Control of a Nonholonomic Mobile Robot Using Neural Networks

R. Fierro and F. L. Lewis, *Fellow, IEEE*

***Abstract*— A control structure that makes possible the integration of a kinematic controller and a neural network (NN) computed-torque controller for nonholonomic mobile robots is presented. A combined kinematic/torque control law is developed using backstepping and stability is guaranteed by Lyapunov theory. 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. Moreover, the NN controller proposed in this work can deal with unmodeled bounded disturbances and/or unstructured unmodeled dynamics in the vehicle. On-line NN weight tuning algorithms do no require off-line learning yet guarantee small tracking errors and bounded control signals are utilized.**

***Index Terms*— Backstepping control, Lyapunov stability, mobile robots, neural networks, nonholonomic systems.**

I. Introduction

MUCH has been written about solving the problem of motion under nonholonomic constraints using the kinematic model of a mobile robot, little about the problem of integration of the nonholonomic kinematic controller and the dynamics of the mobile robot [19]. Moreover, the literature on robustness and control in presence of uncertainties in the dynamical model of such systems is sparse.

Another intensive area of research has been neural-network (NN) applications in closed-loop control. In contrast to classification applications, in feedback control the NN becomes part of the closed-loop system. Therefore, it is desirable to have a NN control with on-line learning algorithms that do no require preliminary off-line tuning [14]. Several groups by now are doing rigorous analysis of NN controllers using a variety of techniques [5], [14]–[18]. In [14] a multilayer NN controller with guaranteed performance has been developed and successfully applied to control of rigid robot manipulators, flexible-link robotic systems and position/force control. In this paper, we present an application of this NN controller to a mobile robot system. Due to the presence of the NN in the control loop, special steps must be taken to guarantee that the entire system is stable and the NN weights stay bounded.

Traditionally the learning capability of a multilayer NN has been applied to the navigation problem in mobile robots [23]–[25]. In these approaches the NN is trained in a preliminary off-line learning phase with navigation pattern behaviors; that is, the mobile robot is taught to exhibit navigation behaviors such as obstacle avoidance, wall following and so forth. Sensor signals (e.g., ultrasonic) are fed to the input layer of the network, and the output provides motor control commands (e.g., turn left). Furthermore the dynamics and nonholonomic motion constraints of the mobile robot are not taken into account. In contrast, the objective of this work is to design an adaptive neuro-controller based on the universal approximation property of NN. The NN learns the *full dynamics* of the mobile robot *on-line*. We still need, of course, a higher-level controller (i.e., trajectory generator) to carry out complex navigation behaviors; this could be provided by techniques such as [23] and [25].

Mobile robot navigation can be classified into three basic problems [4]: tracking a reference trajectory, following a path, and point stabilization. Some nonlinear feedback controllers have been proposed for solving these problems [2]–[4], [10]. The main idea behind these algorithms is to find suitable velocity control inputs which stabilize the closed-loop system.

In the literature, the nonholonomic tracking problem is simplified by neglecting the vehicle dynamics and considering only the steering system. To compute the vehicle control inputs, it is assumed that there is "perfect velocity tracking" [10]. 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 [19]. The *backstepping* control approach [11] proposed in this paper corrects this omission by means of an NN controller. 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 asymptotically stable. Then, an NN 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. The NN controller is independent of the navigation problem because its function is to compute the torque inputs based on approximating the nonlinear dynamics of the cart.

This paper is organized as follows. In Section II, we present some basics of nonholonomic systems and NN. Some struc-

Manuscript received October 29, 1995, revised January 10, 1996, March 17, 1996, and April 11, 1998. This work was performed when the first author was a Graduate Research Assistant at ARRI and a Fulbright scholar at the University of Texas at Arlington. This work was supported by NSF Grant ECS-9521673.

R. Fierro is with Escuela Politécnica Nacional, Facultad de Ingeniería Elééctrica, Casilla Postal 17-01-2759, Quito, Ecuador.

F. L. Lewis is with the Automation and Robotics Research Institute, The University of Texas at Arlington, Fort Worth, TX 76118-7115 USA.

Publisher Item Identifier S 1045-9227(98)04759-6.

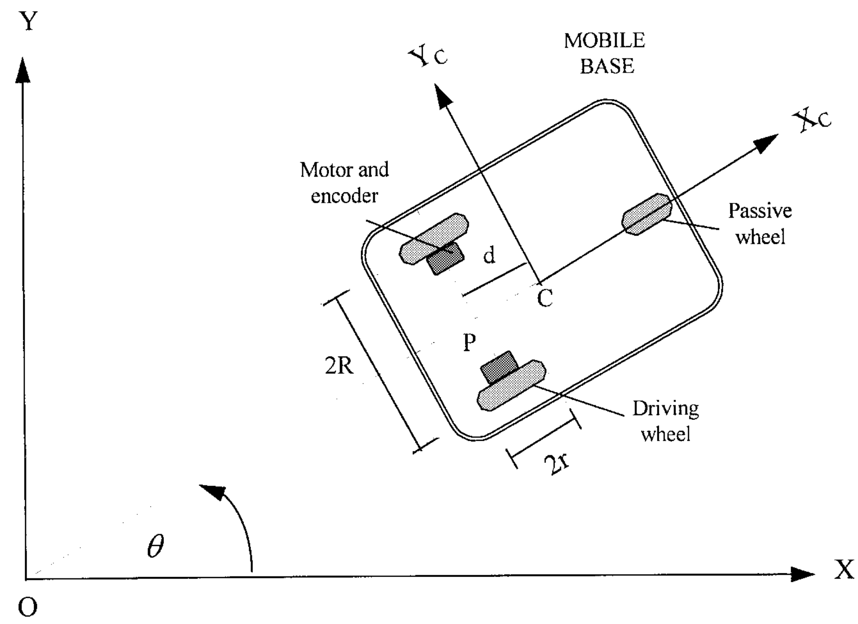

Fig. 1. A nonholonomic mobile platform.

tural properties of the nonholonomic dynamical equations are given including an important "skew-symmetry" property. Section III discusses the nonlinear kinematic-NN backstepping controller as applied to the tracking problem. Stability is proved by Lyapunov theory. Section IV presents some simulation results. Finally, Section V gives some concluding remarks.

II. Preliminaries

A. A Nonholonomic Mobile Robot

A mobile robot system having an n-dimensional configuration space $\mathcal{C}$ with generalized coordinates $(q_1, \cdots, q_n)$ and subject to m constraints can be described by [13] and [20]

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

where $\boldsymbol{M}(q) \in \Re^{n\times n}$ is a symmetric, positive definite inertia matrix, $\boldsymbol{V}_m(q,\dot{q}) \in \Re^{n\times n}$ is the centripetal and coriolis matrix, $\boldsymbol{F}(\dot{q}) \in \Re^{n\times 1}$ denotes the surface friction, $\boldsymbol{G}(q) \in \Re^{n\times 1}$ is the gravitational vector, τ_d denotes bounded unknown disturbances including unstructured unmodeled dynamics, $\boldsymbol{B}(q) \in \Re^{n\times r}$ is the input transformation matrix, $\tau \in \Re^{n\times 1}$ is the input vector, $\boldsymbol{A}(q) \in \Re^{m\times n}$ 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:

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

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

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

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

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

The mobile robot shown in Fig. 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 where xc, yc are the coordinates of the center of mass of the vehicle, and is the orientation of the basis $\{C, XC, YC\}$ 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 nonslipping [1], [21]

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

It is easy to verify that $\boldsymbol{S}(q)$ is given by

$$\boldsymbol{S}(q) = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix}. \tag{6}$$

The kinematic equations of motion (4) of $\boldsymbol{C}$ in terms of its linear velocity and angular velocity are

$$v = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{7}$$

where $|v_1| \leq V_{\max}$ and $|v_2| \leq W_{\max}$. $V_{\max}$ and $W_{\max}$ are the maximum linear and angular velocities of the mobile robot. System (7) is called the steering system of the vehicle.

The Lagrange formalism is used to derived the dynamic equations of the mobile robot. In this case $\boldsymbol{G}(q) = 0$, 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 is given by [13]

$$k_i = \tfrac{1}{2} m_i v_i^T + \tfrac{1}{2}\omega_i^T I_i \omega_i$$
$$K = \sum_{i=1}^{n_i} k_i$$
$$K = \tfrac{1}{2}\dot{q}^T \boldsymbol{M}(q)\dot{q}. \tag{8}$$

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

$$\boldsymbol{M}(q) = \begin{bmatrix} m & 0 & md\sin\theta \\ 0 & m & -md\cos\theta \\ md\sin\theta & -md\cos\theta & I \end{bmatrix}$$
$$\boldsymbol{V_m}(q,\dot{q}) = \begin{bmatrix} 0 & 0 & md\dot{\theta}\cos\theta \\ 0 & 0 & md\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix}$$
$$\boldsymbol{G}(q) = \boldsymbol{0}, \quad \boldsymbol{B}(q) = \frac{1}{r}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ R & -R \end{bmatrix}$$
$$\tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}, \quad \boldsymbol{A}^T(q) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ -d \end{bmatrix}$$
$$\lambda = -m(\dot{x}_c\cos\theta + \dot{y}_c\sin\theta)\dot{\theta}. \tag{9}$$

Similar dynamical models have been reported in the literature; for instance in [21] the mass and inertia of the driving wheels are considered explicitly.

B. Structural Properties of a Mobile Platform

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

$$\dot{q} = \boldsymbol{S}v \tag{10}$$
$$\boldsymbol{S}^T\boldsymbol{M}\boldsymbol{S}\dot{v} + \boldsymbol{S}^T(\boldsymbol{M}\dot{\boldsymbol{S}} + \boldsymbol{V_m}\boldsymbol{S})v + \overline{\boldsymbol{F}} + \overline{\tau}_d = \boldsymbol{S}^T\boldsymbol{B}\tau \tag{11}$$

where $v(t) \in \Re^{n-m}$ is a velocity vector. By appropriate definitions we can rewrite (11) as follows:

$$\overline{\boldsymbol{M}}(q)dotv + \overline{\boldsymbol{V}}_{\boldsymbol{m}}(q,\dot{q})v + \overline{\boldsymbol{F}}(v) + \overline{\tau}_d = \overline{\boldsymbol{B}}\tau \tag{12.a}$$
$$\overline{\tau} \equiv \overline{\boldsymbol{B}}\tau \tag{12.b}$$

where $\overline{\boldsymbol{M}}(q) \in \Re^{r\times r}$ is a symmetric positive definite inertia matrix, $\overline{\boldsymbol{V}}_{\boldsymbol{m}}(q,\dot{q}) \in \Re^{r\times r}$ is the centripetal and coriolis matrix, $\overline{\boldsymbol{F}}(v) \in \Re^{r\times 1}$ is the surface friction, $\overline{\tau}_d$ denotes bounded unknown disturbances including unstructured unmodeled dynamics, and $\overline{\tau} \in \Re^{r\times 1}$ is the input vector. If $r = n - m$, it is easy to verify that $\overline{\boldsymbol{B}}$ is a constant nonsingular matrix that depends on the distance between the driving wheels $\boldsymbol{R}$ and the radius of the wheel $\boldsymbol{r}$ (see Fig. 1). Equation (12) describes the behavior of the nonholonomic system in a new set of *local* coordinates, i.e., $\boldsymbol{S}(q)$ is a Jacobian matrix that transforms velocities in mobile base coordinates v to velocities in Cartesian coordinates $\dot{q}$. Therefore, the properties of the original dynamics hold for the new set of coordinates [13].

Boundedness: $\overline{\boldsymbol{M}}(q)$, the norm of the $\overline{\boldsymbol{V}}_{\boldsymbol{m}}(q,\dot{q})$, and are bounded.

Skew-Symmetry: The matrix $\dot{\overline{\boldsymbol{M}}} - 2\overline{\boldsymbol{V}}_{\boldsymbol{m}}$ is skew symmetric.

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

$$\dot{\overline{\boldsymbol{M}}} = \dot{\boldsymbol{S}}^T\boldsymbol{M}\boldsymbol{S} + \boldsymbol{S}^T\dot{\boldsymbol{M}}\boldsymbol{S} + \boldsymbol{S}^T\boldsymbol{M}\dot{\boldsymbol{S}}$$
$$\overline{\boldsymbol{V}}_{\boldsymbol{m}} = \boldsymbol{S}^T\boldsymbol{M}\dot{\boldsymbol{S}}^T\boldsymbol{V_m}\boldsymbol{S}.$$

Since $\dot{\boldsymbol{M}} - 2\boldsymbol{V_m}$ is skew-symmetric [13], it is straightforward to show that (13) is skew-symmetric also

$$\dot{\overline{\boldsymbol{M}}} - 2\overline{\boldsymbol{V}}_{\boldsymbol{m}} = \dot{\boldsymbol{S}}^T\boldsymbol{M}\boldsymbol{S} - (\dot{\boldsymbol{S}}^T\boldsymbol{M}\boldsymbol{S})^T + \boldsymbol{S}^T(\dot{\boldsymbol{M}} - 2\boldsymbol{V_m})\boldsymbol{S}. \tag{13}$$

□

C. Feedforward Neural Networks

A "two-layer" feedforward NN in Fig. 2 has two layers of adjustable weights. The NN output y is a vector with m components that are determined in terms of the n components of the input vector x by the formula

$$y_i = \sum_{j=1}^{N_b}\left[w_{ij}\sigma\left(\sum_{k=1}^{n} v_{jk}x_k + \theta_{vj}\right) + \theta_{wi}\right]$$
$$i = 1,\cdots,m \tag{14.a}$$

where $\sigma(\cdot)$ are the activation functions and N_h is the number of *hidden-layer neurons*. The inputs-to-hidden-layer interconnection weights are denoted by v_{jk} and the hidden-layer-to-outputs interconnection weights by w_{ij}. The threshold offsets are denoted by θ_{vj}, θ_{wi}.

Many different activation functions $\sigma(\cdot)$ are in common use, including sigmoid, hyperbolic tangent, and Gaussian. In this work we shall use the sigmoid activation function

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{14.b}$$

By collecting all the NN weights v_{jk}, w_{ij} into matrices of weights $\boldsymbol{V}^T, \boldsymbol{W}^T$, one can write the NN equation is terms of vectors as

$$y = \boldsymbol{W}^T\sigma(\boldsymbol{V}^Tx) \tag{15}$$

with the vector of activation functions defined by $\sigma(\boldsymbol{z}) = [\sigma(z_1)\cdots\sigma(z_n)]^T$ for a vector $\boldsymbol{z} \in \Re^n$. The thresholds are included as the first columns of the weight matrices. To accommodate this the vectors x and $\sigma(\cdot)$ need to be augmented by placing a "1" as their first element (e.g., $x \equiv [1\ x_1\ x_2\ x_3 \cdot x_n]^T$). Any tuning of $\boldsymbol{W}$ and $\boldsymbol{V}$ then includes tuning of the thresholds as well.

The main property of a NN we shall be concerned with for controls purposes is the *function approximation property* [6], [8]. Let $f(x)$ be a smooth function from $\Re^n$ to $\Re^m$. Then, it can be shown that, as long as x is restricted to a compact set

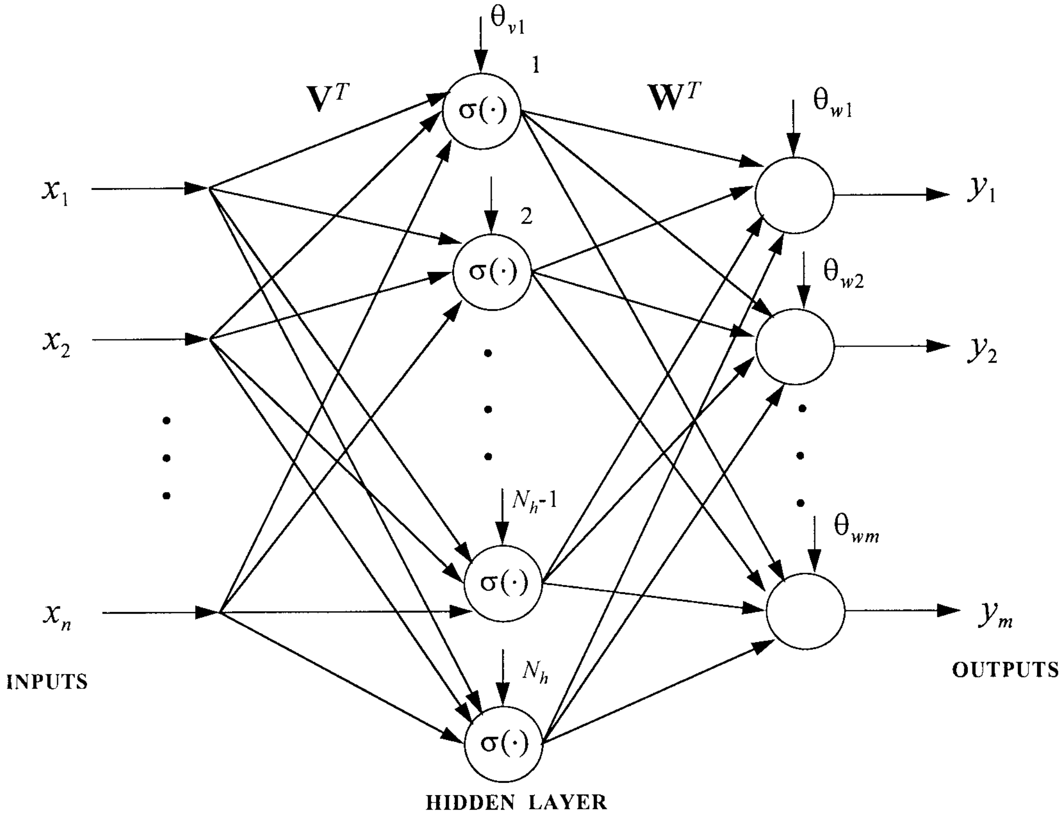

Fig. 2. Multilayer feedforward NN.

U_x of $\Re^n$, for some number of hidden layer neurons N_h, there exist weights and thresholds such that one has

$$f(x) = \boldsymbol{W}^T \sigma(\boldsymbol{V}^T x) + \epsilon. \tag{16}$$

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*. In fact, for any choice of a positive number ϵ_N, one can find a NN such that $\epsilon < \epsilon_N$ in U_x.

For controls purposes, all one needs to know is that, for a specified value of ϵ_N these *ideal* approximating NN weights exist. Then, an estimate of $f(x)$ can be given by

$$\hat{f}(x) = \hat{\boldsymbol{W}}^T \sigma(\hat{\boldsymbol{V}}^T x) \tag{17}$$

where $\hat{\boldsymbol{W}}$ and $\hat{\boldsymbol{V}}$ are estimates of the ideal NN weights that are provided by some on-line weight tuning algorithms.

A common weight tuning algorithm is the gradient algorithm based on the *backpropagated* error [27], where the NN is training off-line to match specified exemplar pairs (x_d, y_d), with x_d the ideal NN input that yields the desired NN output y_d. The continuous-time version of the backpropagation algorithm for the two-layer NN is given by

$$\begin{aligned} \dot{\hat{\boldsymbol{W}}} &= \boldsymbol{F} \sigma(\hat{\boldsymbol{V}}^T x_d) E^T \\ \dot{\hat{\boldsymbol{V}}} &= \boldsymbol{G} x_d (\hat{\sigma}'^T \hat{\boldsymbol{W}} E)^T \end{aligned} \tag{18}$$

where $\boldsymbol{F}$, $\boldsymbol{G}$ are positive definite design parameter matrices governing the speed of convergence of the algorithm. The backpropagated error E is selected as the desired NN output minus the actual NN output $E = y_d - y$. For the scalar sigmoid activation function (14.b), for instance, the hidden-layer output gradient is

$$\frac{\partial \sigma}{\partial z} = \sigma(z)[1 - \sigma(z)] \equiv \sigma'. \tag{19}$$

The hidden-layer output gradient or jacobian may be explicitly computed; for the sigmoid activation functions, it is

$$\hat{\sigma}' \equiv \mathrm{diag}\{\sigma(\hat{\boldsymbol{V}}^T x_d)\}[\boldsymbol{I} - \mathrm{diag}\{\sigma(\hat{\boldsymbol{V}}^T x_d)\}] \tag{20}$$

where $\boldsymbol{I}$ denotes the identity matrix, and $\mathrm{diag}\{z\}$ means a diagonal matrix whose diagonal elements are the components of vector z. One major problem in using backprop tuning in direct closed-loop control applications is that the required gradients [Jacobian (20)] depend on the unknown plant being controlled; this make them impossible or very difficult to compute. Extensive work on confronting this problem has been done by a number of authors using a variety of techniques, see for instance [14]–[18] and the references therein.

III. Control Design

An important result in controllability of nonholonomic systems states that the steering system (10) is *controllable* regardless the nature of the constraints [3]. A review of the controllability properties for the kinematic steering system (10) can be found in [7]. The complete dynamics (10), (11) consist of the kinematic steering system (10) plus some extra dynamics (11).

Backstepping Design: Many approaches exist to selecting a velocity control $v(t)$ for the steering system (10). In this section, we desire to convert such a prescribed control $v(t)$ into a torque control $\tau(t)$ for the actual physical cart. Therefore, our objective is to design an NN control algorithm so that (10), (11) exhibits the desired behavior motivating the specific choice of the velocity $v(t)$.

The nonholonomic navigation problem of steering $v(t)$ 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 stabilization about a desired posture may be solved under the same control structure.

The smooth steering system control, denoted by v_c, can be found by any technique in the literature. Using the algorithm to be derived and proved in Section III-C, 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

$$\begin{aligned} \dot{x}_r &= v_r \cos\theta_r \\ \dot{y}_r &= v_r \sin\theta \\ \dot{\theta}_r &= w_r \\ q_r &= [x_r \quad y_r \quad \theta_r]^T, \qquad v_r = [v_r \quad w_r]^T \end{aligned} \tag{21}$$

with $v_r > 0$ for all t, find a smooth velocity control $v_c = f_c(e_p, v_r, K)$ such that $\lim_{t\to\infty}(q_r - q) = 0$,, where ϵ_p, v_r, and K are the tracking position error, the reference velocity vector and the control gain vector, respectively. Then compute the torque input $\tau(t)$ for (1), such that $v \to v_c$ as $t \to \infty$.

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

Point Stabilization: Given an arbitrary configuration q_r, find a smooth time-varying velocity control input $v_c = f_c(e_p, v_r, K, t)$ such that $\lim_{t\to\infty}(q_r - q) = 0$.. Then compute the torque input $\tau(t)$ for (1), such that $v \to v_c$ 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 which requires that only the angular velocity change in order to decrease the distance between a given geometric path and the mobile robot. *Point stabilization* can be solve using the same controller, but in this case the input control velocities are time varying.

A. NN Control Design for Tracking a Reference Trajectory

The structure for the tracking control system to be derived in Section III-C is presented in Fig. 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 contribution of this paper lies in deriving a suitable $\tau(t)$ from a specific $v_c(t)$ that controls the steering system (10). In the literature, the nonholonomic tracking problem is simplified by neglecting the vehicle dynamics (11) and considering only the steering system (10). That is, a steering system input $v_c(t)$ is determined such that (10) tracks the reference cart trajectory. To compute the vehicle torque $\tau(t)$, it is assumed that there is "perfect velocity tracking" so that $v = v_c$, then (11) is used to compute $\tau(t)$. There are three problems with this approach: first, the perfect velocity tracking assumption does not hold in practice, second, the disturbance τ_d is ignored, and, finally, complete knowledge of the dynamics is needed. A better alternative to this unrealistic approach is the *NN integrator backstepping method* now developed.

To be specific, it is assumed that the solution to the steering system tracking problem in [10] is available. This is denoted as $v_c(t)$. Then, a control $\tau(t)$ for (10), (11) is found that guarantees robust trajectory tracking despite unknown dynamical parameters and bounded unknown disturbances $\overline{\tau}_d(t)$.

The tracking error vector is expressed in the basis of a frame linked to the mobile platform [4], [10] as

$$\begin{aligned} e_p &= \boldsymbol{T}_e(q_r - q) \\ \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} &= \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix}. \end{aligned} \tag{22}$$

An auxiliary velocity control input that achieves tracking for (10) is given by [10]

$$\begin{aligned} v_c &= \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ w_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{bmatrix} \\ v_c &= f_c(e_p, v_r, K). \end{aligned} \tag{23}$$

where $k_1, k_2, k_3 > 0$ are design parameters. If we consider *only* the kinematic model of the mobile robot (4) with velocity input (23), and assume perfect velocity tracking, then the kinematic model is asymptotically stable with respect to a *reference trajectory* (i.e., $e_p \to 0$ as $t \to \infty$) [10], [7].

Given the desired velocity $v_c(t) \in \Re^{n-m}$, define now the auxiliary velocity tracking error as

$$e_c = v_c - v. \tag{24}$$

Differentiating (24) and using (12), the mobile robot dynamics may be written in terms of the velocity tracking error as

$$\overline{\boldsymbol{M}}(a)\dot{e}_c = -\overline{\boldsymbol{V}}_{\boldsymbol{m}}(q,\dot{q})e_c - \overline{\tau} + f(x) + \overline{\tau}_d \tag{25}$$

where the important *nonlinear mobile robot function* is

$$f(x) = \overline{\boldsymbol{M}}(q)\dot{v}_c + \overline{\boldsymbol{V}}_{\boldsymbol{m}}(q,\dot{q})v_c + \overline{\boldsymbol{F}}(v). \tag{26}$$

The vector x required to compute $f(x)$ can be defined as

$$x \equiv [v^T \quad v_c^T \quad \dot{v}_c^T]^T \tag{27}$$

which can be measured.

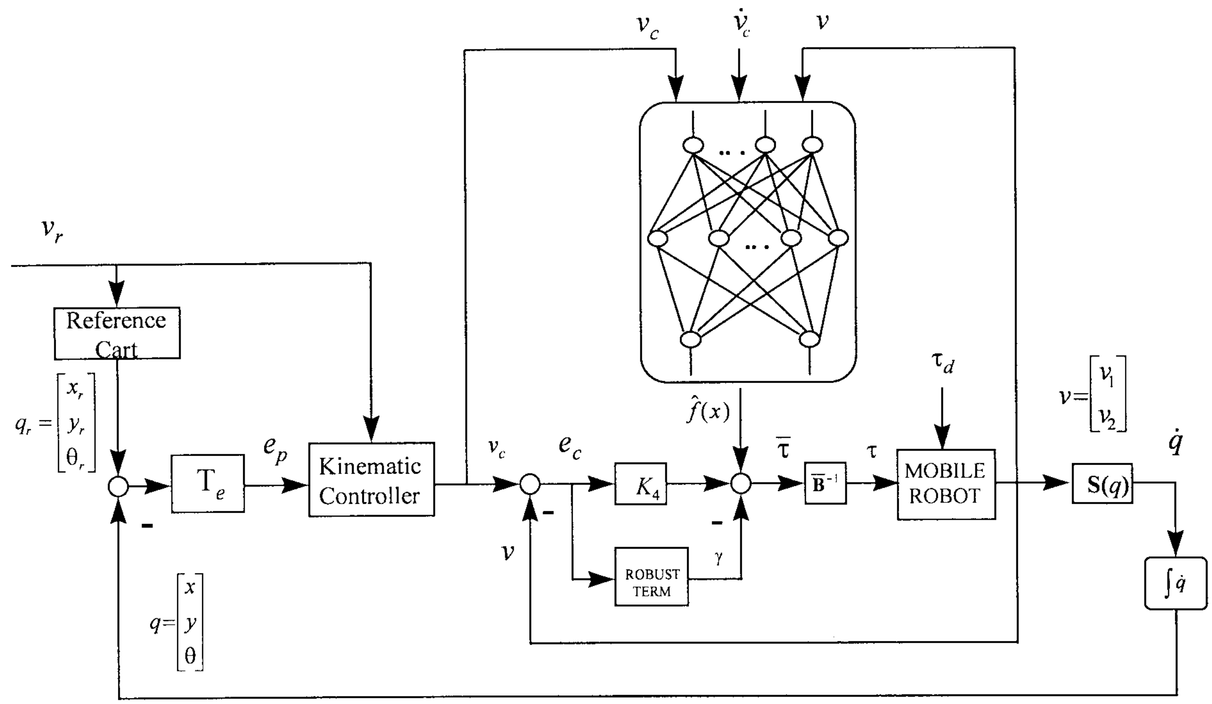

Fig. 3. Tracking by a neural-net control.

Function $f(x)$ 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.

B. Mobile Robot Controller Structure

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

$$\bar{\tau} = \hat{f} + K_4 e_c - \gamma \tag{28}$$

with K_4 a diagonal positive definite gain matrix, and $\hat{f}(x)$ an *estimate* of the robot function $f(x)$ that is provided by the NN. The robustifying signal $\gamma(t)$ is required to compensate the unmodeled unstructured disturbances. Using this control in (25), the closed-loop system becomes

$$\overline{\boldsymbol{M}}\dot{e}_c = -(K_4 + \overline{\boldsymbol{V}}_{\boldsymbol{m}})e_c + \tilde{f} + \bar{\tau}_d + \gamma \tag{29}$$

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

$$\tilde{f} = f - \hat{f}. \tag{30}$$

In computing the control signal, the estimate $\hat{f}$ can be provided by several techniques, including *adaptive control*. The robustifying signal $\gamma(t)$ can be selected by several techniques, including sliding-mode methods and others under the general aegis of *robust control* methods.

C. Neural-Net Controller

By using the controller (28), there is no guarantee that the control $\bar{\tau}$ will make the velocity tracking error small. Thus, the control design problem is to specify a method of selecting the matrix gain K_4, the estimate $\hat{f}$, and the robustifying signal $\gamma(t)$ so that both the error $e_c(t)$ 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 $e_c(t)$ should be in some sense "small enough" because it will affect directly the position tracking error $e_p(t)$. In this section we shall use an NN to compute the estimate $\hat{f}$. A major advantage is that this can always be accomplished, due to the NN approximation property (16). By contrast, in adaptive control approaches it is only possible to proceed if $f(x)$ is linear in the known parameters; moreover, tedious analysis is needed to compute a "regression matrix."

Some definitions are required in order to proceed.

Definition 3.3.1: We say that the solution of a nonlinear system with state $x(t) \in \Re^n$ is *uniformly ultimately bounded (UUB)* if there exists a compact set $U_x \subset \Re^n$ such that for all $x(t_0) = x_0 \in U_x$, there exists a $\delta > 0$ and a number $T(\delta, x_0)$ such that $\|x(t)\| < \delta$ for all $t \geq t_0 + T$.

Definition 3.3.2: We denote by $\|\cdot\|$ any suitable vector norm. When it is required to be specific we denote the p-norm by $\|\cdot\|_p$.

Definition 3.3.3: Given $\boldsymbol{A} = [a_{ij}]$, $\boldsymbol{B} \in \Re^{m\times n}$ the Frobenius norm is defined by

$$\|\boldsymbol{A}\|_F^2 = \mathrm{tr}\{\boldsymbol{A}^T\boldsymbol{A}\} = \sum_{i,j} a_{ij}^2 \tag{31}$$

with $\mathrm{tr}\{\cdot\}$ the trace. The associated inner product is $\langle \boldsymbol{A}, \boldsymbol{B}\rangle_F = \mathrm{tr}\{\boldsymbol{A}^T\boldsymbol{B}\}$. The Frobenius norm cannot be defined as the induced matrix norm for any vector norm, but is *compatible* with the 2-norm so that $\|\boldsymbol{A}\boldsymbol{x}\|_2 \leq \|\boldsymbol{A}\|_F\|\boldsymbol{x}\|_2$, with $\boldsymbol{A} \in \Re^{m\times n}$ and $x \in \Re^n$.

Definition 3.3.4: For notational convenience we define the matrix of all the NN weights as $\hat{\boldsymbol{Z}} \equiv \mathrm{diag}\{\hat{\boldsymbol{W}}, \hat{\boldsymbol{V}}\}$.

Definition 3.3.5: Define the weight estimation errors as $\tilde{\boldsymbol{V}} = \boldsymbol{V} - \hat{\boldsymbol{V}}$, $\tilde{\boldsymbol{W}} = \boldsymbol{W} - \hat{\boldsymbol{W}}$, $\tilde{\boldsymbol{Z}} = \boldsymbol{Z} - \hat{\boldsymbol{Z}}$.

Definition 3.3.6: Define the *hidden-layer output error* for a given x as

$$\tilde{\sigma} = \sigma - \hat{\sigma} = \sigma(\boldsymbol{V}^T x) - \sigma(\hat{\boldsymbol{V}}^T x). \tag{32}$$

The Taylor series expansion of $\sigma(x)$ for a given x may be written as

$$\sigma(\boldsymbol{V}^T x) = \sigma(\hat{\boldsymbol{V}}^T x) + \sigma'(\hat{\boldsymbol{V}}^T x + O(\tilde{\boldsymbol{V}}^T x) \tag{33.a}$$

with

$$\sigma'(\hat{z}) \equiv \left.\frac{\partial \sigma(z)}{\partial z}\right|_{z=\hat{z}} \tag{33.b}$$

the Jacobian matrix and $O(\tilde{\boldsymbol{V}}^T x)$ denoting the higher-order terms in the Taylor series. Denoting $\hat{\sigma}' = \sigma'(\hat{\boldsymbol{V}}^T x)$, we have

$$\tilde{\sigma} = \sigma'(\hat{\boldsymbol{V}}^T x)\tilde{\boldsymbol{V}}^T x + O(\tilde{\boldsymbol{V}}^T x + O(\tilde{\boldsymbol{V}}^T x). \tag{33.c}$$

The importance of this equation is that it replaces $\tilde{\sigma}$, which is nonlinear in $\tilde{\boldsymbol{V}}$, by an expression linear in $\tilde{\boldsymbol{V}}$ plus higher-order terms. This will allow us to determine tuning algorithms for $\hat{\boldsymbol{V}}$ in subsequent derivations. Different bounds may be put on the Taylor series higher-order terms depending on the choice for the activation functions $\sigma(\cdot)$.

The following mild assumptions always hold in practical applications.

Assumption 3.3.1: On any compact subset of $\Re^n$, the ideal NN weights are bounded by known positive values so that $\|\boldsymbol{V}\|_F \le V_M$, $\|\boldsymbol{W}\|_F \le W_M$, or $\|\boldsymbol{Z}\|_F \le Z_M$ with Z_M known.

Assumption 3.3.2: The desired reference trajectory is bounded so that $\|q_r\| \le q_M$ with q_M a known scalar bound, and the disturbances are bounded so that $\|\vec{\tau}_d\| \le d_M$.

Lemma 3.3.1 (Bound on NN Input x): For each time $t, x(t)$ in (27) is bounded by

$$\|x\| \le q_M + c_0\|e_c(t_0)\| + c_2\|e_c(t)\| \le c_1 + c_2\|e_c(t)\| \tag{34}$$

for computable positive constants c_i.

Lemma 3.3.2 (Bounds on Taylor Series Higher-Order Terms): For sigmoid activation functions, the higher-order terms in the Taylor series (33) are bounded by

$$\|O(\tilde{\boldsymbol{V}}^T x)\| \le c_3 + c_4\|\tilde{\boldsymbol{V}}\|_F + c_5\|\tilde{\boldsymbol{V}}\|_F\|e_c\| \tag{35}$$

for computable positive constants c_i.

We will use an NN to approximate $f(x)$ for computing the control in (28). By placing into (28) the NN approximation equation given by (17), the control input then becomes

$$\overline{\tau} = \hat{\boldsymbol{W}}^T \sigma(\hat{\boldsymbol{V}}^T x) + K_4 e_c - \gamma \tag{36}$$

with $\gamma(t)$ a function to be detailed subsequently that provides robustness in the face of robot kinematics and higher-order terms in the Taylor series.

Using this controller, the closed-loop velocity error dynamics become

$$\begin{aligned}\overline{\boldsymbol{M}}\dot{e}_c = &-(K_4 + \overline{\boldsymbol{V}}_m)e_c + \boldsymbol{W}^T\sigma(\boldsymbol{V}_x^T) \\ &- \hat{\boldsymbol{W}}^T\sigma(\hat{\boldsymbol{V}}^T x) + (\epsilon + \overline{\tau}_d) + \gamma.\end{aligned} \tag{37.a}$$

Adding and subtracting $\boldsymbol{W}^T\hat{\sigma}$ yields

$$\overline{\boldsymbol{M}}\dot{e}_c = -(K_4 + \overline{\boldsymbol{V}}_{\boldsymbol{m}})e_c + \tilde{\boldsymbol{W}}^T\hat{\sigma} + \boldsymbol{W}^T\tilde{\sigma} + (\epsilon + \overline{\tau}_d) + \gamma \tag{37.b}$$

with $\hat{\sigma}, \tilde{\sigma}$ defining in (32). Adding and subtracting now $\hat{\boldsymbol{W}}^T\tilde{\sigma}$ yields

$$\begin{aligned}\overline{\boldsymbol{M}}\dot{e}_c = &-(K_4 + \overline{\boldsymbol{V}}_{\boldsymbol{m}})e_c + \tilde{\boldsymbol{W}}^T\hat{\sigma} + \hat{\boldsymbol{W}}^T\tilde{\sigma} + \tilde{\boldsymbol{W}}^T\tilde{\sigma} \\ &+ (\epsilon + \overline{\tau}_d) + \gamma.\end{aligned} \tag{37.c}$$

The key step is the use now of the Taylor series approximation (33.c) for $\tilde{\sigma}$, according to which the error system is

$$\begin{aligned}\overline{\boldsymbol{M}}\dot{e}_c = &-(K_4 + \overline{\boldsymbol{V}}_{\boldsymbol{m}})e_c + \tilde{\boldsymbol{W}}^T(\hat{\sigma} - \hat{\sigma}'\hat{\boldsymbol{V}}^T x) \\ &+ \hat{\boldsymbol{W}}^T\hat{\sigma}'\tilde{\boldsymbol{V}}^T x + x + w + \gamma\end{aligned} \tag{38}$$

where the disturbance terms are

$$w(t) = \tilde{\boldsymbol{W}}^T\hat{\sigma}'\boldsymbol{V}^T x + \boldsymbol{W}^T O(\tilde{\boldsymbol{V}}^T x) + \epsilon + \overline{\tau}_d. \tag{39}$$

It is important to note that the NN reconstruction error $\epsilon(x)$, the disturbance $\overline{\tau}_d$, and the higher-order terms in the Taylor series expansion of $f(x)$ all have exactly the same influence as disturbances in the error system. The next bound is required. Its importance it is in allowing one to overbound $w(t)$ at each time by a known computable function.

Lemma 3.3.3 (Bounds on the Disturbance Term): The disturbance term (39) is bounded according to

$$\|w(t)\| \le (\epsilon_N + d_M + c_3 Z_M) + c_6 Z_M\|\tilde{\boldsymbol{Z}}\|_F + c_7 Z_M\|\tilde{Z}\|_F\|e_c\|$$

or

$$\|w(t)\| \le C_0 + C_1\|\tilde{\boldsymbol{Z}}\|_F + C_2\|\tilde{\boldsymbol{Z}}\|_F\|e_c\| \tag{40}$$

with C_i known positive constants. Note that C_0 becomes larger with increases in the NN estimation error ϵ and the mobile robot dynamics disturbances $\overline{\tau}_d$. Proofs of Lemmas 3.3.1–3 are omitted here, details are discovered in [14].

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

Theorem 3.3.1: Given a nonholonomic system (10), (11) with n generalized coordinates q, m independent constraints, $r = n - m$ actuators, let the following assumptions hold.

Assumption 3.3.3: The reference linear velocity is constant, bounded, and $v_r > 0$ for all t. The angular velocity w_r is bounded.

Assumption 3.3.4: A smooth auxiliary velocity control input $v_c(t)$ is prescribed that solves the trajectory tracking problem for the steering system (10), neglecting the dynamics (11). A sample v_c [10] is given by (23).

Assumption 3.3.5: $K = [k_1\ k_2\ k_3]^T$ is a vector of positive constants.

Assumption 3.3.6: $K_4 = k_4\boldsymbol{I}$, where k_4 is a sufficiently large positive constant.

Take the control $\overline{\tau} \in \Re^r$ for (12) as (36) with robustifying term

$$\gamma(t) = -K_z(\|\hat{Z}\|_F + Z_M)e_c - e_c \tag{41}$$

and gain

$$K_z > C_2 \tag{42}$$

with C_2 the known constant in (40). Let NN weight tuning be provided by (43). Then, for large enough control gain K_4, the velocity tracking error $e_c(t)$, the position error $e_p(t)$, and the NN weight estimates $\hat{\boldsymbol{V}}, \hat{\boldsymbol{W}}$ are UUB. Moreover, the velocity tracking error may be kept as small as desired by increasing the gain K_4

$$\begin{aligned}\dot{\hat{\boldsymbol{W}}} &= \boldsymbol{F}\hat{\sigma}e_c^T - \boldsymbol{F}\hat{\sigma}'\hat{\boldsymbol{V}}^T x e_c^T - \kappa\boldsymbol{F}\|e_c\|\hat{\boldsymbol{W}}\\ \dot{\hat{\boldsymbol{V}}} &= \boldsymbol{G}x(\hat{\sigma}'^T\hat{\boldsymbol{W}}e_c)^T - \kappa\boldsymbol{G}\|e_c\|\hat{\boldsymbol{V}}\end{aligned} \tag{43}$$

where $\boldsymbol{F}, \boldsymbol{G}$ are positive definite design parameter matrices, $\kappa > 0$ and the hidden-layer gradient or Jacobian $\hat{\sigma}'$ is easily computed in terms of measurable signals—for the sigmoid activation function it is given by

$$\hat{\sigma}' \equiv \mathrm{diag}\{\sigma(\hat{\boldsymbol{V}}^T x)\}[\boldsymbol{I} - \mathrm{diag}\{\sigma(\hat{\boldsymbol{V}}^T x)\}] \tag{44}$$

which is just (20) with the constant exemplar x_d replaced by the time function $x(t)$.

Proof: See the Appendix.

The first terms of (43) 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 weights estimates. The middle term in (43) is a *novel term* needed to prove stability.

Theorem 3.3.1 guarantees that the NN weight estimation errors are bounded, and the tracking error can be made arbitrarily small. As time passes the NN updates its weights learning the dynamics of the mobile robot *on-line*.

D. Robustness Considerations

In practical situations the velocity and tracking 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.

The robust-adaptive controller designed in the previous section consists of two subsystems: 1) a kinematic controller and 2) a dynamic controller. The NN-based dynamic controller provides the required torques, so that the mobile robot's velocity tracks a reference velocity input.

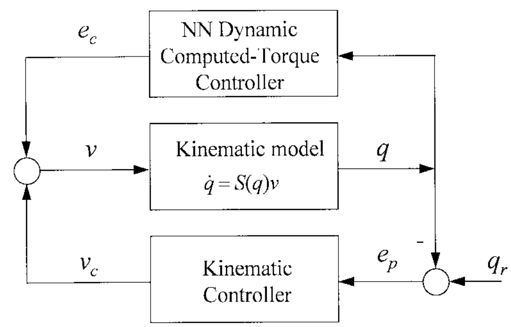

Fig. 4. Closed-loop model of a nonholonomic system.

As "perfect velocity tracking" does not hold in practice, the dynamic controller generates a velocity error e_c which is bounded by some know constant (Theorem 3.3.1). This error can be seen as a disturbance for the kinematic system, see Fig. 4.

The closed-loop kinematic system becomes

$$\begin{aligned}\dot{x} &= (v_{c1} + e_4)\cos\theta - d\sin(v_{c2} + e_5)\\ \dot{y} &= (v_{c1} + e_4)\sin\theta + d\cos(v_{c2} + e_5)\\ \dot{\theta} &= v_{c2} + e_5\end{aligned}$$

where $e_c = [e_4\ e_5]^T$ and $v_c = [v_{c1}\ v_{c2}]^T$ denote the velocity tracking error and the desired velocity control input, respectively. The disturbance e_c satisfies the matching condition [28] i.e., the nonholonomic constraint (5) is not violated. Then, by using standard Lyapunov methods it can be shown that along a system's solution $\|e_p\|$ is bounded, and thus $\|\dot{e}_p\|$ is bounded. The norm of the velocity error affects directly to the norm of the position error. Note that the norm of the velocity error $\|e_c\|$ depends on the NN functional approximation error ϵ and the matrix K_4. Since $\|e_c\|$ can be made arbitrarily small then $\|e_p\|$ can be made arbitrarily small.

IV. Simulation Results

We should like to illustrate the NN control scheme presented in Fig. 3 and compare its performance with two different approaches. For this purpose, three controllers have been implemented and simulated in MATLAB™: 1) a controller that assumes "perfect velocity tracking;" 2) a controller that assumes complete knowledge of the mobile robot dynamics; and 3) an NN backstepping controller which requires no knowledge of the dynamics, not even their structure. We took the vehicle parameters (Fig. 1) as $m = 10$ kg, $I = 5$ kg-m^2, $R = 0.5$ m, $r = 0.05$ m, and $v_r = 0.5$ m/s. The reference trajectory is a straight line with initial coordinates and slope of (1, 2) and 26.56, respectively. The controller gains were chosen so that the closed-loop system exhibits a critical damping behavior: $K = [10\ 5\ 4]^T$, $K_4 = \mathrm{diag}\{40, 40\}$. For the NN, we selected the sigmoid activation functions with $N_h = 10$ hidden-layer neurons, $\boldsymbol{F} = \boldsymbol{G} = \mathrm{diag}\{10, 10\}$ and $\kappa = 0.1$.

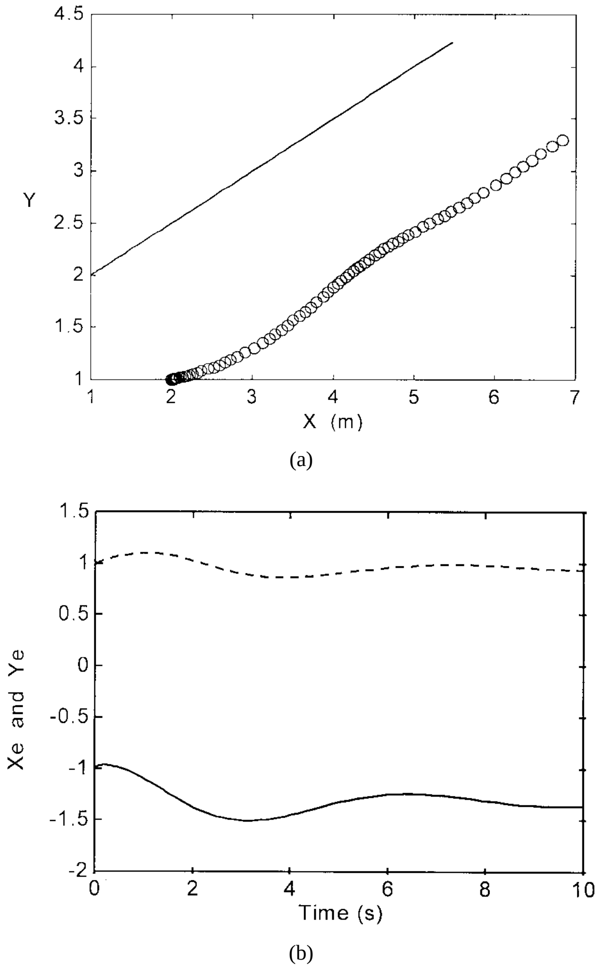

Fig. 5. Perfect velocity tracking assumption. (a) Desired (-) and actual (o) trajectories. Mobile robot initial pisition (2,1), $\Theta_0 = 10°$. (b) Position errors: Xe (—) and Ye (- -).

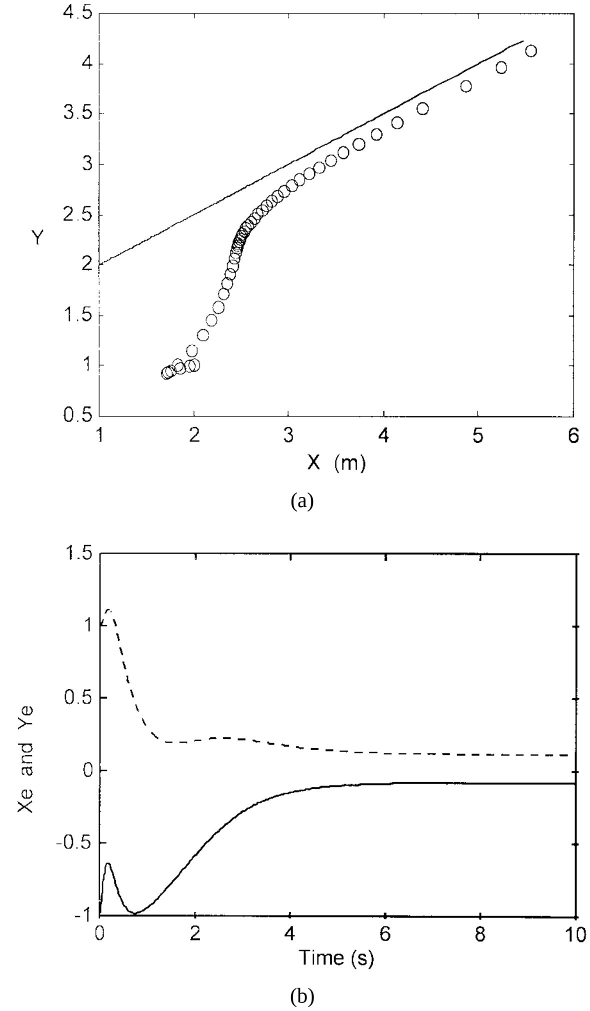

Fig. 6. Conventional computed-torque controller. (a) Desired (-) and actual (o) trajectories. Mobile robot initial pisition (2,1), $\Theta_0 = 10°$. (b) Position errors: Xe (—) and Ye (- -).

A. 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 Fig. 5. 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.

B. Conventional Computed-Torque Controller

The response with this controller is shown in Fig. 6. Since bounded unmodeled disturbances and friction 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 in order 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.

C. NN Backstepping Controller

The response with this controller is shown in Fig. 7. Bounded unmodeled disturbances and nonsymmetric friction were included in this case. It is clear that the performance of the system has been improved with respect to the previous 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 dynamics and disturbances. The validity of the NN controller has been evidently verified.

In both cases 4.2 and 4.3, the mobile base *maneuvers*, i.e., exhibits forward and backward motions (Figs. 6–7), to track the reference trajectory. Note that there is no path planning involved—the mobile base naturally describes a path that satisfies the nonholonomic constraints.

V. CONCLUSIONS

A stable control algorithm capable of dealing with the three basic nonholonomic navigation problems, and that does not require knowledge of the cart dynamics has been derived using an NN backstepping approach. This feedback servo-control scheme is valid as long as the velocity control inputs are

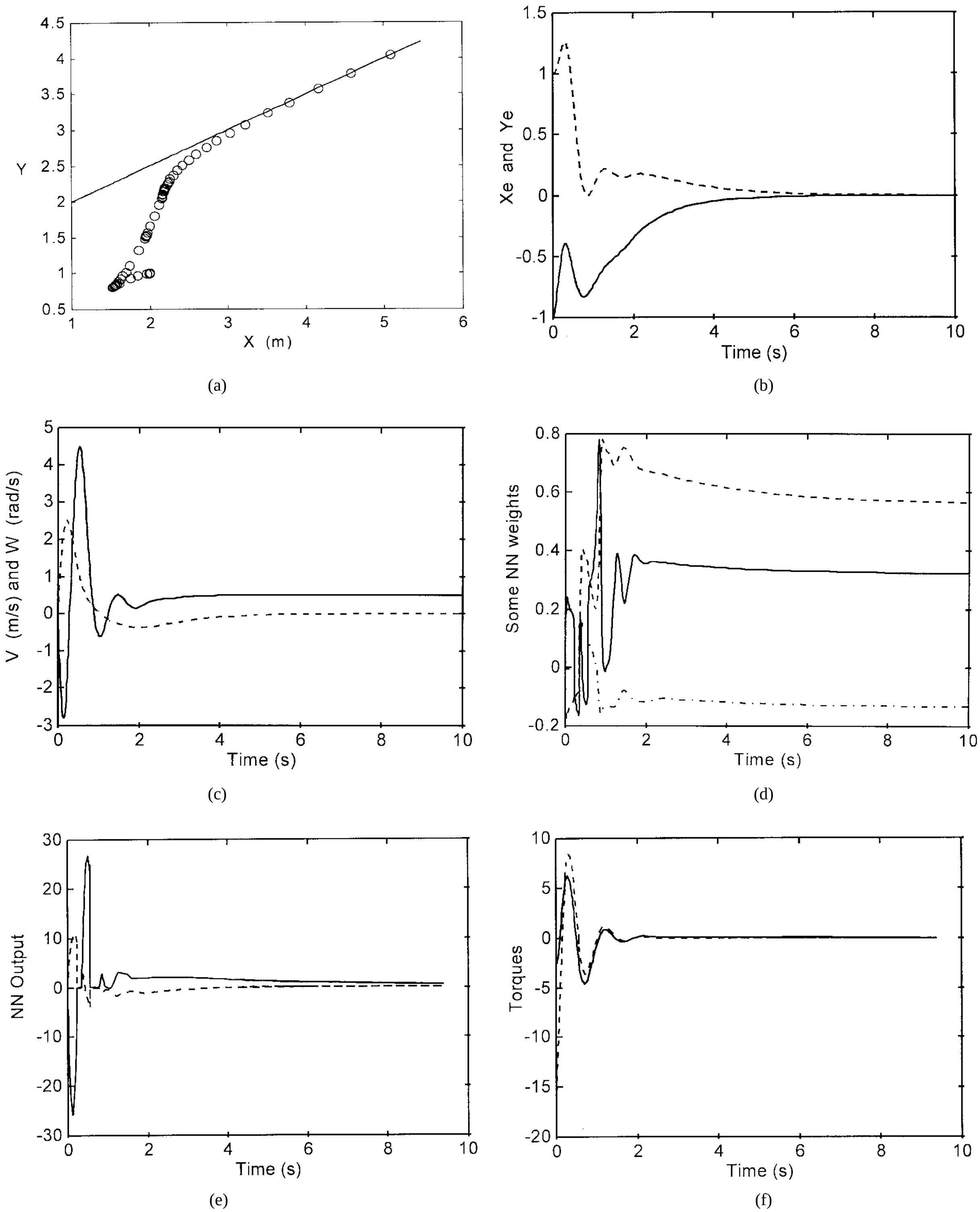

Fig. 7. NN backstepping controller: (a) mobile robot trajectory, (b) position errors. (c) position error, (d) some NN weights. (e) NN outputs, (f) torques.

smooth and bounded, and the disturbances acting on actual cart are bounded.

A key point in developing intelligent systems is the *reusability* of the low-level control algorithms, i.e., the same control algorithm works if the behavior or goal of the system has been modified. This is the case of the control structure reported in this paper. Section III-C considers the case of *trajectory tracking* behavior. Redefining the control velocity input v_c in that section, one may generate a different stable behavior, for instance *path following* behavior, without changing the structure of the controller. Moreover, if the mobile robot is modified or even replaced, the NN controller is still valid.

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, an NN controller with guaranteed performance has been derived.

In summary, an NN dynamic controller together with a well-designed kinematic controller may improve the performance of the mobile robot drastically. There is not need of *a priori* information of the dynamic parameters of the mobile robot, because the NN learns them on-the-fly.

APPENDIX
PROOF OF THEOREM 3.3.1

Let the approximation property (16) hold with a given accuracy ϵ_N for all x in the compact set U_x. Consider the following Lyapunov function candidate:

$$V = k_1(e_1^2 + e_2^2) + 2k_3 v_r(1 - \cos e_3) + V_1 \tag{45}$$

where

$$V_1 = \tfrac{1}{2}[e_c^T \overline{\boldsymbol{M}} e_c + \mathrm{tr}\{\tilde{\boldsymbol{W}}^T \boldsymbol{F}^{-1} \tilde{\boldsymbol{W}}\} + \mathrm{tr}\{\tilde{\boldsymbol{V}}^T \boldsymbol{G}^{-1} \tilde{\boldsymbol{V}}\}]. \tag{46}$$

Differentiating yields

$$\dot{V} = 2k_1(e_1\dot{e}_1 + e_2\dot{e}_2) + 2k_3 v_r \dot{e}_3 \sin e_3 + \dot{V}_1. \tag{47}$$

Differentiating V_1, and substituting now from the error system (38) we obtain

$$\begin{aligned} \dot{V}_1 = &-e_c^T K_4 e_c + \tfrac{1}{2} e_c^T(\dot{\overline{\boldsymbol{M}}} - 2\overline{\boldsymbol{V}}_{\boldsymbol{m}})e_c \\ &+ \mathrm{tr}\{\tilde{\boldsymbol{W}}^T(\boldsymbol{F}^{-1}\dot{\tilde{\boldsymbol{W}}} + \hat{\sigma} e_c^T - \hat{\sigma}'\hat{\boldsymbol{V}}^T x e_c^T)\} \\ &+ \mathrm{tr}\{\tilde{\boldsymbol{V}}^T(\boldsymbol{G}^{-1}\dot{\tilde{\boldsymbol{V}}} + x e_c^T \hat{\boldsymbol{W}}^T \hat{\sigma}')\} + e_c^T(w + \gamma). \end{aligned} \tag{48}$$

The skew symmetry property (Section II-B) makes the second term zero, and since $\dot{\tilde{\boldsymbol{W}}} = -\dot{\hat{\boldsymbol{W}}}$, $\dot{\tilde{\boldsymbol{V}}} = -\dot{\hat{\boldsymbol{V}}}$, the tuning rules yield

$$\begin{aligned} \dot{V}_1 &= -e_c^T K_4 e_c + \kappa\|e_c\| \,\mathrm{tr}\{\tilde{\boldsymbol{W}}^T(\boldsymbol{W} - \tilde{\boldsymbol{W}})\} \\ &\quad + \kappa\|e_c\| \,\mathrm{tr}\{\tilde{\boldsymbol{V}}^T(\boldsymbol{V} - \tilde{\boldsymbol{V}})\} + e_c^T(w+\gamma) \\ &= -e_c^T K_4 e_c + \kappa\|e_c\| \,\mathrm{tr}\{\tilde{\boldsymbol{Z}}^T(\boldsymbol{Z} - \tilde{\boldsymbol{Z}})\} \\ &\quad + e_c^T(w+\gamma). \end{aligned} \tag{49}$$

Since

$$\begin{aligned} \mathrm{tr}\{\tilde{\boldsymbol{Z}}^T(\boldsymbol{Z} - \tilde{\boldsymbol{Z}})\} &= \langle \tilde{\boldsymbol{Z}}, \boldsymbol{Z}\rangle_F - \|\tilde{\boldsymbol{Z}}\|_F^2 \\ &\le \|\tilde{\boldsymbol{Z}}\|_F \|\boldsymbol{Z}\|_F - \|\tilde{\boldsymbol{Z}}\|_F^2 \end{aligned} \tag{50}$$

there results

$$\begin{aligned} \dot{V}_1 &\le -e_c^T K_4 e_c - \kappa\|e_c\| \cdot \|\tilde{\boldsymbol{Z}}\|_F(\|tilde\boldsymbol{Z}\|_F - Z_M) \\ &\quad - K_z(\|\tilde{\boldsymbol{Z}}\|_F + Z_M)\|e_c\|^2 + \|e_c\| \cdot \|w\| - e_c^T e_c \\ &\le -K_{4\min}\|e_c\|^2 - \kappa\|e_c\| \cdot \|\tilde{\boldsymbol{Z}}\|_F(\|\tilde{\boldsymbol{Z}}\|_F - Z_M) \\ &\quad - K_z(\|\tilde{\boldsymbol{Z}}\|_F + Z_M)\|e_c\|^2 \\ &\quad + \|e_c\|[C_0 + C_1\|\tilde{\boldsymbol{Z}}\|_F + C_2\|e_c\| \cdot \|\tilde{\boldsymbol{Z}}\|_F] - e_c^T e_c \\ &\le -\|e_c\| \cdot [K_{4\min}\|e_c\| + \kappa\|\tilde{\boldsymbol{Z}}\|_F(\|\tilde{\boldsymbol{Z}}\|_F - Z_M) \\ &\quad - C_0 - C_1\|\tilde{\boldsymbol{Z}}\|_F] - e_c^T e_c \end{aligned} \tag{51}$$

where $K_{4\min}$ is the minimum singular value of K_4, Lemma 3.3.3 was used, and the last inequality holds due to (42).

The velocity tracking error is

$$\begin{aligned} e_c = \begin{bmatrix} e_4 \\ e_5 \end{bmatrix} &= \begin{bmatrix} v_{c1} - v_1 \\ v_{c2} \quad\quad v_2 \end{bmatrix} \\ &= \begin{bmatrix} v_r \cos e_3 + k_1 e_1 - v_1 \\ w_r + k_2 v_r e_2 + k_3 v_r \sin e_3 - v_2 \end{bmatrix} \end{aligned} \tag{52}$$

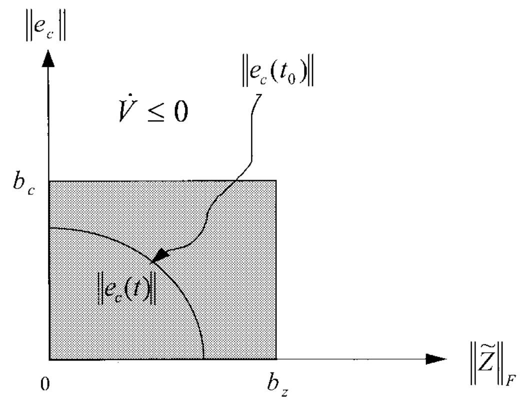

Fig. 8. Graphical representation of Theorem 3.3.1.

by substituting (51) and the derivatives of the position error in (47), we obtain

$$\begin{aligned} \dot{V} &\le 2k_1 e_1(v_2 e_2 - v_1 + v_r \cos e_3) \\ &\quad + 2k_1 e_2(-v_2 e_1 + v_r \sin e_3) + 2k_3 v_r(w_r - v_2)\sin e_3 \\ &\quad - e_c^T e_c - \|e_c\| \cdot [K_{4\min}\|e_c\| + \kappa\|\tilde{\boldsymbol{Z}}\|_F(\|\tilde{\boldsymbol{Z}}\|_F - Z_M) \\ &\quad - C_0 - C_1\|\tilde{\boldsymbol{Z}}\|_F] \end{aligned} \tag{53}$$

by using (52) and defining $k_2 = (k_1/k_3 v_r)$ yield

$$\begin{aligned} \dot{V} &\le -k_1^2 e_1^2 - k_3^2 v_r^2 \sin^2 e_3 - (k_1 e_1 - e_4)^2 \\ &\quad - (k_3 v_r \sin e_3 - e_5)^2 - \|e_c\| \cdot [K_{4\min}\|e_c\| \\ &\quad + \kappa\|\tilde{\boldsymbol{Z}}\|_F(\|\tilde{\boldsymbol{Z}}\|_F - Z_M) - C_0 - C_1\|\tilde{\boldsymbol{Z}}\|_F]. \end{aligned} \tag{54}$$

Since the first four terms in (54) are negative, there results

$$\begin{aligned} \dot{V} &\le -\|e_c\| \cdot \{K_{4\min}\|e_c\| + \kappa\|\tilde{\boldsymbol{Z}}\|_F(\|\tilde{\boldsymbol{Z}}\|_F - Z_M) \\ &\quad - C_0 - C_1\|\tilde{\boldsymbol{Z}}\|_F\}. \end{aligned} \tag{55}$$

Thus, $\dot{V}$ is guaranteed negative as long as the term in braces in (55) is positive. Defining $C_3 = (1/2)(Z_M + (C_1/\kappa))$ and completing the square yields

$$\begin{aligned} &K_{4\min}\|e_c\| + \kappa\|\tilde{\boldsymbol{Z}}\|_F(\|\tilde{\boldsymbol{Z}}\|_F - Z_M) - C_0 - C_1\|\tilde{\boldsymbol{Z}}\|_F \\ &\quad = K_{4\min}\|e_c\| + \kappa(\|\tilde{\boldsymbol{Z}}\|_F - C_3)^2 - C_0 - \kappa C_3^2 \end{aligned}$$

which is guaranteed positive as long as either

$$\|e_c\| > \frac{\kappa C_c^2 + C_0}{K_{4\min}} \equiv b_c \tag{56}$$

or

$$\|\tilde{\boldsymbol{Z}}\|_F > C_3 + \sqrt{C_3^2 + \frac{C_0}{\kappa}} \equiv b_z. \tag{57}$$

Therefore, $\dot{V}$ is negative outside a compact set. According to a standard Lyapunov theory and LaSalle extension, this demonstrates the UUB of both $\|e_c\|$ and $\|\tilde{\boldsymbol{Z}}\|_F$.

Note that $\|e_c\|$ can be kept arbitrarily small by increasing the gain $K_{4\min}$ in (56). Finally, the right-hand sides of (56), (57) can be taken as practical bounds on $e_c(t)$ and the NN weight estimation errors, respectively. Moreover (56) and (57) represent the worst case one can have. In fact, the actual convergence region is a subset of the set given by (56) and (57), see Fig. 8.

Acknowledgment

The authors would like to acknowledge the anonymous referees for their useful comments and constructive suggestions.

References

[1] J. Barraquand and J-C. Latombe, "Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles," in *Proc. IEEE Int. Conf. Robot. Automat.*, Sacramento, CA, 1991, pp. 2328–2335.

[2] A. M. Bloch, M. Reyhanoglu, and N. H. McClamroch, "Control and stabilization of nonholonomic dynamic systems," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 1746–1757, 1992.

[3] G. Campion, B. d'Andréa-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. Berlin, Germany: Springer-Verlag, 1991, pp. 106–124.

[4] C. Canudas de Wit, H. Khennouf, C. Samson, and O. J. Sordalen, "Nonlinear control design for mobile robots," in *Recent Trends in Mobile Robots*, Y. F. Zheng, Ed. Singapore: World Scientific, 1993, pp. 121–156.

[5] F. C. Chen and C. C. Liu, "Adaptively controlling nonlinear continuous-time systems using multilayer neural networks," *IEEE Trans. Automat. Contr.*, vol. 39, no. 6, pp. 1306–1310, 1994.

[6] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Contr. Signals. Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

[7] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics," in *Proc. IEEE Conf. Decision Contr.*, New Orleans, LA, 1995, pp. 3805–3810.

[8] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[9] R. Fierro and F. L. Lewis, "Practical point stabilization of a nonholonomic mobile robot using neural networks," in *Proc. IEEE Conf. Decision Contr.*, Kobe, Japan, 1996, pp. 1722–1727.

[10] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1990, pp. 384–389.

[11] C. M. Kwan and F. L. Lewis, "Robust backstepping control of nonlinear systems using neural networks," in *Proc. European Contr. Conf.*, Rome, 1995, pp. 2772–2777.

[12] F. L. Lewis, "Neural-network control of robot manipulators," *IEEE Expert*, June 1996, pp. 64–75.

[13] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators*, New York: MacMillan, 1993.

[14] F. L. Lewis, A. Yesildirek, and K. Liu "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 7, pp. 1–12, 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. Cambridge, MA: MIT Press, 1991, pp. 115–142.

[16] M. M. Polycarpou and P. O. Ioannu, "Neural networks as on-line approximators of nonlinear systems," in *Proc. IEEE Conf. Dec. Contr.*, Tucson, AZ, 1992, pp. 7–12.

[17] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 400–412, 1994.

[18] N. Sadegh, "A perceptron network for functional identification and control of nonlinear systems," *IEEE Trans. Neural Network*, vol. 4, pp. 982–988, 1993.

[19] C. Samson, "Velocity and torque feedback control of a nonholonomic cart," in *Lecture Notes in Control and Information Science*, C. Canudas de Wit, Ed. Berlin, Germany: Springer-Verlag, 1991, pp. 125–151.

[20] N. Sarkar, X. Yun, and V. Kumar, "Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots," *Int. J. Robot. Res.*, vol. 13, no. 1, pp. 55–69, 1994.

[21] Y. Yamamoto and X. Yun, "Coordinating locomotion and manipulation of a mobile manipulator," in *Recent Trends in Mobile Robots*, Y. F. Zheng, Ed. Singapore: World Scientific, 1993, pp. 157–181.

[22] Y. Zhao and S. BeMent, "Kinematics, dynamics and control of wheeled mobile robots," in *Proc. IEEE Int. Conf. Robot. Automt.*, Nice, France, 1992, pp. 91–96.

[23] K. Berns, R. Dillman, and R. Hofstetter, "An application of a backpropagation network for the control of a tracking behavior," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, Sacramento, CA, 1991, pp. 2426–2431.

[24] K. C. Beom and H. S. Cho, "A sensor-based obstacle avoidance controller for a mobile robot using fuzzy logic and neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, vol. 2, pp. 1470–1475, 1992.

[25] S. Nagata, M. Sekiguchi, and K. Asakawa, "Mobile robot control by a structured hierarchical neural network," *IEEE Contr. Syst.*, Apr. 1990, pp. 69-76.

[26] I. Kolmanovsky and N. H. McClamroch, "Developments in nonholonomic control problems," *IEEE Contr. Syst.*, Dec. 1995, pp. 20–36.

[27] P. J. Werbos, "Backpropagation: Past and future," in *Proc. 1988 Int. Conf. Neural Network*, vol. 1, pp. 1343–1353, 1989.

[28] C. Canudas de Wit and H. Khennouf, "Quasicontinuous stabilizing controllers for nonholonomic systems: Design and robustness considerations," in *Proc. European Contr. Conf.*, Rome, 1995, pp. 2630–2635.

R. Fierro was born in Riobamba, Ecuador. He received the bachelor's degree in electrical engineering at the Escuela Politécnica Nacional, Quito, Ecuador, in 1987. In 1990 he received the Master Degree in Control Engineering from the University of Bradford, U.K. In 1997 he received the Ph.D. degree from The University of Texas at Arlington, where he was employed from 1993 to 1997 as a Graduate Research Assistant at The Automation and Robotics Research Institute.

In 1991, he was a Visiting Research Assistant at the Universidad de Valladolid, Spain. At present, he is an Associate Professor at the Escuela Politécnica Nacional. His areas of interest include hybrid systems, robotics, nonlinear analysis of intelligent control systems (fuzzy and neural), and SCADA systems.

Dr. Fierro is a registered Professional Engineer in Eucador. He received a British Council Scholarship, a Fulbright Scholarship, and an Organization of American States Scholarship. He is a member of the Sigma Xi Research Society.

Frank L. Lewis (S'78–M'81–SM'86–F'94), for a photograph and biography, see this issue, p. 588.