. "A Stable Tracking Control Method for an Autonomous Mobile Robot". Proc. 1990 IEEE Int. Conf. Robotics and Autom, Cincinatti, Ohio, pp. 384-389.

[9] A. Micaelli, C. Samson. "Trajectory Tracking For Unicycle-Type And Two-Steering-Wheels Mobile Robots". Rapport de Recherche N. 2097. Institut National de Recherche en Informatique et en Automatique. 1993.

[10] Richard S. Millman. "Elements of Differential Geometry". Edit. Prentice Hall, Englewood Cliffs, N.J. 1977.

[11] N. Sarkar, X. Yun, V. Kumar. "Control of Mechanical Systems with Rolling Constraints: Application to Dynamic Control of Mobile Robots". The Int. J. Rob. Research. V13.No.1 Feb1994.

[12] C. E. Thorpe. "Vision and Navigation : The Carnegie Mellon Navlab". Ed. by Thorpe. Kluwer Academ. Publ., 1990.