Proceedings of the 38th Conference on Decision & Control
Phoenix, Arizona USA • December 1999

ON ADAPTIVE CRITIC ARCHITECTURES IN FEEDBACK CONTROL

F.L. Lewis, J. Campos, and R. Selmic
Automation and Robotics Research Institute, The University of Texas at Arlington
7300 Jack Newell Blvd. South, Fort Worth, Texas 76118-7115
tel. 817-272-5972, fax. 817-272-5989, Email: flewis@controls.uta.edu

Research supported by ARO Grant 39657-MA

ABSTRACT

Two feedback control systems are designed that employ the adaptive critic architecture, which consists of two neural networks, one of which (the critic) tunes the other. The first application is a deadzone compensator, where it is shown that the adaptive critic structure is a natural consequence of the mathematical problem of inversion of an unknown function. In this situation the adaptive critic appears in the feedforward loop. The second application is the supervisory loop adaptive critic, where it is shown that the critic neural network requires additional dynamics that effectively give it a memory capability.

1 INTRODUCTION

The uses of neural networks (NN) in open-loop applications such as signal processing or system identification are significantly different than their applications in closed-loop feedback control applications. In the latter situation, it is necessary to take into account the interaction between the dynamics of the controlled system and that of the NN weight tuning algorithms, providing rigorous proofs of the boundedness of the tracking error, suitable performance guarantees, and proofs of the boundedness of all the NN weights. The literature of NN in open-loop applications has been rich, mathematically rigorous, and varied for years. By contrast, only during the past few years have stability proofs been given for neurocontrollers [4], [11], [16], [17],[19],[20],[21],[23],[24],[25]. By now, techniques for design and analysis of neurocontrollers are well established, so that one may have confidence in the performance of NN controllers when properly designed.

Most neurocontroller designs have relied on the function approximation property of NN [6], [10]. It would be desirable to use more advanced learning and intelligent features of NN in controls design as suggested in [29]. A particularly intriguing higher-level topology is the adaptive critic [2]. In the adaptive critic architecture there are two neural networks (NN), one of which (the critic) evaluates system performance and tunes the other (the action generating network), which in turn provides the control input signal for the system being controlled.

Papers dealing with control using adaptive critic neural nets are too numerous to mention. Most of these papers have some discussion and simulation results and no stability proofs. In this paper the adaptive critic architecture in feedback control is rigorously examined using nonlinear stability proof techniques. Two applications are given. In the problem of deadzone compensation [7], [8], [22], [27] it is shown that the adaptive critic structure is a natural consequence of the mathematical problem of inversion of an unknown nonlinear function [26]. In this situation the adaptive critic appears in the feedforward loop. The second application is the supervisory loop adaptive critic [2], where it is shown that the critic neural network requires additional dynamics that effectively give it a memory capability [3]. In both cases, the two NN composing the adaptive critic can be viewed as the two layers of a *single augmented NN*. The critic element is effectively the second layer, and the action generating NN the first.

2 BACKGROUND ON NEURAL NETWORKS

One may describe a 2-layer NN mathematically as

$$y = W^T \sigma(V^T x + v_0)$$

where V is a matrix of first layer weights, W is a matrix of second layer weights , and v_0 is a vector of first-layer thresholds. The second-layer thresholds are included as the first column of of the matrix W^T by augmenting the vector activation function $\sigma(w)$ by '1' in the first position. Tuning of the weights W then includes tuning of the second-layer thresholds too. One may similarly include the first-layer thresholds as the first column of matrix V^T by augmenting vector x by '1' in the first position, so that one can write alternatively

$$y = W^T \sigma(V^T x).$$

The main property of NN we are concerned with for control and estimation purposes is the *function approximation property* [6],[10]. Let $f(x)$ be a smooth function from $\Re^n \to \Re^m$. Then it can be shown that if the activation functions are suitably selected, as long as x is restricted to a compact set $S \in \Re^n$, then for some sufficiently large number of hidden-layer neurons L, there exist weights and thresholds such one has

$$f(x) = W^T \sigma(V^T x) + \varepsilon(x).$$

The value of $\varepsilon(x)$ is called *the neural network functional approximation error.* In fact, for any choice of a positive number ε_N, one can find a neural network such that $\varepsilon(x) \le \varepsilon_N$ for all $x \in S$.

If the first-layer weights are fixed, then the NN is linear in the adjustable parameters *W* (LIP). It has been shown that, if the first-layer weights *V* are suitably fixed, then the approximation property can be satisfied by selecting only the output weights *W* for good approximation. For this to occur, $\sigma(V^T x)$ must be a basis:

Definition [24]: Let S be a compact simply connected set of $\Re^n$ and let $\sigma(V^T x)$ be integrable and bounded. Then $\sigma(V^T x)$ is said to provide a basis for $C^m(S)$ if:

1. A constant function on S can be expressed as (8) for finite *L* for some value of *W*.
2. The functional range of neural network (8) is dense in $C^m(S)$ for countable *L*.

In this paper we select LIP NN, tuning only the output weights *W* for good performance. It was shown by Barron [1] that the neural network approximation error $\varepsilon(x)$ for LIP NN is fundamentally bounded below by a term of the order $(1/L)^{2/n}$. This does not limit the tracking performance in our controllers because of the control system structure selected.

3 FEEDFORWARD LOOP ADAPTIVE CRITIC: DEADZONE COMPENSATION

Deadzone compensation [7],[8],[22],[27] relies on designing a precompensator to ameliorate the deleterious effects of deadzone by effectively providing a pre-inverse of the deadzone. In this section it is shown that to invert an unknown nonlinear function, a structure using two neural nets can be used. One NN is in the actual feedforward path and estimates an inverse of the deadzone, while a second NN is on a higher level and tunes the first NN; it effectively estimates the deadzone itself as a sort of observer. The result is an adaptive critic architecture that arises directly from the mathematical considerations inherent in function inversion. In general, this proposed adaptive critic scheme can be used for inverting any continuous invertible function. Therefore it is a powerful result for compensation of general actuator nonlinearities in motion control systems.

3.1. Dynamics of Mechanical Motion Systems with Deadzone

The dynamics of mechanical systems with no vibratory modes can be written [15] as

$$M(\dot{q})\ddot{q} + V_m(q,\dot{q})\dot{q} + G(q) + F(q,\dot{q}) + \tau_d = \tau$$

where $q(t)\in \Re^n$ is a vector describing position and orientation and $\tau_d(t)\in \Re^n$ represents disturbances. The dynamics satisfy some well known physical properties as a consequence of the fact that they are a Lagrangian system. These properties are important in control system design and include the positive definiteness of M(q), the norm boundedness of $V_m(q,\dot{q})$, the skew symmetry of $\dot{M} - 2V_m$, and the boundedness of the disturbance $\tau_d(t)$.

To design a motion controller that causes the mechanical system to track a prescribed trajectory $q_d(t)$, define the *tracking error* by $e(t) = q_d(t) - q(t)$ and the *filtered tracking error* by $r = \dot{e} + \Lambda e$, where $\Lambda=\Lambda^T>0$ is a design parameter matrix. Common usage is to select Λ diagonal with large positive entries.

Differentiating, it is seen that the robot dynamics are expressed in terms of the filtered error as $M\dot{r} = -V_m r + f(x) - \tau + \tau_d$, where the nonlinear robot function is

$$f(x) = M(\dot{q})\ddot{q}_d + V_m(q,\dot{q})(\dot{q}_d + \Lambda e) + G(q) + F(q,\dot{q})$$

Vector x contains all the time signals needed to compute f(x), and may be defined for instance as $x \equiv \begin{bmatrix} e^T & \dot{e}^T & q_d^T & \dot{q}_d^T & \ddot{q}_d^T \end{bmatrix}^T$. It is noted that the function f(x) contains all the potentially unknown functions.

The desired trajectory is assumed bounded so that

$$\left\| \begin{matrix} q_d(t) \\ \dot{q}_d(t) \\ \ddot{q}_d(t) \end{matrix} \right\| \le q_B ,$$

with q_B a known scalar bound.

Let the input to the mechanical system have a deadzone [7], [8] so that $\tau(t)= D(u)$, with $u(t)$ the control input. Fig. 3.1 shows a nonsymmetric deadzone nonlinearity D(u) where u and τ are scalars. In general, u and τ are vectors.

A mathematical model for the deadzone characteristic of Fig. 3.1 is given by

$$\tau = D(u) = \begin{cases} g(u) < 0, & u \le d_- \\ 0, & -d_- < u < d_+ \\ h(u) > 0, & u \ge d_+ \end{cases} .$$

It is assumed that functions h(u) and g(u) are smooth and invertible continuous functions. These functions are very general, so this describes a very general class of deadzone. All of h(u), g(u), d_+, and d_- are assumed unknown, so that compensation is difficult.

3.2. NN Deadzone Precompensator

To offset the deleterious effects of deadzone, one may place a precompensator as illustrated in Fig. 3.3. There, the desired function of the precompensator is to cause the composite throughput from w to τ to be unity. In

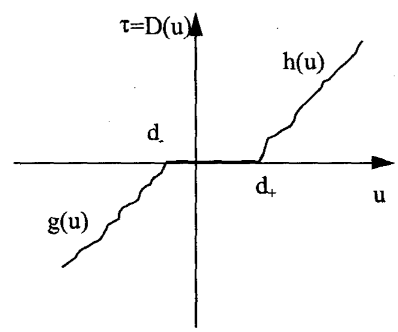

Figure 3.1 Nonsymmetric deadzone nonlinearity.

order to accomplish this, it is necessary to generate the pre-inverse of the deadzone. Techniques for adaptive deadzone compensation are given in a series of papers culminating in a book [22], [27]. Fuzzy and neural techniques are given in [12],[14]. The rigorous neural net compensation technique described herein is amplified in [26].

By assumption, the function D(.) is right invertible, therefore there exists a pre-inverse $D^{-1}(w)$, such that

$$D(D^{-1}(w)) = w \ . \qquad (3.1)$$

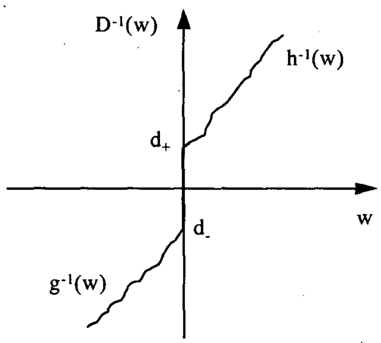

Figure 3.2 Deadzone inverse.

The function $D^{-1}(w)$ is shown in Figure. 3.2.

The mathematical model for the function shown in Fig. 3.2 is given by

$$D^{-1}(w) = \begin{cases} g^{-1}(w), & w < 0 \\ 0, & w = 0 \\ h^{-1}(w), & w > 0 . \end{cases}$$

The deadzone inverse $D^{-1}(w)$ can be expressed in equivalent form as

$$D^{-1}(w) = w + w_{NN}(w) \ ,$$

which has a direct feedforward term plus a correction term, the *modified deadzone inverse* w_{NN} given by

$$w_{NN}(w) = \begin{cases} g^{-1}(w) - w, & w < 0 \\ 0, & w = 0 \\ h^{-1}(w) - w, & w > 0 . \end{cases}$$

Function $w_{NN}(w)$ is discontinuous at zero.

Based on the NN approximation property, one can approximate the deadzone function by a NN so that

$$\tau = D(u) = W^T \sigma(V^T u + v_0) + \varepsilon(u) \qquad (3.2).$$

One can design a second NN for the approximation of the modified inverse function given by

$$w_{NN}(w) = W_i^T \sigma_i (V_i^T w + v_{0i}) + \varepsilon_i(w) \qquad (3.3)$$

In these equations $\varepsilon(u)$, $\varepsilon_i(w)$ are the NN reconstruction error and W, W_i, are ideal target weights. It is assumed that the ideal weights are unknown but bounded such that $\|W\|_F \le W_M$, $\|W_i\|_F \le W_{iM}$, with W_M, and W_{iM} known bounds.

The reconstruction errors are bounded on a compact set by $\|\varepsilon\| < \varepsilon_N$, $\|\varepsilon_i\| < \varepsilon_{Ni}$. The first-layer weights V, V_i, v_0, v_{0i} are fixed, and they must be properly chosen for the approximation property of the NN to be valid [26].

The ideal NN weights are unknown. The approximations of the nonlinear deadzone and modified deadzone inverse functions are given by

$$\hat{\tau} = \hat{D}(u) = \hat{W}^T \sigma(V^T u + v_0) \ , \qquad (3.4)$$

$$\hat{w}_{NN} = \hat{W}_i^T \sigma_i (V_i^T w + v_{0i}) \ . \qquad (3.5)$$

where tilde denotes the actual values of the weights appearing in the two neural nets. If the weights are suitably tuned, these values approximate the unknown ideal weights.

Note that expressions (3.4) and (3.5) represent respectively a NN approximation of the deadzone function and of the modified deadzone inverse. Signal $\hat{w}_{NN}$ is used for the deadzone compensation, and $\hat{\tau}$ represents the estimated value of signal τ.

For deadzone compensation, select the control input as

$$u = w + \hat{w}_{NN}(w) \ . \qquad (3.6)$$

Note we use two NN. The structure of the NN deadzone estimator and deadzone precompensator are shown in Fig. 3.3. The first NN, denoted NN1, has weights W^T and is effectively a sort of 'observer dynamics'. The second NN, denoted NN2, has weights W_i^T and is used as a deadzone compensator; it effectively estimates the deadzone pre-inverse. The requirement for such a two-NN structure is a direct consequence of the expression (3.1), and represents a technique for adaptively inverting any nonlinear invertible functions in industrial motion device actuators. Note that only the output of the NN2 is directly affecting the input u, while NN1 is a kind of higher level 'performance evaluator', the upcoming proof shows that it is used for tuning the NN2.

The next result shows the effectiveness of the proposed NN structure, by providing an expression for the *composite throughput error* of the compensator plus deadzone. It shows that, as the estimates $\hat{W}$, $\hat{W}_i$ approach the actual neural network parameters W, W_i, the NN precompensator effectively provides a *preinverse* for the deadzone nonlinearity. It is shown in the next section how to tune (3.2) and (3.3) so that tracking error is small, and $\hat{W}$ and $\hat{W}_i$ are close to W, W_i.. Define the weight estimation errors as

$$\tilde{W} = W - \hat{W} \ , \ \tilde{W}_i = W_i - \hat{W}_i$$

The proof of the next result [26] is important in that it shows that two NN are needed in the inversion of unknown functions, namely, one NN that estimates the function and another that estimates the function inverse.

Theorem 3.1 (Throughput Error Using NN Deadzone Compensation).

Given the NN deadzone compensator (3.5), (3.6), and the NN observer (3.4), the throughput of the compensator plus the deadzone is given by

$$\tau = w - \hat{W}^T \sigma'(V^T u + v_0) V^T \tilde{W}_i^T \sigma_i (V_i^T w + v_{0i})$$
$$+ \tilde{W}^T \sigma'(V^T u + v_0) V^T \hat{w}_{NN} + d(t)$$

where the modeling mismatch term d(t) is given by

$$d(t) = -\widetilde{W}^T \sigma'(V^T u + v_0) V^T W_i^T \sigma_i (V_i^T w + v_{0i}) - b(t) + \varepsilon(u)$$

with b(t) defined in the proof.

The next result [26] gives us the upper bound of the norm of d(t). It is an important result used in the stability proof.

Lemma

The norm of the modeling mismatching term d(t) is bounded on a compact set by

$$\|d(t)\| \le a_1 \|\widetilde{W}\|_F + a_2 \|\widetilde{W}_i\|_F^2 + a_3 \|\widetilde{W}_i\|_F + a_5 ,$$

where a_1, a_2, a_3, a_4, a_5 are computable constants.

3.3. Tuning the Adaptive Critic Deadzone Compensator

In the previous section, specifically in the proof of Theorem 3.1, it was shown that to estimate the inverse of an unknown function one requires two NN, one of which estimates the function and one of which estimates its inverse. In this section it is shown how to tune or learn the weights of the two NN in (3.4), (3.5) on-line so that the tracking error is guaranteed small and all internal states are bounded. It is assumed, of course, that the actuator output $\tau(t)$ is not measurable.

If f(x) is unknown, it can be estimated using adaptive control techniques, or the neural network controller in [19]. Let $\hat{f}(x)$ be an estimate for f(x). Since the main purposed of this paper is to use NN to compensate for the deadzone, to avoid distractions the estimate $\hat{f}(x)$ is fixed at a known nominal value in this paper and will not be adapted. This is common in robust control techniques [5]. The functional estimation error $\widetilde{f}(x) = f(x) - \hat{f}(x)$ is assumed to be bounded so that $\|\widetilde{f}\| \le f_M(x)$ for some known bounding function $f_M(x)$.

A robust compensation scheme for unknown terms in f(x) is provided by selecting the tracking controller

$$w = \hat{f}(x) + K_V r - v . \tag{3.7}$$

The feedback gain matrix $K_V>0$ is often selected diagonal and v(t) is a robustifying term to be selected for disturbance rejection. Deadzone compensation is provided using

$$u = w + \hat{w}_{NN} = w + \hat{W}_i^T \sigma_i (V_i^T w + v_{0i}) . \tag{3.8}$$

The multiloop control structure implied by this scheme is shown in Fig. 3.3, where $\underline{q} = \begin{bmatrix} q^T & \dot{q}^T \end{bmatrix}^T$, $\underline{q}_d = \begin{bmatrix} q_d^T & \dot{q}_d^T \end{bmatrix}^T$, $\underline{e} = \begin{bmatrix} e^T & \dot{e}^T \end{bmatrix}^T$. The controller has a proportional-derivative (PD) tracking loop with gains $K_V r = K_V \dot{e} + K_V \Lambda e$, where the deadzone effect is ameliorated by the NN feedforward compensator. The estimate $\hat{f}(x)$ is computed by an inner nonlinear control loop.

In order to design a NN system such that the tracking error r(t) is bounded and all internal states are stable, one must examine the *closed-loop error dynamics*

$$M\dot{r} = -V_m r - K_V r + \hat{W}^T \sigma'(V^T u + v_0) V^T \widetilde{W}_i^T \sigma_i (V_i^T w + v_{0i}) - \widetilde{W}^T \sigma'(V^T u + v_0) V^T \hat{w}_{NN} - d(t) + \widetilde{f} + \tau_d + v .$$

The next theorem provides algorithms for tuning the NN weights for the deadzone precompensator with guaranteed closed-loop stability.

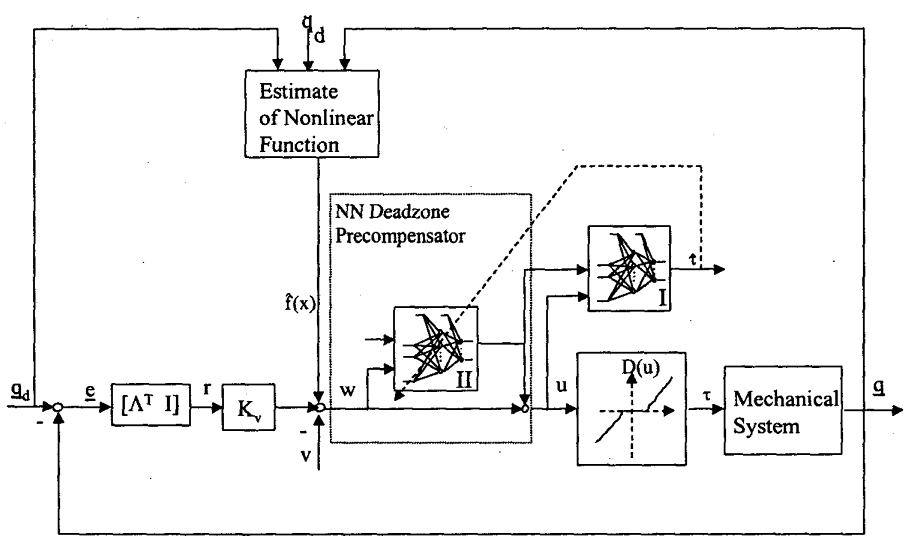

Figure 3.3 Tracking controller with NN deadzone compensation.

Theorem 3.2 (Tuning of NN adaptive deadzone compensator).
Select the tracking control law (3.7), plus the deadzone compensator (3.8). Choose the robustifying signal as

$$v(t) = -\left(f_M(x) + \tau_M\right)\frac{r}{\|r\|} ,$$

where the $f_M(x)$ and τ_M are bounds on functional estimation error and disturbance respectively. Let the estimated NN weights be provided by the NN tuning algorithm

$$\dot{\hat{W}} = -S\sigma'(V^T u + v_0)V^T \hat{W}_i^T \sigma_i(V_i^T w + v_{0i})r^T - k_1 S\|r\|\hat{W}$$

$$\dot{\hat{W}}_i = T\sigma_i(V_i^T w + v_{0i})r^T \hat{W}^T \sigma'(V^T u + v_0)V^T - k_1 T\|r\|\hat{W}_i - k_2 T\|r\|\|\hat{W}_i\|_F \hat{W}_i ,$$

with any constant matrices $S=S^T>0$, $T=T^T>0$, and k_1, $k_2>0$ small scalar design parameters. Then the filtered tracking error r(t) and NN weight estimates $\hat{W}$, $\hat{W}_i$ are UUB. Moreover, the tracking error may be kept as small as desired by increasing the gains K_v. In fact, an effective bound on the tracking error is given by

$$\|r\| \geq \frac{\frac{1}{4}k_1\left(W_M + \frac{a_1}{k_1}\right)^2 + C + a_5}{K_{V\min}} .$$

<u>Proof:</u> The proof relies on selecting the Lyapunov function candidate

$$V = \frac{1}{2}r^T M r + \frac{1}{2}\mathrm{tr}\left[\tilde{W}^T S^{-1}\tilde{W}\right] + \frac{1}{2}\mathrm{tr}\left[\tilde{W}_i^T T^{-1}\tilde{W}_i\right]$$

and demonstrating that $\dot{V}$ is negative outside a certain compact set [26]. This proof technique is by now standard in neural net control, but it is complicated here by the appearance of two NN.

■

The mutual dependence between NN1 and NN2 results in *coupled tuning law equations*. This mathematical result followed from (3.1) which dictates the fact that the information stored in NN1 and NN2 are dependent on each other. Examining the form of the first terms of the tuning laws and having in mind the backpropagation algorithm [28], it is evident that the proposed NN compensator with two NN's can be viewed as a *single NN structure with two layers of tunable weights*. In fact, NN1 can be interpreted as the second layer of this NN and NN2 as its first layer. This structure amounts to an *adaptive critic* architecture with a higher-level 'observer' NN tuning a lower order NN which generates the actual command signals.

4 SUPERVISORY LOOP ADAPTIVE CRITIC

In the previous section it was shown that the mathematical problem of inverting an unknown function leads to an adaptive critic structure with two NN. In fact, the critic NN and the action generating NN can be considered as the second and first layers of a *single NN with two tunable layers*. In this section it is shown that an adaptive critic having a similar structure can be placed in the outer feedback loops to improve the performance of closed-loop systems. In this situation, the critic NN needs some additional internal dynamics that give it a memory capability. Details on this work are provided in [3].

4.1. Dynamics of an mn-th order MIMO system

Consider an mn-th order multi-input and multi-output system given by the Brunovsky form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ &\vdots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= g(x) + u(t) + d(t) \\ y &= x_1 , \end{aligned}$$

with state $x = [x_1\ x_2 \ldots x_n]^T$, control input $u(t)$, output $y(t)$, $d(t)$ a disturbance with a known upper bound b_d, and $g(x): \Re^n \rightarrow \Re^m$ smooth functions. Many physical systems, such as robotic systems, can be represented in this form. It is assumed that the nonlinearity $g(x)$ in the system and the external disturbances $d(t)$ are unknown to the controller.

Let there be prescribed a desired trajectory and its derivatives

$$x_d(t) = \left[x_d \quad \dot{x}_d \quad \ldots \quad x_d^{(n-1)}\right].$$

The desired trajectory is assumed bounded so that

$$\|x_d\| \leq q_B$$

with q_B a known scalar bound.

Define the tracking error as $e(t) = x(t) - x_d(t)$, and the filtered tracking error $r(t) \in R^m$ as

$$r(t) = e^n(t) + \lambda_{n-1}e^{n-1}(t) + \cdots + \lambda_1 e(t) .$$

In matrix form one may write

$$r(t) = [\Lambda^T \quad 1] \cdot e(t),$$

where $e^{(n-1)}(t), \ldots, e^{(1)}(t)$ are the derivative values of the error $e(t)$, and $\lambda_1, \ldots, \lambda_{n-1}$ are constant values selected so that $\left|s^{n-1} + \lambda_{n-1}s^{n-2} + \cdots + \lambda_1\right|$ is Hurwitz. Thus, $e(t) \rightarrow 0$ exponentially as $r(t) \rightarrow 0$.

Using these equations the dynamics can be written in terms of the filtered error as

$$\dot{r} = g(e, x_d^{(n-1)}) + u(t) + d(t),$$

where $g(e, x_d^{(n-1)})$ is a nonlinear function of error vector e and the *(n-1)th* derivative of the trajectory x_d.

4.2. NN Compensation of Unknown Nonlinearity

According to the approximation properties of NN, the continuous nonlinear function $g(e, x_d^{(n-1)})$ can be represented as

$$g(e, x_d^{(n-1)}) = W_2^T \sigma(x_2) + \varepsilon(x_2), \qquad (4.1)$$

where the NN reconstruction error $\varepsilon(x_2)$ is bounded on a compact set by a known constant ε_N. The ideal NN weights W_2 that approximate $g(\cdot)$ are unknown.

Let the NN functional estimate for the continuous nonlinear function $g(e, x_d^{(n-1)})$ be given by a NN as

$$\hat{g}(e, x_d^{(n-1)}) = \hat{W}_2^T \sigma(x_2),$$

where $\hat{W}_2$ are the current weights estimating W_2.

Select now the control input given by

$$u(t) = -K_v r - \hat{W}_2^T \sigma(x_2) + v(t), \qquad (4.2)$$

where the control gain matrix is $K_v = K_v^T > 0$ and $v(t)$ is a robustifying vector that will be used to offset the NN functional reconstruction error $\varepsilon(x)$ and disturbances $d(t)$. One can now rewrite the closed-loop dynamics as

$$\dot{r} = -K_v r + \tilde{W}_2^T \sigma(x_2) + \varepsilon(x_2) + d(t) + v(t),$$

with the weight estimation error $\tilde{W}_2 = W_2 - \hat{W}_2$.

4.3. Adaptive Critic Feedback Controller (ACFC)

The adaptive critic architecture is described in [2],[29]. The tracking error $r(t)$ can be viewed as the real-valued *instantaneous utility function* of the plant performance. When $r(t)$ is small, system performance is good. The NN (4.1) is termed the *action generating NN* and is subsequently called NN2. It is now desired to tune the action generating NN in such a fashion that the tracking error *r(t)* is guaranteed to be small and the control *u(t)* is bounded.

To accomplish this using an adaptive critic architecture, introduce a second *critic NN*, here denoted NN1, that manufactures a *critic signal R* according to

$$R = \hat{W}_1^T \cdot \sigma(r) + \rho,$$

where $\hat{W}_1^T$ are the current weight values of the critic NN. The input to the critic is the signal r(t) which contains information on the performance of the system. The critic signal R must be used to tune the weights $\hat{W}_2^T$ of the action generating NN.

Signal ρ is an auxiliary term which will be detailed later. Its structure is determined by the requirements of the stability proof, and is a key feature in ensuring closed-loop stability with bounded NN weights.

Fig. 4.1 shows the architecture of the proposed ACFC controller, depicting the overall adaptive critic scheme whose details are subsequently derived. In the ACFC, the performance evaluation loop measures the system performance for the current system states by determining the instantaneous utility *r(t)*. This information is provided to the critic NN1 which then supplies the learning signal *R(t)* to tune the action generating NN2. Then, the action generating NN2 generates the counter-signal $\hat{g}(.)$ necessary to overcome the nonlinearities which the performance loop cannot deal with.

It is assumed that the ideal weights of both NN

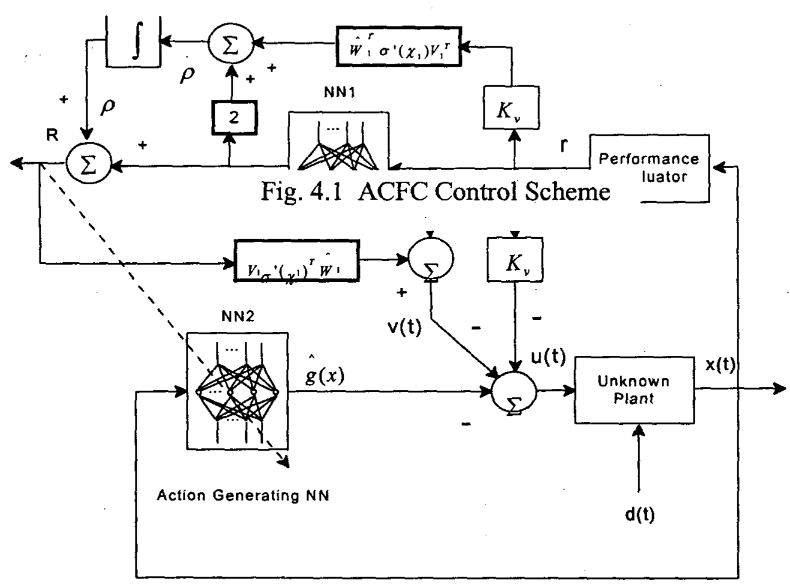

Fig. 4.1 ACFC Control Scheme

Fig 1: ACFC control scheme

are bounded by known positive values so that

$$\|W_1\|_F \le W_{1M}, \quad \|W_2\|_F \le W_{2M}.$$

4.4. ACFC Algorithm Dynamics and Tuning

The next theorem is our main result and shows how to adjust the weights of both NN to guarantee closed-loop stability.

Theorem 4.1 (Tuning of Adaptive Critic Feedback Controller).
Let the control action $u(t)$ be provided by (4.2) and the robustifying term be given by

$$v(t) = -k_z \cdot \frac{V_1\sigma'(x_1)\hat{W}_1 R + r}{\left\|V_1\sigma'(x_1)\hat{W}_1 R + r\right\|}$$

with $k_z \ge b_d$. Let the critic signal be provided by

$$R = \hat{W}_1^T \sigma(x_1) + \rho$$

with $\hat{W}_1^T \sigma(x_1)$ the output of a critic NN and ρ being an auxiliary adaptive term. Let the tuning for the critic and action generating NNs be

$$\dot{\hat{W}}_1 = -\sigma(x_1)R^T - \hat{W}_1$$
$$\dot{\hat{W}}_2 = \Gamma\sigma(x_2)\cdot\left(r + V_1\sigma'(x_1)^T \hat{W}_1 R\right)^T - \Gamma\hat{W}_2$$

with $\Gamma = \Gamma^T > 0$. Finally, let the auxiliary term ρ be dynamically defined by

$$\dot{\rho} = \hat{W}_1^T\left[2\sigma(x_1) + \sigma'(x_1)V_1^T K_v r\right]$$

Then the errors r, $\tilde{W}_1$, $\tilde{W}_2$ are *Uniformly Ultimately Bounded (UUB)*. Moreover, the performance measure $r(t)$ can be made arbitrarily small by increasing the fixed control gains K_v. In fact, an effective bound for the filtered tracking error is given by

$$\|r\| \ge \frac{W_{\max}^2}{\sqrt{2K_{v\min}}}.$$

Proof: [3] ■

Note that the signal $\rho(t)$ is the output of an integrator. This means that the critic NN2 has additional dynamics that effectively give it a *memory capability*. This form is required by the mathematical formulation of the problem if one goes through the details of the proof.

It is very interesting to note the relation between the critic NN1, which has weights W_1, and the action generating NN2, which has weights W_2. Though the two NN are used as distinct and separate networks, in the tuning algorithms they are coupled together. In fact, the first terms of the tuning algorithms are continuous-time versions of backpropagation [28] (note the Jacobian appearing in the update for $\hat{W}_2$). Having this in mind, it appears that the critic NN is effectively performing as the second layer of a single augmented NN with two layers of adjustable weights, which contains the action generating NN as layer number one. The philosophical ramifications of this are still under study.

REFERENCES

[1] R. Barron, “Universal approximation bounds for superpositions of a sigmoidal function,” *IEEE Trans. Info. Theory*, vol. 39, no. 3, pp. 930-945, May 1993.

[2] A.G. Barto, "Reinforcement learning and adaptive critic methods," *Handbook of Intelligent Control*, pp. 469-492, Van Nostrand Reinhold, New York, 1992.

[3] J. Campos and F.L. Lewis, "Adaptive critic neural network for feedforward compensation," *Proc. American Control Conf.*, San Diego, June 1999, to appear.

[4] F.-C. Chen, and H. K. Khalil, “Adaptive control of nonlinear systems using neural networks,” *Int. J. Contr.*, vol. 55, no. 6, pp. 1299-1317, 1992.

[5] M. J. Corless and G. Leitmann, “Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems,” *IEEE Trans. Automat. Contr.*, vol. 26, no. 5, pp. 850-861, 1982.

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

[7] C. A. Desoer and S. M. Shahruz, “Stability of dithered nonlinear systems with backlash or hysteresis,” *Int. J. Contr.*, vol. 43, no. 4, pp. 1045-1060, 1986.

[8] B. Friedland, *Advanced Control System design*, Prentice-Hall, New Jersey, 1996.

[9] B. Igelnik and Y. H. Pao, “Stochastic Choice of Basis Functions in Adaptive Function Approximation and the Functional-Link Net,” *IEEE Trans. Neural Networks*, vol. 6, no. 6, pp. 1320-1329, Nov. 1995.

[10] K. Hornik, M. Stinchombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, pp. 359-366, 1989.

[11] B.S. Kim and A.J. Calise, "Nonlinear flight control using neural networks," *AIAA J. Guidance, Control, Dynamics*, vol. 20, no. 1, pp. 26-33, 1997.

[12] J.-H. Kim, J.-H. Park, S.-W. Lee, and E. K. P. Chong, “Fuzzy precompensation of PD controllers for systems with deadzones,” *J. Int. Fuzzy Syst.*, vol. 1, pp. 125-133, 1993.

[13] Y. Kim and F.L. Lewis, *High-Level Feedback Control with Neural Networks*, World Scientific, Singapore, 1998.

[14] S.-W. Lee and J.-H. Kim, "Control of systems with deadzones using neural-network based learning control," *Proc. IEEE Int. Conf. Neural Networks*, June 1994, pp. 2535-2538.

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

[16] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 7, no. 2, pp. 1-11, Mar. 1996.

[17] F. L. Lewis, K. Liu, and A. Yesilidrek, "Neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 703-715, 1995.

[18] F. L. Lewis, K. Liu, R. R. Selmic, and Li-Xin Wang, "Adaptive fuzzy logic compensation of actuator deadzones," *J. Robot. Sys.*, vol. 14, no. 6, pp. 501-511, 1997.

[19] F.L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor and Francis, Lindon, 1998.

[20] K. S. Narendra, "Adaptive Control Using Neural Networks," *Neural Networks for Control*, pp. 115-142. ed. W. T. Miller, R. S. Sutton, P. J. Werbos, Cambridge: MIT Press, 1991.

[21] M. M. Polycarpou, "Stable adaptive neural control scheme for nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 41, no. 3, pp. 447-451, March 1996.

[22] Recker, P. V. Kokotovic, D. Rhode, and J. Winkelman, "Adaptive nonlinear control of systems containing a dead-zone," *Proc. IEEE Conf. Decis. Control*, 1991, pp. 2111-2115.

[23] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 24, no. 3, pp. 400-412, 1994.

[24] N. Sadegh, "A perceptron network for functional identification and control of nonlinear systems," *IEEE Trans. Neural Networks*, vol. 4, no. 6, pp. 982-988, Nov. 1993.

[25] R. M. Sanner and J. -J. E. Slotine, "Stable adaptive control and recursive identification using radial gaussian networks," *Proc. IEEE Conf. Decis. and Contr.*, Brighton, 1991.

[26] Rastko R. Selmic and F.L. Lewis, "Deadzone compensation in nonlinear systems using neural networks," *Proc. IEREE Conf. Decision and Control*, pp. 513-519, Tampa, Dec. 1998.

[27] G. Tao and P. V. Kokotovic, *Adaptive Control of Systems With Actuator and Sensor Nonlinearities*, John Wiley & Sons, New York, 1996.

[28] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550-1560, 1990.

[29] P. J. Werbos, "Neurocontrol and supervised learning: an overview and evaluation," in Handbook of Intelligent Control, D. A. White and D. A. Sofge, Eds. NY: Van Nostrand Reinhold, 1992, pp. 65-89.