An Input–Output Based Robust Stabilization Criterion for Neural-Network Control of Nonlinear Systems


Abstract—In this paper, a stabilization method based on the input–output conicity criterion is presented. Conventional learning algorithms are applied to adjust the controller dynamics, and robust stability of the closed-loop system is guaranteed by modifying the training patterns which yield unstable behavior. The methodology developed expands the class of nonlinear systems to be controlled using neural control schemes, so that the stabilization of a broad class of neural-network-based control systems, even with unknown dynamics, is assured. Straightforwardness in the application of this method is evident in contrast to the Lyapunov function approach.

Index Terms—Conicity criterion, neural control, robustness, stabilization.

I. INTRODUCTION

Stability is the main property of feedback systems, and it is a crucial feature in the practical design of control systems based on neural networks. In fact, an analytical tool for the stability criterion would facilitate neuro-control applications for many kinds of real-world problems.

The stability theory of nonlinear systems has followed two main trends; stability analysis in the sense of Lyapunov and input–output (I/O) stability [17]. There have been several attempts to establish the relationship between Lyapunov and I/O stability for nonlinear systems, as is proven in [17] and [18] for the case of reachable and uniformly observable systems. In fact, Safonov’s separation stability criterion [12] comprises both these distinct approaches.

There is an extensive literature concerning studies on the local and global stability conditions of neural-network-based dynamic systems using the Lyapunov function approach [5], [14], [16] while I/O-based approaches have experienced an increasing development [7], [15] particularly in the field of fuzzy control systems [4], [11] and neural controlled systems [13].

This paper focuses on the fixed equilibrium stability of neural feedback controlled systems using the I/O-based conicity stability criterion. Specifically, a strategy to improve the design of a neural-based controller previously trained by standard neural learning algorithms is proposed. Starting from an initial controller’s training data set, a new modified training data set is elicited, so that the system output response characteristics of the controlled plant are maintained, while robust stability is assured.

II. STABILITY DEFINITIONS AND CONICITY CRITERION

Consider, following [12], the space $X = L^2$, called extended space, formed by signals $x(t)$ whose time-truncations $x_\tau(t)$ belong, for all $\tau > 0$, to $X = L^2$, the space of square integrable signals. Here, the truncated signal $x_\tau$ is defined by $x_\tau(t) = x(t)$ for $t \leq \tau$ and $x_\tau(t) = 0$ for $t > \tau$. A dynamical system $G$, with inputs $x(t) \in X_0 = L^2$ and outputs $y(t) \in Y_0 = L^2$ is regarded as a relation $G \subset X_0 \times Y_0$ formed by all the pairs $(x(t), y(t))$ compatible with $G$. In this way $G_0 = \{y; (x, y) \in G\}$, and the inverse relation $G^{-1}$ is obtained by interchanging inputs and outputs, and is always well defined. The system $G$ is finite-gain stable, with gain $g(G) = \|G\|$, when

$$g(G) = \sup \left\{ \frac{\|Gx\|}{\|x\|}; x \in X_0, \tau > 0, \|x_\tau\| \neq 0 \right\} < \infty.$$ 

Now consider the closed-loop dynamics of Fig. 1. The feedback relation $F$ maps $(u, v) \rightarrow (x, y)$ and is defined by $F = \{(u, v, x, y); (y, x - u) \in G, (x, y - v) \in H\}$. Finite-gain stability of the feedback system $F$ is addressed in [12] using a general form of the conicity criterion, based on two relations $C, R: X_0 \rightarrow Y_0$, called center and radius (aperture), and using the notion of conic sector. Cone $(C, R) = \{(x, y); \|y - Cx\| \leq \|R(x)\|, \forall \tau > 0\}$. We state here the simplified version [3], [17] obtained when $C, R$ are chosen to be constant matrices $C, R \in \mathbb{R}^{n \times n}$.

Theorem 1 (Conicity Criterion): The feedback system $F$ in Fig. 1 is finite-gain stable if there exist $C, R \in \mathbb{R}^{n \times n}$ such that

$$H$$ lies strictly inside cone $(C, R)$

$$G^{-1}$$ lies outside cone $(C, R)$.

Notice that the conicity conditions are sufficient and that the problem is to find (if they exist) $C, R$ such that the conditions hold. The search is simplified if $R = r_1I$, if $G$ is nonanticipative linear time-invariant, with $(n \times n)$ transfer function $G(s)$, and if $H$ is memoryless, given by a static relation $y = H(x)$. Then, it can be easily shown [3], [12], [17] that the conicity conditions are satisfied if for some $(C, r)$.

Manuscript received February 15, 2000; revised December 18, 2000. This work was supported by CICYT project under Grant TAP99-0926-C04-02.

J. Fernández de Cañete, A. García-Cerezo, and I. García-Moral are with the Departamento de Ingeniería de Sistemas y Automática, ETSI Industriales, Universidad de Málaga, Málaga 29071, Spain (e-mail: canete@ctima.uma.es).

A. Barreiro is with the Departamento de Ingeniería de Sistemas y Automática, ETSI Industriales, Universidad de Vigo, Spain.

Publisher Item Identifier S 1045-9227(01)05539-4.
Fig. 2. Stability conditions using the conicity domains.

1) Conicity of \( H \). There exist \( \epsilon > 0 \) such that, for all \( x \) and \( y = H(x) \):
\[
\|H(x) - Cx\| \leq (r - \epsilon)\|x\|. 
\]

2) Conicity of \( G \). The feedback system \( F(s) = G(s)(I - CG(s))^{-1} \) satisfies
\[
F(s) = G(s)(I - CG(s))^{-1} 
\]
is stable
\[
\|F(s)\| = \sup_{\omega \geq 0} \|F(j\omega)\| \leq \frac{1}{r}, 
\]
and, furthermore
\[
\|F(s)\| = \sup_{\omega \geq 0} \|F(j\omega)\| \leq \frac{1}{r}. 
\]

To deal with these conditions, define [4] the conicity domains for \( H \) and \( G \) as
\[
D_H = \{(C, r); \|H(x) - Cx\| \leq (r - \epsilon)\|x\|, \exists \epsilon > 0, \forall x \in \mathbb{R}^n\} 
\]
\[
D_G = \{(C, r); \|F(j\omega)\| \leq 1/r, \forall \omega \geq 0 \text{ and } F(s) \text{ is stable}\}. 
\]

Introduce now the definitions [4]
\[
r_{H}(C) = \sup_{x \neq 0} \frac{\|H(x) - Cx\|}{\|x\|} 
\]
\[
r_{G}(C) = \left(\sup_{\omega \geq 0} \|F(j\omega)\|\right)^{-1} = \inf_{\omega \geq 0} \|F(j\omega)\|^{-1}. 
\]

where \( r_{G}(C) \) is defined for those \( C \) satisfying (2), that is, those \( C \) taken from the set \( D_G = \{C; F(s) = G(s)(I - CG(s))^{-1} \text{ is stable}\} \). Then, from (1) to (5) it is clear (see Fig. 2) that \( r_{H}(C) \) and \( r_{G}(C) \) define the lower boundary of \( D_H \) and upper boundary of \( D_G \), respectively, in the \( C \times r \) space. If conicity holds for some \((C^*, r^*)\) then \( r_{H}(C^*) < r^* \leq r_{G}(C^*) \). In other words, conicity is equivalent to the nonempty intersection \( D_H \cap D_G \neq \emptyset \). This can also be expressed as in [4]: "if the robustness measure \( r_{G}(C^*) \) for the stable loop \( F \) formed by \( G(s) \) and \( C^* \) is greater than the deviation measure \( r_{H}(C^*) \) for \( H(x) \) compared to \( C^*x \), then the loop formed by \( G(s) \) and \( H(x) \) is stable."

III. STABILITY ANALYSIS

The stability conditions previously given in (1)–(3) apply to a closed-loop system formed by a linear time-invariant plant \( G(s) \) and a nonlinear static (memoryless) controller \( H(x) \), as shown in Fig. 1. Let us see how quite general systems, like (6) below, can be put in the linear-static separation form of Fig. 1. So, assume now a plant in the form
\[
\dot{x} = f(x, u) 
\]
where \( x = (x_1, \ldots, x_m)^T \) is the state vector, which is supposed to be available for the controller, \( u = (u_1, \ldots, u_p)^T \) is the input vector, and \( f: \mathbb{R}^{m+p} \rightarrow \mathbb{R}^m \) is a \( C^2 \) function defined on a neighborhood of the origin \( x = 0, u = 0 \) which is an equilibrium point \( f(0, 0) = 0 \). The control signal \( u \) is generated by a neural controller
\[
u = N N_e(c), \quad c = d - x 
\]
where \( e \) is the error signal with respect to a reference \( d \), that can be assumed \( d = 0 \), so that \( c = -x \). After adding and subtracting a fixed term \( \pm Ax \) we have
\[
\dot{x} = Ax - (Ax - f(x, N N_e(c))). 
\]
Then it is easy to see that the dynamics (8) can be put in the linear-static separation form of Fig. 1, by defining (see Fig. 3) an \((n \times n)\) transfer function (new plant)
\[
G(s) = (sI - A)^{-1} 
\]
connected by negative feedback to a nonlinear system (new controller)
\[
H(e) = Ac + f(-e, NN_e(c)), 
\]
The term \( \pm Ax \) in (8) can be regarded as a “loop transformation” [17], that does not affect stability. To apply the stability condition \( r_{G}(C) > r_{H}(C) \), we need to obtain \( r_{G}(C) \) using (2), (5), (9) and \( r_{H}(C) \) using (4), (10). This gives
\[
r_{H}(C) = \sup_{x \neq 0} \frac{\|f(-e, N N_e(c)) + (A + C)c\|}{\|x\|} 
\]
\[
r_{G}(C) = \inf_{s = 0} \frac{||((sI - A)^{-1}(I - C(sI - A)^{-1})^{-1})^{-1}||^{-1}}{\|x\|} 
\]
\[
= \inf_{s = 0} \frac{||((sI - (A + C))^{-1})^{-1}||^{-1}}{\|x\|}. 
\]

Theorem 2: Given \( f(x, u) \) and \( N N_e(c) \), if there exist square matrices \( A, C \) such that \( A + C \) is a stable matrix and such that \( r_{G}(C) > r_{H}(C) \) for the \( r_{G}(C) \) and \( r_{H}(C) \) given by (11), (12), then we can conclude stability of the system (6)–(8), that can also be written as in Fig. 3.

Remark 1—Choice of \( A \): Notice that one possible choice of \( A \) is the plant’s Jacobian matrix with respect to \( x \), \( D_x f(x, u) \), evaluated at the equilibrium (origin), that is \( A = D_x f(x, u) \big|_{x=0, u=0} \). It should be stressed that the Jacobian choice does not imply any kind of linear validity. The conicity criterion guarantees global stability (as long as the...
conic confinement is global) for the actual nonlinear system (not for its local linearization).

Remark 2—External or I/O Stability: In fact, the conic conditions guarantee stability in the external or I/O sense. This means that, for example, if the loop of Fig. 3 satisfies conicity for, say $C = 0$, then $r_H(0) < r_C(0)$ or $r_H(0)r_C^{-1}(0) < 1$, which is equivalent to the small gain condition

$$r_H(0) \cdot \frac{1}{r_C(0)} = \left( \sup_{e \in \mathcal{W}} \frac{|H(e)|}{|e|} \right) \cdot \left( \sup_{s \in \mathcal{W}} |G(s)| \right) = |H| \cdot |G| < 1. \quad (13)$$

Putting $\gamma_1 = |H|$, $\gamma_2 = |G|$ and $\gamma = \gamma_1 \gamma_2 < 1$, this implies [3], [17] that the $L^2$-norm of the error signal $\|e\|^2 = \int e^T e \, dt$, is bounded, for all $\tau > 0$. By $|e|_\tau \leq (1 - \gamma)^{-1}(\gamma_2 |v|_\tau + |d|_\tau)$. In a similar way, and for other $C \neq 0$, the conic conditions imply bounds in the form $|e|_\tau \leq K_0 |v|_\tau + K_d |d|_\tau$. So, the internal signals of the loop (see Fig. 3) remain bounded provided that the external signals $v, d$ are kept bounded. In particular, if $v, d \to 0$, then $e \to 0$ (local nominal stability).

Remark 3—Internal or Lyapunov Stability: Although internal stability is not explicitly guaranteed in conicity, both internal and external types of stability are, under mild conditions, equivalent [9], [18]. For example, in [9] the following result is stated.

Theorem 3: Consider the closed loop $\{G(s), H\}$ where $G(s) = C(sI - A)^{-1}B$ with $A, B$ controllable and $A, C$ observable, and where $H(\cdot)$ is memoryless. Then, if the conic conditions in the small-gain form $|G| \leq 1$, $|H| < 1$ are satisfied (any conic condition can be expressed in the small-gain form after suitable loop transformation [17]), then there exist positive definite matrices $P, Q > 0$ satisfying the Riccati equation $A^T P + PA + PBB^T P + C^T C + Q = 0$. Furthermore, the closed-loop dynamics $\dot{x} = Ax + BH(\cdot - Cx)$ are globally asymptotically stable and $V(x) = x^T P x$ is a Lyapunov function for this feedback system.

This result applies directly to our approach, so by using conicity we achieve both external (I/O) and internal (Lyapunov) stability. This is true even if the computations of $r_H(C)$ are made only within a finite, compact region, $x = -e \in W \subset \mathbb{R}^n$. In this way $r_H(C) = \sup \{ |H(x) - Cx|/|x| : 0 \neq x \in W \}$ is $W$-dependent. Then, using the previous result [9], we can obtain the Lyapunov function $V(x) = x^T P x$, defining a family of concentric ellipsoids, which is regionally valid within $W$, that is $\dot{V} < 0$ inside $W$. Then, $x(t) \to 0$ at least within the largest ellipsoid of “shape” $x^T P x$ contained inside the working domain $W$ where conic bounds have been checked.

Remark 4—Choice of $C$: Once we carry out the functional separation in (9) and (10), selecting $G(s)$, for simplicity, as the linearized system around the origin, we can analyze the stability of the complete nonlinear system. Notice that the complexity increases with the number of independent entries in $C$. A simplifying choice is $C = BK_f$, where $B$ is the input matrix of the linearized system, and $K_f$ is a vector of feedback gains which can vary in the set of gains which stabilize the linearized system. In this way, we reduce the complexity of the search for centers $C$ which define valid conic conditions.

Remark 5—Neural-Network Model of the Plant: In the case of partially known or unknown dynamics, as is well-known approximation results from [2] and [8] among others, the function $f(x, u)$ in (6) will be replaced by a multilayer feedforward neural network $NN_f(x, u)$ which represents the plant dynamics with a specified degree of approximation, the rest of the analysis remaining unaltered.

IV. STABILIZATION METHOD

As a starting point, we will assume that a neural controller $u = NN_c(e)$ has been trained by a standard learning algorithm [1]. This initial design $NN_c$ is assumed to have some good properties we desire to preserve as long as possible. This is the case, for example, when we obtain a starting solution $u = NN_c(e)$ which solves, only locally, the stabilization problem as well as some performance objectives (good local transients, output tracking error zero, etc.). Then, we would like to extend this solution to a region $e \in W$ as large as possible. But, due to nonlinearities, we are not able to find $C, r$ for which conicity is satisfied and then we are not able to guarantee closed-loop stability.

Bearing this situation in mind, our objective is to change $NN_c \to NN_c$ [and $H \to H$, because $H$ depends via (10) on $NN_c$] so that the new solutions satisfy conic stability. Also, as explained above, we would like to preserve, as long as possible, the initial design $H(e)$. In other words, we would like to make $H(e) \approx H(e)$ or $|H - H| = \sup |H(e) - H(e)|/|e|$ as small as possible. It has been proved [4] that the solution to this problem is, as shown in Fig. 4, to make a compensation based on “pushing” $H(e)$ within the bounds of a cone $(C^*, r^*)$. The center $C^*$ of this working cone is the one which minimizes

$$C^* = \arg \min_{C \in \mathcal{S}_r} \{ r_H(C) - r_C(C) \}$$
and the radius of the working cone is \( r^* = r_C(C^*) \). So, we need to detect for which values of \( e \) the function \( \bar{H}(e) \) “falls outside” the cone and then make the appropriate corrections. This is done in practice by computing a representative set of points \( (\epsilon_i, H(\epsilon_i)) \) with \( i = 1, 2, \ldots, p \) and \( p \) large, so that the \( \epsilon_i \) provide a fine mesh which describes, with a high degree of accuracy, the region where the compensation \( H(e) \rightarrow \bar{H}(e) \) is going to take place. Bearing this in mind, the compensation rule can be stated as: define \( \tilde{H}(\epsilon_i) \) so that

\[
\| \tilde{H}(\epsilon_i, \theta) - C^*\epsilon_i \| = \lambda r^* |\epsilon_i|, \quad \text{with} \ 0 \leq \lambda \leq 1. \tag{14}
\]

Then, using this inequality, it becomes apparent that several choices arise for the selection of \( \tilde{H}(\epsilon_i) \), from the center of the cone \( (\lambda = 0) \) to the surface of the cone \( (\lambda = 1) \), as well as intermediate values \( (0 < \lambda < 1) \). We will use \( \lambda = 1 \) if we wish to maintain as far as possible the features of the starting controller, \( H \approx \bar{H} \). Introducing (10) into (14), and choosing \( \lambda = 1 \)

\[
\| \hat{\delta}(\epsilon_i) - C^*\epsilon_i \| = \| f(-\epsilon_i, \tilde{N}\tilde{N}_C(\epsilon_i, \theta)) - (C^* - A)\epsilon_i\|
\]

where \( \tilde{N}\tilde{N}_C(\epsilon_i, \theta) \) is the modified value of the neural controller, and \( \theta \) has been included to make explicit the dependence on the network parameters. To train the network and derive an adaptation law for \( \theta \) we define a suitable index \( J(\theta) \) based on (15)

\[
J(\theta) = \frac{1}{2} \sum_{i=1}^{p} (\hat{\delta}(\epsilon_i, \theta) - \delta(\epsilon_i))^2 \tag{16}
\]

\[
\hat{\delta}(\epsilon_i, \theta) = \| \bar{H}(\epsilon_i, \theta) - C^*\epsilon_i \|
\]

(17)

\[
\delta(\epsilon_i) = r^* |\epsilon_i|. \tag{18}
\]

The parameters of the controller are updated following a gradient descent technique to minimize the performance function \( J(\theta) \). The problem with these methods is that they could find a local but nonglobal minimum. This drawback is common among other nonlinear optimization methods, but can be avoided using techniques like simulated annealing, combining on-line and batch-mode weight updating along with other modified backpropagation algorithms [6].

In any case, it is important to highlight that the conic confinement is also feasible (i.e., stability is guaranteed) when (14) is true for any \( 0 \leq \lambda \leq 1 \), not only for \( \lambda = 1 \) (see also discussion in Section V-C). This degree of freedom could be used and in some cases might relax the problem and make it feasible. Another way of relaxing the problem is to reduce the “working domain” \( W \) where conicity is tested, which is implicit in the “sup” computations in (4) for \( r_H(C^*) \) and \( r_H(C^*) \). If no solution appears for a domain \( W^f \subset \mathbb{R}^m \), we can try a smaller \( W \subset W^f \) until a solution appears. If the starting solution \( H \) is locally stable, in this way we can always obtain a regionally stable \( \tilde{H} \) for a suitable small region \( W \). If \( W \) is smaller, the problem becomes easier, but it is obvious that we are interested in a compact \( W \) as large as possible, to achieve larger stability regions. See also remark 3 in Section III.

V. ROBUSTNESS ANALYSIS

The stability results assured by the method outlined in the preceding sections are valid when applied to a plant with known dynamics \( \dot{x} = f(x, u) \). If the actual \( f \) is unknown, it can be identified by a neural-network model, \( NN_f \), see Remark 5. It should be noticed that the approximation ability of neural networks holds over compact subsets. This nonglobal, regional feature is also shared by the conic confinement (see Remark 3), so that there exists a compact set \( W \subset \mathbb{R}^m \) inside which all the results of our approach are valid. Now, let us consider robustness against conic-bounded and constant-bounded model errors.

A. Robustness Against Plant Conic Errors

If instead of the actual nonlinear system

\[
\dot{x} = f(x, u) = f(x, NN_c(-x)) =: f(x) \tag{19}
\]

we only have the neural model

\[
\dot{x} = NN_f(x, NN_c(-x)) =: NN_f(x) \tag{20}
\]

Fig. 4. Modification of the neural controller for the scalar case. (a) Unstable system. (b) Stable system.
then it can be shown that the conic stability of the model also implies stability of the actual system, so we have robustness against (small enough) model mismatch. The neural-network identification error can be reduced in different ways. Recalling from [10] the ability of neural networks for approximation over a compact set \( x \in W \subset \mathbb{R}^m \), not only of the function \( f \), but also of its derivatives, one can say that, given any \( \epsilon > 0 \), it is possible to find a neural network such that

\[
||\tilde{f}(x) - \bar{N}N_f(x)|| + ||\bar{f}(x) - \bar{N}_Nf(x)|| \leq \epsilon, \quad \forall x \in W.
\]

Here, the prime denotes the derivatives (gradients or Jacobians) with respect to \( x \), and \( || \cdot || \) are standard Euclidean norms. It should be stressed that the error \( \epsilon \) in (21) can be fixed a priori by the designer, and that the bound (21) can be achieved by using a large enough number \( N \) of network nodes. To be more precise, in [10] it is proved that (21) can be made true using a neural network with a single hidden layer comprising \( N \) hidden units if \( \epsilon \leq c/\sqrt{N} \). The results in [10] hold for a variety of Sobolev norms and activation functions. The coefficient \( c \) depends only on the problem structure (input dimension, activation function, error norm). It is constant for a fixed problem structure, and can be rigorously determined from those data.

To derive conic bounds from (21), let us assume a match at zero \( \bar{f}(0) = \bar{N}_Nf(0) \), which is easy to achieve by bias weights (alternatively, the error at zero can be treated as a small bias as in Section V-B below). Then we have

\[
||\bar{f}(x) - \bar{N}_Nf(x)|| = \left( \bar{f}(0) + \int_0^x \bar{f}(\xi) d\xi \right) - \left( \bar{N}_Nf(0) + \int_0^x \bar{N}_Nf(\xi) d\xi \right) \leq ||\bar{f} - \bar{N}_Nf|| ||\xi|| \leq \epsilon ||x||.
\]

So, derivative bounds imply conic bounds for the modeling error in the form

\[
||\hat{f}(x)|| \leq \epsilon ||x||.
\]

It can be shown, following the ideas in [4] that if the actual system \( \dot{x} = \bar{f}(x) = \bar{N}_Nf(x) + \delta(x) \) is subject to uncertainty \( \Delta \) in (21) bounded by \( ||\Delta|| \leq \epsilon \), then what has to be done is to repeat the same conic confinement process as in Section IV but changing \( r_C(C) \) to \( r_C(C) + \epsilon_+ \), where \( \epsilon_+ \) is the positive value [that is \( (\epsilon_+) = x \) if \( x > 0 \) and \( (\epsilon_+) = 0 \) if \( x < 0 \)]. In other words, the conic modeling errors \( \epsilon \) can be treated by reducing the cone aperture \( \tau \to (\tau - \epsilon) \) for the conic confinement procedure [4].

Remark 6: Compact Supports. It should be said that the fact that the neural approximation (21) is valid only over compact sets does not produce serious practical problems. If \( W_{\infty} \) is the compact set in (21) for which the approximation \( \bar{f} \approx \bar{N}_Nf \) holds, then \( W_{\infty} \) can be chosen as large as desired, covering all the state values which have practical interest.

At the same time, it can be seen that the finiteness of the neural approximation domain does not produce a serious “coupling problem” with the finiteness of the domain where \( r_H(C) \) has been computed, see Remark 3. Following this remark, let \( W_{H_{\infty}} \) be the compact domain where \( r_{H_{\infty}}(C) \) has been computed, with \( H_{\infty}(x) = -Ax + \bar{N}_Nf(x) \).

The presence of possibly different compact supports can be arranged in a simple way: by intersections. Put \( W = W_{\infty} \cap W_{H_{\infty}} \). Then, (21) is true for \( x = -\epsilon \in W \subset W_{\infty} \). Also, the (robust) conicity condition \( r_{H_{\infty}}(C) < r_C(C) - \epsilon \) is imposed for \( x \in W_{H_{\infty}} \), and so is valid for \( x \in W \subset W_{H_{\infty}} \). This implies, as \( ||\bar{H}(x) - H_{\infty}(x)|| \leq \epsilon ||x|| \), that \( r_{H}(C) < r_C(C) \).

So, we can identify \( W \) as the working domain where conic stability and neural approximation are simultaneously guaranteed.

B. Robustness Against Plant Bounded Errors

Suppose that the plant identification error contains a contribution in the form \( ||\delta(x)|| \leq \epsilon_0 \) instead of, or in addition to, \( ||\delta(x)|| \leq \epsilon ||x|| \). That is, we have constant-bound errors. The dynamics are then

\[
\dot{x} = \bar{f}(x) = \bar{N}_Nf(x) + \delta(x),
\]

If \( \delta(x) \) cannot be kept to zero for small \( x \) then it is impossible to maintain \( x = 0 \) as an attractive equilibrium. So, it is natural here to resume the external or I/O setting. Adding and subtracting \( \pm Ax \) in (23), one reaches the system in Fig. 3, where \( \delta(x) \) is identified with the time-varying signal \( u(t) \), e.g., of random nature, bounded in magnitude \( ||u(t)|| \leq \epsilon_0 \). Recalling the conicity properties in the I/O setting (see Remark 2), it is possible to obtain a “gain” \( K < \infty \) [that becomes smaller when \( r_H(C) \) becomes smaller] such that \( ||x||_r \leq K||u||_r \).

\[
\int_0^T ||u(t)||^2 dt \leq K^2 \int_0^T ||u(t)||^2 dt \leq K^2 \epsilon_0^2 T
\]

so that

\[
X_{\text{rms}} = \sqrt{\frac{1}{T} \int_0^T ||x(t)||^2 dt} \leq K \epsilon_0
\]

so the root-mean-square value \( X_{\text{rms}} \) of the state can be made as small as possible just by making the error \( \epsilon_0 \) small enough, and the equilibrium point \( x = 0 \) still remains an attractive equilibrium.

C. Robustness Against Controller Conic Errors

If the confinement problem (15) is solvable for \( \lambda = 1 \), or feasible for \( 0 \leq \lambda \leq 1 \), and \( u \equiv u(e) \) is a nonlinear feasible solution, the fact that this solution is obtained after a training phase where \( \bar{N}_Nc(e) \) approaches \( u(e) \) in a discrete set of training points may induce another source of inaccuracy or loss of robustness. To see this, let us assume, using the theory of neural approximation, that for a small \( \epsilon_c > 0 \)

\[
||u(e) - \bar{N}_Nc(e)|| \leq \epsilon_c ||e||.
\]

This will be true over a compact set \( W_{NN} \). However, this “regional” validity can be treated in a similar way as in Remark 6 so, for brevity, the discussion is omitted. If we assume additionally that \( f(x, \cdot) \) is Lipschitz, of Lipschitz constant \( c < \infty \), then
the deviation of $\hat{H}(e)$ with respect to the cone surface would be given by
\[
\|H(e) - C\| = \|f(-e, \hat{N}_N(e)) + Ae - Ce\| \\
\leq \|f(-e, u) + Ae - Ce\| + \|f(-e, \hat{N}_N(e)) - f(-e, u)\| \\
\leq r_G(C)\|e\| + \alpha_c\|e\| \\
= (r_G(C) + \alpha_c)\|e\|.
\] (27)

Therefore, training errors in $\hat{N}_N(e)$ increase the cone aperture in the same way as training errors do in $NN_f$. Robustness can be recovered by imposing a smaller aperture $r_G(C)$ which can cope with this effect as we described above in Section V-A.

VI. EXAMPLE

In order to show the performance of the neural control scheme, a simulation study was carried out using a cart-pole system whose state-space model is given by
\[
\begin{align*}
\dot{x}_1 &= (M + m)\sin x_1 - ml\sin x_1 \cos x_1 x_2 - mgsin x_1 \cos x_1 + u \\
\dot{x}_2 &= M + m(1 - \cos^2 x_1) x_1 \\
\dot{x}_3 &= ml\sin x_1 x_2 - mgsin x_1 \cos x_1 + u \\
\dot{x}_4 &= M + m(1 - \cos^2 x_1) x_1 \\
u &= NN_c(e).
\end{align*}
\]

The state variables $x_1$, $x_2$, $x_3$, and $x_4$ are angle of the pole to the vertical, rate of change of the angle, the position and the velocity of the cart, respectively. The input signal $u$ is the force applied to the cart's center of mass. The symbols $M$, $m$, and $l$ denote the mass of the cart, mass of the pole, and pole length with values $M = 2$, $m = 0.1$, $l = 0.5$.

In order to demonstrate the robustness of the method outlined here, an identification procedure was carried out, choosing initial values for $x$ uniformly distributed on the compact hypercube described by $x_1 \in [-\pi/2, \pi/2]$, $x_2 \in [-10, 10]$, $x_3 \in [-5, 5]$, and $x_4 \in [-5, 5]$. In fact, parameter uncertainty, even when an explicit model is known, suggests the use of the identification model instead of the known model. We have identified the nonlinear cart-pole system, obtaining a radial basis function neural net $NN_f \in N^{3}_{4,45,4}$ trained with orthogonal least squares and a neural controller $NN_c \in N^{3}_{4,35,1}$, also trained with the same algorithm.

For this case, we have a $4 \times 4$ center matrix, so after selecting all of the possible combinations of entries $c_{ij}$ in $C$, we have obtained $D_H \cap D_G = \emptyset$ for $C \in SC$ [Fig. 5(a)]. Therefore we are not able to guarantee that the closed loop is stable. In fact, one can find trajectories diverging from the origin [Fig. 5(c)]. In order to stabilize the cart-pole system, we have selected a center $C$ which leaves a subset of 127 points of 315 located outside the critical cone when considering $c_{21}$ and $c_{42}$ as the variable parameters of the cone center $C$. The modifications of the neural controller $NN_c(e)$ change the shape of $H(e)$, so that $D_H \cap D_G \neq \emptyset$ for some $C \in SC$ [Fig. 5(b)] and stability is guaranteed [Fig. 5(d)]. Similar performance results can be obtained.
starting from any set of initial conditions, with the modified control scheme showing a slightly degraded performance.

VII. CONCLUSION

A method has been derived based on I/O stability considerations for the stabilization of neural feedback control systems applied to a broad class of nonlinear plants with generally unknown dynamics. The conceptual simplicity of the geometric interpretation of stability through conicity conditions becomes evident in contrast to the Lyapunov function approach.

Another noteworthy advantage of conic stabilization is robustness against plant identification errors, both conic and bounded types. Conic stability also guarantees a domain of attraction that becomes larger as the conic domain becomes larger. In this way, conic-stable systems can tolerate model errors, exogenous disturbances and nonzero initial states. Future work will consider the application of the method described here to the neural control of experimental plants and the stabilization of neurofuzzy control schemes, while the design of the on-line version of the presented stabilization method is still under development.

ACKNOWLEDGMENT

The authors acknowledge the helpful suggestions from the reviewers and the Associate Editor of this TRANSACTIONS.

REFERENCES


