Error Adaptive Tracking for Mobile Robots

F. Díaz del Río, G. Jiménez, J. L. Sevillano, C. Amaya, A. Civit Balcells.
Escuela Técnica Superior de Ingeniería Informática
Universidad de Sevilla.
Av. Reina Mercedes s/n. 41012 Sevilla. SPAIN,
[fdiaz, gaji, sevi, claudio, civit]@atc.us.es

Abstract*— **In mobile robots it is usual that the desired trajectory is memorized or previously generated. When following a trajectory, there are several possibilities attending to the way in which the actual robot state can be related with the whole trajectory. One of them is the extension of the servosystem approach, usually called "trajectory tracking". This is the only possibility if we need strict temporal deterministic requirements. But if not, other possibilities appear. One of them is called "path following", where the path's point to track is the "nearest" (under several conditions) to the actual robot's position. In this paper we present another method suitable for non-deterministic systems, which we may call "error adaptive tracking", because the tracking pace adapts to the errors. Its benefits and advantages are identified. Afterwards, we determine how to construct this method and we apply it to the case of SIRIUS, an advanced wheelchair. Then a control law that ensures asymptotic stability is extracted using the second Lyapunov method and under the error adaptive tracking approach. Finally, we show the benefits of the new method, comparing it with the trajectory tracking approach.

Index Terms-- **mobile robots, nonholonomic constraints, chained systems, path following, trajectory tracking.**

I. Introduction

In robotics it is usual to distinguish between two control problems:

- The stabilization problem, i.e. how to get the system to a fixed point (in its state space).
- The tracking problem, i.e. how to follow a desired trajectory or path.

Both problems have been studied profusely in robotics and they can present very different characteristics. Special cases of robots are those that have nonholonomic constraints, e.g. mobile robots (except for omnidirectional mobile robots), where the number of state coordinates is higher than the number of degrees of freedom DOF. Furthermore, mobile robots are intrinsically nonlinear systems, because of their kinematic model. Due to this, the convergence of a non-omnidirectional mobile robot to a fixed posture (the stabilization problem), can not be achieved through a smooth feedback stabilization control law (a direct result of Brockett's theorem [1]). On the other hand, in mobile robots it is usual that the trajectory or path is memorized, and the point stabilization problem is very different to the path's convergence problem, which is considered here.

There is no doubt that the applications of mobile robots will be large in the next few years, especially in fields such as intelligent transportation systems (ITS), explorer vehicles, and personal or assistant robots. Consequently, in the last decade there has been a great interest in finding controllers for these robots.

Many researchers have studied various tracking methods when the desired path is memorized or previously generated. In the field of mobile robots two main tracking methods have been proposed. In a first group we find those that consider time explicitly in the tracking [2,3,4] (usually called "*trajectory tracking*" (TT)), and try to approach the robot to a moving objective point. It is similar to servosystems, and it is guaranteed that the system will converge to the desired point in a deterministic time. In a second one, we find those that do not consider timing requirements and try to converge to a path [4,5,6,7,8,9] (usually called "*path following*"(PF)). Moreover, we can find several excellent compendia of both methods in some reports or books [2,10]. It has been shown that PF is more suitable for many situations in which time is not a critical parameter (see section II). Unfortunately no PF have been designed that can be applicable for all possible paths (to the authors' knowledge).

In this paper we propose a new tracking policy that tries to overcome the difficulties of the other methods. We have named it *error adaptive tracking* (EAT), because the tracking adapts to the system errors. Its design is similar to that of TT but it is intended for path recovering without strict timing requirements. We will show that it retains the exposed advantages of PF and it can be applied to all sets of paths.

The need for this new tracking method comes to us from the development of advanced wheelchairs. Our research group has been interested during the last years in the improvement of electrical wheelchairs [9,13], which incorporate advanced features. Playing back previously recorded trajectories is considered a very helpful aid. This avoids the user the difficult maneuvering of reverse driving, and may be very useful in small areas like bathrooms. In our group we have developed SIRIUS, an advanced wheelchair prototype that includes path recovery of trajectories, detection and avoidance of obstacles through simple sensors like sonars, intelligent user interfaces with shared control, etc. [9,13]. It is important to mention that in SIRIUS we must contemplate all the possible desired paths that can be made by the user (usually driving his/her joystick), including zero-radius turns.

Once the tracking method has been selected, a convergence law must be found. In mobile robots, the most frequent contributions are those based in the Lyapunov direct method, which we will also use it in this paper. Finally, as we are interested in convergence to a path, we will suppose in this work that the desired trajectory has no end.

In the next sections we will try to analyze and expose the *error adaptive tracking*. In section II we formulate the different tracking methods. In section III we define our robot model, construct the new tracking technique, and apply an asymptotically stable control law. Evaluation is found in section IV and finally we expose the conclusions.

* This work was supported in part by the Spanish "Ministerio de Ciencia y Tecnología" through project TIC-2000-0087-P4-03.

II. TRACKING METHODS' FORMULATION

A *memorized*, reference or desired path (or merely *path*) can be described by a single descriptor parameter [14], namely r, and it can be expressed as a vector of state coordinates $\mathbf{q}_{des}(r)$. As a result, tracking progress can be identified with the progress of r. Although the parameter may be time, in the case where time dependence is not relevant, many others are possible. For example, in differential geometry the natural arc parameter [14], which makes the linear speed equal to 1, is generally preferred. Independently of the selected parameter we can classify the tracking of the path according to the way in which we impose or design (when possible) the progress of parameter r.

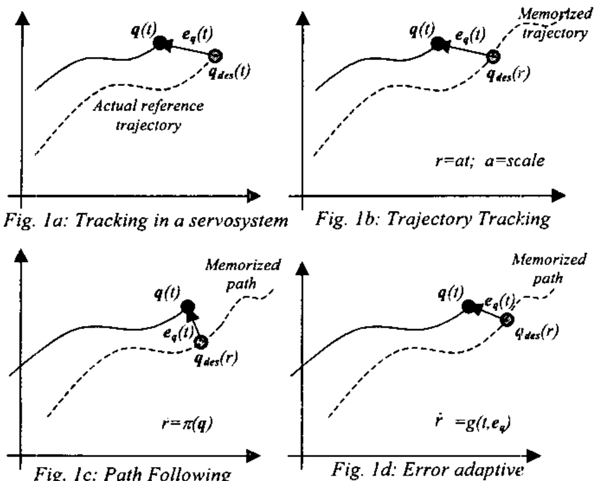

Fig. 1a: Tracking in a servosystem *Fig. 1b: Trajectory Tracking*

Fig. 1c: Path Following *Fig. 1d: Error adaptive tracking*

Fig. 1: Tracking methods regarding to descriptor parameter.

In servosystems we track a mobile system or target at the time it moves. In Fig. 1*a* we show this case. The state coordinates are called $\mathbf{q}(t)$, the desired coordinates $\mathbf{q}_{des}(t)$, and the error coordinates are defined as $\mathbf{e}_q(t)=\mathbf{q}(t)-\mathbf{q}_{des}(t)$. In servosystems where time is critical, this is the only possibility we have. Nonetheless, when we try to track a memorized path, the tracking methodology can be designed openly, as we know a priori the whole trajectory. Of course the classical servosystem tracking can be done just by identifying the parameter r associated to the path with time, that is $r(t)=t$. This is usually called *trajectory tracking* (TT). Furthermore, TT can be extended in a more general assumption than simple servosystems (see Fig. 1*b*): let the parameter r be an increasing function of time $r=r(t)$, for example $r=at$, $a>0$. Therefore, error coordinates are $\mathbf{e}_q(t)=\mathbf{q}(t)-\mathbf{q}_{des}(r(t))$. An asymptotically stable control law guarantees that the system will converge to a point in the desired path in a deterministic time, except for the inherent perturbations that it may suffer.

The best-established alternative in literature is *path following*, which is very suitable when time is not a critical parameter. This is based in some relation between actual system's state $\mathbf{q}(t)$ and the whole memorized path. This relation or *projection* will give us the desired point $\mathbf{q}_{des}(r)$, i.e. the descriptor parameter r as a function of the actual position and the path: $r=\pi(\mathbf{q})$, where π is a some kind of projection of actual position to the path. Then the real system must try to follow this point instead of the one given by the other approach (see Fig. 1*c*). For example, the desired point is usually selected to be the "closest point on the path" to the actual robot's position [14]. The term "closest" means the path's point that makes certain distance criterion minimum. The error coordinates are also $\mathbf{e}_q(t)=\mathbf{q}(t)-\mathbf{q}_{des}(r)$. Of course using this approach, it is not guaranteed that the system will reach a point of the desired trajectory in a deterministic time. But the main problem with PF is that projection uniqueness has not been guarantied yet.

It has been shown that PF is more suitable for many situations. This can be understood if we consider the following example: if perturbations force the system to be at rest, the desired point for TT will move unavoidably. This means that errors will grow up to some value that may introduce instability. On the other hand, if PF were used, the desired point will be the same in spite of these perturbations, because the path's shape and the real robot state remain the same.

Having in mind the difficulties and the goals explained before, we present here the *error adaptive tracking* (EAT). Note that in a pure TT, r is determined exclusively as a function of time t. That is, desired point selection does not contemplate actual robot's posture, because the intention is to have a deterministic tracking. But when time is not so critical, we can "design" the variation of r, instead of stating the parameter r itself. And we can regard robot state, i.e. taking errors into account. Thus, contrasting the TT's rigid variation of r ($\dot{r}=1$ or in general $\dot{r}=f(t)$), we propose $\dot{r}=g(\mathbf{e}_q)$ (see Fig. 1*d*), where $g(\mathbf{e}_q)$ is a "convenient" function of the errors. Here "convenient" is referred to the designer objectives, but also it means that tracking is done correctly and the previously explained advantages of PF are preserved. That is, the function $g(\mathbf{e}_q)$ should fulfil (see Fig. 2):

- If errors are small, $g(\mathbf{e}_q)$ should tend to 1, so the tracking resembles TT, and near deterministic following is done. The real and the reference robot advance at the same pace.
- If errors are large, the reference robot should "wait for" the actual robot. That is, $g(\mathbf{e}_q)$ should be small until a good convergence is reached. Here the tracking rate must not be far from that of PF. In fact when errors are large in PF, we expect that the projecting point on the desired path varies very little, that is $\dot{r}$ tends to zero. Of course in this situation, no deterministic following is expected.

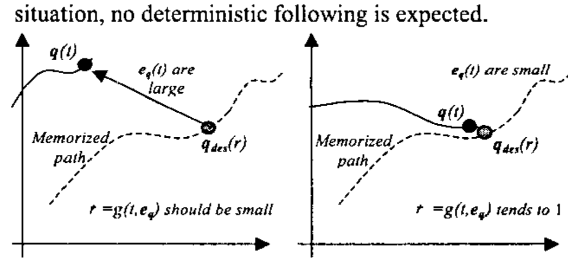

Fig. 2: Rules on the variation of descriptor parameter

The above conditions imply that the tracking will not preserve time determinism. Otherwise, if our design needs some aspects of time determinism, the variation of r can be extended to include the "inaccuracy in the deterministic tracking". That is, the difference between the descriptor parameter r (that indicates the target of our control design $\mathbf{q}_{des}(r)$) and the time t (that indicates the target of the timing requirements $\mathbf{q}_{des}(t)$). To sum up, a more general function for $\dot{r}$ would be $g(t,\mathbf{e}_q)$.

Of course many possible functions $g(t,\mathbf{e}_q)$ can be designed attending to the characteristics and purposes of each system but, in this work, we would explore mainly two:

- g is only a function of errors $\mathbf{e}_q$ and $|g(\mathbf{e}_q)|\leq 1\ \forall \mathbf{e}_q$. With these conditions the tracking can not be deterministic and reminds that of PF.

- $g(t,\mathbf{e_q})$ is a combined function of errors $\mathbf{e_q}$ and the difference between parameter r and time t. This strategy is intended to get a deterministic tracking at the end, although if errors are large, determinism is not applied for a while.

The formulation advantage of EAT is that it obviously can be applied to all sets of paths.

In order to fix ideas, we first analyze a very simple system with two state coordinates $\mathbf{x}=\{x_1, x_2\}$, and whose state equations are:

$$\dot{x}_1 = u_1$$
$$\dot{x}_2 = u_2$$

The goal is to follow a reference path $\mathbf{x_{des}}=\{x_{1des}(r), x_{2des}(r)\}$ made by a virtual robot, which still must fulfil:

$$x'_{1,des} = u_{1,des}(r)$$
$$x'_{2,des} = u_{2,des}(r)$$

where (') holds for derivative with respect to r. Applying chain's law: $u_{j,des}(t)= \dot{r}\ u_{j,des}(r),\ j=1,2$. And then we have $\dot{x}_{j,des} = u_{j,des}(t)\ ,j=1,2$.

As we are interested in the tracking we define the system's errors as the *difference*:

$e_1(t) = x_1(t) - x_{1des}(r(t))\ ;\qquad \dot{e}_1(t)= u_1(t) - \dot{r}\ u_{1des}(r);$

$e_2(t) = x_2(t) - x_{2des}(r(t))\ ;\qquad \dot{e}_2(t) = u_2(t) - \dot{r}\ u_{2des}(r);$

We have expressed explicitly the dependence of these variables because the design methodology must now choose between different forms of $r(t)$.

In a servosystem $r=t$, $\dot{r}=1$ and a very simple convergent control law can be: $u_j(t) = u_{j,des}(t)-K_je_j = \dot{r}\ u_{j,des}(r)-K_je_j$, $j=1,2$, where $K_j>0$. If TT is chosen, the dependence $r=r(t)$ will only add a scale on the election of r, supposing that $\dot{r}>0\ \forall t$. Note that in both cases, the condition $\dot{r}>0\ \forall t$ implies that the reference point $\{x_{1des}(r(t)), x_{2des}(r(t))\}$ will unavoidably advance in spite of real robot motion.

But if we could apply the PF method, the situation would be very different. First we must look for the most suitable projection $r=\pi(\mathbf{x})$, in order to select the descriptor parameter r and hence, the reference point $\{x_{1des}(r), x_{2des}(r)\}$ at any instant. The most obvious choice is the projection that selects the closest point on the path to the robot's position, i.e. r that makes $\sum_{i=1}^{2} e_i^2$ minimum. For example, if the desired path is the line $\{x_{1des}(r)=sr,\ x_{2des}(r)=0,\ s>0\}$ then the closest point will be $\mathbf{x_{des}}=\{x_1,\ 0\}$, and the projection: $r=x_1/s$. Differentiation of projection give us the rate for $r(t)$: $\dot{r}=u_1/s$. Then the tracking will progress only if r increases, i.e. if $u_1>0$ (or $u_1<0$ if the tracking is to be done in reverse order). Then we must impose some condition or "motion exigency"[4] to ensure that the (real and virtual) robots advance, and hence to guarantee that the tracking is being done. The simplest motion exigency is of course $u_1=constant>0$, but other more sophisticated can be proposed. If the trivial motion exigency is preferred (and enough for our control purposes) then $\dot{r}=u_1=constant=K_{mot}>0$, and the previous control law still succeeds (only for input 2):

$$u_2(t) = K_{mot}\, u_{2,des}(r) - K_2\, e_2 = -K_2\, e_2.$$

This is obviously a particular case. If a generic trajectory had to be tracked (as in our application case in section III), then a more general motion exigency is preferable, for example $u_1^2+u_2^2=constant>0$ [4]. This has the additional advantage of maintaining inputs within certain values, avoiding an excessive increase of inputs (which may introduce instability). The imposition of this last condition on the inputs implies that the control law must now be a relation between u_1 and u_2. It can be obtained easily for example if we impose that the derivative of the Lyapunov function $V=e_1^2+e_2^2$ is equal to $-K_V(e_1^2+e_2^2)$, $K_V>0$. Note that in PF it is not predictable when the robot will reach any path's point in the general case. On the other hand PF is in general more stable when perturbations make errors increase.

However, problems arise when we can not guarantee projection uniqueness. For example, if the path were a circle and previous projection want to be applied, then projection uniqueness is broken when the robot approaches the circle center. So it is not fully applicable for all kind of paths, and projection uniqueness should be carefully analyzed (see [4]).

For all stated above, we propose the new tracking method. In EAT we can *design* the most appropriate tracking rate, i.e. an equation for $\dot{r}$ as a function of the errors. As explained before a very simple and interesting possibility would be:

$\dot{r}=g(\mathbf{e_q})=exp\,(-K_r\,(e_1^2+e_2^2))\ ;\ K_r>0$ is the scale factor.

This proposal fulfils the rules of Fig. 2, and tries that errors do not increase greatly. Therefore, it conserves the PF advantages, while avoiding the projection difficulties.

Besides, for this trivial system the previous TT control law still works:

$u_j(t)=exp(-K_r(e_1^2+e_2^2))\ u_{j,des}(r)-K_je_j\ ;\ j=1,2,\ K_j>0.$

The main difference is that when errors are large enough, the convergence resembles that of the stabilization problem. Nevertheless, this is not the case for all systems. We will see in the next section that several assumptions, which are more complex, have to be made for mobile robots.

III. APPLICATION TO MOBILE ROBOTS AND EAT SELECTION.

One of the typical topologies for electric wheelchairs is the so-called unicycle. They include driver motors at each rear wheel, that can turn independently forward or backward. Different speeds at each rear wheel cause the turn of the chair. Let's consider the mobile robot shown in Fig. 3 (whose dimensions are those of SIRIUS) and let $\mathbf{q}=(X,\ Y,\ \phi)^t$ be its state coordinates, which represent the coordinates $(X,\ Y)$ of a certain point $\boldsymbol{P_o}$ (typically the midpoint between rear wheels) in the basis of the fixed frame $(\mathfrak{R})=(\mathbf{O};\ \mathbf{i},\ \mathbf{j})$ and the orientation ϕ of the robot with respect to the fixed frame. Let $\mathbf{u}=(v,\ \omega)^t$ be the pair of input variables which are the linear velocity of point $\mathbf{P_o}$ and the angular velocity of the robot, respectively. The unicycle robot has three state variables but only two degrees of freedom, as a result of the nonholonomic constraint. We assume that the wheels are nondeformable and that they are moving on a horizontal plane without slip in order to hold the constraint.

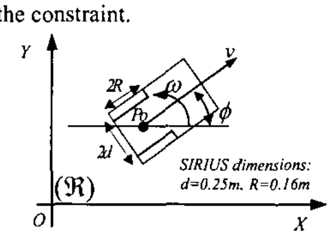

Fig. 3: Unicycle robot model.

For these vector variables the kinematic model of the unicycle robot can be expressed by the equations:

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

In this paper we consider the following state and control transformation of (1) [10]:

$$\begin{aligned} x_1 &= \phi \\ x_2 &= X\cos\phi + Y\sin\phi \\ x_3 &= X\sin\phi - Y\cos\phi \\ v_1 &= \omega \\ v_2 &= v - (X\sin\phi - Y\cos\phi)\omega \end{aligned} \qquad (2)$$

Then, it can be verified that the transformed kinematic takes the form of a very simple chained system:

$$\dot{\mathbf{x}} = \mathbf{B(x)v}\,;\, \mathbf{B(x)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ x_2 & 0 \end{bmatrix};\ \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix};\ \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \qquad (3)$$

A reference or desired path to be followed can be expressed as a vector of desired state coordinates expressed as a function of the descriptor parameter r: $\mathbf{x}_{des}(r)=(x_{1,des}(r), x_{2,des}(r), x_{3,des}(r))^t$. Let $\mathbf{v}_{des}(r)=(v_{1,des}(r), v_{2,des}(r))^t$ be the desired inputs. Besides, applying the chain law we can declare that $\mathbf{v}_{des}(t) = \dot{r}\mathbf{v}_{des}(r)$.

To study the tracking of a path $\mathbf{x}_{des}(r)$ let us define the error coordinates $\mathbf{e}_q=(e_1, e_2, e_3)^t$, that are given by the transformation:

$$\mathbf{e} = \mathbf{T}(\mathbf{x}_{des})(\mathbf{x} - \mathbf{x}_{des}) \qquad (4)$$

We can choose matrix $\mathbf{T}_{des}(\mathbf{x}_{des})$ so that the new state equations on the errors maintain the same dependence on inputs $\mathbf{v}$. That is, if we differentiate (4):

$$\dot{\mathbf{e}} = \dot{\mathbf{T}}\mathbf{T}^{-1}\mathbf{e} + \mathbf{T}(\mathbf{x}_{des})\mathbf{B}(\mathbf{x})\mathbf{v} - \mathbf{T}(\mathbf{x}_{des})\mathbf{B}(\mathbf{x}_{des})\mathbf{v}_{des}$$

And we demand that: $\mathbf{B(e)} = \mathbf{T}(\mathbf{x}_{des})\mathbf{B(x)}$,we get to:

$$\dot{\mathbf{e}} = \dot{\mathbf{T}}\mathbf{T}^{-1}(\mathbf{x}_{des})\mathbf{e} + \mathbf{B(e)v} - \mathbf{B(0)v}_{des}\,;\ \mathbf{T}(\mathbf{x}_{des}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_{2,des} & 0 & 1 \end{pmatrix}$$

The final state equations are:

$$\dot{\mathbf{e}} = \mathbf{B(e)v} - \mathbf{A(e)v}_{des}\,;\ \mathbf{B(e)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ e_2 & 0 \end{bmatrix};\ \mathbf{A(e)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & e_1 \end{pmatrix} \qquad (5)$$

This form agrees with the intuition that error variables must grow with both real and desired posture's advancement.

With these relative coordinates, the nonholonomic constraint takes the form of:

$$\dot{e}_3 - e_2\dot{e}_1 = -v_{2,des}e_1 + v_{1,des}e_2 \qquad (6)$$

Now we can look for a convenient function $\dot{r}=g(t,\mathbf{e})$ to implement an *error adaptive tracking*. Therefore, we will obtain the advantages of the PF method, but without its problems.

Let us first analyze the most elementary case, where g does not depend on t. We have shown that if errors are small, then $g(e)$ should tend to 1, but if they were large, then $g(e)$ would be small. The previous simple idea $g(e)=\exp(-|\mathbf{e}|)$ is not valid now. The nonholonomic constraint of the unicycle prohibits this g (if we want to look for a Lyapunov-based control law). It can be seen that if a Lyapunov function V is the sum of positive definite functions of the relative errors $\mathbf{e}$, then the derivative of V cannot be negative definite, but only a negative semidefinite function. This is a direct consequence of the nonholonomic constraint (6): if (e_1, e_2) are null at a time, then $\dot{e}_3$ must also be null. Therefore e_3 cannot decrease at this moment, and hence V can decrease either. Therefore we can not impose $\dot{V}$ to be negative definite, but only negative semidefinite.

In this situation, any robot movement (satisfying the constraint) will increase e_1 or e_2, while e_3 remains constant, and we can observe that when $\dot{r}$ becomes lower, the convergence will be slower. This is due to the fact that the input v_2 can not exceed $v_{2,des}(t)=\dot{r}\, v_{2,des}(r)$ (to avoid e_2 being increased). To sum up, when $\dot{V}$ is zero, the decrement of V is "connected" or "attached" to the increment of $\dot{r}$, and then if $\dot{V}$ is null, $\dot{r}$ should be maximum, to get a fast convergence. Moreover if the control law will pursuit $\dot{V}$ to be the "closest" to $-KV$, $K>0$, then $g(e)=\exp(K_V\dot{V})$, $K_V>0$ can be a plausible solution. With "closest" we mean that $\dot{V}$ can not be $-KV$ because of the constraint, although it was desirable to get exponential convergence. Considering this, we can design a control law based on the following theorem.

Theorem: For unicycle robots and the Lyapunov function: $V = \frac{1}{2}\sum_{i=1}^{3} e_i^2$, then $\mathbf{e=0}$ is an asymptotically stable equilibrium point, and $\mathbf{v} \to \mathbf{v}_{des}(t) = \dot{r}\mathbf{v}_{des}(r)$, if:

- $\mathbf{v}_{des}(r) \neq \mathbf{0}$.
- $\dot{r}=g(e)=\exp(K_V\dot{V})$, $K_V>0$
- The following control is imposed:

  $v_1 = v_{1,des} - K_1^2 e_1 + a_1$, $K_1>0$

  $v_2 = v_{2,des} - K_2^2 e_2 + a_2$, $K_2>0$

  where the feedforward terms a_1 and a_2 are:

  $a_1 = e_3\, v_{2,des};\ a_2 = -e_3\, v_{1,des} + K_1^2 e_3 e_1 - e_3^2\, v_{2,des}$

Proof: Differentiating V and using state equations and control law, we get to:

$$\dot{V} = -\sum_{i=1}^{2} K_i^2 e_i^2 = -K_1 e_1^2 + K_2 e_2^2$$

Then V is non-increasing, so $\dot{V} \to 0$, and V converges to some limit: $V \to V_{lim} \geq 0$. Therefore by Barbalat's lemma [16] $e_j \to 0, \dot{e}_j \to 0; j=1,2$, and then $e_3 \to e_{3,lim} < \infty$. At this limit, $\dot{r}=1$, and the control law is:

$v_1 = v_{1,des}(r) + e_3 v_{2,des}(r)$

$v_2 = v_{2,des}(r) - e_3\, v_{1,des}(r) - e_3^2\, v_{2,des}(r)$

And the state equations tend to:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} v_1 - \dot{r}v_{1,des}(r) \\ v_2 - \dot{r}v_{2,des}(r) \\ 0 \end{pmatrix} = \begin{pmatrix} e_{3,lim}v_{2,des}(r) \\ -e_{3,lim}v_{1,des}(r) - e_{3,lim}^2 v_{2,des}(r) \\ 0 \end{pmatrix}$$

Therefore, using the first state equation we have that $e_{3,lim}v_{2,des}(r)=0$; and then by the second one: $e_{3,lim}\, v_{1,des}(r)$.

Squaring and adding previous equations we get to:

$$0 = e_{3,lim}^2\left(v_{1,des}^2(r) + v_{2,des}^2(r)\right)$$

that implies (by the first hypothesis) that $e_{3,lim}=0$.

QED

Remark 1: Note that if $g(e)$ tended to zero in the case $e_j \to 0, \dot{e}_j \to 0; j=1,2,\ e_3 \to e_{3,lim} < \infty$, then $\mathbf{v}$ also would tend to zero and convergence would be very slow.

Remark 2: Actually there is not special exigencies on $g(e)$; only it is convenient to tend to one in the case $\dot{V} \to 0$ and to zero when $\dot{V} \to -\infty$. In fact we will use $g(e)=\exp(-K_V\sqrt{-\dot{V}})$, $K_V>0$ in the next section to achieve an adequate convergence.

The results of this proposal will be shown in the next section. Now we will study the case where g has also dependence on t. As mentioned before, we concentrate on a function $g(t,e)$ that is a combined function of the errors and the difference $(t\text{-}r)$, in order to get a "relaxed" deterministic tracking. The dependence of g on errors could be the same as before, because here previous reasons apply also when $r=t$. The intention for introducing the dependence on t is to permit that r remains "at rest" when the robot is far from the path (in spite of the increase of $r\text{-}t$). In a second case, when robot "recuperates" and approaches the path, the difference $r\text{-}t$ should be reduced. Therefore $g(t,e)$ can not be bounded by 1, because in the second case r must approach t. Moreover at the origin $(t\text{-}r=0,\ e=0)$, g must be 1. Thus we might think of a first proposal for this more complex EAT, e.g.:

$$g(t,e)=\exp(\dot{V})\,(1+K_{tr}\arctan(t-r)) \qquad (7)$$

where $K_{tr}>0$ is a scale factor that indicates how fast the convergence of r to t is. Here $g(t,e)$ is upper bounded by $1+K_{tr}\frac{\pi}{2}$, and lower bounded by $1-K_{tr}\frac{\pi}{2}$ or 0.

We assume that the previous choice can not be optimal at all for some applications and more sophisticated g can be tried in future work, but this simple alternative is sufficient for a mobile robot like SIRIUS as a first proposal.

In addition, the proposed function $g(t,e)$ fulfils remark 2 of the previous theorem: it tends to zero when $\dot{V}\to-\infty$, and in the case $\dot{V}\to 0$ it does not tend to one if and only if r is much greater than t (i.e. the robot is in advance with respect to time). But in this last case, the robot will simply be quiet until the time has approached r, and at that moment the convergence will be rapid (however note that this situation is not very usual). Finally if $K_{tr}>2/\pi$, the variation of r can be null or even negative when robot is in advance with respect to the desired point. This situation does not make sense when a wheelchair is required to track a path and it will be avoided in this paper (we will choose exactly $K_{tr}=2/\pi$, so the chair will wait for $\mathbf{x}_{des}(t)$ if necessary).

V. SIMULATION RESULTS.

In this section we show the behavior of our system in several situations. For our real system SIRIUS, the inertia load driven by the motors is important even when gearbox ratio is high (31:1). Due to this it is very important to consider the case where the motors have a response delay (i.e., the real kinematic inputs are delayed with respect to the commanded inputs). In spite of this, we will show that the EAT method ensures that errors will not grow too much when motor response is slow.

Even when asymptotic convergence is ensured, simulation is always a good way to verify and observe the control behavior. This behavior is primarily significant when errors are large, because, if they were small, simulated system's behavior will be similar to that of an exponential convergent system.

We first analyze two contrasting desired paths: a straight line (case 1; $v_{1,des}(r)=0$, $v_{2,des}(r)=1m/s$) and a zero-radius turn (case 2; $v_{1,des}(r)=1rad/s$, $v_{2,des}(r)=0$). Generality of the EAT method is confirmed by these examples. We present different initial conditions (with considerable initial errors), so that the robot has to cope with different situations. The errors for the two cases are: case 1, $\mathbf{e}=(-0.5rad,\ 0.5m,\ 0.5m)$; case 2, $\mathbf{e}=(0,\ 0,\ 0.6m)$.

The values for constant parameters have been tuned to ensure a smooth convergence when errors are small and a fast one when they are large: $K_1^2=0.7\ s^{-1}$; $K_2^2=1.2\ s^{-1}$. Also $K_V=5s^{1/2}m^{-1}$ has been chosen to achieve a satisfactory variation of r when errors are large.

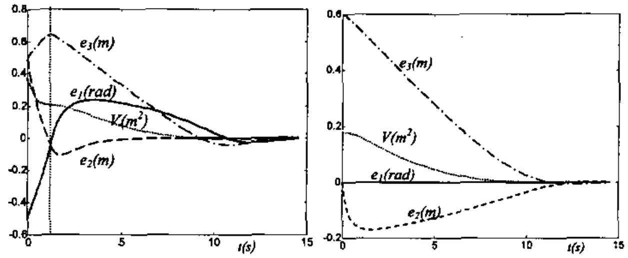

Fig. 4: EAT behavior for case 1 (left) and case 2 (right).

Obviously when errors are small convergence to the path is smooth and fast. However, in spite of the presence of extreme conditions, good convergence of EAT method is observed, as seen in Fig. 4. Note that at the moment in which errors e_1 and e_2 approach to zero (vertical dashed line on case 1), V and e_3 do not decrease as explained above (they have an inflexion point and a maximum, respectively).

In Fig. 5 we can see how $\dot{r}$ evolves for each case: in case 1 $\dot{r}$ begins with a value near to zero because of the big initial errors, and then it "recovers" until it reaches the stationary value of 1.On the other hand, in case 2 $\dot{r}$ begins with a value near to one because e_1 and e_2 are zero, afterwards it decreases (e_2 must grow so that e_3 is allowed to decrease), and finally it reaches the stationary value of 1. Of course the differences between $\dot{r}$ and 1 make r less than t. This will be avoided with $g(t,e)$.

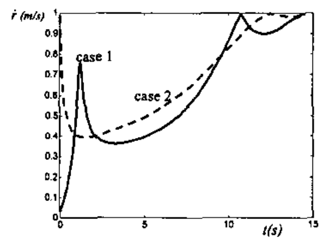

Fig. 5: Variation of r for the two experiments.

In the simulated experiments presented before we have illustrated the nice behavior of EAT method for an ideal model. Similarly, the TT has such a good behavior with an ideal model. However, the above commented high inertia of SIRIUS' motors will not permit instantaneous speed changes of its wheels. We then must go further in the simulations and emulate the inertia of the motors, so that the real speeds change always smoothly. Although the tracking is not perfect for this real situation, our aim is to demonstrate that EAT behaves better than TT under these perturbations.

In the following experiment we emulate the motors inertia like a first order delay, with the equation:

$$\frac{dv}{dt}=\tau_j\,(v_{j,cont}-v_{j,real});\ j=1,2$$

where $\tau_j>0$ are time constants, $v_{j,cont}$ the inputs demanded by the control and $v_{j,real}$ the real ones. With time constants $\tau_1=\tau_2=0.5s$, we repeat the tracking of a line. In Fig. 6, the experiment is prolonged for 14.5s for TT (left figure) and EAT (right figure). The robot starts at the same position of previous case 1: $r=0$, $\mathbf{e}=(-0.5rad,\ 0.5m,\ 0.5m)$, $\mathbf{v}_{real}=(0,0)$.

Note that TT method deviates the system very faraway from the path and the robot oscillations are remarkable, because the desired posture $\mathbf{x_{des}}(t)$ advances continuously. On the other hand EAT method obliges $\mathbf{x_{des}}(t)$ to "wait" for the robot until the errors have reduced sufficiently, so the deviation is not so pronounced (compare with case 1 in Fig. 4). Moreover and due to its large errors, the real speeds demanded by TT are bigger than those of EAT, and bigger than usual for wheelchairs ($2m/s$). It is obvious that speed limitation will degrade even more TT response. This is a well-known advantage of PF [2,7], that the new EAT retains. Nonetheless, determinism has been lost when this pure EAT is applied: simulation gives a final value for parameter r of 7.9 m (instead of 14.5m, given by TT).

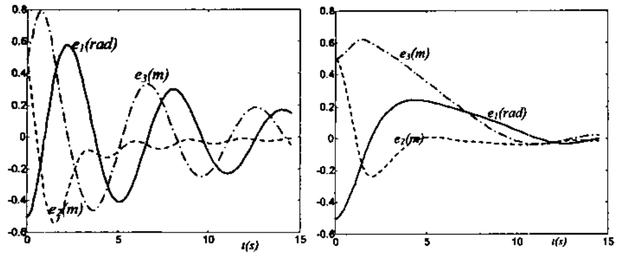

Fig. 6: Line following with response delay for TT (left) and EAT (right).

Although previous simulations are not rigorous demonstrations that EAT is more robust than TT against perturbations or unmodeled dynamics, we think that there is no doubt that the adaptive variation of r facilitates robustness. Furthermore it is important to observe that the qualitative behavior of EAT is similar to that of PF, i.e. $\dot{r}$ reduces in presence of large errors until system approaches the path. Of course both methods are constructed in a very different way, so it is not easy to compare them quantitatively.

Finally we present some results for the EAT that include time in $\dot{r}$. In this work, we concentrate on:

$$g(t, e)=\exp(-K_V\sqrt{-\dot{V}})\,(1+K_{tr}\arctan(t-r)) \tag{8}$$

with $K_V=5s^{1/2}m^{-1}$, $K_{tr}=2/\pi\ m/s$. In Fig. 7 we represent the first 14.5 seconds of the convergence to a straight line using the EAT method (8) (initial conditions are those of previous case 1). Note that, while this tracking is similar to that of previous EAT (same constants are used), from the evolution of $\dot{r}$ we discover that in the first transient $\mathbf{x_{des}}$ "waits" for the robot, while from t=6s approx. $\dot{r}$ exceeds 1, so the final r has almost reached 14.5 m (exactly 14.37m).

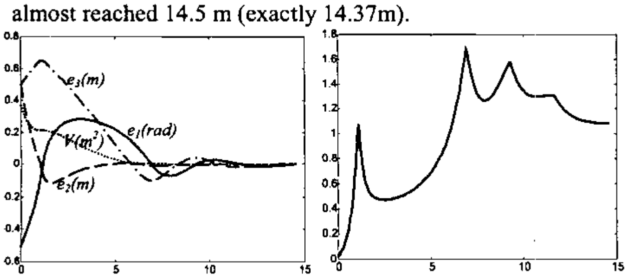

Fig. 7: EAT with $g(t,\mathbf{e})$: Errors (left), $\dot{r}$ evolution (right) for case 1.

The good qualities of this EAT are highlighted when the robot has reached the path (i.e it is on some $\mathbf{x_{des}}(r)$) but it is not upon the point $\mathbf{x_{des}}(t)$. Then $\mathbf{e}=\mathbf{x}(t)-\mathbf{x_{des}}(r)$ is null and input $\mathbf{u}$ is a scale of desired input $\mathbf{u_{des}}$. Hence only the feedforward terms $v_{1,des}$, $v_{2,des}$ of v and ω are not null.

We hope that both EAT methods can soon be implemented successfully in the SIRIUS wheelchair when recovering a path done by the user, in order to confirm simulations results.

VI. CONCLUSIONS.

We present a method for tracking memorized paths: the *error adaptive tracking* (EAT). It is based on the design of the variation of path's descriptor parameter, so that the tracking adapts to the system errors. Based upon this EAT methodology, a control law that ensures asymptotic stability is proposed for mobile robots, and it is compared with classical trajectory tracking. Its benefits are identified and shown through several simulations:

- It conserves most of the advantages of the path following method, while it avoids its main obstacle: the non-uniqueness of the selected path's point.
- It is valid for all possible trajectories.
- Its behavior under large errors or delayed response is much better than that of a trajectory tracking. Overall, simulations reveal that EAT facilitates robustness.
- A variety of EAT that includes time in the variation of path's parameter to preserve deterministic (non-strict) tracking, is also presented. It also behaves better than pure TT and similarly to previous EAT. Future research will investigate this method more deeply.

REFERENCES

1. Brockett, R. W. "Asymptotic stability and feedback stabilization". Differential Geometric Control Theory, Birkhauser, pp. 181-208, 1983.
2. Bloch, M., M. Reyhanoglu and N. H. McClamroch. Control and stabilization of nonholonomic dynamic systems. IEEE Trans. on Aut. Control, 37 (11), pp 1746-1757, 1992.
3. Lamiraux, F., S. Sekhavat, J.P. Laumond, Motion planning and control for Hilare pulling a trailer, IEEE Trans. on Robotics and Automation, Vol 15 No 4. August 1999.
4. Díaz del Río, F., G. Jiménez, J. L. Sevillano, S. Vicente, A. Civit Balcells. A path following control for unicycle robots. Journal of Robotic Systems 18(7), 325-342 (2001).
5. Díaz del Río, F., G. Jiménez, J. L. Sevillano, S. Vicente, A. Civit Balcells. A generalization of path following for mobile robots. Proc. of the IEEE Int. Conf. on Robotics and Aut, ICRA'99. Detroit. May 1999.
6. DeSantis. R. M. Path-tracking for a tractor-trailer-like robot. The International Journal Of Robotics Research. Vol 13. No.6 Dec. 1994.
7. Sarkar, N., X. Yun, V. Kumar. Control of mechanical systems with rolling constraints: application to dynamic control of mobile robots. The Int. J. Rob. Research. 13(1) 1994.
8. Sordalen, O.J. and C. Canudas de Wit. Exponential control law for a mobile robot: extension to path following. IEEE Trans. on Robotics And Autom., V. 9, N 6, Dec. 1993.
9. Samson, C. Control of chained systems: application to path following and time-varying point-stabilization of mobile robots, IEEE Trans. on Autom.Control, 40 (1995), pp. 64-77.
10. De Luca, A., G. Oriolo, and C. Samson. "Feedback control of a nonholonomic car-like robot". Robot Motion Plannning and Control, pages 171-253. Springer-Verlag, Berlin, 1998.
11. Canudas de Wit, C., Khennouf, C. Samson and O. J. Sordalen. Nonlinear control design for mobile robots. In Recent trends in mobile robots. Edited by Y. F. Zheng. World Scientific Series in Robotics and Autom. Systems. 1993.
12. Civit Balcells, A., F. Díaz del Río, G. Jiménez, J. L. Sevillano. A proposal for a low cost advanced wheelchair architecture. The 4th AAATE Conf. Oct 1997. Thessaloniki. Greece.
13. Díaz del Río, F. Analysis and evaluation of mobile robot control: application to electric wheelchairs (in Spanish). Ph. D. Thesis. University of Seville (Spain), 1997.
14. Manfredo do Carmo, P. Differential geometry of curves and surfaces. Prentice-Hall. 1976.
15. Encarnaçao P., A. Pascoal. "Combined trajectory tracking and path following for marine craft". Proceedings of the MED'01. Dubrovnik, Croatia. June 2001.
16. Sastry, S. Nonlinear systems: analysis, stability and Control. Springer-Verlag, New York. 1999.