15th Triennial World Congress, Barcelona, Spain

FEEDBACK LINEARIZATION USING A MINIMIZED STRUCTURE NEURAL NETWORK

Saman Orafa † **M. J. Yazdanpanah ‡**

Department of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran.
†: sa_orafa@yahoo.com
‡: yazdan@sofe.ece.ut.ac.ir

Abstract - For a class of single-input single-output continuous-time nonlinear systems, a three-layer neural network-based controller that feedback linearizes the system is presented. Control action is used to achieve tracking performance for a feedback linearizable but unknown nonlinear system. The control structure consists of a feedback linearization portion provided by two neural networks plus a robustifying portion that keeps the control magnitude bounded. This paper, in some sense, is the contribution of the work done in Yesildirek and Lewis,1995. It is shown that a new look at the weight update formulas makes it possible to obtain very simple network structures with only two neurons in their hidden layers, which results in a reduced number of controller equations without changing the corresponding stability results. This reduces network complexities and makes output tracking faster. It is shown that all the signals in the closed-loop system are uniformly ultimately bounded. No off-line learning phase is needed, Initialization of the network weights is straightforward. *Copyright © 2002 IFAC*

Keywords - Feedback linearization, high gain neuron, robust-adaptive control , Lyapunov stability.

1. INTRODUCTION

A typical control structure for feedback linearization of the state equation $\dot{x} = f(x) + g(x).u$ can be given by $u = \dfrac{-f(x) + \dot{x}_d}{g(x)}$. When dealing with unknown plant dynamics, we must compute the controller with $\hat{f}(x,\theta)$ and $\hat{g}(x,\theta)$. Adaptive schemes such as NN systems should be employed in a manner so that $\hat{g}$ will remain bounded away from zero for all times. Because of this problem, solutions are usually given locally and/or some additional prior knowledge about the system may be needed. For a certain class of nonlinear systems, Yesildirek and Lewis, 1995 introduce a controller structure that avoids the zero division problem regardless of NN weight estimates. They propose a control signal with a switching law and made use of a dead-zone to keep the controller output bounded.

This paper is organized as follows: In section (2), we define the plant dynamics and tracking problem. In section (3), we review briefly controller design equations and NN weight update formulas used in Yesildirek and Lewis,1995. In section (4), we will show that with a little modification in the neuron activation functions the tracking and feedback linearization goals can be achieved with only two neurons in the NN hidden layer. In section (5) simulation results are demonstrated. Finally, section (6) contains the concluding remarks where we discuss the effectiveness of our results and give suggestions for future works.

2. PROBLEM STATEMENT

This section defines the control task and details the underlying structural assumptions which are required in order to construct the controller.

2.1. The class of nonlinear systems to be examined, are single-input single-output state-feedback linearizable systems which are in the following controllability canonical form:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_n &= f(x)+g(x)u+d \\ y &= x_1 \end{aligned} \tag{1}$$

where $x=[x_1,x_2,...,x_n]^T$ and $d(t)$ is an unknown disturbance with a known upper bound b_d, and $f,g:R^n \to R$ are unknown smooth functions with $f(0)=0$ and

$$|g(x)| \geq \underline{g} > 0 \ \forall x \tag{2}$$

with $\underline{g}$, a known lower bound on $g(x)$.

Assumption 1. The sign of $g(x)$ is known.

2.2. *Tracking problem.*

The task of the controller is to force the plant output $y(t)$ and it's derivatives up to order $n-1$ to track a given desired output $y_d(t)$ and it's corresponding derivatives with an acceptable accuracy. (i.e., bounded error tracking), while all the states and controls remain bounded. Define a vector

$$x_d(t) = [y_d, \dot{y}, ..., y_d^{(n-1)}]^T \tag{3}$$

Assumption 2. The desired trajectory vector x_d is continuous, measurable and has a known upper bound $\exists Q$: $\|\mathrm{x_d}(t)\| \leq Q$ (4)

Define a filtered error vector as

$$r = \Lambda^T e \tag{5}$$

where $e = x - x_d$ and $\Lambda = [\lambda_1, \lambda_2, ..., \lambda_{n-1}]^T$ is chosen such that the polynomial $S^{n-1} + \lambda_{n-1}S^{n-2} + ... + \lambda_2 S + \lambda_1$ is Hurwitz. The time derivative of the filtered error can be written as

$$\begin{aligned} \dot{r} &= f(x) + g(x)u + d + Y_d \\ Y_d &= -y_d^{(n)}(t) + \sum_{i=1}^{n-1} \lambda_i e_{i+1} \end{aligned} \tag{6}$$

3. NEURAL NETWORK STRUCTURE

In this section we review briefly some of the main results stated in Yesildirek and Lewis,1995.

A three layer neural network structure is used for functional approximations in the controller. Such a net has two weight matrices $V \in R^{n+1 \times l_h}$ and $W \in R^{l_h+1}$ in hidden and output layers respectively. Also the standard sigmoid functions $\sigma(x) = \frac{1}{1+e^{-x}}$ are used in hidden layer and the output layer has a linear activation function. Let $h(x)$ be a continuous function, then there exist W_h and V_h such that

$$h(x) = W_h^T \sigma(V_h^T x) + \varepsilon_{l_h}(x) \tag{7}$$

with $\varepsilon_{l_h}(x)$ the minimal NN reconstruction error in u, (The NN thresholds are included in augmented x and σ.)

Assumption 3. Given $\sigma(x), x_d(t) \in U_d \subset U$, where $U \subset R^n$ is a compact set and a sufficiently large number of hidden units l_h, let the NN reconstruction error be bounded according to

$$\|\varepsilon_{l_h}(x)\| = \sup_{x\in U}\left|h - W^T\sigma\right| \le \varepsilon_h \qquad \forall x \in U \tag{8}$$

with ε_h a known bound and U_d a compact subset. Define a weight matrix $\Theta_h = \begin{bmatrix} V_h & 0 \\ 0 & W_h \end{bmatrix}$.

Assumption 4. The ideal NN weights W_h and V_h are bounded by some known constants, or

$$\|\Theta_h\| \le \Theta_{hm} \tag{9}$$

Since *h(x)* is a continuous in a compact set U , there exist C_3 and C_4 such that

$$|h(x)| = \left|W_h^T\sigma(V_h^T x) + \varepsilon_{l_h}(x)\right| \le C_3 + C_4|r| \qquad \forall x \in U$$

It was shown by Lewis et al. (1993) that for any continuous function *h(.), the* approximation error $\tilde{h}(x) = h(x) - \hat{h}(x)$ with the estimate $\hat{h}(x) = \hat{W}_h^T\sigma(\hat{V}_h^T x)$ can be written by using a Taylor series expansion of $\sigma(V_h^T x)$ as

$$\tilde{h}(x) = \tilde{W}_h^T(\hat{\sigma}_h - \hat{\sigma}_h{}'\hat{V}_h^T x) + \hat{W}_h^T\hat{\sigma}_h{}'\tilde{V}_h^T x + w_h \tag{10}$$

where $\hat{\sigma}_h = \sigma(\hat{V}_h^T x)$ and $\hat{\sigma}_h{}' \equiv [\partial\sigma_h(z)/\partial z|_{z=\hat{z}}$ is the Jacobian matrix. A bound on w_h is given by

$$|w_h(t)| \le C_0 + C_1\|\tilde{\Theta}_h\| + C_2|r|\|\tilde{\Theta}_h\| \tag{11}$$

where the C_i are computable constants.

4. ADAPTIVE NEURO CONTROLLER DESIGN

4.1. Proposed controller.

In the case of known functions with no disturbances the control law

$$u(x) = \frac{1}{g(x)}\left[-f(x) - K_v r - Y_d\right] \tag{12}$$

would bring r(t) to zero exponentially for any positive K_v, but we must use the approximations of ***f*** and ***g*** functions constructed by NN's as:

$$u_c(x) = \frac{-\hat{f}(\hat{\Theta}_f, x) - K_v r - Y_d}{\hat{g}(\hat{\Theta}_g, x)} \tag{13}$$

The control law (13) is not well defined when $\hat{g}(\hat{\Theta}_g, x) = 0$, therefore some attention must be taken to guarantee the boundedness of the controller a well.

To ensure the stability of the closed-loop system with a well defined control input, the following action has been proposed in Yesildirek and Lewis,1995:

$$u = u_c + u_d = u_c + \begin{cases} \frac{1}{2}(u_r - u_c)e^{\gamma(|u_c| - s)} & \text{if } I = 1 \\ (u_r - u_c)(1 - \frac{1}{2}e^{-\gamma(|u_c| - s)}) & \text{if } I = 0 \end{cases} \tag{14}$$

where $s>0$ is a design parameter, $\gamma < (\ln 2)/s$ and u_c is as defined in (13), with the time-varying gain given by:

$$K_v(t) = K_N + K_z\left[(\|\hat{\Theta}_f(t)\| + \Theta_{fm}) + s((\|\hat{\Theta}_g(t)\| + \Theta_{gm})\right] \tag{15}$$

with $K_N > 0$ *and* $K_z >$ *max*$\{C_2, C_4/s\gamma e\Theta_{gm}\}$ constant design parameters. The known bounds Θ_{im} for *i=f,g* are defined as in (9). The robustifying control term is

$$u_r = -\mu\frac{|\hat{g}|}{\underline{g}}|u_c|\,\mathrm{sgn}(r), \qquad \mu \ge 2, \tag{16}$$

and the indicator *I* is defined as

$$I = \begin{cases} 1 & \text{if } \hat{g} \ge \underline{g} \text{ and } |u_c| \le s \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

4.2. Stability analysis.

Theorem 1. Assume that the feedback-linearizable system is in the controllability canonical form and control input given by (1), Let (2) and assumptions 1-4 hold. Let the neural net weights update laws be provided by (18) and (19). Then the filtered tracking error r(t), neural net weight errors $\tilde{\Theta}_{f,g}(t)$ and control input are Uniformly

Ultimately Bounded with specific bounds given in (9). Moreover, the filtered tracking error r(t) can be made arbitrarily small by increasing the gain K_N .(note: $M_{f,g}, N_{f,g}$ are positive-definite matrices.)

The proof in Yesildirek and Lewis,1995 has been omitted due to the lack of space.

$$\begin{cases} \hat{f}(x) = \hat{w}_f^T \sigma\left(\hat{v}_f^T x\right) \\ \dot{\hat{w}}_f = M_f\left(\hat{\sigma}_f - \hat{\sigma}'_f \hat{v}_f^T x\right)r - K|r| M_f \hat{w}_f \\ \dot{\hat{v}}_f = N_f r x \hat{w}_f^T \hat{\sigma}'_f - K|r| N_f \hat{v}_f \end{cases} \quad (18)$$

$$\begin{cases} \hat{g}(x) = \hat{w}_f^T \sigma(\hat{v}_f^T x) \\ \dot{\hat{w}}_g = IM_g\left[(\hat{\sigma}_g - \hat{\sigma}'_g \hat{v}_g^T x)u_c r - \kappa|r||u_c|\hat{w}_g\right] \\ \dot{\hat{v}}_g = IN_g\left[u_c r x \hat{w}_g^T \hat{\sigma}'_g - \kappa|r||u_c|\hat{v}_g\right] \end{cases} \quad (19)$$

5. NEW STRUCTURE

Now, we are in a place to make our modifications in the system. As mentioned earlier, stability of the closed-loop system is shown without making any assumption on the initial NN weights. They are not required to be in the neighborhood of some ideal weights, which are unknown even for known dynamical models.

If we choose suitable initial values of NN weights $\hat{w}_i$ and $\hat{v}_i$ and positive-definite matrices M_i and N_i $(i = f,g)$ in (18) and (19), we can convert matrix (vector) differential equations of $\hat{v}_i$ $(\hat{w}_i)$ into the vector (scalar) ones.

To explain this, consider, for example, the first equation in (18) for updating $\hat{w}_f$ vector. For a short time, ignore the bias of hidden neuron. Let us choose initial conditions of $\hat{w}_f$ and $\hat{v}_f$ to be a vector of equal elements and a matrix of equal rows, respectively, i.e., let

$$\hat{w}_f^T(0) = [w, w, ..., w]_{1\times l_h} \quad (20)$$

$$\hat{v}_f^T(0) = \begin{bmatrix} v_1, v_2, ..., v_{n+1} \\ v_1, v_2, ..., v_{n+1} \\ \vdots \\ v_1, v_2, ..., v_{n+1} \end{bmatrix} \quad (21)$$

Choosing such a $\hat{v}_f$, we can see that all elements of the vector $(\hat{\sigma}_f - \hat{\sigma}'_f.\hat{v}^T.x)$ have equal initial values. (We have already chosen the standard sigmoid functions in all the hidden neurons.)

Next, the matrix M_f , can take a form such that each term in the right hand side of $\hat{w}_f$ equation becomes a vector with equal elements. The only restriction on the matrix M_f is it's P-D condition. A suitable candidate for this purpose is the identity matrix multiplied by a gain. Note that none of these settings affects the correctness of Theorem (1).

The above discussion leads to the following results: By selecting initial values of $\hat{w}_f$, $\hat{v}_f$ and matrix M_f as stated before, initial values of $\dot{\hat{w}}_f$ elements will be all equal. The same results hold true for the second equation in (18), i.e., for $\dot{\hat{v}}_f$, if the N_f matrix is choosen as in the M_f case. As the time proceeds, next values of $\hat{w}_f$ and $\hat{v}_f$ have the same structure as previous ones, i.e., $\hat{w}_f$ remains a vector of equal elements and $\hat{v}_f$ a matrix of equal vectors. As the following relations show, there is no need to further work on these repeatative structures:

$$\hat{f}(x) = \hat{W}^T(\sigma(\hat{V}^T x)) = [w, w, ..., w].\sigma\left(\begin{bmatrix} v_1, v_2, ..., v_{n+1} \\ v_1, v_2, ..., v_{n+1} \\ \vdots \\ v_1, v_2, ..., v_{n+1} \end{bmatrix}.x\right) \quad (22)$$
$$= K_f.w.\sigma(V_1^T x) = K_f.\hat{f}_1(x).$$

where $V_1^T = [v_1, v_2, ..., v_{n+1}]$ and K_f is the number of hidden layer neurons.

The structure looks like a 3-layer NN with one hidden neuron for approximation of $\hat{f}_1(x)$ multiplied by an adjustable gain. Stated otherwise, in this new scheme, “the output neuron gain”, plays the same role as the hidden neurons in previous one.

Now, we are able to present our next results from the mentioned re-arrangement of the NN structure. Making use of functions $\hat{f}(x) = K_f.\hat{f}_1(x)$ and $\hat{g}(x) = K_g.\hat{g}_1(x)$ in (13), we obtain:

$$u_c = \frac{-\hat{f}(x) - K_v r - Y_d}{\hat{g}(x)} = \frac{-K_f.\hat{f}_1(x) - K_v r - Y_d}{K_g.\hat{g}_1(x)} \quad (23)$$

Yesildirek ensures us to keep $\hat{g}(x)$ away from zero so $\hat{g}_1(x) > 0$ in any simulation.

Suppose in a given problem, we have gained a minimum value $\hat{g}_{\min} \succ 0$ for function $\hat{g}$ during the simulation time by suitable settings of design parameters, considering Yesildirek and Lewis method. The same problem can be solved easier as follows,Select K_g so that :

$$K_g \geq \frac{\underline{g}}{\hat{g}_{\min}} \tag{24}$$

So we have:

$$\hat{g}(x) = K_g . \hat{g}_1(x) \geq \frac{\hat{g}_1(x)}{\hat{g}_{\min}} . \underline{g} \geq \underline{g} \qquad \forall x \in R \tag{25}$$

Similarly, we can choose parameter K_g large enough such that the quantity u_c never goes beyond s, i.e.,

$$|u_c| \leq s \qquad \forall x \in R \tag{26}$$

Now, according to condition (17), parameter I will be always equal to 1. So there in no need to check this condition later, and the equations 18,19 follows:

$$u = u_c + \frac{1}{2}(u_r - u_c)\exp(\gamma(|u_c| - s)) \tag{27}$$

$$\dot{\hat{w}}_g = M_g[(\hat{\sigma}_g - \hat{\sigma}'_g \hat{V}_g^T x)u_c r - \kappa |r||u_c| \hat{w}_g]$$

$$\dot{\hat{V}}_g = N_g[u_c r x \hat{w}_g \hat{\sigma}_g^T - \kappa |r||u_c| \hat{V}_g] \tag{28}$$

Thus, we have eliminated the dead-zone in updating equation of $\hat{w}_g$ and $\hat{v}_g$, and this results in a faster convergence and tracking goal with simpler design equations. The cost we pay is to get larger bounds for final weight errors. We can also improve the approximation capability of the NN by adding a bias for hidden neuron.

6. SIMULATION RESULTS

In order to do the comparison easily, we pose the same example as in Yesildirek and Lewis,1995 for the purpose of simulation.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = (1 - x_1^2)x_2 - x_1 + \left(1 + x_1^2 + x_2^2\right)u \end{cases} \tag{29}$$

figure (1) shows the results obtained in Yesildirek and Lewis,1995 all of design parameters have been chosen in the same way as in Yesildirek and Lewis,1995,i.e.,

$s = 10, \gamma = 0.05, K_N = 20, \lambda_1 = 5$

$M_i = N_i = 20, \kappa = 0.1$ and $\mu = 4$.

with the rest set equal to 1. Initial Conditions are $\tilde{\Theta}_f(0) = 0$ and $\tilde{\Theta}_g(0) = 0.4$ and $x_1(0) = x_2(0) = 1$.

The desired trajectory is defined as $y_d(t) = \sin(t)$

Figure (2) shows the results by new method. Two neural networks with only one neuron (plus a bias) in their hidden layers are used to approximate f and g functions. The output neuron gains K_f and K_g are selected to be 10 and 1000 respectively, all the other parameters are the same as above. Figure (2) shows faster convergence of states to their desired references. Also the control signal has smaller initial value. These are due to the use of high output neuron gain in the new method which results in simplified equations (27) and (28) and could not be performed by the first method.

7. CONCLUDING REMARKS

This paper, in some sense, is the contribution of the work done in Yesildirek and Lewis,1995. It is shown that a new look at the weight update formulas makes it possible to obtain very simple neural network structures with only two neurons in their hidden layers, which results in a reduced number of controller equations without changing the corresponding stability results. This reduces network complexities and makes output tracking faster. It is shown that all the signals in the closed-loop system are uniformly ultimately bounded. It is also suggested that the neural approximation be used to approximate $\hat{g}_{-1} = \frac{1}{\hat{g}(x)}$ directly to avoid zero division problems.

REFERENCES

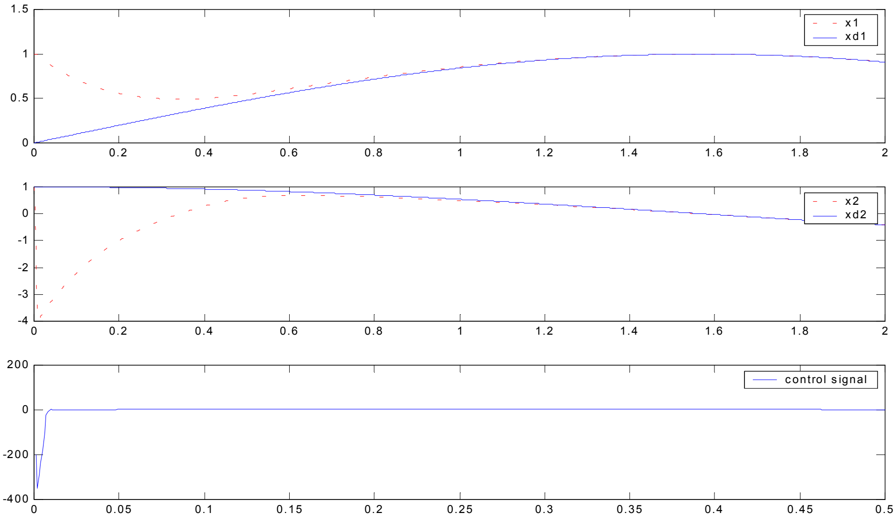
Yesildirek, A. and F. L. Lewis (1995). Feedback Linearization Using Neural networks. *Automatica* , **Vol.31**, No.11, pp. 1659-1664.

Yesildirek A. and F. L. Lewis (1994). Feedback Linearization Using Neural Networks. *IEEE proc. of Neural Networks*, pp. 2539-2544.

Omid Omidvar and D. L. Elliott (1997). *NEURAL SYSTEMS FOR CONTROL*, Academic press. Chap. 7, pp. 161-207.

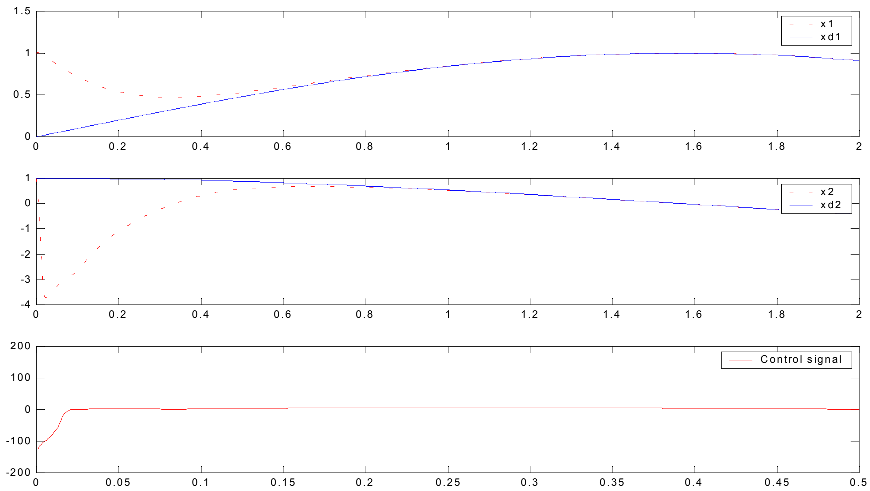
Hagan M. T. and B. Demuth (1999). Neural Networks for Control. *IEEE Proc. of American Control conf.*,

pp. 1642-1656.



Time(sec)

Fig. 1. Actual (x) and desired (xd) states and control signal obtained by Yesildirek and Lewis method.



Time(sec)

Fig. 2. Actual (x) and desired (xd) states and control signal obtained by new method.