PRACTICAL POINT STABILIZATION OF A NONHOLONOMIC MOBILE ROBOT USING NEURAL NETWORKS

R. Fierro and F. L. Lewis
Automation and Robotics Research Institute, The University of Texas at Arlington
7300 Jack Newell Blvd. South, Fort Worth, Texas 76118-7115
rfierro@arrl.uta.edu, flewis@arrl.uta.edu

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. This control algorithm is applied to the practical point stabilization problem i.e., stabilization to a small neighborhood of the origin. The NN controller proposed in this work can deal with unmodelled bounded disturbances and/or unstructured unmodelled 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.

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 [9]. Some preliminary results on nonholonomic system with uncertainties are given in [3], [8].

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 [11]. Several groups by now are doing rigorous analysis of NN controllers using a variety of techniques [4], [11], [12], [13]. In this paper, we present a robust-adaptive kinematic/neuro-controller based on the universal approximation property of NN [7]. 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. 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.

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. Stability is proved by Lyapunov theory. Section 4 presents some simulation results. Finally, Section 5 gives some concluding remarks.

2. Preliminaries
A Nonholonomic System 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 in the d'Alembert-Lagrange form [10],

\[
M(q) \ddot{q} + V_a(q, \dot{q}) \dot{q} + F(q) + G(q) + \tau_d = B(q) \tau - A^T(q) \lambda, \tag{1}
\]

\[
A(q) \dot{\lambda} = 0, \tag{2}
\]

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

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

\[
S^T(q)A^T(q) = 0. \tag{3}
\]

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

\[
\dot{q} = S(q) \nu(t). \tag{4}
\]

Equations (1)-(4) are well-known results in nonholonomic system theory, see for instance [1], [9], [14]. A car-like mobile robot is a typical example of a nonholonomic mechanical system [6].

The system (1) is now transformed into a more appropriate representation for controls purposes [15].

\[
\dot{q} = Sv, \tag{5}
\]
\[ S^T M \dot{v} + S^T (M \ddot{v} + V_m \dot{v}) + \ddot{\tau}_d = S^T B \tau, \]  
(6)

where \( v(t) \in \mathbb{R}^{n \times m} \) is a velocity vector. By appropriate definitions we can rewrite equation (6) as follows

\[ \bar{M}(q) \ddot{q} + \bar{V}_m(q, \dot{q}) \dot{q} + \bar{F}(v) + \ddot{\tau}_d = \ddot{\tau}, \]  
(7)

where \( \bar{M}(q) \in \mathbb{R}^{2 \times 2} \) is a symmetric, positive definite inertia matrix, \( \bar{V}_m(q, \dot{q}) \in \mathbb{R}^{2 \times 2} \) is the centripetal and coriolis matrix, \( \bar{F}(v) \in \mathbb{R}^{2 \times 1} \) is the surface friction, \( \ddot{\tau}_d \) denotes bounded unknown disturbances including unstructured unmodelled dynamics, and \( \ddot{\tau} \in \mathbb{R}^{2 \times 1} \) is the input vector.

**Feedforward Neural Networks** [11] A 'three-layer' feedforward NN 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

\[ y_i = \sum_{j=1}^{N_h} w_{ij} \sigma \left( \sum_{k=1}^{N_h} v_{jk} x_k + \theta_{0j} \right) + \theta_{1i}; \quad i = 1, ..., m \]  
(8)

where \( \sigma(\cdot) \) are the activation functions and \( N_h \) is the number of hidden-layer neurons. The first-to-second-layer interconnection weights are denoted by \( v_{jk} \) and the second-to-third-layer interconnection weights by \( w_{ij} \). The threshold offsets are denoted by \( \theta_{0j}, \theta_{1i} \).

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

\[ y = W^T \sigma(V^T x), \]  
(9)

The thresholds are included as the first columns of the weight matrices. Any tuning of \( W \) and \( V \) 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* [5], [7]. Let \( f(x) \) be a smooth function from \( \mathbb{R}^n \) to \( \mathbb{R}^m \). Then, it can be shown that, as long as \( x \) is restricted to a compact set \( U_x \) of \( \mathbb{R}^n \), for some number of hidden layer neurons \( N_h \), there exist weights and thresholds such that one has

\[ f(x) = W^T \sigma(V^T x) + \epsilon. \]  
(10)

This equation means that a NN can approximate any function in a compact set. The value of \( \epsilon \) is called the NN *functional approximation error*. In fact, for any choice of a positive number \( \epsilon_N \), 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 \( \epsilon_N \) these *ideal* approximating NN weights exist. Then, an estimate of \( f(x) \) can be given by

\[ \hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x), \]  
(11)

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

**3. Control Design**

Many approaches exist to selecting a velocity control \( \nu(t) \) for the steering system (4). In this section, we desire to convert such a prescribed control \( \nu(t) \) into a torque control \( \tau(t) \) for the actual physical cart. Therefore, our objective is to design a NN control algorithm so that (5), (6) exhibits the desired behavior motivating the specific choice of the velocity \( \nu(t) \).

The smooth steering system control, denoted by \( \nu_c \), can be found by any technique in the literature. Using the algorithm to be derived and proved in Section 3.3, the practical point stabilization problem is solved as follows:

**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

\[ \dot{x}_r = \nu_r \cos \theta_r, \quad \dot{y}_r = \nu_r \sin \theta_r, \quad \dot{\theta}_r = \omega_r, \]  
(12)

\[ q_r = [x_r y_r \theta_r]^T, \quad \nu_r = [\nu_r \omega_r]^T. \]

As in [5] it is assumed that the reference cart moves along the \( x \)-axis, i.e.,

\[ \dot{x}_r = \nu_r, \quad q_r = [x_r 0 0]^T, \quad \nu_r = [\nu_r 0]^T. \]  
(13)

Therefore, the point stabilization problem consists of finding a smooth time-varying velocity control input \( \nu(t) \) such that

\[ \lim_{t \to \infty} (q - q_r) = 0 \text{ and } \lim_{t \to \infty} x_r = 0. \]

Then compute the torque input \( \tau(t) \) for (7), such that \( \nu \to \nu_c \) as \( t \to \infty \).

**3.1 NN Control Design for Tracking a Trajectory**

The structure for the point stabilization system to be derived in Section 3.3 is presented in Fig. 1. In this figure, no knowledge of the dynamics of the cart is assumed. The function of the NN is to reconstruct the dynamics (7) by learning it on-line.

![Fig. 1. Practical point stabilization using NN.](image-url)

The contribution of this paper lies in deriving a suitable \( \tau(t) \) from a specific \( \nu_c(t) \) that controls the steering system (5). In the literature, the nonholonomic point stabilization problem is simplified by neglecting the vehicle dynamics (6) and considering only the steering
system (5). To compute the vehicle torque $\tau(t)$, it is assumed that there is ‘perfect velocity tracking’ so that $v = v_c$, then (6) 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 $\tau_d$ 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 [2] is available. This is denoted as $v_y(t)$. Then, a control $\tau(t)$ for (5), (6) is found that guarantees robust practical point stabilization despite unknown dynamical parameters and bounded unknown disturbances $\tau_d(t)$. The position error is expressed in the basis of a frame linked to the mobile platform as

$$ e = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}, $$

and the derivative of the error is

$$ \dot{e} = \begin{bmatrix} v_x e_y - v_y \cos e_x \\ -v_x e_y + v_y \sin e_x \\ -v_z \end{bmatrix}. $$

An auxiliary velocity control input that achieves point stabilization for (5) is given by [2]

$$ v_c = f_c(e, v, \tau, \theta) = \begin{bmatrix} v_x e_y + k_1 e_2 \\ v_y \sin e_x \sin e_1 \end{bmatrix}, $$

The derivative of $v_c$ becomes

$$ \dot{v}_c = \begin{bmatrix} e_x e_y - v_x \cos e_x \\ e_y e_x - v_y \sin e_x \\ k_1 \dot{v}_x + k_2 \dot{v}_y \end{bmatrix}, $$

where $v_x = -k_3 x + g(e, t)$, and

$$ g(e, t) = \|e\|^2 \sin t, $$

Different time-varying functions $g(e, t)$ are available in the literature, see [9] and the references therein. We have chosen the approach presented in [2].

Given the desired velocity $v_c(t) \in \mathbb{R}^2$, define now the auxiliary velocity tracking error as

$$ e_c = v_c - v, $$

Differentiating (20) and using (7), the mobile robot dynamics may be written in terms of the velocity tracking error as

$$ \dot{e}_c = -\dot{v}_c - \ddot{v} + f(x) + \tau_d, $$

where the important nonlinear mobile robot function is

$$ f(x) = \dot{v}_c + \ddot{v} + \dddot{v}, $$

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

$$ x = \begin{bmatrix} v^T \\ v_c^T \end{bmatrix}, $$

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

$$ \tau = \hat{f} + K_4 e_c - \gamma, $$

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 neural network. The robustifying signal $\gamma(t)$ is required to compensate the unmodelled unstructured disturbances. Using this control in (21), the closed-loop system becomes

$$ \ddot{e}_c = -(K_4 + \dddot{v}) e_c + \dddot{f} + \tau_d + \gamma, $$

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

$$ \dddot{f} = f - \hat{f}. $$

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.

### 3.3 Neural Net Controller

By using the controller (24), there is no guarantee that the control $\tau$ 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 error $e(t)$. In this section we shall use a NN to compute
the estimate \( \hat{f} \). A major advantage is that this can always be accomplished, due to the NN approximation property (10).

Some definitions are required in order to proceed:

d.1) We say that the solution of a nonlinear system with state \( x(t) \in \mathbb{R}^n \) is uniformly ultimately bounded (UUB) if there exists a compact set \( U_\delta \subset \mathbb{R}^n \) such that for all \( x(t_0) = x_0 \in U_\delta \), there exists a \( \delta > 0 \) and a number \( T(\delta, x_0) \) such that \( \| x(t) \| < \delta \) for all \( t \geq t_0 + T \).

d.2) Given \( A = [a_{ij}], B \in \mathbb{R}^{m \times n} \), the Frobenius norm is defined by

\[
\| A \|_F = \text{tr} \{ A^T A \} = \sum_{i,j} a_{ij}^2,
\]

with \( \text{tr} \{ \cdot \} \) the trace. The associated inner product is \( \langle A, B \rangle_F = \text{tr} \{ A^T B \} \).

d.3) For notational convenience we define the matrix of all the NN weights as \( \hat{Z} = \text{diag} \{ \hat{\mathbf{W}}, \hat{\mathbf{V}} \} \), and the weight estimation errors as \( \hat{\mathbf{V}} = \mathbf{V} - \hat{\mathbf{V}} \), \( \hat{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}} \), \( \hat{\mathbf{Z}} = \mathbf{Z} - \hat{\mathbf{Z}} \).

d.4) Define the hidden layer output error for a given \( x \) as

\[
\sigma = \mathbf{v} - \hat{\mathbf{v}} = \sigma(\mathbf{V}^T x) - \sigma(\hat{\mathbf{V}}^T x).
\]

The Taylor series expansion for a given \( x \) is

\[
\sigma(\mathbf{V}^T x) = \mathbf{v}(\mathbf{V}^T x) + \sigma'(\hat{\mathbf{V}}^T x) \hat{\mathbf{V}}^T x + O(\hat{\mathbf{V}}^T x),
\]

where \( O(\hat{\mathbf{V}}^T x) \) denotes the higher-order terms in the Taylor series.

The following mild assumptions always hold in practical applications:

a.1) The ideal weights are bounded by known positive values so that \( \| \mathbf{Z} \|_F \leq Z_{uu} \) with \( Z_{uu} \) known.

a.2) The desired velocity control input is bounded in the sense, for instance, that for some known constant \( V_c \in \mathbb{R} \),

\[
\| \mathbf{v}_c \|_F \leq V_c.
\]

The following are facts, and do not restrict the class of systems susceptible to our approach:

f.1) For each time \( t \), \( x(t) \) in (23) is bounded by

\[
\| x(t) \| \leq c_1 + c_2 \| e_c \|.
\]

for computable positive constants \( c_1 \).

f.2) For the sigmoid activation functions the higher-order terms in the Taylor series (29) are bounded by

\[
\| O(\hat{\mathbf{V}} x) \| \leq c_1 + c_2 \| \hat{\mathbf{V}} \|_F + c_3 \| \hat{\mathbf{V}} \|_F \| e_c \|,
\]

for computable positive constants \( c_2 \).

We will use a neural net to approximate \( f(x) \) for computing the control in (24). By placing into (24) the neural network approximation equation given by (11), the control input then becomes

\[
\mathbf{v}_t = \hat{\mathbf{W}}^T \sigma(\hat{\mathbf{V}}^T x) + K_e e_c - \gamma,
\]

and the velocity error dynamics is given by

\[
\dot{\mathbf{v}}_e = -(K_v + \mathbf{V}_e) e_c + \hat{\mathbf{W}}^T (\hat{\mathbf{v}} - \hat{\mathbf{v}}^T x) + \hat{\mathbf{W}}^T \sigma(\hat{\mathbf{V}}^T x) + (e + \epsilon_e) + \gamma.
\]

Adding and subtracting some terms and using the previously given definitions, assumptions and facts, the error system becomes

\[
\dot{\mathbf{v}}_e = -(K_v + \mathbf{V}_e - \mathbf{W}^T \sigma(\hat{\mathbf{V}}^T x) + \hat{\mathbf{W}}^T \sigma(\hat{\mathbf{V}}^T x) + w + \gamma.
\]

where the disturbance terms are

\[
w(t) = \hat{\mathbf{W}}^T \hat{\mathbf{v}}^T x + \mathbf{W}^T \mathbf{O}(\hat{\mathbf{V}}^T x) + \epsilon + \mathbf{z}_d.
\]

f.3) The disturbance term (35) is bounded according to

\[
\| w(t) \| \leq C_1 + C_2 \| \hat{\mathbf{Z}} \|_F + C_3 \| \hat{\mathbf{Z}} \|_F \| e_c \|.
\]

with \( C_i \) known positive constants. Details are discovered in (11).

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

**Theorem 1** Given a nonholonomic mobile robot (5), (6). Let the following assumptions hold:

a.3) A smooth time-varying auxiliary velocity control input \( v_c(t) \) is prescribed that solves the point stabilization problem for the steering system (5), neglecting the dynamics (6). A sample \( v_c \) [2] is given by (16).

a.4) \( K = [k_1 k_2 k_3 k_4]^T \) is a vector of positive constants.

a.5) \( K_a = k_4 I \), where \( k_4 \) is a sufficiently large positive constant.

Take the control \( \mathbf{v}_t \in \mathbb{R}^2 \) for (7) as (32) with robustifying term

\[
\gamma(t) = -K_a (\| \hat{\mathbf{Z}} \|_F + Z_{uu}) e_c - e_c
\]

and gain

\[
K_a > C_4
\]

with \( C_2 \) the known constant in (36). Let NN weight tuning be provided by (39). Then, the velocity tracking error \( e_c(t) \), the position error \( e(t) \), and the NN weight estimates \( \hat{\mathbf{v}}, \hat{\mathbf{w}} \) are UUB. Moreover, the velocity tracking error may be kept as small as desired by increasing the gain \( K_a \).

\[
\dot{\hat{\mathbf{W}}} = F \hat{\mathbf{v}} e_c^T - F \hat{\mathbf{v}} \hat{\mathbf{v}}^T x e_c^T - \kappa \hat{\mathbf{W}} \| e_c \| \hat{\mathbf{W}},
\]

\[
\dot{\hat{\mathbf{V}}} = G_x (\hat{\mathbf{v}}^T \hat{\mathbf{W}} e_c e_c^T - \kappa G \| e_c \| \hat{\mathbf{V}},
\]

where \( F, G \) are positive definite design parameter matrices, \( \kappa > 0 \) and the hidden-layer gradient \( \hat{\mathbf{v}} \) is easily computed—for the sigmoid activation function it is given by

\[
\hat{\mathbf{v}} = \text{diag} \{ \sigma(\hat{\mathbf{V}}^T x_d) \} \{ I - \text{diag} \{ \sigma(\hat{\mathbf{V}}^T x_d) \} \}
\]

**Proof:** See the Appendix.

**Remarks** The first terms of (39) are nothing but the standard backpropagation algorithm. The last terms correspond to the \( e \)-modification [12] from adaptive control theory; they must be added to ensure bounded NN weights estimates. The middle term in (39a) is a novel term needed to prove stability.

**4. Simulation Results**

We should like to illustrate the NN control scheme presented in Fig. 1. Note that the NN controller does not
require knowledge of the dynamics, not even their structure. The controller gains were chosen so that the closed-loop system exhibits a critical damping behavior: $K = [10 \ 5 \ 4 \ 1]^T$, $K_i = \text{diag}(25, 25)$. For the NN, we selected the sigmoid activation functions with $N_h = 10$ hidden-layer neurons, $F = G = \text{diag}(10, 10)$ and $\kappa = 0.1$.

It is well-known that the rates of convergence provided by smooth periodic velocity inputs are at most $1/\sqrt{I}$ [14], i.e., nonexponential. In order to have an acceptable closed-loop performance, we may use feedback laws which are smooth everywhere except at the boundary of a small neighborhood of the origin. The following choice has been proposed in [2]

$$g(e, t) = \begin{cases} \sin t & \|e\| \geq \varepsilon > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (41)$$

Although asymptotic convergence of the mobile robot cannot be guaranteed, the reference cart can be proven to be asymptotically stable. Therefore, the mobile robot can be stabilized to an arbitrarily small neighborhood of the origin. Simulation results that verify the validity of the combined kinematic/NN controller are depicted in Fig. 2-5.

![Fig. 2. Trajectory of the mobile robot.](image)

![Fig. 3. Position and orientation.](image)

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.

![Fig. 4. Some NN weights.](image)

![Fig. 5. Applied Torques: Right and (--) Left Wheels.](image)

**5. Conclusions**

A stable control algorithm for practical point stabilization of a nonholonomic mobile robot, and that does not require knowledge of the cart dynamics has been derived using a NN approach. 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.

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.

**APPENDIX: PROOF OF THEOREM 1**

Let the approximation property (10) hold with a given accuracy $\varepsilon_N$ for all $x$ in the compact set $U_x$.

Consider the following Lyapunov function candidate

$$V = \frac{k_2}{2}(e_1^2 + e_2^2) + \frac{e_3^2}{2} + V_1, \quad (42)$$

where

$$V_1 = \frac{1}{2}[e_1^T M e_1 + \nu (\tilde{W}^T F^{-1} \tilde{W}) + \nu (\tilde{V}^T G^{-1} \tilde{V})]. \quad (43)$$

Differentiating yields

$$\dot{V} = k_2 (e_1 \dot{e}_1 + e_2 \dot{e}_2) + e_3 \dot{e}_3 + \dot{V}_1. \quad (44)$$

Differentiating $V_1$, and using the skew symmetry property and the tuning rules give

$$\dot{V}_1 = -e_1^T K e_1 + \kappa \|e_2\| \|r (\tilde{Z}^T (Z - \tilde{Z})) + e_3 (\nu + \gamma) \quad (45)$$
since

\[ \text{tr}(\tilde{Z}(Z - \bar{Z})) = (\bar{Z}, \bar{Z})_F - \|\bar{Z}\|_F^2 \leq (\bar{Z}, \|Z\|_F - \|\bar{Z}\|_F^2) \] (46)

there results

\[ \dot{V} \leq -k_1 \|K_{\text{max}} \|e_1 + k_2 \|Z\|_F (\|Z\|_F - \|\bar{Z}\|_F) - C_1 \|Z\|_F - \|\bar{Z}\|_F \] (47)

where \( K_{\text{max}} \) is the minimum singular value of \( K \) and the last inequality holds due to (38). By substituting (47) and (15) in (44), we obtain

\[ \dot{V} \leq k_1 e_1 (v_e e_1 - v_e + \cos e_1) + k_2 (v_e e_1 + v_e - \sin e_1)
\]

\[ + e_2 (v_e - \|e_2\|_F^2 - \|e_2\|_F (\|Z\|_F - \|\bar{Z}\|_F)) + \kappa \|\bar{Z}\|_F (\|Z\|_F - \|\bar{Z}\|_F) \] (48)

\[-C_0 - C_1 \|\bar{Z}\|_F. \]

Defining \( k_1 = \frac{k_2}{4} \), \( k_2 > 0 \), \( k_3 > \frac{1}{4} \), yield

\[ \dot{V} \leq -k_3 (k_1 - \frac{k_2}{2}) e_1^2 - (k_3 - \frac{k_2}{4}) e_1^2 - (e_2 - \frac{k_3}{2}) e_1^2
\]

\[-(e_2 - \|e_2\|_F) (\|e_2\|_F - \|e_2\|_F) - C_0 - C_1 \|\bar{Z}\|_F \] (49)

Since the first four terms in (49) are negative, there results

\[ \dot{V} \leq -k_0 \|e_2\|_F \|e_2\|_F + \kappa \|\bar{Z}\|_F (\|Z\|_F - \|\bar{Z}\|_F) - C_0 - C_1 \|\bar{Z}\|_F. \] (50)

Thus, \( \dot{V} \) is guaranteed negative as long as either

\[ \|e_2\|_F > \frac{k_0 C_0 + C_2}{K_{\text{min}}} b_2 \] (51)

or

\[ \|Z\|_F > C_3 + \sqrt{C_2^2 + \frac{k_0 C_0}{\kappa}} \approx b_2 \] (52)

where

\[ C_2 = \frac{1}{2} (Z_0 + \frac{C_0}{\kappa}). \] (53)

Therefore, \( \dot{V} \) is negative outside a compact set. According to a standard Lyapunov theory extension, this demonstrates the UUB of both \( \|e_2\|_F \) and \( \|\bar{Z}\|_F \).

Finally from (18) the reference cart equation is given by

\[ \dot{x}_r = -k_0 x_r + g(e, t). \] (54)

Since \( g(e, t) \) tends to zero for the ideal case, and \( g(e, t) = 0 \) if \( \|e\| < \varepsilon \) for the practical case, it can be shown that \( x_r \rightarrow 0 \) as \( t \rightarrow \infty \).

\[ \square \]

References


