NEURAL NETWORK APPROACHES FOR FEEDBACK LINEARIZATION

YIANNIS S. BOUTALIS

Automatic Control Systems Laboratory
Department of Electrical & Computer Engineering
Democritus University of Thrace
GR-67100 XANTHI, Greece
Tel. +30 25410 79504, FAX +30 25410 79512, e-mail: ybout@ee.duth.gr

Abstract: *In this paper a recent approach is reported, which performs feedback linearization of uncertain nonlinear systems using Artificial Neural Networks (ANNs). Also, a new ANN approach is presented, which tackles a special case of feedback linearization using ANNs. Instead of using the ANN as an estimator of the uncertain system dynamics the ANN is used as a compensator to the effects of the model uncertainties, which appear in the linearizing control law. The updating of the neural weights is carried out on-line using a conventional back propagation scheme, where the error to be minimized is chosen such that it ensures the stability of the tracking error system. The proposed method is tested on a well known nonlinear system and its application on a fermentation process is reported.*

Keywords: f*eedback control methods, output feedback, neural networks, feedback linearization, fermentation process*

1. Introduction

Most physical systems operations are nonlinear in nature and hence they should be described by means of nonlinear mathematical models. Since nonlinear models are not convenient for control purposes, due to both theoretical and computational reasons, they are often linearized by using appropriate exact or approximate linearization techniques [1], [2]. Among them, feedback linearization [2] is the most theoretically rigorous method; it consists in finding a feedback control law and a state variable transformation (diffeomorphism) such that the closed loop system model becomes linear, in the new coordinate variables. Feedback linearization techniques require the system to be controllable and the nonlinearities must satisfy some conditions to guarantee that the solution to the closed loop system exists. Moreover the application of this approach to feedback control of actual systems is quite limited because it relies on exact knowledge of system nonlinearities. In case the original system is characterized by uncertain parameters, external disturbances and unmeasured state variables, as it appears often in real life control systems, the linearization problem becomes particularly complex and almost inextricable.

General nonlinear systems of order n occur in the form

$$\begin{aligned} \dot{x} &= F(x,u) \\ \dot{y} &= H(x,u) \end{aligned} \qquad (1.1)$$

Where $x(t) \in \mathbb{R}^n$ is the internal state vector,

$u(t)\in\mathbb{R}^m$ is the control input and $y(t)\in\mathbb{R}^p$ is the measured system output. A state-space transformation applied to the system yields a linear model in state space canonical form, of order equal to the relative degree r of the original nonlinear system [2]. In this model, the r-th state equation is of the form

$$\dot{x}_r = f(x)+g(x)u+d \quad \left\{\begin{array}{c} x: \textit{state vector} \\ u: \textit{control input} \\ d: \textit{external disturbance} \end{array}\right\} \quad (1.2)$$

This equation is expressed in terms of the controller components, and, in some cases, in terms of the effect of the system's zero dynamics, which implies an additional first order nonlinear differential equation depending on the system's non-linearity and uncertainty. In case *r* equals the order of the system, *n,* the original system (1.1) is transformed to the well known Brunovsky canonical form (BCF)

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_n &= f(x)+g(x)u+d \\ y &= h(x) \end{aligned} \qquad (1.3)$$

Many systems occur naturally in Brunovsky form. For general systems of the form (1.1) it is often possible to be transformed to BCF, provided that an appropriate state transformation followed by an input transformation can be found. For this purpose, a reachability condition and an involutivity condition have to be satisfied by the initial system [2,3]. If these conditions are satisfied, then there exists a state space transformation that puts the system in Brunovsky form. It is not always easy to find the transformation. In a recent work [4] the ANN approximation properties are used to effectively estimate this transformation, so that NN controllers can be designed for a large class of nonlinear systems that are more general than Brunovsky form.

The reachability condition is usually satisfied for practical systems, but some classes of systems fail to be involutive. If the involutivity condition fails to hold, then it is still possible to perform partial feedback linearization, or input-output feedback linearization. In this case, there may exist some zero dynamics that must be dealt with [5]. A certain pathological condition may also hold with regard to the relative degree of the system. In this ill-defined relative degree situation even greater care is needed and 'almost' feedback linearization techniques are needed [6].

In this paper we show how to design ANN controllers for systems in Brunovsky form, assuming that, if the initial system is not in this form, there exists a transformation that puts it there. We are mainly concerned with the presence of incomplete knowledge about the model and/or uncertainties regarding its parameters. This means that functions $f(x)$ and $g(x)$ in (1.3) are not exactly known. In this equation, the control law, *u*, is usually designed to contain desired values of state x_n (considered as the output we try to control) and eliminates $f(x)$ and $g(x)$. Thus, it is of the general form

$$u = \frac{1}{g(x)}\left(-f(x)+\upsilon-d\right) \qquad (1.4)$$

where υ, as we will see in the next section, represents an appropriate function of the desired state values and possibly of some of its derivative. Under this control law, description (1.3) is usually converted to a linear error dynamical system in canonical form. Ensuring the stability of this system guarantees that the tracking error reaches zero in the steady state. When there is uncertainty about the model, functions *f(x)* and *g(x)* appearing in the above control low are replaced by their estimates $\hat{f}(x)$ and $\hat{g}(x)$. In this case *f(x)* and *g(x)* are not completely eliminated in (1.3) and thus the tracking error does not reach zero in the staedy state even if the error system is stable.

Last years, Artificial Neural Networks (ANNs) are often used in control problems of uncertain nonlinear systems [5, 7-10]. A problem often arising in control problems using ANNs with conventional training algorithms (e.g. Backpropagation) is that these algorithms do not guarantee that the performance of NNs will not deteriorate the stability of the whole process. This is overcome by describing the control objective as a tracking stability problem and the weights of the NNs are obtained on-line so that the tracking stability is ensured. Consequently, it is ensured that the ANN performance will not cause instability of the error tracking system.

For the general description (1.3), mentioned above, both cases of only *f(x)* or both *f(x)* and *g(x)* being unknown can be encountered in real problems. In this paper two neural network approaches are presented, which consider the case of only *f(x)* being unknown. In the first approach Neural Network estimators of *f(x)* are used in the control law [10]. It is assumed that all states, *x*, of the system are available for feedback. The weight updating of the two NNs is performed on-line and the weight updating rules arrise after a Lyapunov stability analysis of the error dynamical system. However, this approach is rather complicated (especially if the ANN estimators are not single layer) and some strict bounds on the weight of the NNs have to be set before hand. In the second approach, which is the author's proposition, a relatively simpler approach is proposed, where the neural network is not used as an estimator of *f(x),* but rather as a compensator to the effect of an uncertain function $\hat{f}(x)$ used by the linearizing control law. Moreover, the weight updating of the neural weights is carried out on-line using a conventional back propagation scheme, avoiding thus the derivation of a special purpose weight updating rule. Nevertheless, the stability of the tracking error system and the decay of the tracking error to zero are guaranteed by defining an appropriate error signal which has to be minimized by the trained ANN.

The paper is organized as follows: In section 2, the tracking controller and the error dynamics arising from the application of the linearizing control law in (1.3) are described. In section 3, the NN scheme for approximating function *f(x)* and the structure of the resulting linearizing controller is described. In section 4, the author's proposition for the structure of the linearizing controller and the use of the ANN is given. The simple training strategy that ensures the stability of the tracking error system is also presented. Simulation examples of the proposed method are given in section 5. Two simulation examples are given: the first one refers to the application of the method on a simple well known problem of the literature, while the second one is a brief report of the application of the method on the control of a fermentation process. Finally, section 6 presents the conclusions and thoughts for future work.

2. Output tracking Controller and Error Dynamics

Feedback linearization will be used to perform output tracking. Assume that, in (1.3) $y(t) = x_1(t)$. The control objective is defined as follows: given the desired output, $y_d(t)$, find a control law u(t) so that $y(t) = x_1(t)$ follows the desired trajectory with an acceptable accuracy, while all the states and control remain bounded. To design a tracking controller some mild assumptions are made, which are widely used [5]. Defining the desired trajectory vector as

$$x_d(t) \equiv \begin{bmatrix} y_d & \dot{y}_d & \cdots & y_d^{(n-1)} \end{bmatrix}^T \quad (2.1)$$

we assume that it is continuous, available for measurement and

$$\|x_d(t)\| \le Q \quad (2.2)$$

with Q a known bound.

Define a state error vector

$$e = x - x_d \quad (2.3)$$

with state $x(t) \equiv \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$, and a filtered tracking error as

$$r = \Lambda^T e \quad (2.4)$$

where $\Lambda \equiv \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_{n-1} & 1 \end{bmatrix}^T$ is an appropriately chosen coefficient vector so that $e \to 0$ exponentially as $r \to 0$ (i.e. $s^{n-1} + \lambda_{n-1}s^{n-2} + \ldots + \lambda_1$ is asymptotically stable).

Considering (1.3) and defining $e_{i+1} = y^{(i)} - y_d^{(i)}$, for $i = 1, 2, \ldots, n-1$ and $\overline{\Lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_{n-1} \end{bmatrix}^T$, the time derivative of the filtered error is written as

$$\dot{r} = f(x) + g(x)u + d + Y_d \quad (2.5)$$

where

$$Y_d = -x_d^{(n)} + \sum_{i=1}^{n-1} \lambda_i e_{i+1} = -x_d^{(n)} + \begin{bmatrix} 0 & \overline{\Lambda}^T \end{bmatrix} e \quad (2.6)$$

is a known signal.

In case the exact form of the nonlinear functions *f(x)* and *g(x)* is known, then the feedback linearization control law

$$u = \frac{1}{g(x)}\left(-f(x) - K_v r - Y_d - d\right) \quad (2.7)$$

when applied on (2.5) results in the following equation

$$\dot{r} = -K_v r \qquad (2.8)$$

which is a first order dynamic equation of the filtered tracking error. Any positive K_v is appropriate for making (2.8) stable and $r \to 0$. Then, since (2.4) is a stable system, *e* is bounded and the tracking performance is achieved. In (2.7) the disturbance term, *d,* was considered known, which however is not usually true in practice. If this term is unknown, one should remove it from (2.7) and take care for its effect on the stability of (2.8).

If *f(x)* and *g(x)* are not exactly known, then their estimates $\hat{f}(x)$, $\hat{g}(x)$ should be put in (2.7) and the control law becomes

$$u_c = \frac{1}{\hat{g}(x)}\left(-\hat{f}(x) - K_v r - Y_d\right) \qquad (2.9)$$

An important issue in this kind of controller is to guarantee the boundedness of $\hat{g}$ away from zero and take a well defined control problem [5]. However, this issue is not of our concern, since in this paper *g(x)* is considered to be known.

In the next two sections two alternative NN approaches are presented to tackle the case of *f(x)* being unknown. In the first approach the ANN is used as an estimator of the unknown function *f(x)*, while in the second approach, which is the author's proposition the ANN is introduced in the formation of the control law to act as a compensator to the uncertainties introduced by the incomplete knowledge of *f(x)*.

3. Feedback linearizing Controller using NN approximation of function *f(x)*.

Let the unknown function *f(x)* is approximated by a neural network of one hidden layer. This implies that on a compact set of $\mathbb{R}^n$ there exist ideal target NN weights so that

$$f(x) = W^T \sigma(V^T x) + \varepsilon \qquad (3.1)$$

where the functional estimation error is bounded so that

$$\|\varepsilon\| < \varepsilon_N \qquad (3.2)$$

for a known constant ε_N. The neural weights can be denoted in a more compact form by the weight matrix

$$\Theta = \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \qquad (3.3)$$

It is assumed, [5], that the ideal weights for the approximation of the continuous function *f(x)* are unknown and possibly non-unique, but they are bounded in any compact subset $\mathbb{R}^n$ according to

$$\|\Theta\| \le \Theta_m \qquad (3.4)$$

with Θ_m known.

Estimates of the nonlinear function *f(x)* can now be given by the NN as

$$\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x) \qquad (3.5)$$

with ^ denoting the actual values of the NN weights as provided by the tuning algorithm. Using a Taylor Series expansion [5] one may write the functional approximation error as

$$\tilde{f}(x) = \tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x) + \hat{W}^T \hat{\sigma}'\tilde{V}^T x + w \qquad (3.6)$$

where $\tilde{\bullet} = \bullet - \hat{\bullet}$, $\hat{\sigma}'$ is the Jacobian matrix and the higher order terms are bounded according to

$$\|w(t)\| \le C_0 + C_1\|\tilde{\Theta}\|_F + C_2\|r\|\,\|\tilde{\Theta}\|_F \qquad (3.7)$$

for computable positive constants C_0, C_1, C_2.

The NN estimate is now used to select the control law as [10]

$$\begin{aligned} u_c &= \frac{1}{g(x)}\left(-\hat{f}(x) - K_v r - Y_d + u_r\right) \\ &= \frac{1}{g(x)}\left(-\hat{W}^T\sigma(\hat{V}^T x) - K_v r - Y_d + u_r\right) \end{aligned} \qquad (3.8)$$

where $u_r(t)$ is an auxiliary robustifying term. The structure of this controller is shown in Fig. 3.1. Substituting this control law into the filtered error dynamics (2.5) yields

$$\begin{aligned} &\dot{r} = -K_v r + \tilde{f} + u_r + d \quad \text{or} \\ &\dot{r} = -K_v r + \tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x) + \hat{W}^T\hat{\sigma}'\tilde{V}^T x + u_r + d + w \end{aligned} \qquad (3.9)$$

Performing Lyapunov stability analysis and using the filtered error dynamics (3.9), adaptation formulae for the neural weights are derived in [10], which guarantees the UUB of both $|r|$ and $\|\Theta\|$, provided that some conditions are fulfilled. These formulae along with the required signals and design parameters are given in Table 1, bellow.

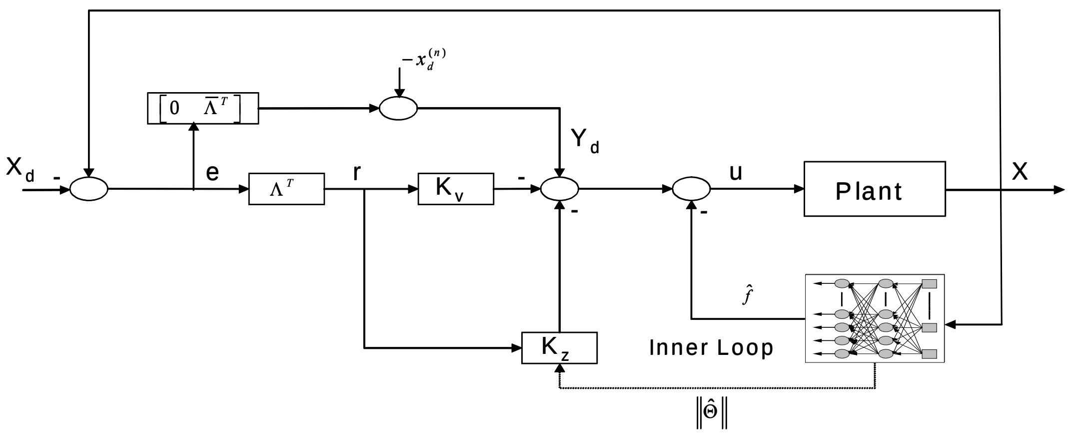


Figure 3.1. Neural network controller, where the ANN acts as estimator of *f(x)*.

Table 1. Formulae of Feedback Linearization Controller using NN estimator [10]

NN controller: According to formula (3.8)

Robustifying Term: Computed according to

$$u_r = -K_z(\|\hat{\Theta}\| + \Theta_m) \qquad (3.10)$$

NN Weight Updating: The weights are online updated according to the formulae

$$\dot{\hat{W}} = M(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x)r - k|r|M\hat{W} \quad and$$
$$\dot{\hat{V}} = Nrx\hat{W}^T\hat{\sigma}' - k|r|N\hat{V} \qquad (3.11)$$

Required Signals:

$e(t) = x(t) - x_d(t)$ Tracking error vector (3.12)

$r(t) = \Lambda^T e(t)$ Filtered tracking error (3.13)

$Y_d = -x_d^{(n)} + \begin{bmatrix}0 & \bar{\Lambda}^T\end{bmatrix} e$

Desired trajectory feedforward signal (3.14)

Design Parameters:

Gains K_v, K_z positive

Λ a coefficient vector of a Hurwitz function

Θ_m a bound on the unknown target weight norms

Tunning matrices *M, N* symmetric and positive definite

Scalar $k>0$.

In the above approach the boundedness of both $|r|$ and $\|\Theta\|$ depends on the appropriate choice of the design parameters and on the bounds of the NN reconstruction error (e_N), the bound of the disturbance term *d* and the bound *Q* of the desired trajectory. A more detailed discussion on these issues is presented in [5]. The updating of the NN weights is based on the filtered tracking error signal r(t) and require the computation of the Jacobian $\hat{\sigma}'$, which can be computed in terms of measurable signals in the closed –loop system [5]. In the next section a theoretically simpler approach is presented, which can perform equally well without requiring the strict bounds to be initially set.

4. Feedback linearizing Controller using NN compensation of function $\hat{f}(x)$

Let an approximate estimate $\hat{f}(x)$ **of** the unknown function *f(x)* be known. If this estimate is used, the linearizing control law (2.7) becomes

$$u_c = \frac{1}{g(x)}\left(-\hat{f}(x) - K_v r - Y_d\right) \qquad (4.1)$$

Which, differs from (2.7) in the presence of the known estimate $\hat{f}(x)$ and the absence of the disturbance term *d*, which in this case is considered unknown. Applying (4.1) to (2.5) the filtered tracking error dynamics become

$$\dot{r} = -K_v r + \tilde{f} + d \qquad (4.2)$$

where $\tilde{f}(x)$ is the unknown functional estimation error and *d* is the unknown disturbance term. The presence of $\tilde{f}$ and *d* affects the stability of (4.2) in a way that it cannot be easily compensated by appropriate choice of the positive constant K_v. Therefore the desired tracking performance cannot be achieved. Next, a NN scheme is proposed to compensate for the uncertainty about *f(x)* and the presence of *d*.

Instead of using the NN as an estimator of the unknown function *f(x)*, it is rather introduced in the linearizing control law as follows

$$u_c = \frac{1}{g(x)}\left(-\hat{f}(x) + \Phi_c(x) - K_v r - Y_d\right) \qquad (4.3)$$

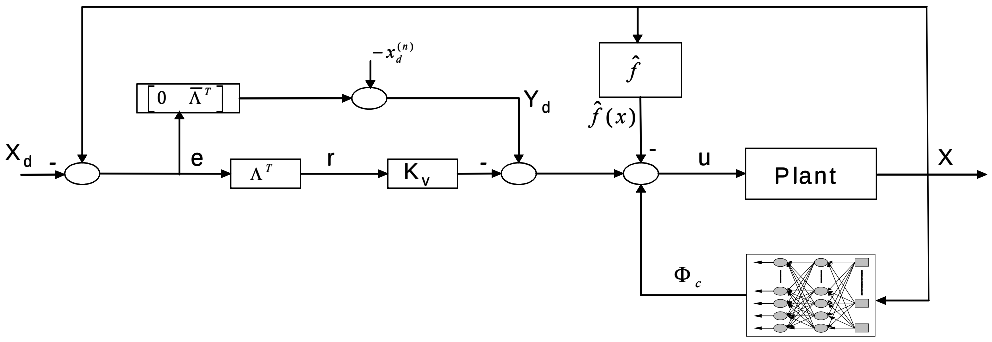


Figure 4.1. Neural network controller, where the ANN acts as compensator to the effects of $\hat{f}(x)$.

where $\Phi_c(x)$ is the output of the NN compensator, which is in general a function of the state x. The structure of the proposed controller is shown in Fig. 4.1. In this approach there is no restriction about the number of the hidden layers of the neural network.
Applying (4.3) to (2.5) the filtered tracking error dynamics become

$$\dot{r} = -K_v r + \tilde{f} + d - \Phi_c \tag{4.4}$$

In the ideal case, when the compensation is complete, equation (4.4) becomes

$$v = \dot{r} + K_v r = \tilde{f} + d - \Phi_c = 0 \tag{4.5}$$

Therefore, the objective of the NN compensator is to learn $\tilde{f} + d$, or equivalently to lead the signal v to zero (Fig. 4.2). In the proposed scheme, since $\tilde{f} + d$ is unknown, we use the signal v as the output error signal for training the Neural Network. The minimization of the new error signal v will result in the minimization of the filtered tracking error signal and in the steady state $r \to 0$. Therefore according to section 2, $e \to 0$ as well and the objectives of the control are fulfilled.

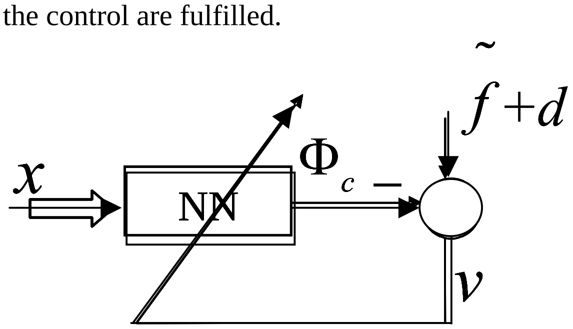


Figure 4.2. Learning signal of the neural network

Training of the Neural Network

The weight updating law minimizes the objective function I, which is a quadratic function of the training signal v

$$I = \frac{1}{2} v^2 \tag{4.6}$$

Differentiating (4.6) and using (4.5) yields the gradient of I in respect to the NN weights w as follows

$$\frac{\partial I}{\partial w} = \frac{\partial v}{\partial w} v = -\frac{\partial \Phi_c}{\partial w} v \tag{4.7}$$

The weight updating of the ANN is performed on-line by using the conventional back-propagation-updating rule with a momentum term [11] as follows

$$\Delta w(t) = \eta \frac{\partial v}{\partial w} v + \alpha \Delta w(t-1) \tag{4.8}$$

Where η is the update rate and α is the momentum coefficient. The proposed NN controller along with the required signals and design parameters are given in Table 2, bellow.

Table 2. Formulae of Feedback Linearization Controller using NN compensator

NN controller:

$$u_c = \frac{1}{g(x)}\left(-\hat{f}(x) + \Phi_c(x) - K_v r - Y_d\right) \tag{4.9}$$

NN Weight Updating: The weights are online updated using the conventional backpropagation scheme according to the formulae

$$\frac{\partial I}{\partial w} = \frac{\partial v}{\partial w} v = -\frac{\partial \Phi_c}{\partial w} v \tag{4.10}$$

$$\Delta w(t) = \eta \frac{\partial v}{\partial w} v + \alpha \Delta w(t-1) \qquad (4.11)$$

Required Signals:

$e(t) = x(t) - x_d(t)$ Tracking error vector (4.12)

$r(t) = \Lambda^T e(t)$ Filtered tracking error (4.13)

$Y_d = -x_d^{(n)} + \begin{bmatrix} 0 & \overline{\Lambda}^T \end{bmatrix} e$

Desired trajectory feedforward signal (4.14)

$v = \dot{r}(t) + K_v r(t)$ NN error training signal (4.15)

Design Parameters:

Gain K_v positive

Λ a coefficient vector of a Hurwitz function

Scalars $0 < \eta, \alpha < 1$. well known quantities from the backpropagation algorithm

In the proposed scheme the initial NN weights are set to zero. Then, according to Fig. 4.1 the controller consists of a PD tracking loop, which holds the system bounded stable until the NN begins to learn. After that point the tracking performance will continuously improve.

5. Simulation examples of the proposed method

In this section we perform simulation experiments on a well known simple nonlinear system (Van der Pol's oscilator). We also report on the application of a simplified version of the proposed controller to control a fermentation process [12].

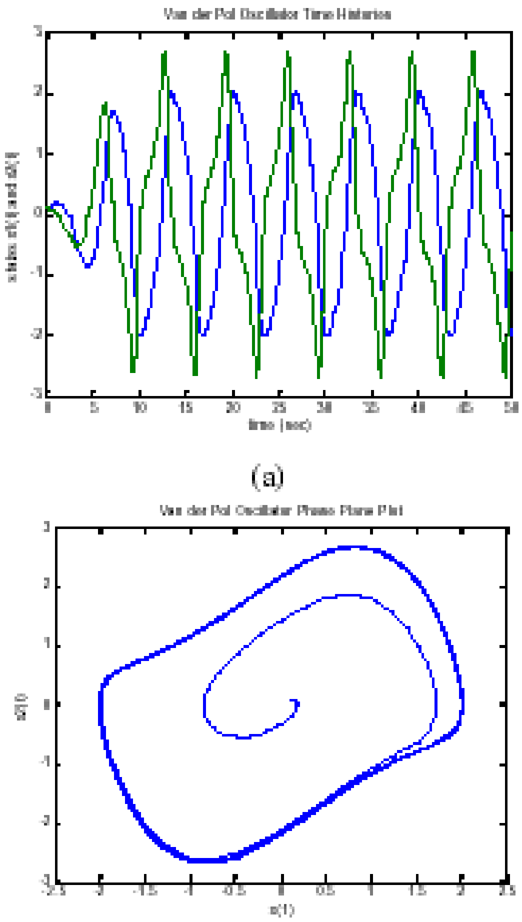
Example 1.

Let the Van der Pol's system described by the following set of nonlinear equations

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= (1 - x_1^2) x_2 - x_1 + u \end{aligned} \qquad (5.1)$$

which is already in the Brunovsky canonical form, where $g(x) = 1$, known, and $f(x) = (1 - x_1^2) x_2 - x_1$, possibly erroneously known. Note that the system has an unstable equilibrium point at the origin x= (0,0) and a stable limit cycle. An open loop trajectory ($u = 0$) for the system is illustrated in Fig. 5.1(a) and the phase plane plot of the states x_1, x_2 is shown in Fig. 5.1(b). The initial state values were x_0 = (0.1, 0.1).

The desired trajectory for the closed loop system is defined as $y_d(t) = x_{1d}(t) = \sin t$. Design parameters are set to K_v = 5, $\Lambda = \begin{bmatrix} 5 & 1 \end{bmatrix}$ and the initial state values are now set to x_0 = (1, 1).

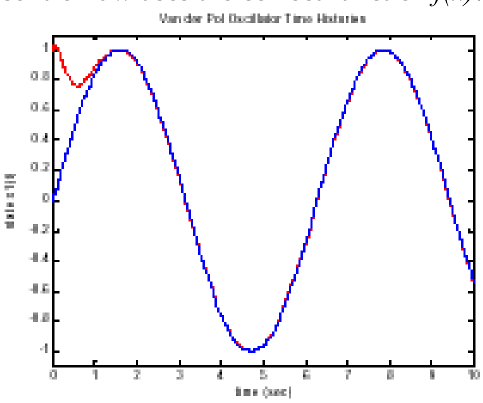


(b)

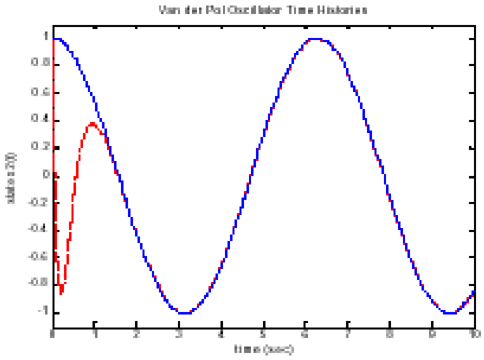
Figure 5.1. Time history and phase plane plots of the states x_1, x_2 of the open loop Van der Pol's oscillator.

(5.1)

Figures 5.2(a) and (b) illustrate the desired and actual state trajectories of the closed loop system, when the function *f(x)* is considered to be equal to the one appearing in (5.1) and the control law uses the correct function *f(x)*.



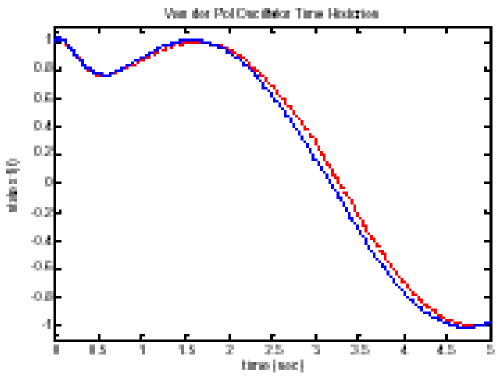
(a)



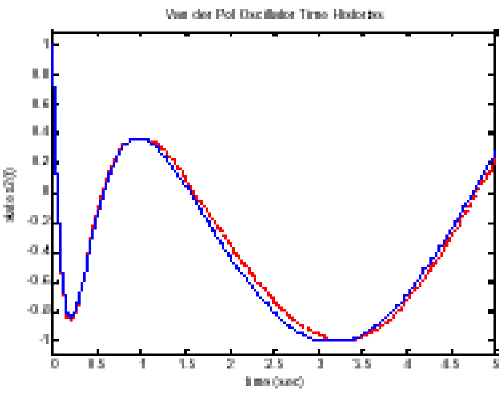
(b)

Figure 5.2. Time histories of the desired (blue solid) and the actual (red dashed) states x_1, x_2 of the closed loop Van der Pol's oscillator, when an accurate $f(x)$ is used.

To display the effect of the uncertainties in the operation of the plant we ran again the system equations (5.1), but in the control law a spurious version, $\hat{f}$, of function f was used. Actually, $\hat{f}$ was selected to differ about 10% (in magnitude) from the actual function f. Figures 5.3 (a) and (b) show the plots of the nominal (with the actual function f) (blue – solid)) and the degraded (with the spurious function $\hat{f}$) (red-dashed)) state values for the two states.



(a)

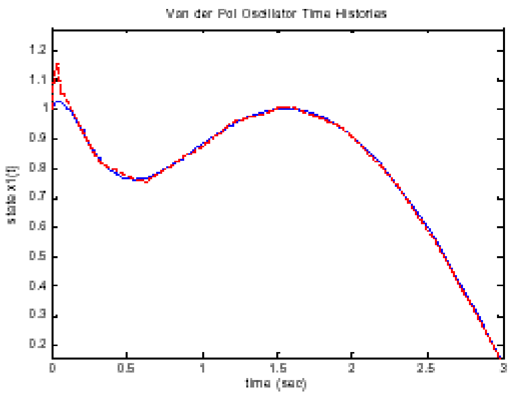


(b)

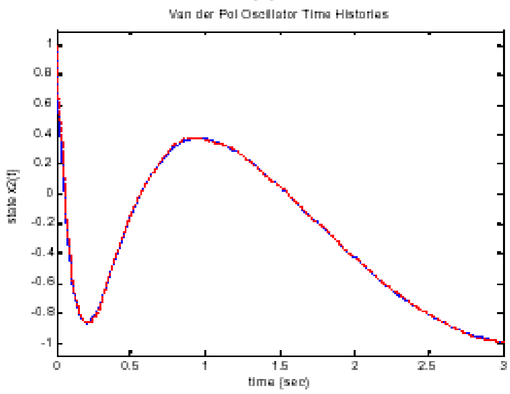
Figure 5.3. Nominal (blue solid) and degraded (red dashed) trajectories of the states x_1, x_2.

It is apparent that in the presence of uncertainties the control objectives are not reached.

The NN compensator, proposed in section 4, is in the sequel used to eliminate the effect of the uncertainties. To simulate the ANN compensator, a (3-8-1) multilayer perceptron is used. The number of inputs is 3 (states x_1, x_2 and the desired output y_d). The activation function of the hidden units was a Sigmoid, while the activation function of the last (output) unit was linear. The design parameters of the updating rule were η=0.01 and α = 0.9. The NN compensated control law uses the spurious version, $\hat{f}$, of function f., where $\hat{f}$ was selected to differ about 10% (in magnitude) from the actual function f. The initial weights of the last (linear) layer of the network were selected to be 0. Figure 5.4 shows the nominal (when the exact function f is used) and the NN compensated values of the states x_1, x_2, when $K_v = 5$. It is apparent that the NN compensated control law makes the states to approach to the nominal values after a number of samples.



(a)



(b)

Figure 5.4. Nominal (blue solid) and NN Compensated (red dashed) trajectories of the states x_1, x_2.

This number is required in order the NN to learn its goal (The NN is trained on-line according to section 4).

Example 2.

A simplified version of the proposed method is also used for the control of a fermentation process, in presence of uncertainty [12]. The process under study is the production of Saccharomyces Cerevisiae (SC- baker's yeast), which is produced from the fermentation of glucose. Due to the presence of bacteria, specific conditions, such as temperature, pH, and concentrations of the different substances, are required [13, 14]. The fermentation process takes lieu in a reactor whose input and output are glucose and SC, respectively. Measured values of different quantities (ethanol, glucose, acetate) can also be considered as outputs. Three physiological modes are possible [15]:

- glucose fermentation produces SC, ethanol and acetate (mode 1)
- glucose respiration produces only SC (mode 2)
- ethanol and acetate respiration produce SC (mode 3)

Transition from one mode to another is made according to activation and/or inhibition laws and depends on the milieu's composition. Since there is no limitation from the oxygen, the whole process works in an aerobic way. Thus, the corresponding kinetic expressions are used to describe growth rates of the different substances. On the other hand, the matter balance laws allow the determination of their concentrations at the different modes. These concentrations are the state variables of the process:

X_1 : SC concentration in mode 1

X_2: SC concentration in mode 2

X_3: SC concentration in mode 3

E: ethanol concentration

S: glucose concentration

A: acetate concentration

Denoting as X the total concentration of SC one has

$$X = X_1 + X_2 + X_3 \qquad (5.2.a)$$

$$\dot{X} = R_1 + R_2 + R_3 - DX \qquad (5.2.b)$$

where R_i, i=1,2,3 are the growth rates of SC in modes 1,2,3, respectively, and D is the dilution rate of glucose at the system input.

The following state-space model has been developed in [16] and is used in this paper.

$$\dot{X}_1 = R_1 + R_5 + R_9 - R_4 - R_8 - DX_1 \qquad (5.3.a)$$

$$\dot{X}_2 = R_2 + R_4 + R_7 - R_5 - R_6 - DX_2 \qquad (5.3.b)$$

$$\dot{X}_3 = R_3 + R_8 + R_6 - R_9 - R_7 - DX_3 \qquad (5.3.c)$$

$$\dot{E} = \frac{45}{14}R_1 - \frac{20}{11}R_3 - DE \qquad (5.3.d)$$

$$\dot{S} = -\frac{50}{7}R_1 - 2R_2 + D(S_{in} - S) \qquad (5.3.e)$$

$$\dot{A} = \frac{1}{14}R_1 - DA \qquad (5.3.f)$$

where S_{in} is the glucose concentration at the input and $R_1, R_2 \dots, R_9$ are the growth rates at the transitions from one physiological mode to another. They are in general nonlinear functions of the states of the system. Analytic expressions for these functions can be found in [17].

The control objective is to regulate the glucose concentration at the input in order to optimize, in some sense, the production of SC with respect to the glucose consumption. It has been found that this is achieved whenever the glucose concentration takes the value $S \equiv S_c = 0.071 g/l$. Thus, a set-point regulation problem has to be solved. Since a nonlinear model describes the process, a possible control law consists in using pole placement for a linearizing control law [16]. Obviously, the only output to be regulated is the glucose concentration S. Thus, the linearizing control law is given by means of the following expression:

$$D = -\frac{1}{S_{in} - S}\left(-\frac{50}{7}R_1 - 2R_2 + \beta_s(S - S_c)\right) \qquad (5.4)$$

where β_s is a positive gain constant specified by the designer.

The input should always be positive. In practice, one starts applying the control action at the instant t_0 when $E < 0.05 g/l$ and $S < 0.05 g/l$. Then, $S(t_0) < S_c$ and hence $D > 0$. From (4.e) one has $\dot{S} = -\beta_s(S - S_c) >$ 0, and thus S exponentially tends to S_c from inferior values.

Equation (5.4) is of the form of equation (1.4), with $g(x) = S_{in} - S$, $f(x) = -\frac{50}{7}R_1 - 2R_2$ and *d=0*.. In this description, *g(x)* can be easily considered known, but *f(x)* may carry uncertainties related to incomplete knowledge of the model parameters. A simplified version of the proposed NN compensator controller, where $\Lambda = 1$ is given bellow.

$$D = -\frac{1}{S_{in} - S}\ (\hat{f}(x) + \Phi_c + \beta_s (S - S_c)) \qquad (5.5)$$

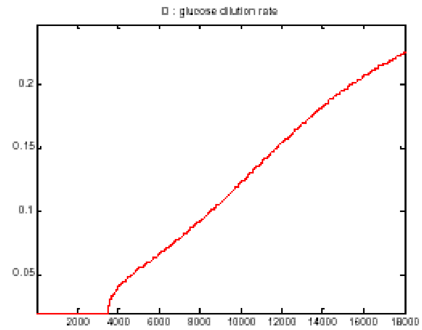
Applying the linearizing control law, (5.5), in (5.3.e), defining the tracking error $e = S - S_c$, and taking into account that S_c is constant one can deduce the following first order dynamic equation of the tracking error e.

$$\dot{e} = \tilde{f} - \Phi_c - \beta_s e \qquad (5.6)$$

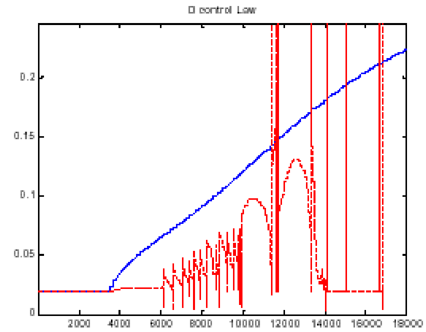
In this case, the NN output is used to cancel the effect of the uncertainty $\tilde{f}$ about function f and the NN training error is defined as

$$v = \dot{e} + \beta_s e \qquad (5.7)$$

The nominal values of the six states and the input (D) were produced by the model equations (5.3a – 5.3f) and the control law (5.5). The sampling period was considered to be T = 0.0009 h. The initial values of the state variables were considered according to [16, 18] to be [X1(0), X2(0), X3(0), E(0), S(0), A(0)] = [14.7, 0, 0, 1.8, 2.5, 0] g/l, and Sin=300 g/l. The input D was computed by eq. (5.3) with $\beta_s = 1$, but D >= D_{min} = 0.02 ℓ / h. It should be mentioned that β_s corresponds to the design parameter K_v in section 4. The control law is for the first time applied when $E < 0.05 g/l$ and $S < 0.05 g/l$. Figure 5.5.(a) shows the plot of the control law D. The plot contain 18.000 samples of the control variable, which correspond to 16.2 hours of continuous operation of the plant. It can be seen that the control law is actually applied after the 3498 sample which corresponds to about 3.15 hours of operation. Figure 5.5(b) illustrates the nominal (with the actual function *f*) and the false (with the spurious function $\hat{f}$) values of the control law D for the case where $\beta_s = 1$. It is apparent that the stability of the process is deteriorated. This condition can be improved by increasing the stabilizing constant β_s, but the process will become finally unstable.



(a)



(b)

Figure 5.5. (a). Nominal values of the control law D for 16.2 hours of continuous operation (b) Nominal and degraded values of control law variable D

Finally, in order to test the proposed method, we used a 4-8-1 multilayer perceptron to simulate the ANN compensator. The number of inputs is 4 (states E, S, A and the desired glucose concentration Sc). The activation function of the 8 hidden units was a Sigmoid, while the activation function of the last (output) unit was linear. The parameters of the updating rule were η=0.01 and α = 0.9. The initial weights of the last (linear) layer of the network were selected to be 0. Figure 5.6. shows the nominal and the NN compensated values of the control law, D, in case where $\beta_s = 1$. It is apparent that the NN compensated control law approaches the nominal values after a number of samples. This number is required in order the NN to learn its goal(The NN is trained on-line). Figure 5.7 shows a plot of v^2 (the squared error signal v), where the progress of the training of the NN compensator is apparent.

It should be mentioned here, that the states X_1, X_2, X_3, are not available for measurement and, therefore, they are not used as inputs to the ANN. Nevertheless, the ANN emulates the functional error $\tilde{f}(x)$ quite satisfactory.

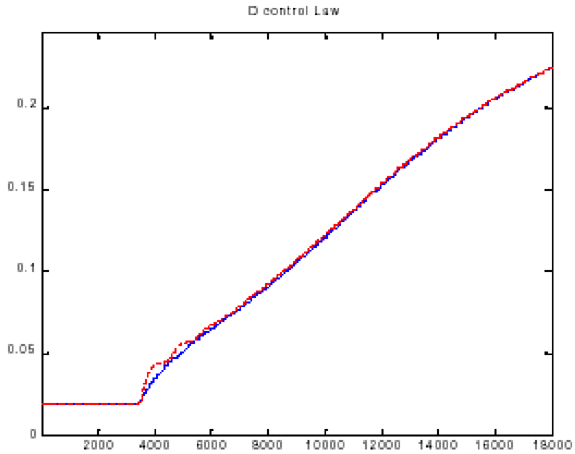


Figure 5.6. Nominal and NN compensated values of control law variable D. The stabilizing constant is $\beta_s = 1$

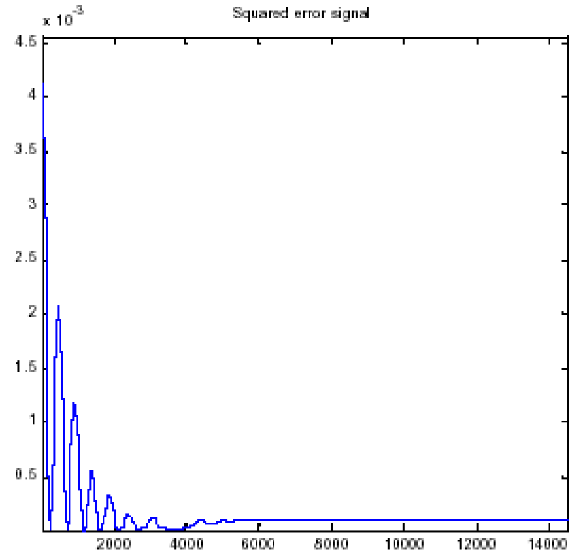


Figure 5.7. Squared error signal (v^2). The signal is recorder just after the NN compensated control law is applied

6. Conclusions

Two alternative approaches of feedback linearization of an uncertain system using Artificial Neural Networks are presented in this paper. The first approach, reported from the recent bibliography uses the ANN as an estimator of the unknown or erroneously known function *f(x)* appearing in the linearizing control law. In the second approach the author's proposition is presented to treat the same problem in a conceptually simpler manner. In this approach the ANN is not used as an estimator of the unknown function but as a compensator to the effects of the incomplete knowledge $\hat{f}$ about *f*. It is shown that in this case the ANN is actually learning the functional error $\tilde{f}$ and the disturbance term. The ANN is trained on-line using the conventional backpropagation, without any a-priori assumptions about the bounds of its weights. The error to be minimized by the training algorithm is chosen so that the stability of the tracking error system is ensured. Future work might include a complete investigation on the robustness of the proposed approach when the system states are not all available and its extension to tackle the more general case where both functions *g(x)* and *f(x)* of the control law are erroneously known.

---

7. References

[1].G.O. Guardabassi and S.M. Savaresi, "Approximate Linearization via Feedback – an Overview", Automatica, vol. 37, pp. 1-15, 2001.

[2].A.Isidori, "Nonlinear Control Systems", Springer-Verlag, 1995.

[3].J. Slotine and W. Li, "Applied Nonlinear Control", Prentice-Hall, New Jersey, 1991

[4].T. Zhang, S. S. Ge and C.C. Hang, "Adaptive output feedback control for general nonlinear systems using multilayer neural networks", Proc. Amer. Control Conf., pp. 520-524, Philadelphia, 1998.

[5].F. L. Lewis, S. Jagannathan and A. Yesildirek, Neural Network Control of Robot Manipulators and Nonlinear Systems, Taylor & Francis, 1999.

[6].J. Hauser, S. Sastry and P. Kokotovic, "Nonlinear control via approximate input-output linearization", *IEEE Trans. Automat. Control,* vol 37, no. 3, pp. 392-398, 1992.

[7].K. S. Narendra and K. Parthasarathy, Identification and Control of Dynamical [8].Systems Using Neural Networks, IEEE Trans. on Neural Networks, vol. 1, no. 1, pp 4-27, 1990.

[8].A. U. Levin, and K. S. Naredra, Control of nonlinear dynamical systems using neural networks: controlability and stabilization, *IEEE Trans. Neural Networks*, **4**, 192 – 206, 1993

[9].J. D. Boskovic and K. S. Narendra, Comparison of Linear, Nonlinear and Neural Network based Adaptive Controllers for a Class of Fed-batch Fermentation Process, *Automatica*. **31** (6), 817-840, 1995

[10].A. Yesildirek and F. L. Lewis, Feedback Linearization using Neural Networks, *Automatica*, **31** (11), pp. 1659-1664, 1995.

[11].D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Internal Representations by Error Propagation", in Parallel Distributed Processing, D.E. Rumelhart and J.L. McLelland, Eds., Cambridge, MA: MIT Press, 1986.

[12].Y. S. Boutalis and O. I. Kosmidou, "A Feedback Linearization Technique by Using Neural Networks: Application to Bioprocess Control", Proceedings of the International Conference of Computational Methods in Sciences and Engineering – ICCMSE 2003, pp. 90-92, Kastoria – Greece, 12-16 September 2003

[13].G. Bastin and G. Dochain, "On line Estimation and Adaptiva Control of Bioreactors", Elsevier, 1990.

[14].G. Bastin and J.F. Van Impre, "Nonlinear and Adaptive Control in Biotechnology: A Tutorial", European Journal of Control, vol. 1, 1995, pp. 37-53.

[15].A. Rajab, J.-M. Engasser, P. Germain and A. Miclo, "A Physiological Model of Yeast Aerobic Growth", 3rd European Congress on Biotechnology, München, Germany, 1984.

[16].M.Pengov, "Applications des observateurs non linéaires à la commande des bioprocédés", Thèse de Docteur en Sciences, Université de Metz, France, 1998

[17].Boutalis Y., S., and Kosmidou O. I., Using neural network observers for bioprocess control, *Control Engineering and Applied Informatics*, **3**, pp. 51-58, 2001.

[18].F. Bicking, "Définition et mise au point d'un nouvel algorithme de type génétique: application a la conduite d'un bioprocédé semi-continu", Thèse INPL, Nancy, France, 1991.