

A Neural Network Approach of Input-Output Linearization of Affine Nonlinear Systems

Wei-Song Lin, Hong-Yue Shue and Chi-Hsiang Wang
Department of Electrical Engineering,
National Taiwan University, Taiwan, R.O.C.
e-mail : weisong@mailbox.ee.ntu.edu.tw

Abstract

For practical reasons, in the technique of feedback linearization, the requirements of mathematical modeling and access of internal states of complicated nonlinear systems should be removed. This paper demonstrates that, simply using output feedback, the input-output linearization of affine nonlinear systems with zero dynamics being exponentially stable can be accomplished by using multilayer neural network to estimate the instantaneous values of the nonlinear terms appearing in the feedback linearizing control law. Neither mathematical model nor internal state of the nonlinear system is required. The configuration for training the multilayer neural network as a device of the input-output linearizing controller is established. An example of affine nonlinear system is studied by computer simulations for various cases linearizing control.

1. Introduction

Feedback linearization based on concepts from differential geometry [2,6,9] can be adopted as a design methodology of nonlinear control systems. Using nonlinear coordinate transformation and feedback control, the original nonlinear model can usually be transformed into an equivalent linear one. With the terminology's, input-state and input-output linearizations refer respectively to complete [5,7,12] and partial linearizations [4,8,10]. For the linearized system, the well-developed design techniques of linear systems can be applied to solve for the appropriate controller. This approach has been successfully applied to a number of practical control plants such as high performance aircraft's, industrial robots and biomedical devices. Even though, there still exists major challenges for the feedback linearization design such as the requirement of precise mathematical model, the necessity of state feedback, and no guarantee of robustness in the appearance of parameter uncertainty or unmodeled dynamics [11].

Identifying the nonlinearity of control systems by neural networks through learning contributes another concept of nonlinear system design [1]. It is shown in this paper that, simply using output feedback and without the mathematical model of the system, the input-output linearization of an affine nonlinear system can be accomplished by estimating the decoupling matrix appearing in the approximated incremental linearizing control law with multilayer neural networks.

2. The Problem Of Neural-Network-Based Input-Output Linearization

Consider a square, affine nonlinear system with exponentially stable zero dynamics described by the following nonlinear differential equations

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t)) + \sum_{j=1}^m \mathbf{g}_j(\mathbf{x}(t))u_j(t) \\ \mathbf{y}(t) &= \mathbf{h}(\mathbf{x}(t)) \end{aligned} \quad (1)$$

where $\mathbf{x} \in \mathbf{R}^n, u_j \in \mathbf{R}, \mathbf{y} \in \mathbf{R}^m, \mathbf{f}: \mathbf{R}^n \rightarrow \mathbf{R}^n, \mathbf{g}_j: \mathbf{R}^n \rightarrow \mathbf{R}^n, \mathbf{h}: \mathbf{R}^n \rightarrow \mathbf{R}^m$ and \mathbf{f}, \mathbf{g}_j and \mathbf{h} are smooth functions of \mathbf{x} . (The zero-

dynamics of a nonlinear system is the dynamics of the system when the outputs are constrained to be identically equal to zero.) For simplicity the arguments regarding to time, unless noticed, are omitted by the expressions in the following context. In the designs of input-output linearization, each output, y_i , of equation (1) is differentiated with respect to time repeatedly until at least one of the control inputs appears in the output equation. If r_i represents the relative degree associated with y_i . Then the r_i^{th} -order derivative of y_i with respect to time can be written as follows :

$$y_i^{(r_i)} = L_f^{r_i} h_i(\mathbf{x}) + \sum_{j=1}^m L_{g_j} L_f^{r_i-1} h_i(\mathbf{x}) u_j \quad (2)$$

where h_i denotes the i^{th} component of \mathbf{h} , $L_f h_i(\mathbf{x}): \mathbf{R}^n \rightarrow \mathbf{R}$ and $L_{g_j} h_i(\mathbf{x}): \mathbf{R}^n \rightarrow \mathbf{R}$ stand for the Lie derivatives of $h_i(\mathbf{x})$ with respect to $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}_j(\mathbf{x})$, respectively. Equation (2) can be rewritten in vector-matrix form as follows

$$\mathbf{y}^{(r)} = \mathbf{c}(\mathbf{x}) + \mathbf{D}(\mathbf{x})\mathbf{u} \quad (3)$$

where $\mathbf{y}^{(r)} = [y_1^{(r_1)}, y_2^{(r_2)}, \dots, y_m^{(r_m)}]^T$ with r_1, r_2, \dots, r_m being the relative degrees of the system, the decoupling matrix is

$$\mathbf{D}(\mathbf{x}) = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(\mathbf{x}) & \dots & L_{g_m} L_f^{r_1-1} h_1(\mathbf{x}) \\ \vdots & & \vdots \\ L_{g_1} L_f^{r_m-1} h_m(\mathbf{x}) & \dots & L_{g_m} L_f^{r_m-1} h_m(\mathbf{x}) \end{bmatrix}$$

and

$$\mathbf{c}(\mathbf{x}) = [L_f^{r_1} h_1(\mathbf{x}), \dots, L_f^{r_m} h_m(\mathbf{x})]^T.$$

If the mathematical model of the nonlinear system is given, the design of input-output linearization is to solve for the control law, \mathbf{u} , so that the closed-loop characteristics of the system characterized by equation (3) is linearized. Unfortunately the availability of mathematical model is not the practical case of most complicated nonlinear systems. For many real world applications, the mathematical models are usually severe difficult to be identified. This reality keeps the feedback linearization approach of many complicated nonlinear systems remaining in the theoretical phase. Instead of using the mathematical model, our consideration is to accommodate the nonlinearity required in the linearizing control law by the learning ability of artificial neural networks. Fig. 1 shows a block diagram of this substitutive configuration. Comparing with equation (3), the neural network may provide the estimations of nonlinear functions, $\mathbf{D}(\mathbf{x})$ and $\mathbf{c}(\mathbf{x})$, through an appropriate learning process. This configuration releases the designer from the necessity of mathematically modeling the nonlinear plants. The goal is to establish the design procedures and show the feasibility of neural-network-based input-output linearization of nonlinear systems. Only square, affine nonlinear systems with zero dynamics being exponentially stable are considered.

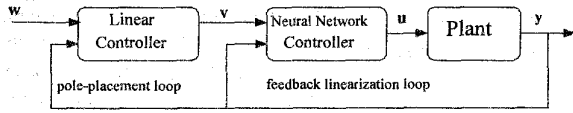


Fig. 1 A block diagram of neural network based feedback linearization.

3. The Incremental Linearizing Control Law

If the decoupling matrix, $\mathbf{D}(\mathbf{x})$, of the nonlinear system characterized by (1) is nonsingular, the exact linearizing control law is

$$\mathbf{u} = \mathbf{D}^{-1}(\mathbf{x})[-\mathbf{c}(\mathbf{x}) + \mathbf{v}] \quad (4)$$

where $\mathbf{v} \in \mathbf{R}^m$ is the auxiliary input vector. Applying (4) to (3) yields the following m decoupled linear, SISO, systems

$$\mathbf{y}^{(r)} = \mathbf{v}. \quad (5)$$

To stabilize the linearized system described by (5), many well-developed linear design methods can be applied. Pole-placement by output feedback may give control laws as

$$v_i = w_i - \alpha_{i1}y_i^{(r-1)} - \dots - \alpha_{i,r-1}\dot{y}_i - \alpha_{ir}y_i \quad (6)$$

for $i=1, 2, \dots, m$ to achieve closed-loop characteristics governed by

$$y_i^{(r)} + \alpha_{i1}y_i^{(r-1)} + \dots + \alpha_{i,r-1}\dot{y}_i + \alpha_{ir}y_i = w_i \quad (7)$$

where w_i 's are reference inputs, α_{ij} 's, $j=1, 2, \dots, r$ are the parameters to be chosen so that $s^r + \alpha_{i1}s^{r-1} + \dots + \alpha_{ir}$ are Hurwitz polynomials.

For the nonlinear system under consideration is with $\mathbf{c}(\mathbf{x})$ and $\mathbf{D}(\mathbf{x})$ being uniformly continuous functions of time over the interval $0 \leq t \leq L$. Then $\mathbf{c}(\mathbf{x})_{(k)} \rightarrow \mathbf{c}(\mathbf{x})_{(k-1)}$ and $\mathbf{D}(\mathbf{x})_{(k)} \rightarrow \mathbf{D}(\mathbf{x})_{(k-1)}$ are true for sufficiently small sampling interval of $T = t_k - t_{k-1}$, where $0 \leq T \leq L$ and the subscripts denote the time instant at t_k . This knowledge motivates the time-delay control law of input-output linearization as follows [13],

$$\mathbf{u}_{(k)} = \mathbf{D}^{-1}(\mathbf{x})_{(k)}[-\mathbf{y}_{(k-1)}^{(r)} + \mathbf{D}(\mathbf{x})_{(k-1)}\mathbf{u}_{(k-1)} + \mathbf{v}_{(k)}] \quad (8)$$

It has been shown that if one chooses an appropriate $\hat{\mathbf{D}}$ such that

$$\|\mathbf{I} - \mathbf{D}(\mathbf{x})\hat{\mathbf{D}}^{-1}\| < 1 \quad (9)$$

and further, keeps the auxiliary inputs, \mathbf{v} , as uniformly continuous functions of time, where \mathbf{I} represents an identity matrix with appropriate dimension. Then the system defined by (1) can be input-output linearized by applying the following approximated time-delay control law,

$$\mathbf{u}_{(k)} = \hat{\mathbf{D}}^{-1}[-\mathbf{y}_{(k-1)}^{(r)} + \hat{\mathbf{D}}\mathbf{u}_{(k-1)} + \mathbf{v}_{(k)}] \quad (10)$$

Here, using $\mathbf{D}(\mathbf{x})_{(k)} = \mathbf{D}(\mathbf{x})_{(k-1)}$ and after rearrangement of (8), we obtain the incremental linearizing control law as follows,

$$\Delta \mathbf{u}_{(k)} = \mathbf{u}_{(k)} - \mathbf{u}_{(k-1)} = \mathbf{D}^{-1}(\mathbf{x})_{(k)}[\mathbf{v}_{(k)} - \mathbf{y}_{(k-1)}^{(r)}], \quad \mathbf{u}_{(0)} = \mathbf{0}. \quad (11)$$

Using multilayer neural network to give the estimation of $\mathbf{D}(\mathbf{x})_{(k)}$ as $\hat{\mathbf{D}}_{(k)}$ at every sampling instant, the approximation of the incremental linearizing control law is obtained as follows,

$$\Delta \mathbf{u}_{(k)} = \mathbf{u}_{(k)} - \mathbf{u}_{(k-1)} = \hat{\mathbf{D}}^{-1}_{(k)}[\mathbf{v}_{(k)} - \mathbf{y}_{(k-1)}^{(r)}], \quad \mathbf{u}_{(0)} = \mathbf{0}. \quad (12)$$

In the digital realization of the linearizing controller, the uniformly continuous property of the auxiliary input is usually not guaranteed. Under this consideration, the following theorem shows the situations of the tracking error in applying the approximated incremental linearizing control law.

Theorem 1 Consider the nonlinear system and its controller being well defined by (1) and (12), respectively and $\mathbf{c}(\mathbf{x})$ and $\mathbf{D}(\mathbf{x})$

being uniformly continuous over the time interval $0 \leq t \leq L$. Then the tracking error, $\|\mathbf{v}_{(k)} - \mathbf{y}_{(k)}^{(r)}\|$, of the system satisfies the following conditions,

(i) if $\|\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}\| = 0$ and $\|\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}\|_i \leq \rho < 1$, then the tracking error converges exponentially to zero,

(ii) if $\|\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}\| \leq \|\mathbf{v}_{(k-1)} - \mathbf{y}_{(k-1)}^{(r)}\|$ and

$$\|\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}\|_i \leq \rho < \frac{1}{2}, \text{ then the tracking error converges exponentially to zero,}$$

(iii) if $\|\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}\| > \|\mathbf{v}_{(k-1)} - \mathbf{y}_{(k-1)}^{(r)}\|$, then the tracking error is bounded as

$$\|\mathbf{v}_{(k)} - \mathbf{y}_{(k)}^{(r)}\| \leq 2 \|\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}\|_i \|\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}\|.$$

Proof: Substituting (12) into (3), we have

$$\mathbf{y}_{(k)}^{(r)} = \mathbf{c}(\mathbf{x})_{(k)} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}\mathbf{y}_{(k-1)}^{(r)} + \mathbf{D}(\mathbf{x})_{(k)}\mathbf{u}_{(k-1)} + \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}\mathbf{v}_{(k)} \quad (13)$$

Using the continuous conditions of $\mathbf{c}(\mathbf{x})$ and $\mathbf{D}(\mathbf{x})$, we deduce

$$\mathbf{c}(\mathbf{x})_{(k)} + \mathbf{D}(\mathbf{x})_{(k)}\mathbf{u}_{(k-1)} = \mathbf{c}(\mathbf{x})_{(k-1)} + \mathbf{D}(\mathbf{x})_{(k-1)}\mathbf{u}_{(k-1)} = \mathbf{y}_{(k-1)}^{(r)} \quad (14)$$

and the substitution of this expression into (13) yields

$$\mathbf{y}_{(k)}^{(r)} = [\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}]\mathbf{y}_{(k-1)}^{(r)} + \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}\mathbf{v}_{(k)}. \quad (15)$$

After inserting $[\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}]\mathbf{v}_{(k-1)} - [\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}]\mathbf{v}_{(k-1)}$ into the right hand side of (15) and taking summation with

$$\mathbf{v}_{(k)} = \mathbf{v}_{(k-1)} + [\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}], \text{ we obtain}$$

$$[\mathbf{v}_{(k)} - \mathbf{y}_{(k)}^{(r)}] = [\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}][\mathbf{v}_{(k-1)} - \mathbf{y}_{(k-1)}^{(r)}] + [\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}][\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}] \quad (16)$$

Taking norm on vectors and induced norm on matrices of equation (16), the following inequality is obvious

$$\|\mathbf{v}_{(k)} - \mathbf{y}_{(k)}^{(r)}\| \leq \|\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}\|_i \|\mathbf{v}_{(k-1)} - \mathbf{y}_{(k-1)}^{(r)}\| + \|\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}\|_i \|\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}\| \quad (17)$$

case (i)- In this case, since $\|\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}\| = 0$ and

$$\|\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}\|_i \leq \rho < 1, \text{ (17) becomes}$$

$$\|\mathbf{v}_{(k)} - \mathbf{y}_{(k)}^{(r)}\| \leq \|\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}\|_i \|\mathbf{v}_{(k-1)} - \mathbf{y}_{(k-1)}^{(r)}\| \leq \rho \|\mathbf{v}_{(k-1)} - \mathbf{y}_{(k-1)}^{(r)}\| \leq \rho^k \|\mathbf{v}_{(1)} - \mathbf{y}_{(1)}^{(r)}\| \quad (18)$$

Since $\rho < 1$, (18) demonstrates that the tracking error converges exponentially to zero for sufficiently large value of k .

case (ii)- In this case since $\|\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}\| \leq \|\mathbf{v}_{(k-1)} - \mathbf{y}_{(k-1)}^{(r)}\|$, the

substitution of this relationship for $\|\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}\|$ on the right side of (17), we obtain

$$\|\mathbf{v}_{(k)} - \mathbf{y}_{(k)}^{(r)}\| \leq 2 \|\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}\|_i \|\mathbf{v}_{(k-1)} - \mathbf{y}_{(k-1)}^{(r)}\| \quad (19)$$

Using the condition of $\|\mathbf{I} - \mathbf{D}(\mathbf{x})_{(k)}\hat{\mathbf{D}}^{-1}_{(k)}\|_i \leq \rho < \frac{1}{2}$, (19) becomes

$$\|\mathbf{v}_{(k)} - \mathbf{y}_{(k)}^{(r)}\| \leq 2\rho \|\mathbf{v}_{(k-1)} - \mathbf{y}_{(k-1)}^{(r)}\| \leq (2\rho)^k \|\mathbf{v}_{(1)} - \mathbf{y}_{(1)}^{(r)}\| \quad (20)$$

Since $2\rho < 1$, (20) depicts that the tracking error converges exponentially to zero for sufficiently large value of k .

case (iii)- When $\|\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}\| > \|\mathbf{v}_{(k-1)} - \mathbf{y}_{(k-1)}^{(r)}\|$, the substitution of $\|\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}\|$ for $\|\mathbf{v}_{(k-1)} - \mathbf{y}_{(k-1)}^{(r)}\|$ on the right side of (17), we obtain

$$\|\mathbf{v}_{(k)} - \mathbf{y}_{(k)}^{(r)}\| \leq 2 \|\mathbf{I} - \mathbf{D}(\mathbf{x}) \hat{\mathbf{D}}_{(k)}^{-1}\| \|\mathbf{v}_{(k)} - \mathbf{v}_{(k-1)}\| \quad (21)$$

This proves the last case. ♣

For nonlinear systems being exactly linearized, the tracking error should always be kept at zero. However this is only possible when perfect model of the system is available. Alternatively, by using the approximated incremental linearizing control law, theorem 1 has revealed that if the decoupling matrix can be estimated accurately to some extent and the auxiliary input is manipulated finely. The tracking error of the linearizing control can either be constrained within a bound as in case (iii), or even converge exponentially to zero as in the cases of (i) and (ii). As a result, instead of (5), the imperfect input-output linearizing control results in a perturbed linear system described by

$$\mathbf{y}^{(r)} = \mathbf{v} + \xi \quad (22)$$

where ξ represents the vector of perturbation. For case (iii), the perturbation may be so large that the linearizing control being divergent, if the decoupling matrix and the step size of the auxiliary inputs are not chosen appropriately.

4. Neural Network Based Linearization 4.1. Issues of The Multilayer Neural Networks

A neural network is a massively parallel, interconnected network of elementary units called neurons. A neural network with multiple neurons is usually organized into a sequence of layers, called multilayer neural network. The general structure of an $(q+1)$ multilayer neural network with n_0 inputs and n_q outputs can be illustrated as shown in Fig. 2 and Fig. 3. The input layer usually acts as an input data holder and signals flow from the input layer through the hidden layers to the output layer. The output of each neuron in the p^{th} layer can be expressed as

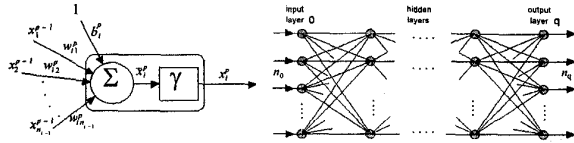


Fig. 2 A hidden neuron i of layer p .

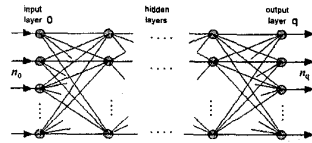


Fig. 3 A multilayer neural network.

$$x_i^p(t) = \gamma \left(\sum_{j=1}^{n_{p-1}} w_{ij}^p x_j^{p-1}(t) + b_i^p \right) \quad (23)$$

In applications of system modeling, it is common for the dynamic range of output data to be greater than 1, the activating function of the output node is therefore chosen to be linear. Thus the i^{th} output node performs a weighted sum of its inputs as follows

$$x_i^q(t) = \sum_{j=1}^{n_{q-1}} w_{ij}^q x_j^{q-1}(t) + b_i^q \quad (24)$$

4.2. Training of The Multilayer Neural Network for Input-Output Linearization

Training of the neural network is to determine w 's and b 's such that $x_i^q(t)$ of (24) is as close to the desired output as possible. Using Stone-Weierstrass theorem [3] it can be shown that a given nonlinear function under certain conditions can be represented by

a corresponding series such as Volterra series or Wiener series. The practical consequence of Stone-Weierstrass theorem is that an infinitely large neural network can model arbitrary piecewise continuous function. A finite network, however, may only accurately model such functions over a subset of the domain. Our interest is mainly in networks which permit on-line identification and control of dynamic systems in terms of finite dimensional nonlinear differential equations.

The method commonly used to evaluate the gradient of a performance function with respect to a weight vector of multilayer neural networks is called back propagation. If $J(\theta)$ is the performance index which has to be optimized with respect to the parameter vector θ . Then θ can be adjusted by according to the steepest decent method as the following equation

$$\theta(t) = \theta(t-1) - \eta \frac{\partial J(\theta)}{\partial \theta} \quad (25)$$

where $0 < \eta < 1$, the step size, is a chosen parameter. In a multilayer neural network, the performance index J is usually

chosen as $\sum_{i=1}^{n_q} (\chi_i - x_i^q)^2$ where χ_i 's are the desired values of x_i^q 's.

Thus the weights and the thresholds are respectively updated according to

$$\begin{aligned} w_{ij}^p(t) &= w_{ij}^p(t-1) + \Delta w_{ij}^p(t) \\ b_i^p(t) &= b_i^p(t-1) + \Delta b_i^p(t) \end{aligned} \quad (26)$$

with the increments $\Delta w_{ij}^p(t)$ and $\Delta b_i^p(t)$ being given by

$$\Delta w_{ij}^p(t) = \eta_w \delta_i^p(t) x_j^{p-1}(t) + \alpha_w \Delta w_{ij}^p(t-1) \quad (27)$$

$$\Delta b_i^p(t) = \eta_b \delta_i^p(t) + \alpha_b \Delta b_i^p(t-1)$$

where the subscripts w and b represent the weight and threshold respectively, α_w and α_b are momentum constants, η_w and η_b represent the learning rates and $\delta_i^p(t)$ is the error signal of the i^{th} neuron of the p^{th} layer. When the activating function of the output neuron is linear, the error signal at an output node is

$$\delta_i^q(t) = \chi_i(t) - x_i^q(t) \quad (28)$$

and for the neurons in the hidden layer

$$\delta_i^p(t) = \dot{\gamma}(\bar{x}_j^p(t)) \sum_{i=1}^{n_{p+1}} \delta_i^{p+1}(t) w_{ij}^{p+1}(t-1), \quad p = q-1, \dots, 2, 1 \quad (29)$$

where $\dot{\gamma}(\bar{x})$ denotes the first derivative of $\gamma(\bar{x})$ with respect to \bar{x} .

To realize the estimations of nonlinear functions $\mathbf{D}(\mathbf{x})$ and $\mathbf{c}(\mathbf{x})$ in equation (3) for input-output linearizing control, Fig. 4 shows a configuration for training the multilayer neural network. In order to calculate the derivatives of plant outputs correctly, the training input $\mathbf{u}(t)$ are generated by sending sequences of random signals through digital lowpass filters with bandwidth ω and sampling period T subject to the constraint $\omega T = 0.1$ rad. The lowpass filter is of 2nd-order Butterworth type and can be expressed as

$$\begin{aligned} \mathbf{u}(k) &= 1.8588\mathbf{u}(k-1) - 0.8681\mathbf{u}(k-2) \\ &\quad + 0.00233(\mathbf{r}(k) + 2\mathbf{r}(k-1) + \mathbf{r}(k-2)) \end{aligned} \quad (30)$$

where $\mathbf{r}(k)$ are random signals uniformly distributed in the operating range. The weights of the neural network are updated by using the steepest decent method represented by (27) with appropriate selections of the step size η and the momentum α to minimize the performance index

$$\begin{aligned} J &= \sum_{i=1}^m (y_i^{(n)} - \hat{y}_i^{(n)})^2 \\ &= \sum_{i=1}^m (y_i^{(n)} - \hat{c}_i - \sum_{j=1}^m \hat{a}_{ij} u_j)^2 \end{aligned} \quad (31)$$

where \hat{c}_i and \hat{d}_{ij} are the elements of $\hat{\mathbf{c}}$ and $\hat{\mathbf{D}}$, respectively, $\hat{y}_i^{(r)}$ is the i^{th} component of $\hat{\mathbf{y}}^{(r)}$ (the approximation of $\mathbf{y}^{(r)}$) and r_i is the i^{th} relative degree of the plant. Because the outputs of the neural network are $\hat{\mathbf{c}}$ and $\hat{\mathbf{D}}$ rather than $\hat{\mathbf{y}}^{(r)}$. The error term described by (28) for the back propagation learning algorithm are modified to

$$\delta_{c_i}^q = [y_i^{(r)} - \hat{y}_i^{(r)}], \text{ for } i = 1, 2, \dots, m \quad (32)$$

$$\delta_{d_{ij}}^q = [y_i^{(r)} - \hat{y}_i^{(r)}]u_j, \text{ for } i = 1, 2, \dots, m \text{ and } j = 1, 2, \dots, m$$

where $\delta_{c_i}^q$, $\delta_{d_{ij}}^q$ are the error signals for \hat{c}_i and \hat{d}_{ij} , respectively. The algorithms for training the weights of hidden layers of the neural network are kept unchanged. The most important thing of this training algorithm is that it requires only the measurements of outputs. This feature is practically necessary.

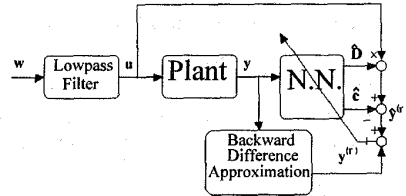


Fig. 4 The configuration for training the neural network.

4.3. Estimation of Relative Degree

To obtain the form described by (3) for the design of input-output linearization, the knowledge of relative degrees is indispensable. This paragraph presents a numerically feasible method to determine the relative degrees of a nonlinear system.

Theorem 2 Assume that a nonlinear system characterized by equation (1) has continuous states and with relative degrees r_1, r_2, \dots, r_m . Let the control input $u_p(t)$ associated with the relative degree of the i^{th} output y_i be a piecewise continuous function with a discontinuity at $t = t_i$, and $u_j(t) = 0$ for all $j \neq p$ of $\mathbf{u}(t)$. Then y_i and its derivatives to the order of $r_i - 1$ are continuous function of time, except that the r_i^{th} order derivative, $y_i^{(r)}$, has a discontinuity at $t = t_i$.

proof: Differentiating the i^{th} output equation of system (1) up to r_i order, we have

$$y_i^{(k)} = c_i^k(\mathbf{x}) \quad \text{for } k = 0, 1, \dots, r_i - 1; \text{ and} \quad (33)$$

$$y_i^{(r)} = c_i^{r_i}(\mathbf{x}) + \mathbf{d}_i(\mathbf{x})\mathbf{u} \quad (34)$$

where $c_i^k(\mathbf{x}) = L_{\mathbf{g}_i}^k h_i(\mathbf{x})$ for $k = 0, 1, \dots, r_i$, and

$\mathbf{d}_i(\mathbf{x}) = [L_{\mathbf{g}_i}^{r_i-1} h_i(\mathbf{x}) \quad \dots \quad L_{\mathbf{g}_i}^{r_i-1} h_i(\mathbf{x})]$ with $L_{\mathbf{g}_i}^{r_i-1} h_i(\mathbf{x}) \neq 0$ are smooth functions of \mathbf{x} . For continuous $c_i^k(\mathbf{x})$ and from (33), we obtain

$$\begin{aligned} & y_i^{(k)}(t_i^+) - y_i^{(k)}(t_i^-) \\ &= c_i^k(\mathbf{x}(t_i^+)) - c_i^k(\mathbf{x}(t_i^-)) = 0 \end{aligned} \quad \text{for } k = 0, 1, \dots, r_i - 1. \quad (35)$$

Similarly, from (34) and the continuity property of $c_i^k(\mathbf{x})$ and $\mathbf{d}_i(\mathbf{x})$, we achieve

$$\begin{aligned} & y_i^{(r)}(t_i^+) - y_i^{(r)}(t_i^-) \\ &= c_i^{r_i}(\mathbf{x}(t_i^+)) - c_i^{r_i}(\mathbf{x}(t_i^-)) + \mathbf{d}_i(\mathbf{x}(t_i^+))\mathbf{u}(t_i^+) - \mathbf{d}_i(\mathbf{x}(t_i^-))\mathbf{u}(t_i^-) \\ &= L_{\mathbf{g}_i}^{r_i-1} h_i(\mathbf{x}(t_i^+)) [u_j(t_i^+) - u_j(t_i^-)] \neq 0 \end{aligned} \quad *$$

Practically, to discover the relative degrees of a nonlinear system, one can drive the system with step or square wave inputs, then investigate the outputs and their derivatives for the appearances of discontinuities. The lowest order of the derivative of the i^{th} output which appears discontinuity corresponding to the jump or jumps

in any one input is the i^{th} relative degree of the system. A special case of SISO systems is that the nonlinear function $\mathbf{D}(\mathbf{x})$ at the occurrence of discontinuity can be calculated from (36) as

$$\mathbf{D}(\mathbf{x}(t_i)) = [y^{(r)}(t_i^+) - y^{(r)}(t_i^-)] / [u(t_i^+) - u(t_i^-)] \quad (37)$$

5. Simulation Results

Given a third order system described by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ \mathbf{c}(\mathbf{x}) + \mathbf{D}(\mathbf{x})\mathbf{u} \\ 5x_1 - 10x_3 \end{bmatrix} \quad (38)$$

$$y = x_1$$

where

$$\mathbf{c}(\mathbf{x}) = -8x_3 + 8 \frac{\exp(-x_2/2) - 1}{\exp(-x_2/2) + 1}$$

$$\mathbf{D}(\mathbf{x}) = 4 - \exp(-0.15x_1^2 + 1.1)$$

A three-layer neural network ($q=2$) with 10 neurons in the hidden layer is adopted as the learning device of all the following simulations. The training algorithm updates the weights of the neural network by the steepest decent method and using step size $\eta=0.01$ and momentum $\alpha=0.01$.

case (a)- using fixed value for all $\hat{\mathbf{D}}_{(k)}$.

With a unit step input $u(t) = u_s(t)$, the value of $\mathbf{D}(\mathbf{x})$ at $t = 0$ can be estimated by using (37). The result is

$$\begin{aligned} \hat{\mathbf{D}}_{(0)} &= (\dot{y}(0^+) - \dot{y}(0^-)) / (u(0^+) - u(0^-)) \\ &= (0.9957 - 0) / (1 - 0) \approx 1 \end{aligned} \quad (39)$$

For the desired closed-loop dynamics being described by,

$$\ddot{y}_d + 4\dot{y}_d + 6y_d = 6w \quad (40)$$

the controller with the approximated incremental linearizing control is

$$\Delta u_{(k)} = \hat{\mathbf{D}}_{(k)}^{-1} [v_{(k)} - \dot{y}_{(k-1)}] \quad (41)$$

where

$$v_{(k)} = -4\dot{y}_{(k-1)} - 6y_{(k-1)} + 6w_{(k)} \quad (42)$$

Using $\hat{\mathbf{D}}_{(k)} = 1$ for all time in (41), the simulation result is shown in Fig. 5 with the sampling period $T = 0.01$ sec., initial state $\mathbf{x}_0 = [0.5 \quad -1 \quad 0.2]^T$, and $w_{(k)} = 1$. Fig. 5 shows that the tracking of desired value, y_d , is in good quality. However, if $w_{(k)} = 4$, the feedback linearizing control system becomes unstable as shown in Fig. 6. This demonstrates the case (iii) of theorem 1.

case (b)- estimating $\mathbf{D}(\mathbf{x})$ on-line by neural network

Consider that the sequence of training input for the neural network is generated by sending the random signal $r(k)$ uniformly distributed in the interval $[-26, 34]$ through the lowpass filter (30) with sampling period $T = 0.01$ sec. $\mathbf{D}(\mathbf{x})$ is estimated by training the neural network for 30000 steps. Fig. 7(a) shows the satisfactory response of the system subject to the controller (40) even for $w_{(k)} = 4$. Fig. 7(b) shows that $\hat{\mathbf{D}}_{(k)}$ is not precisely equal to $\mathbf{D}(\mathbf{x})$.

case (c)- effect of noise

To investigate the effect of noise on the approximated incremental linearizing control, a noise $n(t) = 0.3\sin(20t)$ is superposed on the output of the plant. Fig. 8 shows the output response of the system by using the control law

$$\Delta u_{(k)} = \hat{\mathbf{D}}_{(k)}^{-1} [-\dot{y}_{(k-1)} + 40w_{(k)} - 10\dot{y}_{(k-1)} - 40y_{(k-1)}] \quad (43)$$

where $w_{(k)} = \sum_{i=0}^5 u_s(kT - i)$ and $\hat{D}_{(k)}$ is the estimation of $D(x)_{(k)}$ being obtained by the neural network. The desired closed-loop characteristics is described by

$$\ddot{y}_d + 10\dot{y}_d + 40y_d = 40w \quad (44)$$

The result depicts that the approximated incremental linearizing control can achieve good performance even in the presence of severe output noise.

6. Conclusion

Using output feedback and the learning ability of multilayer networks, it has been shown that the input-output linearization of affine nonlinear systems can be succeeded without mathematically modeling the nonlinear behavior. The configurations for the training and control of approximated exact and incremental linearizing control laws have been established. The incremental linearizing control is able to relax the tolerance of estimating error introduced by the neural network due to absence of internal states, disturbance of noise or inadequate capacity of the network itself. All the establishments are based upon that the affine nonlinear system being with exponentially stable zero dynamics. Given a nonlinear problem, there are still questions about what capacity of a neural network is necessary by the neural-network-based design left to be answered. As a conclusion, the approach with output feedback and neural learning has made feedback linearizing control more practical.

Acknowledgments - Financial support for this research from National Science Council of Taiwan R.O.C. under NSC 82-0404-E002-259 is gratefully acknowledged.

References

- [1] Bavarian, B. (1988). Introduction to neural networks for intelligent control. *IEEE Control Systems Magazine*, 8, 4, 3-7.
- [2] Byrness, C. I., and A. Isidori (1985). Global feedback stabilization of nonlinear systems. *Proceedings of the 24 IEEE CDC*, Ft. Lauderdale, FL., 1031-1037.
- [3] Cotter, N. E. (1990). The Stone-Weierstrass theorem and its application to neural networks. *IEEE Trans. Neural Networks*, 1, 290-295.
- [4] Freund, E. (1975). The structure of decoupled nonlinear system. *Int. J. Control*, 21, 443-450.
- [5] Hunt, L. R., R. Su, and G. Meyer (1983). Global transformations of nonlinear systems. *IEEE Trans. autom. Control*, 28, 24-31.
- [6] Isidori, A. (1989). *Nonlinear Control Systems, 2nd Edition*. Springer, New York.
- [7] Jakubczyk, B. and W. Respondek (1980). On linearization of control systems. *Bull. Acad. Polon. Sci. (Math.)*, 28, 517-522.
- [8] Krener, A. J., and A. Isidori (1983). Linearization by output injection and nonlinear observer. *Syst. Control Lett.*, 3, 47-52.
- [9] Nijmeijer, H., and A. J. van der Schaft (1990). *Nonlinear Dynamical Control Systems*. Springer, New York.
- [10] Porter, W. A. (1970). Diagonalization and inverses for nonlinear systems. *Int. J. Control*, 11, 67-76.
- [11] Slotine, J.-J. E., and W. Li (1991). *Applied Nonlinear Control*. Prentice-Hall, Englewood Cliffs, NJ.
- [12] Su, R. (1982). On the linear equivalents of nonlinear systems. *Syst. Control Lett.*, 2, 48-52.
- [13] Youcef-Toumi, K., and S. T. Wu (1992). Input/output linearization using time delay control. *ASME J. Dyn. Syst. and Meas.*, 114, 10-19.

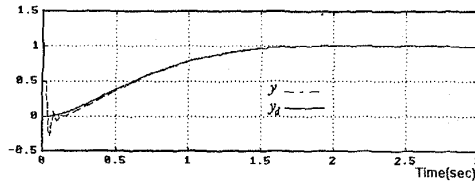


Fig. 5 Response when $\hat{D}_{(k)} = 1$, $w=1$.

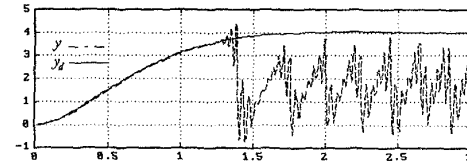


Fig. 6 Unstable results when $\hat{D}_{(k)} = 1$, $w=4$.

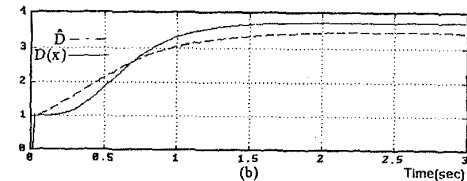
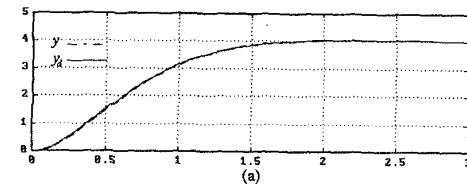


Fig. 7 Stable results when estimating $D(x)$ as $\hat{D}_{(k)}$.

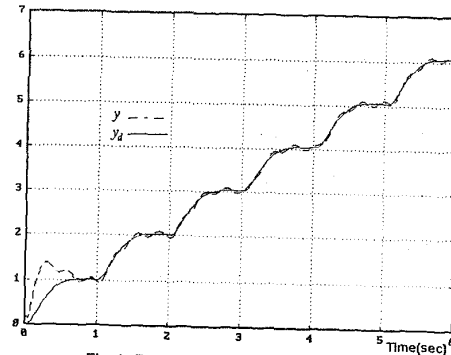


Fig. 8 Response when disturbed by noise.