Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics

R. Fierro* and F. L. Lewis
Automation and Robotics Research Institute
The University of Texas at Arlington
7300 Jack Newell Blvd. South
Forth Worth, TX 76118-7115
e-mail rfierro@arri.uta.edu

Received April 26, 1995; revised July 12, 1996
accepted September 20, 1996

A dynamical extension that makes possible the integration of a kinematic controller and a torque controller for nonholonomic mobile robots is presented. A combined kinematic/torque control law is developed using backstepping, and asymptotic stability is guaranteed by Lyapunov theory. Moreover, this control algorithm can be applied to the three basic nonholonomic navigation problems: tracking a reference trajectory, path following, and stabilization about a desired posture. The result is a general structure for controlling a mobile robot that can accommodate different control techniques, ranging from a conventional computed-torque controller, when all dynamics are known, to robust-adaptive controllers if this is not the case. A robust-adaptive controller based on neural networks (NNs) is proposed in this work. The NN controller can deal with unmodeled bounded disturbances and/or unstructured unmodeled dynamics in the vehicle. On-line NN weight tuning algorithms that do not require off-line learning yet guarantee small tracking errors and bounded control signals are utilized. *© 1997 John Wiley & Sons, Inc.*

*To whom correspondence should be addressed at present address.

Journal of Robotic Systems 14(3), 149–163 (1997)

CCC 0741-2223/97/030149-15

ここでは、ノンホロノミック·モービル·ロボットで使う運動コントローラーとトルク·コントローラを統合することを可能にする、力学的な拡張について説明する。バックステッピングを使って統合型運動／トルク制御則を開発し、近似的な安定性をLyapunov理論によって確保する。さらに、この制御アルゴリズムは、化に適用できる。モービル·ロボットを制御するための一般的な構造は、すべての力学が既にわかっている場合の従来型の計算トルク·コントローラから、そうでない場合の堅牢型適応制御まで、異なる制御法を組み合わせることができる。この研究においては、ニューラル·ネットワーク（NN）をベースにした堅牢型適応コントローラはを提案する。NNコントローラは、モデル化されていない領域化された障害、および車両の構造化とモデル化がされていない力学を扱うことができる。オンラインNN荷重調整アルゴリズムでは、オフラインでの学習を行わなくても十分小さいトラッキング誤差を保証し、領域化された制御信号を利用できる。

1. INTRODUCTION

A mobile robot is suitable for a variety of applications in unstructured environments where a high degree of autonomy is required. This desired autonomous or *intelligent* behavior has motivated an intensive research in the last decade. Much has been written about solving the problem of motion under nonholonomic constraints using the kinematic model of a mobile robot, but little about the problem of integration of the nonholonomic kinematic controller and the dynamics of the mobile robot. Moreover, the literature on robustness and control in presence of uncertainties in the dynamical model of such systems is sparse.[1] Some preliminary results of nonholonomic system with uncertainties are given in refs. 2 and 3.

The navigation problem may be divided into three basic problems:[4] tracking a reference trajectory, following a path, and point stabilization. Some nonlinear feedback controllers have been proposed in the literature for solving the first problem.[5] The main idea behind these algorithms is to define velocity control inputs that stabilize the closed-loop system. A reference cart generates the trajectory that the mobile robot is supposed to follow. In path following, as in the previous case, we need to design velocity control inputs that stabilize a car-like mobile robot in a given *xy*-geometric path; see ref. 4 for references. The hardest problem is stabilization about a desired posture. One way to solve this problem is given in ref. 6, where the velocity control inputs are time-varying functions.

All these controllers consider only the kinematic model (e.g., ''steering system'') of the mobile robot, and ''perfect velocity'' tracking is assumed to generate the actual vehicle control inputs.[5] There are three problems with this approach; first, the perfect velocity tracking assumption does not hold in practice; second, disturbances are ignored; and, finally, complete knowledge of the dynamics is needed.[7] The *backstepping* control approach (c.f., refs. 8–10) proposed in this article corrects this omission. It provides a rigorous method of taking into account the specific vehicle dynamics to convert a steering system command into control inputs for the actual vehicle. First, feedback velocity control inputs are designed for the kinematic steering system to make the position error asymptoticaly stable. Then, a computed-torque controller is designed such that the mobile robot's velocities converge to the given velocity inputs. This control approach can be applied to a class of *smooth* kinematic system control velocity inputs. Therefore, the same design procedure works for all of the three basic navigation problems mentioned above.

A different approach has been developed in refs. 11 and 12. This approach is based on the fact that a nonholonomic system is not input-state linearizable. Nevertheless, it is input-output linearizable if a proper output is selected. The tracking problem is addressed in ref. 12, and an extension to path following is given in ref. 11. The problem of point stabilization has not been considered.

Another intensive area of research has been neural network (NN) applications in closed-loop control. Several groups by now are doing rigorous analysis of NN controllers using a variety of techniques.[13–16] In this article, we design a robust-adaptive kinematic/neuro-controller based on the universal approximation property of NN.[17] The NN learns the *full dynamics* of the mobile robot *on-line,* and the kinematic controller stabilizes the state of the system in a small neighborhood of the origin.

The remainder of the article is organized as follows. Section 2 provides the theoretical background of a nonholonomic mobile robot, and some structural properties of the nonholonomic dynamical equations are given. In section 3 we consider the case when the dynamics of the mobile robot is fully known, and apply our control method to the trajectory tracking navigation problem. The stability of the closed-loop system is proven by Lyapunov theory. In section 4 we develop a robust-adaptive controller based on neural networks. The NN controller can deal with unmodeled bounded disturbances and/or unstructured unmodeled dynamics in the vehicle. Section 5 presents some simulation results. Finally, section 6 gives some concluding remarks.

2. PRELIMINARIES

2.1. A Nonholonomic Mobile Robot

A mobile robot system having an n-dimensional configuration space $\mathscr{S}$ with generalized coordinates $(q_1, \ldots, q_n)$ and subject to m constraints can be described by[11,18]

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V_m}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}(\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_d = \mathbf{B}(\mathbf{q})\boldsymbol{\tau} - \mathbf{A}^T(\mathbf{q})\boldsymbol{\lambda}, \tag{1}$$

where $\mathbf{M}(\mathbf{q}) \in \Re^{n\times n}$ is a symmetric, positive definite inertia matrix, $\mathbf{V_m}(\mathbf{q}, \dot{\mathbf{q}}) \in \Re^{n\times n}$ is the centripetal and coriolis matrix, $\mathbf{F}(\dot{\mathbf{q}}) \in \Re^{n\times 1}$ denotes the surface friction, $\mathbf{G}(\mathbf{q}) \in \Re^{n\times 1}$ is the gravitational vector, τ_d denotes bounded unknown disturbances including unstructured unmodeled dynamics, $\mathbf{B}(\mathbf{q}) \in \Re^{n\times r}$ is the input transformation matrix, $\tau \in \Re^{n\times 1}$ is the input vector, $\mathbf{A}(\mathbf{q}) \in \Re^{m\times n}$ is the matrix associated with the constraints, and $\lambda \in \Re^{m\times 1}$ is the vector of constraint forces.

We consider that all kinematic equality constraints are independent of time, and can be expressed as follows

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0}. \tag{2}$$

Let $\mathbf{S}(\mathbf{q})$ be a full rank matrix $(n - m)$ formed by a set of smooth and linearly independent vector fields spanning the null space of $\mathbf{A}(\mathbf{q})$, i.e.,

$$\mathbf{S}^T(\mathbf{q})\mathbf{A}^T(\mathbf{q}) = \mathbf{0}. \tag{3}$$

According to (2) and (3), it is possible to find an auxiliary vector time function $\mathbf{v}(t) \in \Re^{n-m}$ such that, for all t

$$\dot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\mathbf{v}(t). \tag{4}$$

The mobile robot shown in Figure 1 is a typical example of a nonholonomic mechanical system. It consists of a vehicle with two driving wheels mounted on the same axis, and a front free wheel. The motion and orientation are achieved by independent actuators, e.g., DC motors providing the necessary torques to the rear wheels.

The position of the robot in an inertial Cartesian frame {O, X, Y} is completely specified by the vector $\mathbf{q} = [x_c y_c \theta]^T$ where x_c, y_c are the coordinates of the center of mass of the vehicle, and θ is the orientation of the basis $\{C, X_c, Y_c\}$ with respect to the inertial basis.

The nonholonomic constraint states that the robot can only move in the direction normal to the axis of the driving wheels, i.e., the mobile base satisfies the conditions of *pure rolling and non slipping*[12,19]

$$\dot{y}_c \cos\theta - \dot{x}_c \sin\theta - d\dot{\theta} = 0. \tag{5}$$

It is easy to verify that the kinematic equations of motion (4) of $\mathbf{C}$ in terms of its linear velocity and angular velocity are

$$\mathbf{S}(\mathbf{q}) = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{v} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix},$$

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \tag{6}$$

where $|v_1| \le V_{max}$ and $|v_2| \le W_{max}$. V_{max} and W_{max} are the maximum linear and angular velocities of the mobile robot. System (6) is called the *steering system* of the vehicle.

The Lagrange formalism is used to find the dynamic equations of the mobile robot. In this case $\mathbf{G}(\mathbf{q}) = \mathbf{0}$, because the trajectory of the mobile base is constrained to the horizontal plane, i.e., since the system cannot change its vertical position, its potential energy, U, remains constant. The kinetic energy K_E is given by[18]

$$k_E^i = \frac{1}{2} m_i v_i^T v_i + \frac{1}{2} \omega_i^T I_i \omega_i, K_E = \sum_{i=1}^{n_i} k_E^i = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}. \tag{7}$$

The dynamical equations of the mobile base in Figure 1 can be expressed in the matrix form (1) where

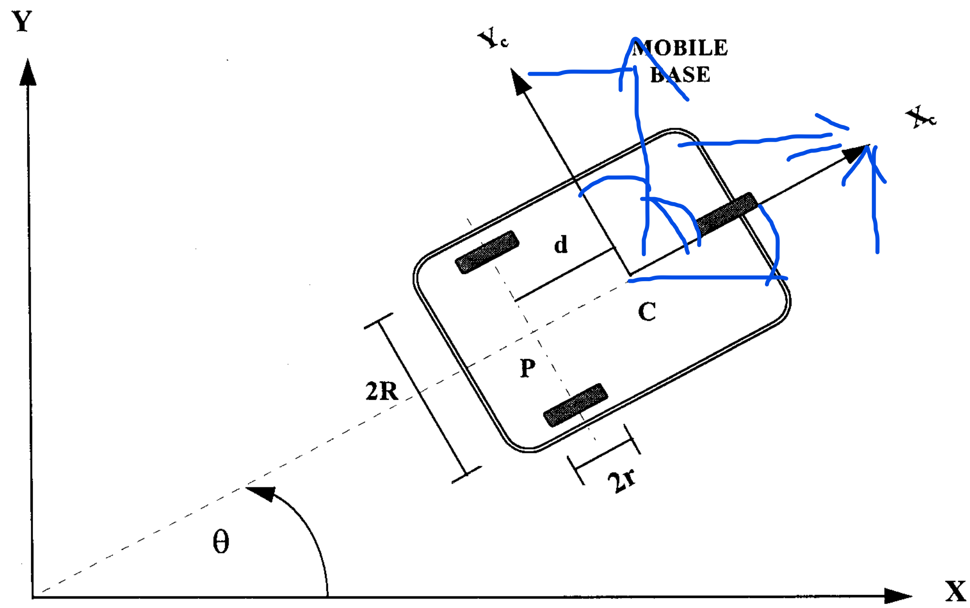

Figure 1. A nonholonomic mobile platform.

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} m & 0 & md\sin\theta \\ 0 & m & -md\cos\theta \\ md\sin\theta & -md\cos\theta & I \end{bmatrix},$$

$$\mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) = \begin{bmatrix} md\dot{\theta}^2\cos\theta \\ md\dot{\theta}^2\sin\theta \\ 0 \end{bmatrix},$$

$$\mathbf{G}(\mathbf{q}) = \mathbf{0},\ \mathbf{B}(\mathbf{q}) = \frac{1}{r}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ R & -R \end{bmatrix},$$

$$\tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}, \qquad \mathbf{A}^T(\mathbf{q}) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ -d \end{bmatrix},$$

$$\lambda = -m(\dot{x}_c\cos\theta + \dot{y}_c\sin\theta)\dot{\theta} \tag{8}$$

Similar dynamical models have been reported in the literature; for instance in ref. 12 the mass and inertia of the driving wheels are considered explicitly.

2.2. Structural Properties of a Mobile Platform

The system (1) is now transformed into a more appropriate representation for control purposes. Differentiating Eq. (4), substituting this result in Eq. (1), and then multiplying by $\mathbf{S}^T$, we can eliminate the constraint matrix $\mathbf{A}^T(\mathbf{q})\boldsymbol{\lambda}$. The complete equations of motion of the nonholonomic mobile platform are given by

$$\dot{\mathbf{q}} = \mathbf{S}\mathbf{v}, \tag{9}$$

$$\mathbf{S}^T\mathbf{M}\mathbf{S}\dot{\mathbf{v}} + \mathbf{S}^T(\mathbf{M}\dot{\mathbf{S}} + \mathbf{V}_m\mathbf{S})\mathbf{v} + \overline{\mathbf{F}} + \overline{\tau}_d = \mathbf{S}^T\mathbf{B}\tau, \tag{10}$$

where $\mathbf{v}(t) \in \mathfrak{R}^{n-m}$ is a velocity vector. By appropriate definitions we can rewrite Eq. (10) as follows

$$\overline{\mathbf{M}}(\mathbf{q})\dot{\mathbf{v}} + \overline{\mathbf{V}}_m(\mathbf{q},\dot{\mathbf{q}})\mathbf{v} + \overline{\mathbf{F}}(\mathbf{v}) + \overline{\tau}_d = \overline{\mathbf{B}}\tau, \tag{11}$$

$$\overline{\tau} \equiv \overline{\mathbf{B}}\tau, \tag{12}$$

where $\overline{\mathbf{M}}(\mathbf{q}) \in \mathfrak{R}^{r\times r}$ is a symmetric, positive definite inertia matrix, $\overline{\mathbf{V}}_m(\mathbf{q},\dot{\mathbf{q}}) \in \mathfrak{R}^{r\times r}$ is the centripetal and coriolis matrix, $\overline{\mathbf{F}}(\mathbf{v}) \in \mathfrak{R}^{r\times 1}$ is the surface friction, $\overline{\tau}_d$ denotes bounded unknown disturbances including unstructured unmodeled dynamics, and $\overline{\tau} \in \mathfrak{R}^{r\times 1}$ is the input vector. If $r = n - m$, it is easy to verify that $\overline{\mathbf{B}}$ is a constant nonsingular matrix that depends on the distance between the driving wheels $\mathbf{R}$ and the radius of the wheel $\mathbf{r}$ (See Fig. 1). Eq. (11) describes the behavior of the nonholonomic system in a new set of *local* coordinates, i.e., $\mathbf{S}(\mathbf{q})$ is a Jacobian matrix that transforms velocities in mobile base coordinates $\mathbf{v}$ to velocities in Cartesian coordinates $\dot{\mathbf{q}}$. Therefore, the properties of the original dynamics hold for the new set of coordinates.[18]

Boundedness: $\overline{\mathbf{M}}(\mathbf{q})$, the norm of the $\overline{\mathbf{V}}_{\mathrm{m}}(\mathbf{q},\dot{\mathbf{q}})$, and $\bar{\tau}_d$ are bounded.

Lemma 2.1. *The matrix* $\dot{\overline{\mathbf{M}}} - 2\overline{\mathbf{V}}_{\mathrm{m}}$ *is skew symmetric.*

Proof: The derivative of the inertia matrix and the centripetal and coriolis matrix are given by

$$\dot{\overline{\mathbf{M}}} = \dot{\mathbf{S}}^T\mathbf{M}\mathbf{S} + \mathbf{S}^T\dot{\mathbf{M}}\mathbf{S} + \mathbf{S}^T\mathbf{M}\dot{\mathbf{S}},$$
$$\overline{\mathbf{V}}_{\mathrm{m}} = \mathbf{S}^T\mathbf{M}\dot{\mathbf{S}} + \mathbf{S}^T\mathbf{V}_{\mathrm{m}}\mathbf{S}, \tag{13}$$

Since $\dot{\mathbf{M}} - 2\mathbf{V}_{\mathrm{m}}$ is skew-symmetric, it is straightforward to show that (14) is skew-symmetric also.

$$\dot{\overline{\mathbf{M}}} - 2\overline{\mathbf{V}}_{\mathrm{m}} = \dot{\mathbf{S}}^T\mathbf{M}\mathbf{S} - (\dot{\mathbf{S}}^T\mathbf{M}\mathbf{S})^T + \mathbf{S}^T(\dot{\mathbf{M}} - 2\mathbf{V}_{\mathrm{m}})\mathbf{S}. \tag{14}$$

■

2.3. A Note on Controllability of Nonholonomic Systems

The complete dynamics (9), (10) consists of the kinematic steering system (9) plus some extra dynamics (10). Standard approaches to nonholonomic controls design deal only with (9), ignoring the actual vehicle dynamics. In this article we correct this omission.

Let $\mathbf{u}$ be an auxiliary input, then by applying the nonlinear feedback.

$$\tau = f_\tau(\mathbf{q},\dot{\mathbf{q}}, \mathbf{v}, \mathbf{u}) = \overline{\mathbf{B}}^{-1}(\mathbf{q})[\overline{\mathbf{M}}(\mathbf{q})\mathbf{u} + \overline{\mathbf{V}}_{\mathrm{m}}(\mathbf{q},\dot{\mathbf{q}})\mathbf{v} + \overline{\mathbf{F}}(\mathbf{v})], \tag{15}$$

one can convert the dynamic control problem into the kinematic control problem

$$\dot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\mathbf{v},$$
$$\dot{\mathbf{v}} = \mathbf{u}. \tag{16}$$

Eq. (16) represents a state-space description of the nonholonomic mobile robot and constitutes the basic framework for defining its nonlinear control properties.[20,21]

In performing the input transformation (15), it is assumed that all the dynamical quantities (e.g., $\overline{\mathbf{M}}(\mathbf{q})$, $\overline{\mathbf{F}}(\mathbf{v})$, $\overline{\mathbf{V}}_{\mathrm{m}}(\mathbf{q},\dot{\mathbf{q}})$) of the vehicle are exactly known and $\bar{\tau}_d = 0$. Defining $\mathbf{x} = [\mathbf{q}^T\mathbf{v}^T]^T$, Eq. (16) can be rewritten as

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}. \tag{17}$$

As the system (16) satisfies the Accessibility Rank Condition at $\mathbf{x}_0$, it is locally weakly controllable at $\mathbf{x}_0$. For the cart-mobile robot, local weak controllability implies controllability.[19] Since, the involutivity condition is not satisfied, the system (16) is not input-state linearizable by a state feedback. Nevertheless, it is input-output linearizable if an adequate output function is selected.[12]

Although a nonlinear system can be controllable, a stabilizable smooth state feedback may not exist. Unfortunately, this is the case of the system (16), where the equilibrium point $\mathbf{x}_e = 0$ cannot be made asymptotically stable by any smooth time-invariant state-feedback.[20]

2.4. Backstepping Controller Design

Many approaches exist to selecting a velocity control $\mathbf{v}(t)$ for the steering system (9). In this section, we desire to convert such a prescribed control $\mathbf{v}(t)$ into a torque control $\tau(t)$ for the actual physical cart. Therefore, our objective is to select $\tau(t)$ in (15) so that (16) exhibits the desired behavior motivating the specific choice of the velocity $\mathbf{v}(t)$. This allows the steering system commands $\mathbf{v}(t)$ in the literature to be converted to torques $\tau(t)$ that take into account the mass, friction, etc., parameters of the actual cart.

The nonholonomic navigation problem of steering $\mathbf{v}(t)$ may be divided into three basic problems: tracking a reference trajectory, following a path, and point stabilization. It is desirable to have a common design algorithm capable of dealing with these three basic navigation problems. This algorithm can be implemented by considering that each one of the basic problems may be solved by using adequate smooth velocity control inputs. If the mobile robot system can track a class of velocity control inputs, then tracking, path following and point stabilization may be solved under the same control structure.

The smooth steering system control, denoted by $\mathbf{v}_c$, can be found by any technique in the literature. Using the algorithm to be derived and proven in the next section, the three basic navigation problems are solved as follows:

Tracking: The trajectory tracking problem for nonholonomic vehicles is posed as follows. Let there be prescribed a reference cart

$$\dot{x}_r = v_r \cos\theta_r,\ \dot{y}_r = v_r \sin\theta_r,\ \dot{\theta}_r = \omega_r,$$
$$\mathbf{q}_r = [x_r y_r \theta_r]^T,\ \mathbf{v}_r = [v_r \omega_r]^T, \tag{18}$$

with $v_r > 0$ for all t, find a smooth velocity control $\mathbf{v}_c(t) = f_c(\mathbf{e}_p, \mathbf{v}_r, \mathbf{K})$ such that $\lim_{t\to\infty}(\mathbf{q}_r - \mathbf{q}) = 0$, where

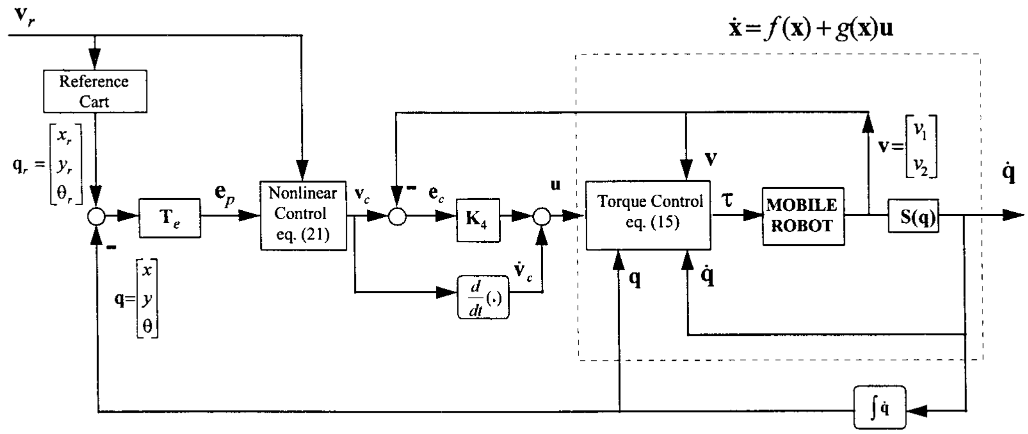

Figure 2. Tracking control structure.

$\mathbf{e}_p$, $\mathbf{v}_r$, and $\mathbf{K}$ are the tracking error, the reference velocity vector, and the controller gain vector, respectively. Then compute the torque input $\tau(t)$ for (1), such that $\mathbf{v} \to \mathbf{v}_c$ as $t \to \infty$.

Path Following: Given a path P in the plane and the mobile robot linear velocity $v(t)$, find a smooth (angular) velocity control input $\mathbf{v}_c(t) = f_c(e_\theta, v, b, \mathbf{K})$ such that $\lim_{t\to\infty} e_\theta = 0$ and $\lim_{t\to\infty} b(t) = 0$, where e_θ and $b(t)$ are the orientation error and the distance between a reference point in the mobile robot and the path P, respectively. Then compute the torque input $\tau(t)$ for (1), such that $\mathbf{v} \to \mathbf{v}_c$ as $t \to \infty$.

Point Stabilization: Given an arbitrary configuration $\mathbf{q}_r$, find a smooth time-varying velocity control input $\mathbf{v}_c(t) = f_c(\mathbf{e}_p, \mathbf{v}_r, \mathbf{K}, t)$ such that $\lim_{t\to\infty}(\mathbf{q}_r - \mathbf{q}) = 0$. Then compute the torque input $\tau(t)$ for (1), such that $\mathbf{v} \to \mathbf{v}_c$ as $t \to \infty$.

As an example to illustrate the validity of the method we have chosen the *trajectory tracking* problem. Note that *path following* is a simpler problem that requires that only the angular velocity change to decrease the distance between a given geometric path and the mobile robot. *Point stabilization* is solved in section 4 by using the same controller structure, but in this case the input control velocities are time-varying, and the control torques are provided by a neural network.

3. TRACKING A REFERENCE TRAJECTORY

A general structure for the tracking control system is presented in Figure 2. In this figure, complete knowledge of the dynamics of the cart is assumed, so that (15) is used to compute $\tau(t)$ given $\mathbf{u}(t)$. The contribution of this section lies in deriving a suitable $\mathbf{u}(t)$ and $\tau(t)$ from a specific $\mathbf{v}_c(t)$ that controls the steering system (16). It is common in the literature to address the problem by assuming "perfect velocity tracking," which may not hold in practice. A better alternative to this unrealistic assumption is the *integrator backstepping method* now developed.

To be specific, it is assumed that the solution to the steering system tracking problem in ref. 5 is available. This is denoted by $\mathbf{v}_c(t)$. The tracking error vector is expressed in the basis of a frame linked to the mobile platform[4]

$$\mathbf{e}_p = \mathbf{T}_e(\mathbf{q}_r - \mathbf{q}),$$

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r & -x \\ y_r & -y \\ \theta_r & -\theta \end{bmatrix}, \tag{19}$$

and the derivative of the error is

$$\dot{\mathbf{e}}_p = \begin{bmatrix} v_2 e_2 - v_1 + v_r \cos e_3 \\ -v_2 e_1 + v_r \sin e_3 \\ \omega_r - v_2 \end{bmatrix}. \tag{20}$$

The auxiliary velocity control input that achieves tracking for (16) is given by

$$\mathbf{v}_c = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{bmatrix},$$

$$\mathbf{v}_c = f_c(\mathbf{e}_p, \mathbf{v}_r, \mathbf{K}), \qquad \mathbf{K} = [k_1\, k_2\, k_3]^T. \tag{21}$$

The derivative of $\mathbf{v}_c$ becomes

$$\dot{\mathbf{v}}_c = \begin{bmatrix} \dot{v}_r \cos e_3 \\ \dot{\omega}_r + k_2 \dot{v}_r e_2 + k_3 \dot{v}_r \sin e_3 \end{bmatrix} + \begin{bmatrix} k_1 & 0 & -v_r \sin e_3 \\ 0 & k_2 v_r & k_3 v_r \cos e_3 \end{bmatrix} \dot{\mathbf{e}}_p, \quad (22)$$

and, assuming that the linear and angular reference velocities are constants, we obtain

$$\dot{\mathbf{v}}_c = \begin{bmatrix} k_1 & 0 & -v_r \sin e_3 \\ 0 & k_2 v_r & k_3 v_r \cos e_3 \end{bmatrix} \dot{\mathbf{e}}_p. \quad (23)$$

Then the proposed nonlinear feedback acceleration control input is

$$\mathbf{u} = \dot{\mathbf{v}}_c + \mathbf{K}_4(\mathbf{v}_c - \mathbf{v}), \quad (24)$$

where $\mathbf{K}_4$ is a positive definite, diagonal matrix given by

$$\mathbf{K}_4 = k_4 \mathbf{I}. \quad (25)$$

Note that Eq. (24) is also valid for the case when $v_r(t)$ and $\omega_r(t)$ are time-varying functions. It is common in the literature to assume simply that $\mathbf{u} = \dot{\mathbf{v}}_c$, called ''perfect velocity tracking,'' which cannot be assured to yield tracking for the actual cart.

Theorem 3.1. *Given a nonholonomic system with n generalized coordinates* $\mathbf{q}$*, m independent constraints, r actuators, let the following assumptions hold:*
a.1. The number of actuators is equal to the number of degrees of freedom (i.e., $r = n - m$).
a.2. The reference linear velocity is nonzero and bounded, $v_r > 0$ for all t. The angular velocity ω_r is bounded.
a.3. A smooth auxiliary velocity control input $\mathbf{v}_c$ is given by (21).
a.4. $\mathbf{K} = [k_1\ k_2\ k_3]^T$ is a vector of positive constants.
Let the nonlinear feedback control $\mathbf{u} \in \Re^{n-m}$ given by (24) be used and the vehicle input commands be given by (15). Then, the origin $\mathbf{e}_p = \mathbf{0}$ is uniformly asymptotically stable, and the velocity vector of the mobile base satisfies $\mathbf{v} \to \mathbf{v}_c$ as $t \to \infty$.

Proof: Define an auxiliary velocity error

$$\mathbf{e}_c = \mathbf{v} - \mathbf{v}_c, \quad (26)$$

$$\mathbf{e}_c = \begin{bmatrix} e_4 \\ e_5 \end{bmatrix} = \begin{bmatrix} v_1 - v_{c1} \\ v_2 - v_{c2} \end{bmatrix} = \begin{bmatrix} v_1 - v_r \cos e_3 - k_1 e_1 \\ v_2 - \omega_r - k_2 v_r e_2 - k_3 v_r \sin e_3 \end{bmatrix},$$

by using (24), we obtain

$$\dot{\mathbf{e}}_c = -\mathbf{K}_4 \mathbf{e}_c, \quad (27)$$

then the velocity vector of the mobile base satisfies $\mathbf{v} \to \mathbf{v}_c$ as $t \to \infty$.

Consider the following Lyapunov function candidate:

$$V = k_1(e_1^2 + e_2^2) + \frac{2k_1}{k_2}(1 - \cos e_3) + \frac{1}{2k_4}\left(e_4^2 + \frac{k_1}{k_2 k_3 v_r} e_5^2\right), \quad (28)$$

where $V \geq 0$, and $V = 0$ only if $\mathbf{e}_p = 0$ and $\mathbf{e}_c = 0$. Furthermore, by using (20), (26) and (27)

$$\dot{V} = 2k_1 e_1 \dot{e}_1 + 2k_1 e_2 \dot{e}_2 + \frac{2k_1}{k_2} \dot{e}_3 \sin e_3 - e_4^2 - \frac{k_1}{k_2 k_3 v_r} e_5^2, \quad (29)$$

$$\dot{V} = -(v_1 - v_r \cos e_3)^2 - k_1^2 e_1^2 - \frac{k_1}{k_2 k_3 v_r}(v_2 - \omega_r - k_2 v_r e_2)^2 - \frac{k_1 k_3}{k_2} v_r \sin^2 e_3, \quad (30)$$

and, considering (26) again, we obtain

$$\dot{V} = -k_1^2 e_1^2 - \frac{k_1 k_3}{k_2} v_r \sin^2 e_3 - (e_4 + k_1 e_1)^2 - \frac{k_1}{k_2 k_3 v_r}(e_5 + k_3 v_r \sin e_3)^2, \quad (31)$$

clearly, $\dot{V} \leq 0$ and the entire error $\mathbf{e} = [\mathbf{e}_p^T\ \mathbf{e}_c^T]^T$ is bounded. Using Eqs. (20), (26), (31), and assumption (*a.3*), one deduces that $\|\mathbf{e}\|$ and $\|\dot{\mathbf{e}}\|$ are bounded, so that $\|V\| < \infty$, i.e., $\dot{V}$ is uniformly continuous. Since $V(t)$ does not increase and converges to some constant value, by Barbalat's lemma, $\dot{V} \to 0$ as $t \to \infty$. Considering that $\mathbf{e}_c = [e_4\ e_5]^T \to 0$ as $t \to \infty$, then in the limit

$$0 = k_1 e_1^2 + \frac{k_3}{k_2} v_r \sin^2 e_3. \quad (32)$$

Eq. (32) implies that $[e_1\ e_3]^T \to 0$ as $t \to \infty$. From (20) we have

$$\omega_r - v_2 = 0, \quad (33)$$

and considering that $e_5 \to 0$ in (26), it yields

$$v_2 - \omega_r - k_2 v_r e_2 - k_3 v_r \sin e_3 = 0, \quad (34)$$

$$-k_2 v_r e_2 = 0. \quad (35)$$

By assumption $v_r > 0$, then $e_2 \to 0$ as $t \to \infty$. Therefore, the equilibrium point $\mathbf{e} = \mathbf{0}$ is uniformly asymptotically stable. ■

4. POINT STABILIZATION USING NEURAL NETWORKS

In this section we present a robust-adaptive kinematic/neuro-controller that can deal with unmodeled bounded disturbances and/or unstructured unmodeled dynamics in the nonholonomic mobile robot. On-line NN weight tuning algorithms that do not require off-line learning yet guarantee small tracking errors and bounded control signals are utilized.

4.1. Feedforward Neural Networks

The neural network output $\mathbf{y}$ is a vector with m components that are determined in terms of the n components of the input vector $\mathbf{x}$ by the formula

$$\mathbf{y}_i = \sum_{j=1}^{N_h} \left[\mathbf{w}_{ij}\,\sigma\left(\sum_{k=1}^{n} \mathbf{v}_{jk}\mathbf{x}_k + \theta_{vj} \right) + \theta_{wi} \right),\ i = 1, \ldots, m \tag{36}$$

where $\sigma(\cdot)$ are the activation functions and N_h is the number of hidden-layer neurons. The first-to-second-layer interconnection weights are denoted by $\mathbf{v}_{jk}$ and the second-to-third-layer interconnection weights by $\mathbf{w}_{ij}$. The threshold offsets are denoted by θ_{vj}, θ_{wi}. By collecting all the NN weights $\mathbf{v}_{jk}$, $\mathbf{w}_{ij}$ into matrices of weights $\mathbf{V}^T$, $\mathbf{W}^T$, one can write the NN equation in terms of vectors as

$$\mathbf{y} = \mathbf{W}^T\sigma(\mathbf{V}^T\mathbf{x}). \tag{37}$$

The thresholds are included as the first columns of the weight matrices. Any tuning of $\mathbf{W}$ and $\mathbf{V}$ then includes tuning of the thresholds as well.

The main property of an NN we shall be concerned with for controls purposes is the *function approximation property*.[17,22] Let $f(\mathbf{x})$ be a smooth function from $\Re^n$ to $\Re^m$. Then, it can be shown that, as long as $\mathbf{x}$ is restricted to a compact set U_x of $\Re^n$ for some number of hidden layer neurons N_h, there exist weights and thresholds such that one has

$$f(\mathbf{x}) = \mathbf{W}^T\sigma(\mathbf{V}^T\mathbf{x}) + \varepsilon. \tag{38}$$

This equation means that an NN can approximate any function in a compact set. The value of ε is called the *NN functional approximation error.* Then, an estimate of $f(\mathbf{x})$ can be given by

$$\hat{f}(\mathbf{x}) = \hat{\mathbf{W}}^T\sigma(\hat{\mathbf{V}}^T\mathbf{x}), \tag{39}$$

where $\hat{\mathbf{W}}$, $\hat{\mathbf{V}}$ are estimates of the ideal NN weights that are provided by some on-line weight tuning algorithms. For a more detailed discussion the reader is referred to ref. 23.

4.2. Feedback Stabilization of Nonholonomic Systems

Feedback stabilization consists of finding feedback laws such that an equilibrium point of the closed-loop system is asymptotically stable. Unfortunately, the linearization of nonholonomic systems about any equilibrium point is not asymptotically stabilizable. Moreover, there exists *no smooth static (dynamic) time-invariant state-feedback* that makes an equilibrium point of the closed-loop system locally asymptotically stable.[1,4,20] Therefore, feedback linearization techniques cannot be applied to nonholonomic systems directly.

A variety of techniques have been proposed in the nonholonomic literature to solve the asymptotic stabilization problem. In ref. 1 these techniques are classified as (1) continuous time-varying stabilization, (2) discontinuous time-invariant stabilization, and (3) hybrid stabilization. This section is concerned with the former.

4.2.1. Time-Varying Stabilization

Time-varying control laws for nonholonomic mobile robots were introduced by Samson.[6] Unfortunately, the rates of convergence provided by smooth time-periodic feedback laws are at most $t^{-1/2}$, i.e., nonexponential.[4] Thus feedback laws with faster rates of convergence are desirable for practical purposes. These feedback laws are necessarily non-smooth.

In this section we use a *hybrid* strategy; that is, a continuous time-periodic static state-feedback that is smooth everywhere except at the boundary of a small neighborhood of the origin.

Point Stabilization as an Extension of the Tracking Problem: The trajectory tracking problem for nonholonomic vehicles is given by (18). As in ref. 4 it is assumed that the reference cart moves along the x-axis, i.e.,

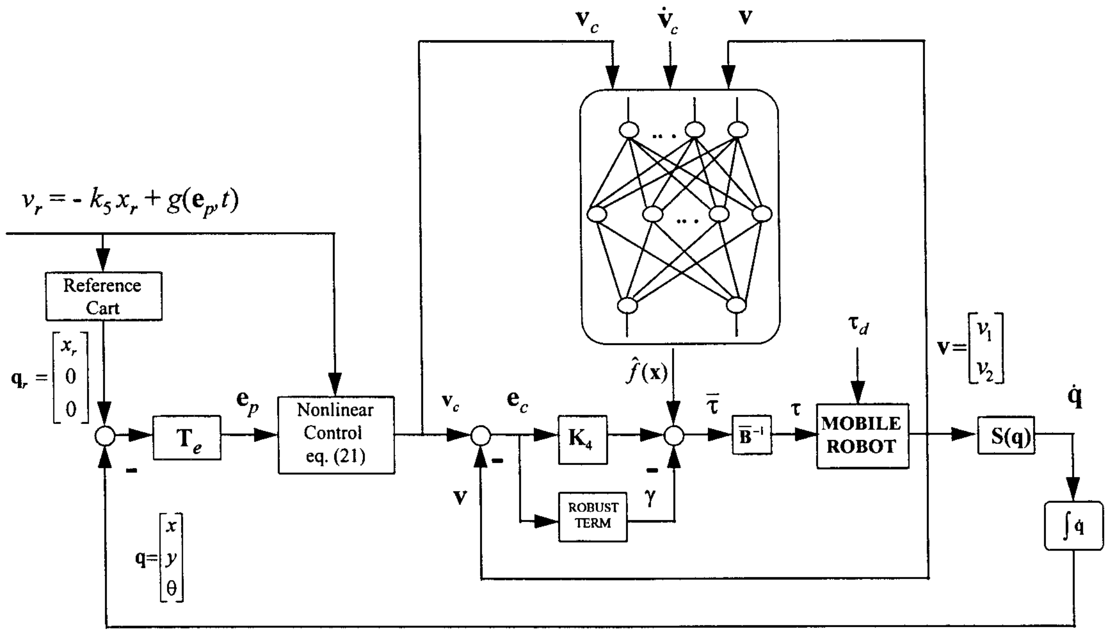

Figure 3. Practical point stabilization using NN.

$$\dot{x}_r = v_r,\ \mathbf{q}_r = [x_r\ 0\ 0]^T,\ \mathbf{v}_r = [v_r\ 0]^T. \tag{40}$$

Therefore, the point stabilization problem consists of finding a smooth time-varying velocity control input $\mathbf{v}_c(t)$ such that $\lim_{t\to\infty}(\mathbf{q}_r - \mathbf{q}) = 0$ and $\lim_{t\to\infty} x_r = 0$. Then compute the torque input $\tau(t)$ for (11), such that $\mathbf{v} \to \mathbf{v}_c$ as $t \to \infty$.

The structure for the point stabilization system is given in Figure 3. In this figure, *no* knowledge of the dynamics of the cart is assumed. The function of the NN is to reconstruct the dynamics (11) by learning it on-line.

The design method is the same as in section 3. However, in this case

$$v_r = -k_5 x_r + g(\mathbf{e}_p,t), \tag{41}$$

and

$$g(\mathbf{e}_p,t) = \|\mathbf{e}_p\|^2 \sin t, \tag{42}$$

where $k_5 > 0$. Different time-varying functions $g(\mathbf{e}_p,t)$ are available in the literature, see ref. 1 and the references therein.

Given the desired velocity $\mathbf{v}_c(t) \in \Re^2$, define now the auxiliary velocity tracking error as

$$\mathbf{e}_c = \mathbf{v}_c - \mathbf{v}. \tag{43}$$

Differentiating (43) and using (11), the mobile robot dynamics may be written in terms of the velocity tracking error as

$$\overline{\mathbf{M}}(\mathbf{q})\dot{\mathbf{e}}_c = -\overline{\mathbf{V}}_m(\mathbf{q},\dot{\mathbf{q}})\mathbf{e}_c - \bar{\tau} + f(\mathbf{x}) + \bar{\tau}_d, \tag{44}$$

where the important *nonlinear mobile robot function* is

$$f(\mathbf{x}) = \overline{\mathbf{M}}(\mathbf{q})\dot{\mathbf{v}}_c + \overline{\mathbf{V}}_m(\mathbf{q},\dot{\mathbf{q}})\mathbf{v}_c + \overline{\mathbf{F}}(\mathbf{v}). \tag{45}$$

The vector $\mathbf{x}$ required to compute $f(\mathbf{x})$ can be defined as

$$\mathbf{x} \equiv [\mathbf{v}^T\ \mathbf{v}_c^T\ \dot{\mathbf{v}}_c^T]^T, \tag{46}$$

which can be measured.

Function $f(\mathbf{x})$ contains all the mobile robot parameters such as masses, moments of inertia, friction coefficients, and so on. These quantities are often imperfectly known and difficult to determine.

4.3. Mobile Robot Controller Structure

In applications the nonlinear robot function $f(\mathbf{x})$ is at least partially unknown. Therefore, a suitable control input for velocity following is given by the computed-torque like control

$$\bar{\tau} = \hat{f} + \mathbf{K}_4\mathbf{e}_c - \gamma, \tag{47}$$

with $\mathbf{K}_4$ a diagonal, positive definite gain matrix, and $\hat{f}(\mathbf{x})$ an *estimate* of the robot function $f(\mathbf{x})$ that is provided by the neural network. The robustifying signal $\gamma(t)$ is required to compensate the unmodeled unstructured disturbances. Using this control in (44), the closed-loop system becomes

$$\overline{\mathbf{M}}\dot{\mathbf{e}}_c = -(\mathbf{K}_4 + \overline{\mathbf{V}}_{\mathbf{m}})\mathbf{e}_c + \tilde{f} + \overline{\tau}_d + \gamma, \tag{48}$$

where the velocity tracking error is driven by the *functional estimation error*

$$\tilde{f} = f - \hat{f}. \tag{49}$$

By using the controller (47), there is no guarantee that the control $\overline{\tau}$ will make the velocity tracking error small. Thus, the control design problem is to specify a method of selecting the matrix gain $\mathbf{K}_4$, the estimate $\hat{f}$, and the robustifying signal $\gamma(t)$ so that both error $\mathbf{e}_c(t)$ and the control signals are bounded. It is important to note that the latter conclusion hinges on showing that the estimate $\hat{f}$ is bounded. Moreover, for good performance, the bound on $\mathbf{e}_c(t)$ should be in some sense "small enough" because it will affect directly the position error $\mathbf{e}_p(t)$.

In this section we will use an NN to provide the estimate $\hat{f}$ for computing the control in (47). By placing into (47) the neural network approximation equation given by (39), the control input then becomes

$$\overline{\tau} = \hat{\mathbf{W}}^T\sigma(\hat{\mathbf{V}}^T\mathbf{x}) + \mathbf{K}_4\mathbf{e}_c - \gamma, \tag{50}$$

and the velocity error dynamics is given by

$$\overline{\mathbf{M}}\dot{\mathbf{e}}_c = -(\mathbf{K}_4 + \overline{\mathbf{V}}_{\mathbf{m}})\mathbf{e}_c + \mathbf{W}^T\sigma(\mathbf{V}^T\mathbf{x}) - \hat{\mathbf{W}}^T\sigma(\hat{\mathbf{V}}^T\mathbf{x}) + (\varepsilon + \overline{\tau}_d) + \gamma. \tag{51}$$

It remains now to show how to select the tuning algorithms for the NN weights, and the robustifying term $\gamma(t)$ so that robust stability and tracking performance are guaranteed.

Definition: We say that the solution of a nonlinear system with state $\mathbf{x}(t) \in \Re^n$ is uniformly ultimately bounded (UUB) if there exists a compact set $U_x \subset \Re^n$ such that for all $\mathbf{x}(t_0) = \mathbf{x}_0 \in U_x$, there exists a $\delta > 0$ and a number $T(\delta, \mathbf{x}_0)$ such that $\|\mathbf{x}(t)\| < \delta$ for all $t \geq t_0 + T$.

Definition: For notational convenience we define the matrix of all the NN weights as $\mathbf{Z} \equiv diag\{\mathbf{W}, \mathbf{V}\}$. Assume that the ideal weights are bounded, i.e., $\|\mathbf{Z}\|_F \leq Z_M$ with Z_M known.

Take the control $\overline{\tau} \in \Re^2$ for (11) as (50) with robustifying term

$$\gamma(t) = -K_z\mathbf{e}_c, \tag{52}$$

where K_z is a known positive constant that depends on both Z_M and the disturbance magnitude. Note that disturbances acting on the mobile robot are assumed to be bounded by some known constants.

A Lyapunov theoretic approach was used in ref. 24 to prove that the controller (50), the robustifying term (52), and the following NN weight tuning laws (53) make the velocity tracking error $\mathbf{e}_c(t)$, the position error $\mathbf{e}_p(t)$, and the NN weight estimates $\hat{\mathbf{V}}, \hat{\mathbf{W}}$ UUB.

$$\dot{\hat{\mathbf{W}}} = \mathbf{F}\hat{\sigma}\mathbf{e}_c^T - \mathbf{F}\hat{\sigma}'\hat{\mathbf{V}}^T\mathbf{x}\mathbf{e}_c^T - \kappa\mathbf{F}\|\mathbf{e}_c\|\hat{\mathbf{W}}, \tag{53.a}$$

$$\dot{\hat{\mathbf{V}}} = \mathbf{G}\mathbf{x}(\hat{\sigma}'^T\hat{\mathbf{W}}\mathbf{e}_c)^T - \kappa\mathbf{G}\|\mathbf{e}_c\|\hat{\mathbf{V}}, \tag{53.b}$$

where $\mathbf{F}$, $\mathbf{G}$ are positive definite design parameter matrices, and $\kappa > 0$. The first terms of (53) are nothing but the standard backpropagation algorithm. The last terms correspond to the *e*-modification[15] from adaptive control theory; they must be added to ensure bounded NN weight estimates. The middle term in (53.a) is a *novel term* needed to prove stability.

In practical situations the velocity and position errors are not exactly equal to zero. The best we can do is to guarantee that the error converges to a neighborhood of the origin. If external disturbances drive the system away from the convergence compact set, the derivative of the Lyapunov function become negative and the energy of the system decreases uniformly; therefore, the error becomes small again.

5. SIMULATION RESULTS

5.1. Tracking a Reference Trajectory

The control algorithm developed in section 3 was implemented in MATLAB. We took the vehicle parameters (See Fig. 1) as m = 10 kg, I = 5 kg-m^2, R = 0.5 m, r = 0.05 m, and initial position $[x_0\ y_0\ \theta_0]$ = [2 2 0]. The reference trajectory is given by $x_r = 1$, $y_r = v_r t$, $\theta_r = 90°$. In some cases, the mobile base *maneuvers,* i.e., exhibits forward and backward motions to track the reference trajectory (See Figs. 4–6). Note that there is no path planning involved—the mobile base naturally describes a path that satisfies the nonholonomic constraints.

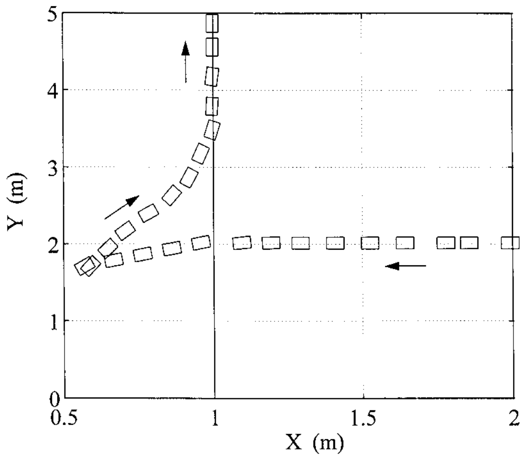

Figure 4. Mobile robot trajectory.

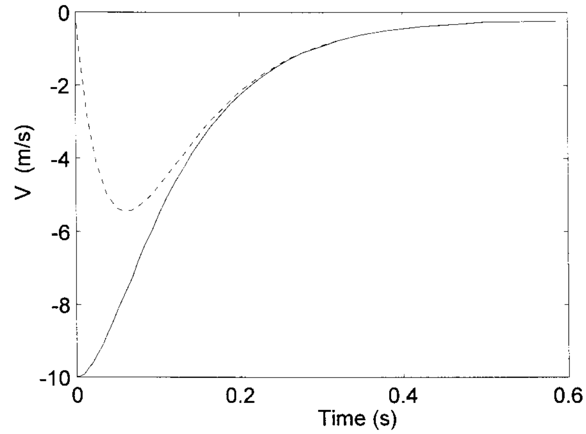

Figure 6. Control (—) and actual (--) linear velocities v (m/s).

5.2. Point Stabilization Using Neural Networks

We should like to illustrate the NN control scheme presented in section 4. Note that the NN controller *does not* require knowledge of the dynamics of the mobile robot. The controller gains were chosen so that the closed-loop system exhibits a critical damping behavior: $\mathbf{K} = [k_1\ k_2\ k_3]^T = [10\ 5\ 4]^T$, $\mathbf{K}_4 = diag\{25, 25\}$ and $k_5 = 1$. For the NN, we selected the sigmoid activation functions with $N_h = 10$ hidden-layer neurons, $\mathbf{F} = \mathbf{G} = diag\{10, 10\}$ and $\kappa = 0.1$.

To have an acceptable closed-loop performance, we may use feedback laws that are smooth everywhere except at the boundary of a small neighborhood of the origin. The following choice has been proposed in ref. 4

$$g(\mathbf{e}_p, t) = \begin{cases} \sin t & \text{if } \|\mathbf{e}_p\| \geq \varepsilon_1 > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{54}$$

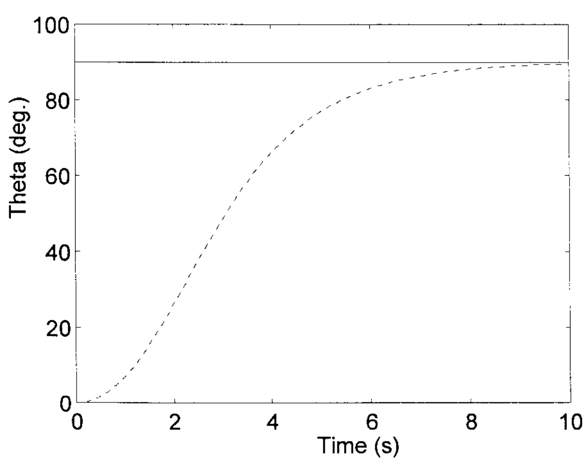

Figure 5. Reference angle (—) and heading angle (--).

Although asymptotic convergence of the mobile robot cannot be guaranteed, the reference cart can be proven to be asymptotically stable. Therefore, the mobile robot can be stabilized to an arbitrarily small neighborhood of the origin. Simulation results that verify the validity of the combined kinematic/NN controller are depicted in Figures 7–9.

5.3. A Comparison Study

For comparison purposes, three controllers have been implemented and simulated in MATLAB: (1) a controller that assumes "perfect velocity tracking," (2) the controller presented in section 3, which assumes complete knowledge of the mobile robot dynamics, and (3) the NN backstepping controller developed in section 4, which requires *no* knowledge of the dynamics. The reference trajectory is a straight line with initial coordinates and slope of (1,2) and 26.56°, respectively.

Controller with Perfect Velocity Tracking Assumption: The "perfect velocity tracking" assumption is made in the literature to convert steering system inputs into actual vehicle commands. The response with a controller designed using this assumption is shown in Figure 10. Although unmodeled disturbances were not included in this case, the performance of the closed-loop system is quite poor. In fact, this result reveals the need of a more elaborate control system, which should provide a velocity tracking inner loop.

Backstepping Computed-Torque Controller: The response with this controller is shown in Figure 11. Since bounded unmodeled disturbances and friction

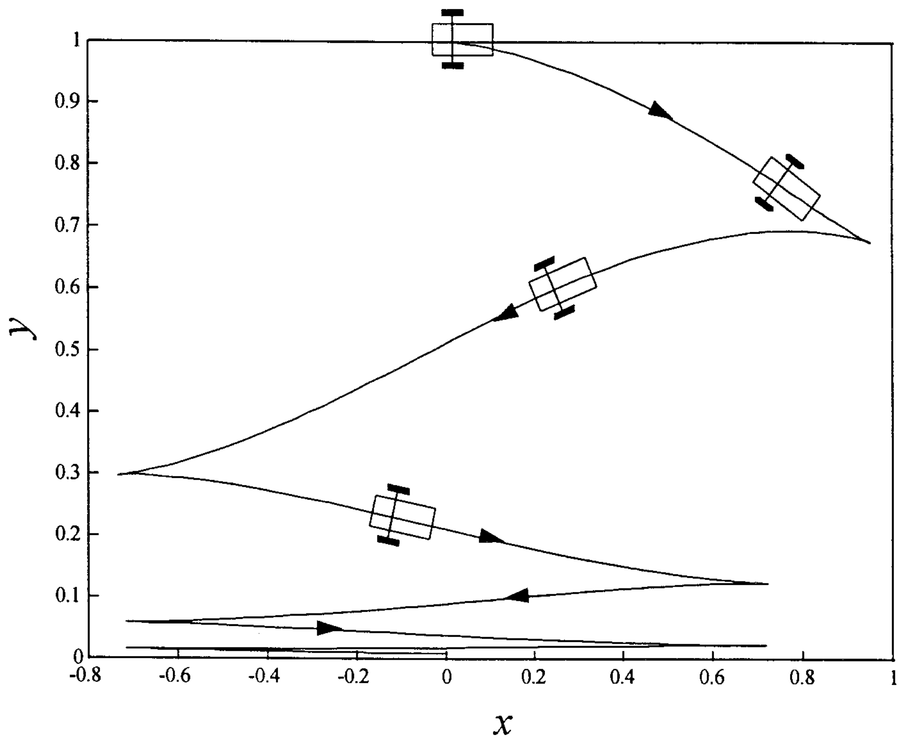

Figure 7. Mobile robot trajectory.

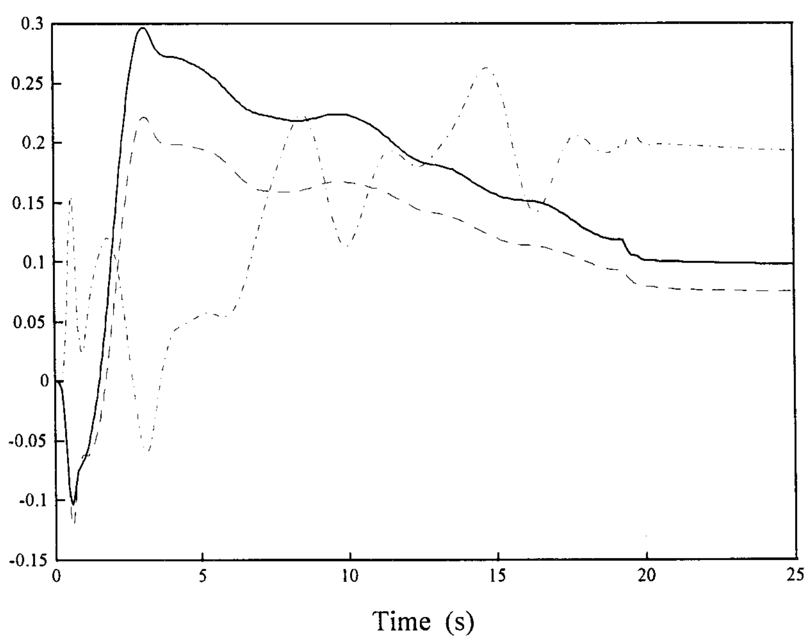

Figure 8. Some NN weights.

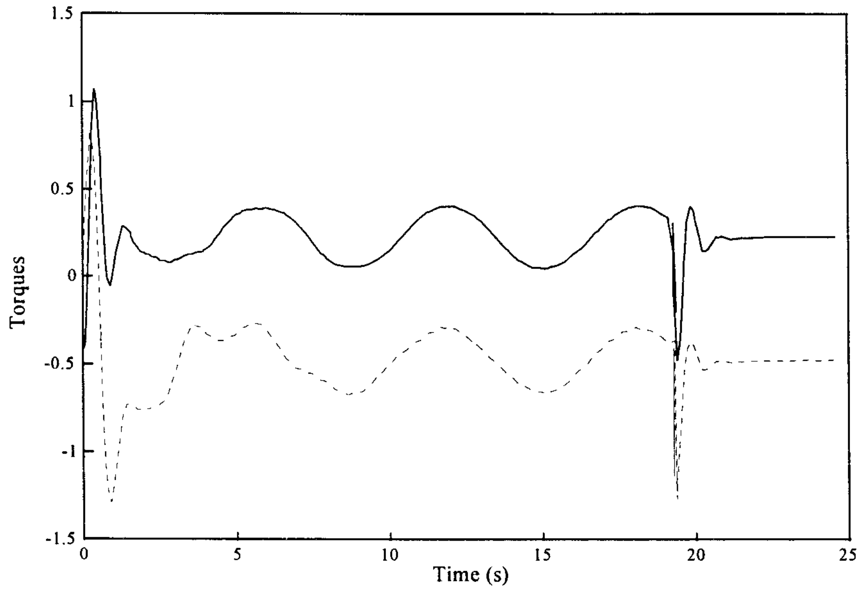

Figure 9. Applied torques: (—) right and (--) left wheels.

were included in this case, the response exhibits a steady-state error. Note that this controller requires exact knowledge of the dynamics of the vehicle to work properly. Since this controller includes a velocity tracking inner loop, the performance of the closed loop system is improved with respect to the previous case.

NN Backstepping Controller: The response with this controller is shown in Figure 12. It is clear that the performance of the system has been improved with respect to the above cases. Moreover, the NN controller requires no prior information about the dynamics of the vehicle. As the conventional computed-torque controller, the NN controller provides a velocity tracking inner loop. The robustifying term deals with unstructured unmodeled disturbances. The validity of the NN controller has been evidently verified.

6. CONCLUSIONS

A stable control algorithm capable of dealing with the three basic nonholonomic navigation problems, and that considers the complete dynamics of a mobile robot, has been derived using backstepping. This

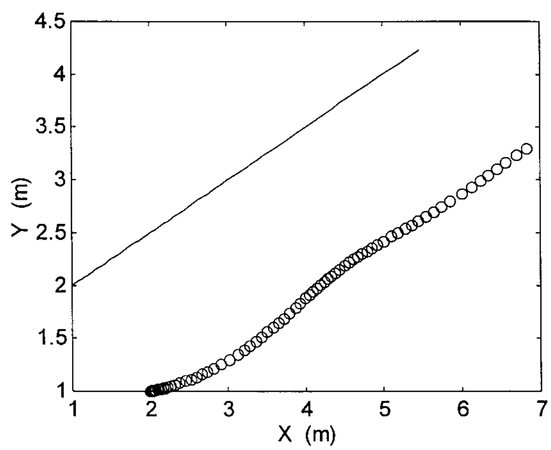

Figure 10. Perfect velocity tracking assumption. Desired (—) and actual (o) trajectories.

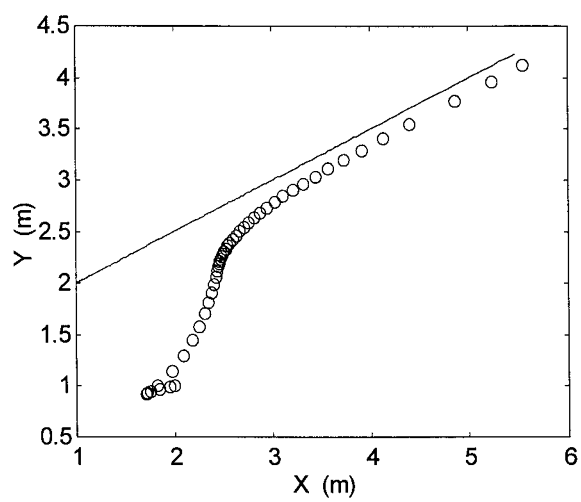

Figure 11. Backstepping controller (section 3). Desired (—) and actual (o) trajectories.

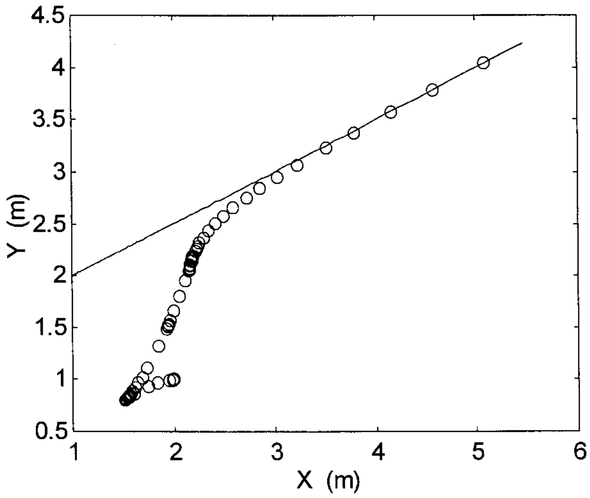

Figure 12. NN backstepping controller (section 4). Desired (—) and actual (o) trajectories.

feedback servo-control scheme is valid as long as the velocity control inputs are smooth and bounded, and the dynamics of the actual cart are completely known.

In fact, perfect knowledge of the mobile robot parameters is unattainable, e.g., *friction* is very difficult to model by conventional techniques. To confront this, a robust-adaptive controller based on neural networks has been developed. There is no need of *a priori* information of the dynamic parameters of the mobile robot, because the NN learns them on-the-fly.

A key point in developing intelligent systems is the reusability of the low-level control algorithms. This is the case of the control structure reported in this article. Section 3 and section 4 consider the case of *trajectory tracking* behavior and *point stabilization* behavior, respectively. Redefining the control velocity input $\mathbf{v}_c$, one may generate a different stable behavior, for instance, *path following,* without changing the structure of the controller. Moreover, if the mobile robot is modified or even replaced, the NN controller is still valid.

The simulation studies have shown that the proposed backstepping control design can effectively stabilize a nonholonomic mobile robot about a reference trajectory or about an equilibrium point.

REFERENCES

1. I. Kolmanovsky and N. H. McClamroch, ''Developments in nonholonomic control problems,'' *IEEE Control Syst. Mag.,* December, 20–36, 1995.
2. C. Canudas de Wit and H. Khennouf, ''Quasi-continuous stabilizing controllers for nonholonomic systems: design and robustness considerations,'' *Proc. Eur. Control Conf.,* Rome, 1995, pp. 2630–2635.
3. Z. P. Jiang and J.-B. Pomet, ''Combining backstepping and time-varying techniques for a new set of adaptive controllers,'' *Proc. IEEE Conf. Dec. Control,* Lake Buena Vista, FL, 1994, pp. 2207–2212.
4. C. Canudas de Wit, H. Khennouf, C. Samson, and O. J. Sørdalen, ''Nonlinear control design for mobile robots,'' in *Recent Trends in Mobile Robots,* Y. F. Zheng, Ed., World Scientific, 1993, pp. 121–156.
5. Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, ''A stable tracking control method for an autonomous mobile robot,'' *Proc. IEEE Int. Conf. Rob. Autom.,* 1990, pp. 384–389.
6. C. Samson, ''Time-varying feedback stabilization of car-like wheeled mobile robots,'' *Int. J. Rob. Res.,* **12**(1), 55–64, 1993.
7. C. Samson, ''Velocity and torque feedback control of a nonholonomic cart,'' in *Lecture Notes in Control and Information Science,* C. Canudas de Wit, Ed., Springer-Verlag, New York, 1991, pp. 125–151.
8. I. Kolmanovsky and N. H. McClamroch, ''Application of integrator backstepping to nonholonomic control problems,'' *Proc. IFAC Nonlinear Control Syst. Des. Symp.,* Tahoe City, CA, 1995, pp. 747–752.
9. R. T. M'Closkey and R. M. Murray, ''Extending exponential stabilizers for nonholonomic systems from kinematic controllers to dynamic controllers,'' *Proc. IFAC Symp. Rob. Control,* Capri, Italy, 1994, pp. 211–216.
10. C. M. Kwan and F. L. Lewis, ''Robust backstepping control of nonlinear systems using neural networks,'' *Proc. Eur. Control Conf.,* Rome, 1995, pp. 2772–2777.
11. N. Sarkar, X. Yun, and V. Kumar, ''Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots,'' *Int. J. Rob. Res.,* **13**(1), 55–69, 1994.
12. Y. Yamamoto and X. Yun, ''Coordinating locomotion and manipulation of a mobile manipulator,'' in *Recent Trends in Mobile Robots,* Y. F. Zheng, Ed., World Scientific, 1993, pp. 157–181.
13. F.-C. Chen and C.-C. Liu, ''Adaptively controlling nonlinear continuous-time systems using multilayer neural networks,'' *IEEE Trans. Autom. Control,* **39**(6), 1306–1310, 1994.
14. F. L. Lewis, ''Neural network control of robot manipulators,'' *IEEE Expert,* June, 64–75, 1996.
15. K. S. Narendra, ''Adaptive control using neural networks,'' in *Neural Networks for Control,* W. T. Miller, R. S. Sutton, and P. W. Werbos, Eds., MIT Press, Cambridge, MA, 1991, pp. 115–142.
16. M. M. Polycarpou and P. O. Ioannu, ''Neural networks as on-line approximators of nonlinear systems,'' *Proc. IEEE Conf. Dec. Control,* Tucson, AZ, 1992, pp. 7–12.
17. K. Hornik, M. Stinchcombe, and H. White, ''Multilayer feedforward networks are universal approximators,'' *Neural Networks,* **2,** 359–366, 1989.
18. F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators,* MacMillan, New York, 1993.
19. J. Barraquand and J.-C. Latombe, ''Nonholonomic multibody mobile robots: Controllability and motion

planning in the presence of obstacles,'' *Proc. IEEE Int. Conf. Rob. Autom.*, Sacramento, CA, 1991, pp. 2328–2335.

20. A. M. Bloch, M. Reyhanoglu, and N. H. McClamroch, ''Control and stabilization of nonholonomic dynamic systems,'' *IEEE Trans. Autom. Control,* **37**(11), 1746–1757, 1992.
21. G. Campion, B. d'Andrea-Novel, and G. Bastin, ''Controllability and state feedback stabilizability of nonholonomic mechanical systems,'' in *Lecture Notes in Control and Information Science,* C. Canudas de Wit, Ed., Springer-Verlag, 1991, pp. 106–124.
22. G. Cybenko, ''Approximation by superpositions of a sigmoidal function,'' *Math. Control Signals Syst.,* **2**(4), 303–314, 1989.
23. F. L. Lewis, A. Yesildirek, and K. Liu ''Multi-layer neural-net robot controller with guaranteed tracking performance,'' *IEEE Trans. Neural Networks,* **7**(2), 1–12, 1996.
24. R. Fierro and F. L. Lewis, ''Practical point stabilization of a nonholonomic mobile robot using neural networks,'' *Proc. IEEE Conf. Dec. Control,* to appear December 1996.
25. W. Oelen, H. Berghuis, H. Nijmeijer, and C. Canudas de Wit, ''Hybrid stabilizing control on a real mobile robot,'' *IEEE Rob. Autom. Mag.,* June, 16–23, 1995.

This is Blank Page 164