Journal of Intelligent and Robotic Systems **20:** 295–317, 1997.

Robust Practical Point Stabilization of a Nonholonomic Mobile Robot Using Neural Networks*

RAFAEL FIERRO and FRANK L. LEWIS
Automation and Robotics Research Institute, The University of Texas at Arlington, 7300 Jack Newell Blvd. South, Fort Worth, Texas 76118-7115, U.S.A.; e-mail: rfierro@arri.uta.edu flewis@arri.uta.edu

(Received: 12 March 1997; accepted: 16 June 1997)

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 and stability is guaranteed by Lyapunov theory. This control algorithm is applied to the practical point stabilization problem i.e., stabilization to a small neighborhood of the origin. 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.

Key words: nonholonomic systems, mobile robots, neural networks.

1. 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. Moreover, the literature on robustness and control in presence of uncertainties in the dynamical model of such systems is sparse (Kolmanovsky et al., 1995). Some preliminary results on nonholonomic system with uncertainties are given in (Canudas de Wit et al., 1995; Jiang et al., 1994).

Another intensive area of research has been neural networks 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 (Lewis et al., 1996b). Several groups by now are doing rigorous analysis of NN controllers using a variety of techniques (Chen et al., 1994; Narendra et al., 1991; Polycarpou et al., 1992; Rovithakis et al., 1994; Sadegh et al., 1993). In (Lewis et al., 1996) a multilayer NN controller with guaranteed performance has been developed and successfully applied to control of rigid

* Research supported by NSF Grant ECS-9521673.

robot manipulators, flexible-link robotic systems and position/force control. In this paper, we present an application of this NN controller to a nonholonomic 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 (Berns et al., 1991; Nagata et al., 1990). In these approaches the NN is trained in a preliminary *off-line* learning phase with navigation pattern behaviors, for instance *obstacle avoidance*. In contrast, the objective of this work is to design an adaptive kinematic/neuro-controller based on the universal approximation property of NN (Hornik et al., 1989). 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.

In the literature, the nonholonomic navigation 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' (Kanayama et al., 1990). 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 (Samson, 1991). The approach proposed in this paper corrects this omission by means of a 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, a 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.

This paper is organized as follows. In Section 2, we present some basics of nonholonomic systems and NN. Section 3 discusses the nonlinear kinematic-NN controller as applied to the point stabilization problem. In this section, we also consider some stability and robustness issues. Section 4 presents some simulation results. Finally, Section 5 gives some concluding remarks.

2. Preliminaries

2.1. A NONHOLONOMIC MOBILE ROBOT

Wheeled vehicles and car-like mobile robots are typical examples of nonholonomic mechanical systems. Unfortunately many researchers treat the problem of motion under nonholonomic constraints using only the kinematic model of a mobile robot. This simplified representation does not correspond to reality of moving vehicle which has unknown masses, frictions, drive train compliance, and backlash effects. In this paper we provide a framework that brings together two camps: nonholonomic control results that deal with a kinematic 'steering system', and full servo-level feedback control that takes into account the mobile robot dynamics.

A generalized mechanical system having an n-dimensional configuration space $\mathcal{C}$ with generalized coordinates $(q_1, \ldots, q_n)$ and subject to m nonholonomic constraints can be described by Sarkar et al. (1994),

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

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

where **M** is a symmetric, positive definite inertia matrix, $\mathbf{V}_m$ is a centripetal and coriolis matrix, **F** is a friction vector, **G** is a gravity vector, τ_d is a vector of disturbances including unmodeled dynamics, **B** is an input transformation matrix, τ is a control input vector, **A** is a matrix associated with the constraints, and λ is a vector of constraint forces. The dynamics of the driving and steering motors should be included in the robot dynamics, along with any gearing.

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

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

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

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

In fact, $v(t)$ often has physical meaning, consisting of two components – the commanded vehicle linear velocity $v_L(t)$, and angular velocity $\omega(t)$ or heading angle θ. Matrix $\mathbf{S}(q)$ is easily determined independently of the dynamics (1) from the wheel configuration of the mobile robot. Thus, Equation (4) is the kinematic equation that express some simplified relations between motion $q(t)$ and a velocity vector $v(t) = [v_L\ \omega]^T$. It does not include dynamical effects, and is known in the nonholonomic literature as the *steering system*. In the case of omnidirectional vehicles, $\mathbf{S}(q)$ is a square matrix and corresponds to the Newton's law model $F = ma$.

A typical nonholonomic platform shown in Figure 1 consists of a differential drive vehicle (e.g., *LabMate* manufactured by TRC). The motion and orientation are achieved by independent actuators, e.g., DC motors providing the necessary torques to the driving wheels. Another common configuration uses the front wheel for driving and steering. The position of the robot in an inertial Cartesian frame $\{O, X, Y\}$ is completely specified by the vector $q = [x_c\ y_c\ \theta]^T$, where x_c, y_c and θ are the coordinates of the reference point C, and the orientation of the basis $\{C, X_C, Y_C\}$ with respect to the inertial basis respectively.

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* (Barraquand et al., 1991)

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

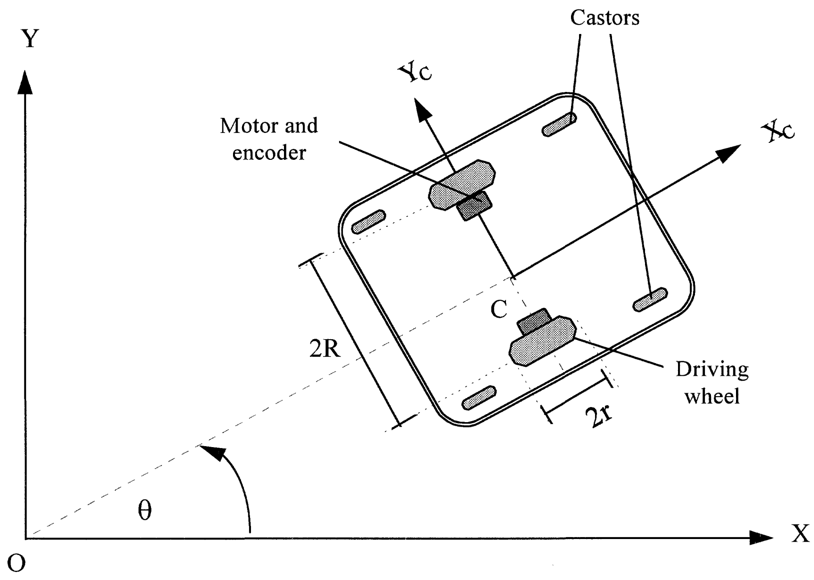

Figure 1. A nonholonomic mobile platform.

It is easy to verify that the kinematic equations of motion (4) of C in terms of its linear velocity and angular velocity may given by

$$\mathbf{S}(q) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}, \qquad \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_L \\ \omega \end{bmatrix}, \tag{6}$$

where $|v_L| < V_{\max}$ and $|\omega| < W_{\max}$. $V_{\max}$ and $W_{\max}$ are the maximum linear and angular velocities of the mobile robot. The dynamics of the nonholonomic mobile base in Figure 1 can be found in (Fierro et al., 1995).

2.2. STRUCTURAL PROPERTIES OF A MOBILE PLATFORM

The system (1) is now transformed into a more appropriate representation for controls purposes (Yamamoto et al., 1993). Differentiating (4), substituting this result in (1), and then multiplying by $\mathbf{S}^T$, we can eliminate the constraint term $\mathbf{A}^T(q)\lambda$. The complete equations of motion of the nonholonomic mobile platform are given now by

$$\dot{q} = \mathbf{S}v, \tag{7}$$

$$\mathbf{S}^T\mathbf{M}\mathbf{S}\dot{v} + \mathbf{S}^T(\mathbf{M}\dot{\mathbf{S}} + \mathbf{V}_m\mathbf{S})v + \overline{\mathbf{F}} + \overline{\tau}_d = \mathbf{S}^T\mathbf{B}\tau. \tag{8}$$

By appropriate definitions we can rewrite equation (8) as follows

$$\overline{\mathbf{M}}(q)\dot{v} + \overline{\mathbf{V}}_m(q,\dot{q})v + \overline{\mathbf{F}}(v) + \overline{\tau}_d = \overline{\mathbf{B}}\tau. \tag{9}$$

The true model of the vehicle is thus given by combining both (7) and (9). However, in the latter equation it turns out that $\overline{\mathbf{B}}$ is square and invertible, so that standard computed-torque techniques can be used to compute the required vehicle control τ. Moreover, the properties of the original dynamics hold for the new set of coordinates, i.e., *Boundedness*: $\overline{\mathbf{M}}(q)$, the norm of $\overline{\mathbf{V}}_m(q,\dot{q})$, and $\overline{\tau}_d$ are bounded. *Skew-symmetry*: The matrix $\dot{\overline{\mathbf{M}}} - 2\overline{\mathbf{V}}_m$ is skew symmetric.

2.3. FEEDFORWARD NEURAL NETWORKS

A 'two-layer' feedforward NN in Figure 2 has two layers of adjustable weights. The neural network 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

$$y_i = \sum_{j=1}^{N_h} \left[w_{ij}\sigma\left(\sum_{k=1}^{n} v_{jk}x_k + \theta_{vj} \right) + \theta_{wi} \right]; \quad i = 1, \ldots, m, \tag{10a}$$

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, gaussian, etc. In this work we shall use the sigmoid activation function

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}}, \tag{10b}$$

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

$$y = \mathbf{W}^T\sigma(\mathbf{V}^T x), \tag{11}$$

with the vector of activation functions defined by $\sigma(\mathbf{z}) = [\sigma(z_1)\ldots\sigma(z_n)]^T$ for a vector $\mathbf{z} \in \Re^n$. 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., $x \equiv [1\ x_1\ x_2\ x_3 \ldots x_n]^T$). Any tuning of $\mathbf{W}$ and $\mathbf{V}$ 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* (Cybenko, 1989; Hornik et al., 1989). 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 U_x of $\Re_n$, for some number of hidden layer neurons N_h, there exist weights and thresholds such that one has

$$f(x) = \mathbf{W}^T\sigma(\mathbf{V}^T x) + \varepsilon. \tag{12}$$

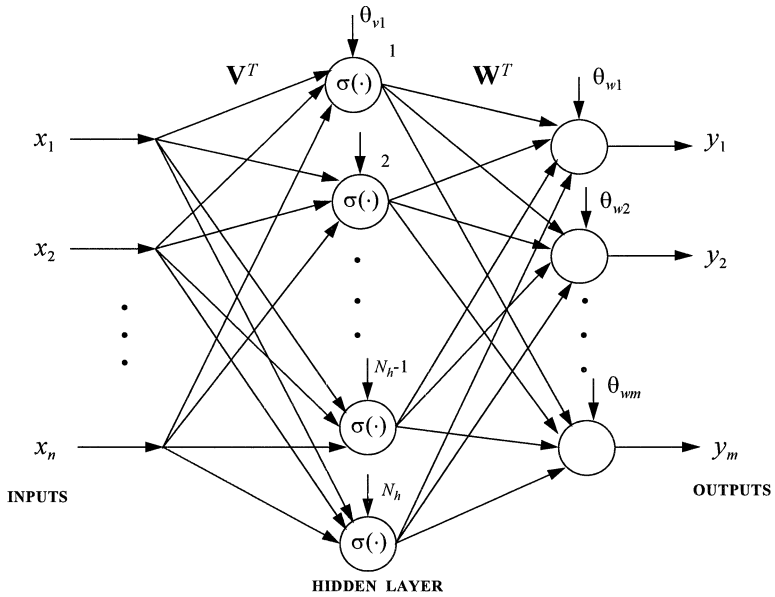

Figure 2. Multilayer feedforward neural network.

This equation means that a 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 $\varepsilon < \varepsilon_N$ 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

$$\widehat{f}(x) = \widehat{\mathbf{W}}^T \sigma(\widehat{\mathbf{V}}^T x), \tag{13}$$

where $\widehat{\mathbf{W}}, \widehat{\mathbf{V}}$ 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 (Werbos, 1989), 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

$$\begin{aligned} \dot{\widehat{\mathbf{W}}} &= \mathbf{F}\sigma(\widehat{\mathbf{V}}^T x_d)E^T, \\ \dot{\widehat{\mathbf{V}}} &= \mathbf{G}x_d(\widehat{\sigma}'^T \widehat{\mathbf{W}} E)^T, \end{aligned} \tag{14}$$

where $\mathbf{F}, \mathbf{G}$ 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 (10b), for instance, the hidden-layer output gradient is

$$\frac{\partial \sigma}{\partial z} = \sigma(z)[1 - \sigma(z)] \equiv \sigma'. \tag{15}$$

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

$$\widehat{\sigma}' \equiv \mathrm{diag}\{\sigma(\widehat{\mathbf{V}}^T x_d)\}\big[\mathbf{I} - \mathrm{diag}\{\sigma(\widehat{\mathbf{V}}^T x_d)\}\big], \tag{16}$$

where **I** denotes the identity matrix, and 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 (Jacobians (16)) 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 (Lewis et al., 1996; Narendra, 1991; Polycarpou et al., 1992; Rovithakis et al., 1994; Sadegh, 1993) and the references therein.

3. NN Point Stabilization of Nonholonomic Systems

Feedback stabilization deals with finding feedback control 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 time-invariant state-feedback* which makes an equilibrium point of the closed-loop system locally asymptotically stable (Brockett, 1983). 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. A comprehensive summary of these techniques and other nonholonomic issues are given in (Kolmanovsky et al., 1995). These techniques can be classified as (1) continuous time-varying stabilization (CTVS), (2) discontinuous time-invariant stabilization (DTIS), and (3) hybrid stabilization (HS). In CTVS the feedback control signals are smooth and time-periodic. In contrast, DTIS uses piecewise continuous controllers and sliding mode controllers. HS consists of designing a discrete-event supervisor and a set of low-level continuous-time controllers. The discrete event-supervisor coordinates (mode switching) the low-level controllers to make an equilibrium point asymptotically stable. In this section, we shall discuss CTVS as an extension of the tracking problem.

Point Stabilization as an Extension of the Tracking Problem

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

$$\begin{aligned} \dot{x}_r &= v_r \cos\theta_r, \qquad \dot{y}_r = v_r \sin\theta_r, \qquad \dot{\theta}_r = w_r, \\ q_r &= [x_r \ y_r \ \theta_r]^T, \qquad \nu_r = [v_r \ w_r]^T. \end{aligned} \tag{17}$$

As in (Canudas et al., 1993) it is assumed that the reference cart moves along the x-axis, i.e.,

$$\dot{x}_r = v_r, \qquad q_r = [x_r \ 0 \ 0]^T, \qquad \nu_r = [v_r \ 0]^T. \tag{18}$$

Therefore, the point stabilization problem consists of finding a *smooth time-varying* velocity control input $\nu_c(t)$ such that $\lim_{t\to\infty}(q_r - q) = 0$ and $\lim_{t\to\infty}(x_r) = 0$. Then compute the torque input $\tau(t)$ for (9), such that $\nu \to \nu_c$ as $t \to \infty$.

3.1. NN CONTROL DESIGN FOR TRACKING A REFERENCE TRAJECTORY

The structure for the point stabilization system to be derived in Section 3.3 is presented 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 (9) by learning it on-line.

The contribution of this paper lies in deriving a suitable $\tau(t)$ from a specific $\nu_c(t)$ that controls the steering system (7). In the literature, the nonholonomic point stabilization problem is simplified by neglecting the vehicle dynamics (8) and considering only the steering system (7). To compute the vehicle torque $\tau(t)$, it is assumed that there is ‘perfect velocity tracking’ so that $\nu = \nu_c$, then (8) 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 adaptive NN controller now developed.

To be specific, it is assumed that the solution to the steering system point stabilization problem in (Canudas de Wit et al., 1993) is available. This is denoted as $\nu_c(t)$. Then, a control $\tau(t)$ for (7), (8) is found that guarantees robust practical point stabilization despite unknown dynamical parameters and bounded unknown disturbances $\overline{\tau}_d(t)$. The (position) error is expressed in the basis of a frame linked to the mobile platform (Kanayama et al., 1990) as

$$\begin{aligned} e_p &= \mathbf{T}_e(q_r - q), \\ \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} &= \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ -y \\ -\theta \end{bmatrix}. \end{aligned} \tag{19}$$

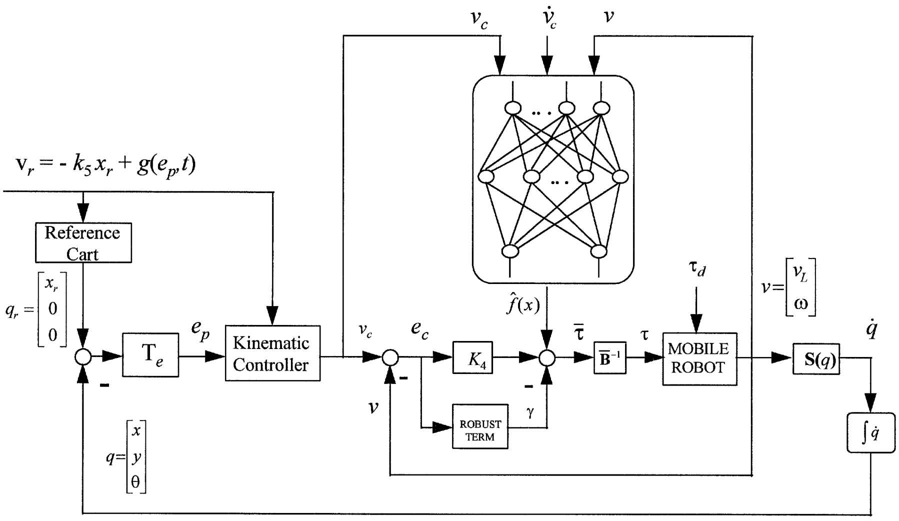

Figure 3. Practical point stabilization using a NN control.

and the derivative of the error is

$$\dot{e}_p = \begin{bmatrix} \omega e_2 - \nu_L + v_r \cos e_3 \\ -\omega e_1 + v_r \sin e_3 \\ -\omega \end{bmatrix}. \tag{20}$$

An auxiliary velocity control input that achieves point stabilization for (7) is given by

$$\nu_c = f_c(e_p, v_r, K) = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ k_2 v_r \dfrac{\sin e_3}{e_3} e_2 + k_3 e_3 \end{bmatrix}. \tag{21}$$

The derivative of ν_c becomes

$$\begin{aligned} \dot{\nu}_c &= \begin{bmatrix} \dot{v}_r \cos e_3 \\ k_2 \dot{v}_r \dfrac{\sin e_3}{e_3} e_2 \end{bmatrix} \\ &\quad + \begin{bmatrix} k_1 & 0 & -v_r \sin e_3 \\ 0 & k_2 v_r \dfrac{\sin e_3}{e_3} & k_2 v_r \dfrac{e_3 \cos e_3 - \sin e_3}{e_3^2} e_2 + k_3 \end{bmatrix} \dot{e}_p, \end{aligned} \tag{22}$$

where

$$v_r = -k_5 x_r + g(e_p, t). \tag{23a}$$

Therefore

$$\dot{x}_r = -k_5 x_r + g(e_p, t), \tag{23b}$$

with (Canudas de Wit et al., 1993)

$$g(e_p, t) = \begin{cases} \sin t & \text{if } \|e_p\| \geqslant \varepsilon > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{23c}$$

The gains $k_1, k_2, k_3, k_5 > 0$ are design parameters. They can be chosen to satisfy certain performance criteria. If the mobile robot is able to follow the desired velocity (21), then the position error e_p converges to a neighborhood of the origin. Different time-varying functions $g(e_p, t)$ are available in the literature, see (Kolmanovsky et al., 1995) and the references therein.

Given the desired velocity $\nu_c(t) \in \Re^2$, define now the auxiliary velocity tracking error as

$$e_c = \nu_c - \nu. \tag{24}$$

Differentiating (24) and using (9), the mobile robot dynamics may be written in terms of the velocity tracking error as

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

where the important *nonlinear mobile robot function* is

$$f(x) = \overline{\mathbf{M}}(q)\dot{\nu}_c + \overline{\mathbf{V}}_m(q, \dot{q})\nu_c + \overline{\mathbf{F}}(\nu). \tag{26}$$

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

$$x \equiv [\nu^T \ \nu_c^T \ \dot{\nu}_c^T]^T, \tag{27}$$

which can be measured.

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.

3.2. 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

$$\overline{\tau} = \widehat{f} + K_4 e_c - \gamma, \tag{28}$$

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

$$\overline{\mathbf{M}}\dot{e}_c = -(K_4 + \overline{\mathbf{V}}_m)e_c + \widetilde{f} + \overline{\tau}_d + \gamma, \tag{29}$$

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

$$\widetilde{f} = f - \widehat{f}. \tag{30}$$

In computing the control signal, the estimate $\widehat{f}$ 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.

3.3. NEURAL NET CONTROLLER

By using the controller (28), 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 K_4, the estimate $\widehat{f}$, 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 fhat 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 a NN to compute the estimate $\widehat{f}$. A major advantage is that this can always be accomplished, due to the NN approximation property (12). By contrast, in adaptive control approaches it is only possible to proceed if $f(x)$ is linear in the known parameters.

Some definitions are required in order to proceed:

DEFINITION 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 $t \geqslant t_0 + T$.

DEFINITION 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. Given $\mathbf{A} = [a_{ij}],\ \mathbf{B} \in \Re^{m\times n}$ the Frobenius norm is defined by

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

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

DEFINITION 4. For notational convenience we define the matrix of all the NN weights as $\widehat{\mathbf{Z}} \equiv \mathrm{diag}\{\widehat{\mathbf{W}}, \widehat{\mathbf{V}}\}$.

DEFINITION 5. Define the weight estimation errors as $\widetilde{\mathbf{V}} = \mathbf{V} - \widehat{\mathbf{V}}$, $\widetilde{\mathbf{W}} = \mathbf{W} - \widehat{\mathbf{W}}$, $\widetilde{\mathbf{Z}} = \mathbf{Z} - \widehat{\mathbf{Z}}$.

DEFINITION 6. Define the hidden-layer output error for a given x as

$$\widetilde{\sigma} = \sigma - \widehat{\sigma} = \sigma(\mathbf{V}^T x) - \sigma(\widehat{\mathbf{V}}^T x). \tag{32}$$

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

$$\sigma(\mathbf{V}^T x) = \sigma(\widehat{\mathbf{V}}^T x) + \sigma'(\widehat{\mathbf{V}}^T x)\widetilde{\mathbf{V}}^T x + \mathrm{O}(\widetilde{\mathbf{V}}^T x), \tag{33a}$$

with

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

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

$$\widetilde{\sigma} = \sigma'(\widehat{\mathbf{V}}^T x)\widetilde{\mathbf{V}}^T x + \mathrm{O}(\widetilde{\mathbf{V}}^T x) = \widehat{\sigma}'\widetilde{\mathbf{V}}^T x + \mathrm{O}(\widetilde{\mathbf{V}}^T x). \tag{33c}$$

The importance of this equation is that it replaces $\widetilde{\sigma}$, which is nonlinear in $\widetilde{\mathbf{V}}$, by an expression linear in $\widetilde{\mathbf{V}}$ plus higher-order terms. This will allow us to determine tuning algorithms for $\widehat{\mathbf{V}}$ 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 1. On any compact subset of $\Re^n$, the ideal NN weights are bounded by known positive values so that $\|\mathbf{V}\|_F \leqslant V_M$, $\|\mathbf{W}\|_F \leqslant W_M$, or $\|\mathbf{Z}\|_F \leqslant Z_M$ with Z_M known.

ASSUMPTION 2. The desired reference trajectory is bounded so that $\|q_r\| < q_M$ with q_M a known scalar bound, and the disturbances are bounded so that $\|\overline{\tau}_d\| \leqslant d_M$.

LEMMA 1 (Bound on NN input x). *For each time* $t, x(t)$ *in* (27) *is bounded by*

$$\|x\| \leqslant q_M + c_0\|e_c(t_0)\| + c_2\|e_2(t)\| \leqslant c_1 + c_2\|e_c(t)\| \tag{34}$$

for computable positive constants c_i.

LEMMA 2 (Bounds on Taylor series higher-order terms). *For sigmoid activation functions, the higher-order terms in the Taylor series* (33) *are bounded by*

$$\|\mathrm{O}(\widetilde{\mathbf{V}}^T x)\| \leqslant c_3 + c_4\|\widetilde{\mathbf{V}}\|_F + c_5\|\widetilde{\mathbf{V}}\|_F\|e_c\|, \tag{35}$$

for computable positive constants c_i.

We will use a neural net to approximate $f(x)$ for computing the control in (28). By placing into (28) the neural network approximation equation given by (13), the control input then becomes

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

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

$$\overline{\mathbf{M}}\dot{e}_c = -(K_4 + \overline{\mathbf{V}}_m)e_c + \mathbf{W}^T\sigma(\mathbf{V}^T x) - \widehat{\mathbf{W}}^T \sigma(\widehat{\mathbf{V}}^T x) + (\varepsilon + \overline{\tau}_d) + \gamma. \tag{37a}$$

Adding and subtracting $\mathbf{W}^T\widehat{\sigma}$ yields

$$\overline{\mathbf{M}}\dot{e}_c = -(K_4 + \overline{\mathbf{V}}_m)e_c + \widetilde{\mathbf{W}}^T\widehat{\sigma} + \mathbf{W}^T\widetilde{\sigma} + (\varepsilon + \overline{\tau}_d) + \gamma \tag{37b}$$

with $\widehat{\sigma},\ \widetilde{\sigma}$ defined in (32). Adding and subtracting now $\widehat{\mathbf{W}}^T\widetilde{\sigma}$ yields

$$\overline{\mathbf{M}}\dot{e}_c = -(K_4 + \overline{\mathbf{V}}_m)e_c + \widetilde{\mathbf{W}}^T\widehat{\sigma} + \mathbf{W}^T\widetilde{\sigma} + \widetilde{\mathbf{W}}^T\widetilde{\sigma} + (\varepsilon + \overline{\tau}_d) + \gamma. \tag{37c}$$

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

$$\overline{\mathbf{M}}\dot{e}_c = -(K_4 + \overline{\mathbf{V}}_m)e_c + \widetilde{\mathbf{W}}^T(\widehat{\sigma} - \widehat{\sigma}'\widetilde{\mathbf{V}}^T x) + \mathbf{W}^T\widetilde{\sigma}'\widetilde{\mathbf{V}}^T x + w + \gamma, \tag{38}$$

where the disturbance terms are

$$w(t) = \widetilde{\mathbf{W}}^T\widehat{\sigma}'\mathbf{V}^T x + \mathbf{W}^T \mathrm{O}(\widetilde{\mathbf{V}}^T x) + \varepsilon + \overline{\tau}_d. \tag{39}$$

It is important to note that the NN reconstruction error $\varepsilon(x)$, 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 (Bounds on the disturbance term). *The disturbance term* (39) *is bounded according to*

$$\|w(t)\| \leqslant (\varepsilon_N + d_M + c_3 Z_M) + c_6 Z_M\|\widetilde{\mathbf{Z}}\|_F + c_7 Z_M\|\widetilde{\mathbf{Z}}\|_F\|e_c\|$$

or

$$\|w(t)\| \leqslant C_0 + C_1\|\widetilde{\mathbf{Z}}\|_F + C_2\|\widetilde{\mathbf{Z}}\|_F\|e_c\|, \tag{40}$$

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 1–3 are omitted here, details are discovered in (Lewis et al., 1996).

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

THEOREM 1. *Given a nonholonomic mobile robot* (7), (8). *Let the following assumptions hold:*

ASSUMPTION 3. *A smooth time-varying auxiliary velocity control input* $\nu_c(t)$ *is prescribed that solves the point stabilization problem for the steering system* (7), *neglecting the dynamics* (8). *A sample* ν_c *is given by* (21).

ASSUMPTION 4. $K = [k_1\ k_2\ k_3\ k_5]^T$ *is a vector of positive constants.*

ASSUMPTION 5. $K_4 = k_4\mathbf{I}$, *where* k_4 *is a sufficiently large positive constant.*

Take the control $\overline{\tau} \in \Re^2$ *for* (9) *as* (36) *with robustifying term*

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

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, the velocity tracking error* $e_c(t)$, *the position error* $e_p(t)$, *and the NN weight estimates* $\widehat{\mathbf{V}}$, $\widehat{\mathbf{W}}$ *are* UUB. *Moreover, the velocity tracking error may be kept as small as desired by increasing the gain* K_4.

$$\dot{\widehat{\mathbf{W}}} = \mathbf{F}\widehat{\sigma}e_c^T - \mathbf{F}\widehat{\sigma}'\widehat{\mathbf{V}}^T x e_c^T - \kappa\mathbf{F}\|e_c\|\widehat{\mathbf{W}}, \tag{43a}$$

$$\dot{\widehat{\mathbf{V}}} = \mathbf{G}x(\widehat{\sigma}'^T\widehat{\mathbf{W}}e_c)^T - \kappa\mathbf{G}\|e_c\|\widehat{\mathbf{V}}, \tag{43b}$$

where $\mathbf{F}, \mathbf{G}$ *are positive definite design parameter matrices,* $\kappa > 0$ *and the hidden-layer gradient* $\widehat{\sigma}'$ *is easily computed – for the sigmoid activation function it is given by*

$$\widehat{\sigma}' \equiv \mathrm{diag}\{\sigma(\widehat{\mathbf{V}}^T x)\}\left[\mathbf{I} - \mathrm{diag}\{\sigma(\widehat{\mathbf{V}}^T x)\}\right], \tag{44}$$

which is just (16) *with the constant exemplar* x_d *replaced by the time function* $x(t)$.

Proof. Let the approximation property (12) hold with a given accuracy ε_N for all x in the compact set U_x. Consider the following Lyapunov function candidate

$$V = \frac{k_2}{2}(e_1^2 + e_2^2) + \frac{e_3^2}{2} + V_1, \tag{45}$$

where

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

Differentiating yields

$$\dot{V} = k_2(e_1\dot{e}_1 + e_2\dot{e}_2) + e_3\dot{e}_3 + \dot{V}_1. \tag{47}$$

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

$$\begin{aligned}\dot{V}_1 \;=\; & -e_c^T K_4 e_c + \tfrac{1}{2} e_c^T(\dot{\overline{\mathbf{M}}} - 2\overline{\mathbf{V}}_m)e_c + \mathrm{tr}\{\widetilde{\mathbf{W}}^T(\mathbf{F}^{-1}\dot{\widetilde{\mathbf{W}}} + \widehat{\sigma} e_c^T - \widehat{\sigma}'\widehat{\mathbf{V}}^T x e_c^T)\} \\ & + \mathrm{tr}\{\widetilde{\mathbf{V}}^T(\mathbf{G}^{-1}\dot{\widetilde{\mathbf{V}}} + x e_c^T \widehat{\mathbf{W}}^T \widehat{\sigma}')\} + e_c^T(w+\gamma).\end{aligned} \tag{48}$$

The skew symmetry property (Section 2.2) makes the second term zero, and since $\dot{\widetilde{\mathbf{W}}} = -\dot{\widehat{\mathbf{W}}}$, $\dot{\widetilde{\mathbf{V}}} = -\dot{\widehat{\mathbf{V}}}$, the tuning rules yield

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

Since

$$\mathrm{tr}\{\widetilde{\mathbf{Z}}^T(\mathbf{Z} - \widetilde{\mathbf{Z}})\} = \langle\widetilde{\mathbf{Z}}, \mathbf{Z}\rangle_F - \|\widetilde{\mathbf{Z}}\|_F^2 \leqslant \|\widetilde{\mathbf{Z}}\|_F\|\mathbf{Z}\|_F - \|\widetilde{\mathbf{Z}}\|_F^2, \tag{50}$$

there results

$$\begin{aligned}\dot{V}_1 \;\leqslant\; & -e_c^T K_4 e_c - \kappa\|e_c\|\cdot\|\widetilde{\mathbf{Z}}\|_F(\|\widetilde{\mathbf{Z}}\|_F - Z_M) - \\ & - K_z(\|\widetilde{\mathbf{Z}}\|_F + Z_M)\|e_c\|^2 + \|e_c\|\cdot\|w\| - e_c^T e_c, \\ \leqslant\; & -K_{4\min}\|e_c\|^2 - \kappa\|e_c\|\cdot\|\widetilde{\mathbf{Z}}\|_F(\|\widetilde{\mathbf{Z}}\|_F - Z_M) - \\ & - K_z(\|\widetilde{\mathbf{Z}}\|_F + Z_M)\|e_c\|^2 + \\ & + \|e_c\|\big[C_0 + C_1\|\widetilde{\mathbf{Z}}\|_F + C_2\|e_c\|\cdot\|\widetilde{\mathbf{Z}}\|_F\big] - e_c^T e_c, \\ \leqslant\; & -\|e_c\|\cdot\big[K_{4\min}|e_c\| + \kappa\|\widetilde{\mathbf{Z}}\|_F(\|\widetilde{\mathbf{Z}}\|_F - Z_M) - \\ & \qquad - C_0 - C_1\|\widetilde{\mathbf{Z}}\|_F\big] - e_c^T e_c,\end{aligned} \tag{51}$$

where $K_{4\min}$ is the minimum singular value of K_4, Lemma 3 was used, and the last inequality holds due to (42).

The velocity tracking error is

$$e_c = \begin{bmatrix} e_4 \\ e_5 \end{bmatrix} = \begin{bmatrix} \nu_{c1} - \nu_L \\ \nu_{c2} - \omega \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 - \nu_L \\ k_2 v_r \dfrac{\sin e_3}{e_3}\, e_2 + k_3 e_3 - \omega \end{bmatrix}, \tag{52}$$

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

$$\begin{aligned}\dot{V} \;\leqslant\; & k_2 e_1(\omega e_2 - \nu_L + v_r \cos e_3) + k_2 e_2(-\omega e_1 + v_r \sin e_3) + \\ & + e_3(-\omega) - e_c^T e_c - \|e_c\| \cdot \big[K_{4\min}\|e_c\| + \kappa\|\widetilde{\mathbf{Z}}\|_F\big(\|\widetilde{\mathbf{Z}}\|_F - Z_M\big) \\ & - C_0 - C_1\|\widetilde{\mathbf{Z}}\|_F\big],\end{aligned} \tag{53}$$

by using (52) and defining $k_1 > k_2/4,\ k_2 > 0,\ k_3 > 1/4$ yield

$$\begin{aligned}\dot{V} \;\leqslant\; & -k_2\big(k_1 - \tfrac{k_2}{4}\big)e_1^2 - \big(k_3 - \tfrac{1}{4}\big)e_3^2 - \big(e_4 - \tfrac{k_2}{2}e_1\big)^2 - \big(e_5 - \tfrac{1}{2}e_3\big)^2 - \\ & - \|e_c\|\big[K_{4\min}\|e_c\| + \kappa\|\widetilde{\mathbf{Z}}\|_F\big(\|\widetilde{\mathbf{Z}}\|_F - Z_M\big) - C_0 - C_1\|\widetilde{\mathbf{Z}}\|_F\big].\end{aligned} \tag{54}$$

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

$$\dot{V} \leqslant -\|e_c\| \cdot \{K_{4\min}\|e_c\| + \kappa\|\widetilde{\mathbf{Z}}\|_F\big(\|\widetilde{\mathbf{Z}}\|_F - Z_M\big) - C_0 - C_1\|\widetilde{\mathbf{Z}}\|_F\}. \tag{55}$$

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

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

which is guaranteed positive as long as either

$$\|e_c\| > \frac{\kappa C_3^2 + C_0}{K_{4\min}} \equiv b_c \tag{56}$$

or

$$\|\widetilde{\mathbf{Z}}\|_F > C_3 + \sqrt{C_3^2 + \tfrac{C_0}{\kappa}} \equiv b_z. \tag{57}$$

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 $\|\widetilde{\mathbf{Z}}\|_F$.

Finally from (23) the reference cart can be proven to be asymptotically stable. Since $g(e_p, t)$ tends to zero for the ideal case, and $g(e_p, t) = 0$ if $\|e_p\| < \varepsilon$ for the practical case, it can be shown that $x_r \to 0$ as $t \to \infty$. Therefore, the mobile robot can be stabilized to an arbitrarily small neighborhood of the origin. □

Remarks. Note that $\|e_c\|$ can be kept arbitrarily small by increasing the gain $K_{4\min}$ in (56). The right-hand sides of (56), (57) can be taken as practical bounds on $e_c(t)$ and the NN weight estimation errors, respectively.

The first terms of (43) are nothing but the standard backpropagation algorithm. The last terms correspond to the e-modification (Narendra, 1991) from adaptive control theory; they must be added to ensure bounded NN weight estimates. The middle term in (43a) is a *novel term* needed to prove stability.

3.4. ROBUSTNESS CONSIDERATIONS

Theorem 1 guarantees that all signals in the closed-loop mobile robot system are bounded and the tracking error can be made arbitrarily small. As time passes the NN learns the nonlinear dynamics of the nonholonomic mobile robot *on-line*.

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. Additionally, for good performance, the bound on $e_c(t)$ should be in some sense 'small enough' because it will affect directly the position error $e_p(t)$. Thus, the nonholonomic control system consists of two subsystems: (1) a kinematic controller, and (2) a NN dynamic controller. The dynamic controller provides the required torques, so that the robot's velocity tracks a reference velocity input. As 'perfect velocity tracking' does not hold in practice, the dynamic controller generates a velocity error $e_c(t)$ which is assumed to be bounded by some know constant (Theorem 1). This error can be seen as a disturbance for the kinematic system, see Figure 4.

The closed-loop kinematic system becomes

$$\begin{aligned} \dot{x}_c &= (\nu_{c1} + e_4)\cos\theta, \\ \dot{y}_c &= (\nu_{c1} + e_4)\sin\theta, \\ \dot{\theta} &= \nu_{c2} + e_5, \end{aligned} \tag{58}$$

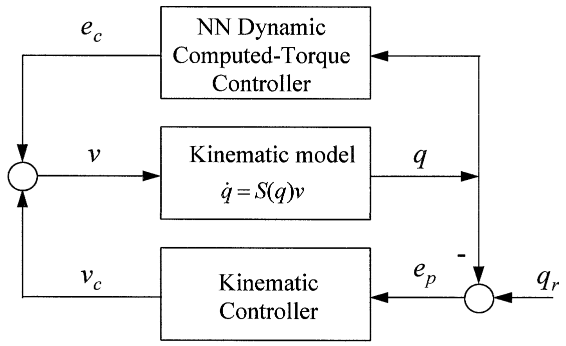

Figure 4. Closed-loop model of a nonholonomic system.

where $e_c = [e_4 \ e_5]^T$ and $\nu_c = [\nu_{c1} \ \nu_{c2}]^T$ denote the velocity tracking error and the desired velocity control input, respectively. The disturbance ec satisfies the matching condition (Canudas de Wit et al., 1995), i.e., the nonholonomic constraint (5) is not violated.

The derivative of the position error in terms of the velocity error is given by

$$\dot{e}_p = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} -k_1 e_1 + k_2 v_r \frac{\sin e_3}{e_3} e_2^2 + k_3 e_2 e_3 + e_4 - e_2 e_5 \\ -k_2 v_r \frac{\sin e_3}{e_3} e_1 e_2 - k_3 e_1 e_3 + v_r \sin e_3 + e_1 e_5 \\ -k_2 v_r \frac{\sin e_3}{e_3} e_2 - k_3 e_3 + e_5 \end{bmatrix}. \tag{59}$$

Let us consider the following Lyapunov function candidate

$$V = \frac{k_2}{2}(e_1^2 + e_2^2) + \frac{e_3^2}{2}, \tag{60}$$

differentiating yields

$$\begin{aligned} \dot{V} \ =\ & k_2 e_1\Big(- k_1 e_1 + k_2 v_r \frac{\sin e_3}{e_3} e_2^2 + k_3 e_2 e_3 + e_4 - e_2 e_5\Big) \\ & + k_2 e_2\Big(- k_2 v_r \frac{\sin e_3}{e_3} e_1 e_2 - k_3 e_1 e_3 + \ v_r \sin e_3 + e_1 e_5\Big) + \\ & + e_3\Big(- k_2 v_r \frac{\sin e_3}{e_3} e_2 - k_3 e_3 + e_5\Big). \end{aligned} \tag{61}$$

After some work, we obtain

$$\dot{V} = -k_1 k_2 e_1^2 - k_3 e_3^2 + k_2 e_1 e_4 + e_3 e_5. \tag{62}$$

The last equation can be rewritten as follows

$$\dot{V} = -e_{13}^T Q e_{13} + e_{13}^T P e_c, \tag{63}$$

where

$$e_{13} = [e_1 \ e_3]^T, \qquad Q = \begin{bmatrix} k_1 k_2 & 0 \\ 0 & k_3 \end{bmatrix}, \quad \text{and} \quad P = \begin{bmatrix} k_2 & 0 \\ 0 & 1 \end{bmatrix}.$$

The first term in the right-hand side is negative definite, while the second term is indefinite. But the second term satisfies

$$|e_{13}^T P e_c| \leqslant \|e_{13}\|_2 \, \|P\|_2 \, \|e_c\|_2, \tag{64}$$

there results

$$\dot{V} \leqslant -e_{13}^T Q e_{13} + \|e_{13}\|_2 \, \|P\|_2 \, \|e_c\|_2, \tag{65}$$

and

$$\dot{V} \leqslant \lambda_{\min}(Q) \, \|e_{13}\|_2^2 + \|e_{13}\|_2 \, \|P\|_2 \, \|e_c\|_2. \tag{66}$$

Thus, $\dot{V}$ is guaranteed negative as long as

$$\|e_{13}\| > \frac{\|P\|_2 \, \|e_c\|_2}{\lambda_{\min}(Q)} \equiv b_{13}. \tag{67}$$

Remark. Along a system's solution $\|e_p\|$ is bounded, and thus $\|\dot{e}_p\|$ is bounded. As we expected, 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 eps. and the matrix K_4. Since $\|e_c\|$ can be made arbitrarily small then $\|e_p\|$ can be made arbitrarily small.

3.5. NN EXPONENTIAL STABILIZER

It is well-known that the rates of convergence provided by smooth time-periodic laws are at most $t^{-1/2}$, i.e., nonexponential. Thus nonsmooth feedback laws with exponential rate of convergence have been proposed in the literature, see for instance (M'Closkey et al., 1994). In this approach the velocity control input is smooth everywhere except at the origin; therefore, the NN control structure in Figure 3 can be applied to this class of exponential feedback stabilization.

The following change of coordinates

$$\begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \\ \sin\theta & -\cos\theta & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ \theta \end{bmatrix} \tag{68}$$

is used to transform the kinematic model of the mobile robot given in (6) to a *chained* form

$$\begin{aligned} \dot{\eta}_1 &= \nu_1, \\ \dot{\eta}_2 &= \nu_2, \\ \dot{\eta}_3 &= n_1 \nu_2. \end{aligned} \tag{69}$$

A periodic time-varying control law that renders the equilibrium of (69) globally exponentially stable is given by

$$\nu_c(t) = \begin{bmatrix} \nu_{c1} \\ \nu_{c2} \end{bmatrix} = \begin{bmatrix} -\eta_1 + \dfrac{\eta_3}{\rho(\eta)} \cos t \\ -\eta_2 - \dfrac{\eta_3^2}{\rho^3(\eta)} \sin t \end{bmatrix}, \tag{70}$$

where

$$\rho(\eta) = (\eta_1^4 + \eta_2^4 + \eta_3^2)^{1/4}. \tag{71}$$

4. Simulation Results

We should like to illustrate the NN control scheme presented in Figure 3. Note that the NN controller *does not* require knowledge of the dynamics. We took

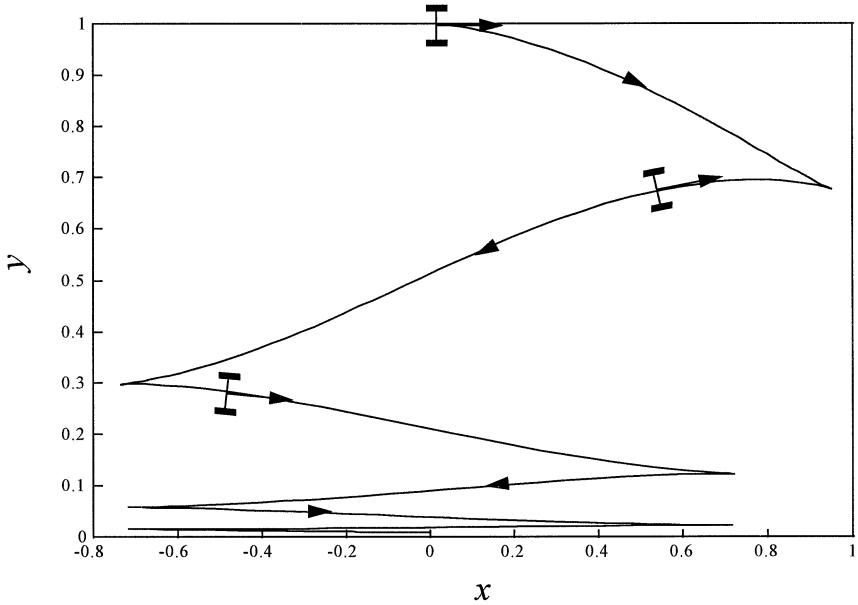

Figure 5. Mobile robot's trajectory: Smooth time-varying feedback.

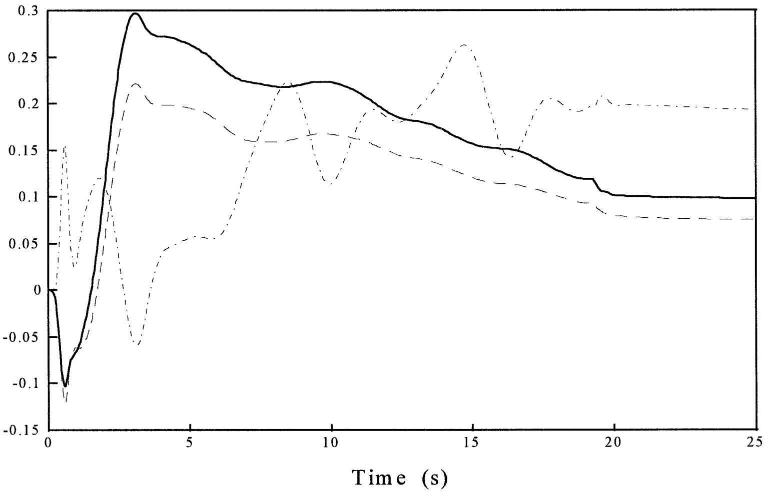

Figure 6. Some NN weights.

the vehicle parameters (Figure 1) as $m = 10$ kg, $I = 5$ kg-m^2, $R = 0.5$ m, and $r = 0.05$ m, $K_4 = \text{diag}\{25, 25\}$. For the NN, we selected the sigmoid activation functions with $N_h = 10$ hidden-layer neurons, $\mathbf{F} = \mathbf{G} = \text{diag}\{10, 10\}$ and $\kappa = 0.1$.

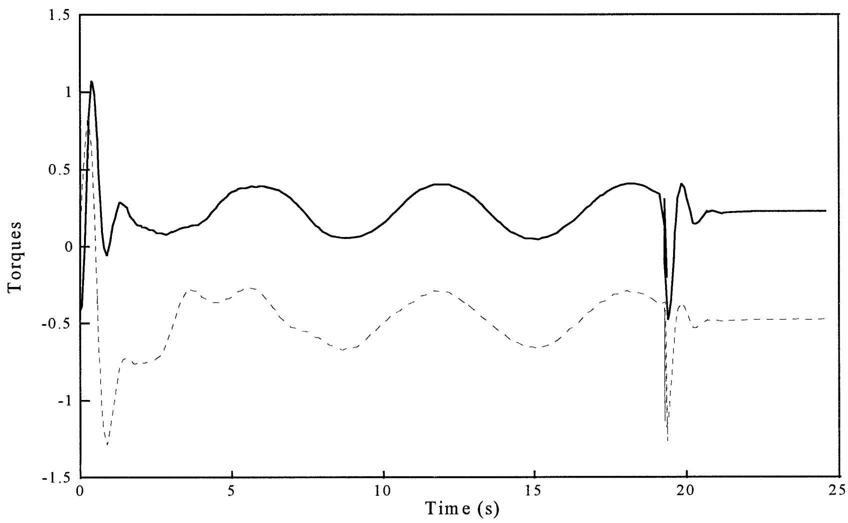

Figure 7. Applied torques: (–) right and (- -) left wheels.

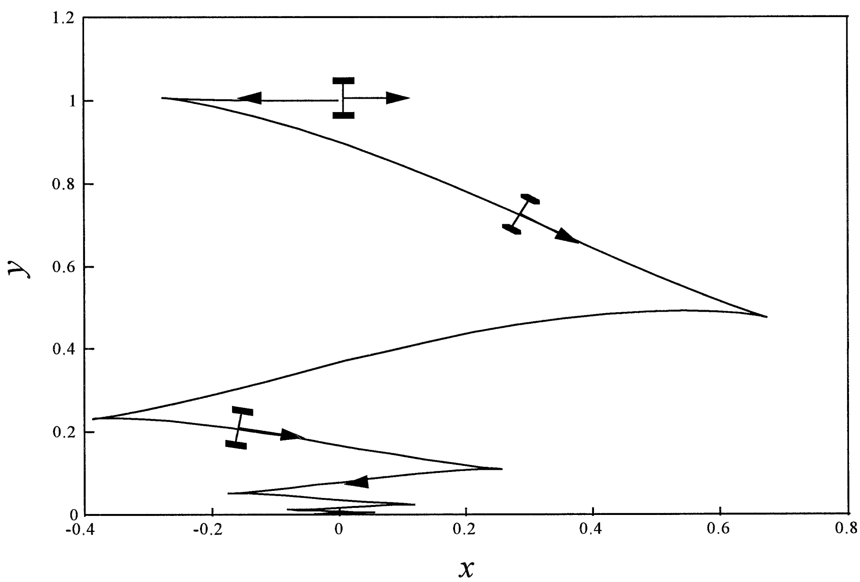

Figure 8. Mobile robot's trajectory: Exponential feedback law.

Simulation results using $\nu_c(t)$ in (21) are depicted in Figures 5–7. The controller gains were chosen so that the closed-loop system exhibits a critical damping behavior: $K = [10\ 5\ 4\ 1]^T$.

A simulation using the exponential feedback law (70) is shown in Figure 8. Compare with the robot's performance presented in Figure 5.

The mobile base *maneuvers*, i.e., exhibits forward and backward motions to reach the origin. Note that there is no path planning involved – the mobile base naturally describes a path that satisfies the nonholonomic constraints.

5. Conclusions

A robust/adaptive control algorithm for practical point stabilization of a nonholonomic mobile robot has been derived using NNs. This feedback servo-control scheme is valid as long as the velocity control inputs are smooth and bounded, and the disturbances acting on actual cart are bounded.

The control structure proposed in this paper can be applied to different navigation problems, e.g., tracking a reference trajectory and path following. Redefining the control velocity input v_c, one may generate a different stable behavior without changing the structure of the controller.

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 *neural network* controller with guaranteed performance has been derived. There is not need of *a priori* information of the dynamic parameters of the mobile robot, because the NN learns them on-the-fly.

References

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

Berns, K., Dillman, R., and Hofstetter, R.: 1991, An application of a backpropagation network for the control of a tracking behavior, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Vol. 3, pp. 2426–2431, Sacramento, CA, April 1991.

Bloch, A. M., Reyhanoglu, M., and McClamroch, N. H.: 1992, Control and stabilization of nonholonomic dynamic systems, *IEEE Trans. Autom. Control* **37**(11), 1746–1757.

Brockett, R. W.: 1983, Asymptotic stability and feedback stabilization, in: R. W. Brockett, R. S. Millman, and H. J. Sussmann (eds.), *Differential Geometric Control Theory*, Boston, Birkhäuser, pp. 181–191.

Campion, G., d'Andrréa-Novel, B., and Bastin, G.: 1991, Controllability and state feedback stabilizability of nonholonomic mechanical systems, in: C. Canudas de Wit (ed.), *Lecture Notes in Control and Information Science*, Springer-Verlag, Berlin, pp. 106–124.

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

Canudas de Wit, C. and Khennouf, H.: 1995, quasi-continuous stabilizing controllers for nonholonomic systems: design and robustness considerations, in: *Proc. Eur. Control Conf.*, Rome, 1995, pp. 2630–2635.

Chen, F. C. and Liu, C. C.: 1994, Adaptively controlling nonlinear continuous-time systems using multilayer neural network, *IEEE Trans. Autom. Control* **39**(6), 1306–1310.

Cybenko, G.: 1989, Approximation by superposition of a sigmoidal function, *Math. Controls Sig. Syst.* **2**(4), pp. 303–314.

Fierro, R. and Lewis, F. L.: 1995, Control of nonholonomic mobile robot: Backstepping kinematics into dynamics, in: *Proc. IEEE Conf. Dec. Control*, New Orleans, LA, 1995, pp. 3805–3810.

Hornik, K., Stinchcombe, M., and White, H.: 1989, Multilayer feedforward networks are universal approximators, *Neural Network* **2**, 359–366.

Jiang, Z. P. and Pomet, J.-B.: 1994, Combining backstepping and time-varying techniques for a new set of adaptive controllers, in: *Proc. IEEE Conf. Dec. Control*, Lake Buena Vista, FL, 1994, pp. 2207–2212.

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

Kolmanovsky, I. and McClamroch, N. H.: 1995, Developments in nonholonomic control problems, in: *IEEE Control Systems*, Dec. 1995, pp. 20–36.

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

Lewis, F. L.: 1996, Neural network control of robot manipulators, in: 1996 June, *IEEE Expert*, pp. 64–75.

Lewis, F. L., Yesildirek, A., and Liu, K.: 1996, Multi-layer neural-net roobot controller with guaranteed tracking performance, *IEEE Trans. Neural Netw.* **7**(2), 1–20.

M'Closkey, R. T. and Murray, R. M.: 1994, Extending exponential stabilizers for nonholonomic systems from kinematic controlllers to dynamic controllers, in: *Proc. IFAC Symp. Robot Control*, Capri, Italy, 1994, pp. 211–216.

Murray, R. M., Li, Z., and Sastry, S. S.: 1994, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Boca Raton, FL.

Nagata, S., Sekiguchi, M., and Asakawa, K.: 1990, Mobile robot control by a structured hierarchical neural network, *IEEE Control Systems*, 69–76.

Narendra, K. S.: 1991, Adaptive control using neural networks, in: W. T. Miller, R. S. Sutton, and P. W. Werbos (eds), *Neural Networks for Control*, MIT Press, Cambridge, pp. 115–142.

Oelen, W., Berghuis, H., Nijmeijer, H., and Canudas de Wit, C.: 1995, Hybrid stabilizing control on a real mobile robot, *IEEE Robot. automat. Magazine*, 16–23.

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

Rovithakis, G. A. and Christodoulou, M. A.: 1994, Adaptive control of unknown plants using dynamical neural networks, *IEEE Trans. Syst. Man Cyber.* **24** (3), 400–412.

Sadegh, N.: 1993, A perceptron network for functional identification and control of nonlinear systems, *IEEE Trans. Neural Netw.* **4**(6), 982–988.

Samson, C.: 1991, Velocity and torque feedback control of a nonholonomic cart, in: C. Canudas de Wit (ed.), *Lecture Notes in Control and Information Science*, Springer-Verlag, Berlin, pp. 125–151.

Sarkar, N. Yun, X., and Kumar, V.: 1994, Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots, *Int. J. Robot. Res.* **13**(1), 55–69.

Werbos, P. J.: 1989, Backpropagation: past and future, in: *Proc. 1988 Int. Conf. Neural Netw.*, Vol. 1, 1989, pp. 1343–1353.

Yamamoto, Y. and Yun, X.: 1993, Coordinating locomotion and manipulation of a mobile manipulator, in: Y. F. Zheng (ed.), *Recent Trends in Mobile Robots*, World Scientific, Singapore, pp. 157–181.