Iranian Journal of Chemical Engineering
Vol. 3, No. 1 (Winter), 2006, IAChE

Application of Affine Gray-Box Neural Models for Nonlinear Control of Chemical Processes

A. Bazaei and V. Johari Majd[*]
Electrical Engineering Department, Tarbiat Modarres University, Tehran, Iran

Abstract

In this paper, an affine neural model is used to model the unknown part of SISO processes with un-modeled actuator dynamics. It is assumed that a partially known first-principles based model of the process, which is invertible with respect to the unknown part, is available. Using this available knowledge, I/O training data of the process, and affine neural networks, a serial gray-box model is generated which is suitable for applying feedback linearization. Hence, the resulting nonlinear controller works in a large operating region. The superiority of the gray-box over the black-box approach is investigated for a fermentor using the experimental data borrowed from the literature. Simulation results of our case study show that the proposed affine gray-box method is superior to the conventional affine black-box method and preserves extrapolation property.

Keywords: *Feedback Linearization, Neural Modeling, Gray-Box, Affine Modeling, Non-linear Control*

Introduction

Achievements in the field of differential geometry have led to the emergence of interesting control techniques in the nonlinear system theory including feedback linearization (FL) [1, 2]. Henson and Kravaris have addressed the application of FL to chemical processes in continuous-time domain [3-6]. Control schemes based on FL provide a larger dynamic operation range than the conventional Jacobian linearization method about an operating point. Furthermore, the benefits of linear control techniques can be utilized via FL.

However, when major parametric or structural uncertainties are present in the system, the states of the plant could be neither accurately measured nor estimated, or when some necessary controllability and involutivity conditions are not satisfied, an exact FL is not possible.

Neural networks can approximate any continuous function to an arbitrary accuracy [7]; hence they are suitable tools for modeling nonlinear uncertain plants provided that appropriate input/output data is available [8]. Model-based FL method using neural networks has received attention since it has the ability of handling plant uncertainties just by input-output data. FL has also been successfully applied to continuous-time recurrent neural networks [9]. In another work, a model- based FL in discrete-time via neural networks has been developed [10]. In all of

[*] - E-mail: majd@modares.ac.ir

the FL schemes mentioned above, the neural network is used as a black-box model and no part of the plant dynamic equations is assumed known. Hence, the black-box neural network model may have poor extrapolation properties and its validity may remain within the range of the training data.

The models totally generated by the first principles knowledge (*a priori* information based on general physical rules) governing the plants are called *white-box* models [1]. Although these models have good extrapolation properties, they are hard to find since they require reliable knowledge about the plant. Moreover, converting the available plant information into standard dynamic equations may be difficult or impossible. Hence, generating white-box models (when possible) is expensive in general.

It may be possible to generate models based on a combination of first principles knowledge of the plant and neural or fuzzy models, resulting in what is called *first-principles-based gray-box*, *hybrid*, or *semi-physical models* [11-15]. These models include black-box elements as parts of the white-box model derived from the first principles knowledge and can be classified into parallel and serial types. In a parallel gray-box model, the neural network is placed parallel to the first principles model [16-17]. This type may have better interpolation properties in comparison with the black-box one. In serial gray-box models, the neural network is trained to model only the unknown or uncertain part of the plant dynamics [11-14]. This type yields better extrapolation properties than the black-box type while requiring a lower amount of training data and training time. Hence, using serial gray-box models provides an efficient approach, especially when data acquisition is a serious bottleneck, as is the case for chemical and biochemical processes. Moreover, using hybrid models prevents large oscillations in the predicted output.

In [15], an input/output linearizing control law has been applied to a non-neural and non-fuzzy first-principles-based gray-box model of an industrial-scale batch polymerizetion reactor in continuous time framework; however, the gray-box model is only valid for the desired output profile that has been used during the training. Furthermore, to obtain the training data, it is necessary that a suitable controller (PI controller in that case) be available to generate the desired output profile.

Another scheme for incorporating *a priori* knowledge in neural/fuzzy models and producing gray-box models has been considered in [18-20]. In this approach, some qualitative *a priori* information such as the smoothness of system behavior, open-loop stability, steady state gain, settling time, and nonlinearity are used as constraints on the black-box model parameters in the form of linear inequalities. In [21], an internal model control (IMC) scheme has been applied to the gray-box fuzzy model of a pH process. The inverse model used in this approach (Equation (21) in [21]) does not have straightforward mapping and needs computation and decision for all simplices in order to determine the inverse model output. Evidently, for a large number of simplices the computational overhead will be high. Furthermore, the type of *a priori* knowledge that can be incorporated in the gray-box model is limited to that which can be represented or transformed into linear inequalities on the parameters of the fuzzy model. Also, if some known or unknown parameters of the plant change after training, the trained gray-box fuzzy model will become invalid because it is not possible to incorporate the known parameters of the plant in the fuzzy model such that their variations can be directly reflected into the model. (Notice that this type of gray-box model is not based on the first principles knowledge of the plant). In that work, if the number of rules in the fuzzy model is not large enough, the gray-box model may exhibit poor performance compared to a black-box model with the same number of rules. (See table II in [2]).

In this paper, we apply FL techniques to the first-principles-based serial gray-box model of an uncertain biochemical plant via affine neural networks; hence, we take the advantages of the FL technique, first-principles knowledge, and serial gray-box neural modeling to improve the control performance. We will also have a comparative discussion about the model-based FL control schemes via black-box and serial gray-box affine neural network models in *one-step-ahead* (OSA) and *free-run* (FR) modes. Furthermore, we will compare the simulation results of the conventional and the proposed methods for a fermentation process.

The Theory

Consider an n-th order nonlinear uncertain SISO process, described by the following discrete-time dynamic equation:

$$\begin{aligned} y_{k+1} &= f(y_k, y_{k-1}, \ldots, y_{k-n+1}, u_k, u_{k-1}, \ldots, u_{k-m}) \\ &= f(X_k, u_k) \qquad (1) \end{aligned}$$

where $X_k = [y_k, y_{k-1}, \ldots, y_{k-n+1}, u_{k-1}, \ldots, u_{k-m}]^T$,y_k and u_k are the output and the input (actuator) of the plant respectively, and f is an unknown function because of the un-modeled actuator dynamics. We want to develop a serial gray-box neural model for the plant using input/output data. The gray-box model will be used to control the plant via FL schemes in different modes.

Assume that, using the available first principles physical knowledge about the plant, the right hand side of Equation (1) could be described as:

$$y_{k+1} = f(X_k, u_k) = g(X, h, \beta)\big|_{X=X_k, u=u_k} \qquad (2)$$

where $g(.,.,.)$ is a known function of its explicit arguments, $\beta \in \Re^p$ is a vector containing known parameters, and h is an unknown scalar valued function of X_k and u_k with an unknown structure which does not depend on the parameter vector β. Thus, we have:

$$h = h(X, u)\big|_{X=X_k, u=u_k} \qquad (3)$$

We state that the gray-box structure is *invertible* if the unknown function h can be extracted from Equation (2) such that we can express it in the form of:

$$h(X_k, u_k) = g_h^{-1}(y_{k+1}, X_k, \beta) \qquad (4)$$

where $g_h^{-1}(.,.,.)$ is a known function given that y_{k+1} is known. For an invertible gray-box model, if some experimental I/O plant data are available, one can use Equation (4) to obtain the I/O data that corresponds to the unknown scalar function h. This data can be used for training a neural network to model the unknown function $h(X_k, u_k)$. We describe this trained neural network by the following equation:

$$\hat{h} = \hat{h}(X, u)\big|_{X=X_k, u=u_k} \qquad (5)$$

In this manner, we obtain a gray-box model of the plant in one-step-ahead (OSA) mode that can be described by the following equation:

$$\hat{y}_{k+1} = \hat{f}(X_k, u_k) = g[X_k, \hat{h}(X_k, u_k), \beta] \qquad (6)$$

By OSA we mean that the true plant output is used for evaluating the right hand side Equation (6).

Free-Run (FR) mode: If the measurement of the plant output is not available, we use the previous values of the estimated output of the gray-box model. In this case, the vector:

$$\hat{X}_k = [\hat{y}_k, \hat{y}_{k-1}, \ldots, \hat{y}_{k-n+1}, u_{k-1}, \ldots, u_{k-m}]^T \qquad (7)$$

is used in place of X_k in Equations (6).

Affine Modeling of the Unknown Part

In affine gray-box modeling, the unknown function $h(X,u)$ is approximated by two neural functions, and is described by the following equation:

$$\hat{h} = \hat{h}(X_k, u_k) = N_1(X_k) + u_k \cdot N_2(X_k) \qquad (8)$$

where functions $N_1(X_k)$ and $N_2(X_k)$ represent the outputs of two separate neural networks. Notice that the control input u_k appears linearly in the overall model output $\hat{h}$. Evidently, training of this model requires simultaneous learning of the two neural networks $N_1(X_k)$ and $N_2(X_k)$ using a cost function that considers $\hat{h}$ as the overall output. In this manner, the affine gray-box model is described by Equations (6) and (8) in OSA mode. For invertible structures the input-output training data for the overall network is arranged in the form of $[X_k^T, u_k; h(X_k, u_k)]$, where the desired output data $h(X_k,u_k)$ is obtained from Equation (4). Figure 1 shows the block diagram for training the affine neural model $\hat{h}$.

Feedback Linearization (FL)

Recalling that $g(\cdot,\cdot,\cdot)$ is a known function and h can be extracted in the form of Equation (4) from Equation (2), u_k can be extracted from Equations(4),(6) and (8) as below:

$$u_k = [g_h^{-1}(\hat{y}_{k+1}, X_k, \beta) - N_1(X_k)] / N_2(X_k) \qquad (9)$$

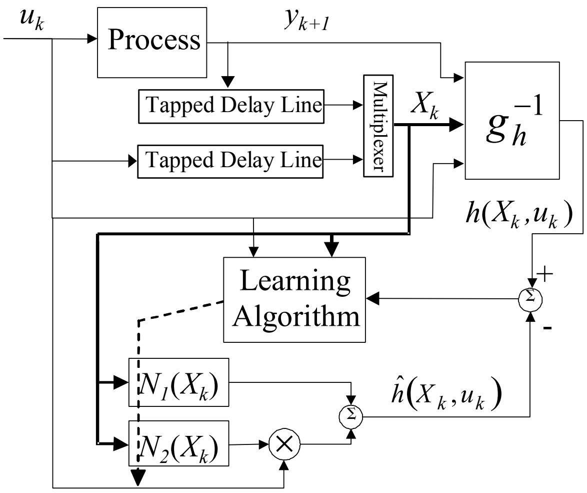

Figure 1. Training Diagram for the Affine Neural Model

By selecting an appropriate stable dynamic equation such as the following for the output error:

$$e_{k+1} + \alpha_l e_k + \alpha_{l-1} e_{k-1} + \cdots + \alpha_0 e_{k-l} = 0 \qquad (10)$$

where $e_k = \hat{y}_k - r_k$ and scalar parameters $\alpha_0, \cdots, \alpha_l$ are selected such that the following characteristic equation has only stable roots:

$$z^{l+1} + \alpha_l z^l + \cdots + \alpha_1 z + \alpha_0 = 0 \qquad (11)$$

the following control law is obtained in OSA mode:

$$u_k = \frac{g_h^{-1}\left(r_{k+1} - \sum_{i=0}^{l} \alpha_{l-i} e_{k-i}, X_k, \beta \right) - N_1(X_k)}{N_2(X_k)}$$

(12)

In FR mode $\hat{X}_k$ must be used in place of X_k in above control law.

Case Study

To further explore the results of different methods in previous sections and to investigate their advantages in comparison with the black-box modeling, we focus on a biochemical process as a case study. The plant that we would like to control is a fermentor described in [1] by Van Can et. al. At the bottom of the vessel, air is blown with input flow rate u_2 and we assume that it is kept equal to a known constant. The position of a pneumatic valve in the outlet gas pipe (u_1), expressed by the percentage that it is closed, controls the pressure (y) in the vessel. We denote u_1 as the control input and y as the output of the plant to be controlled. The plant dynamics in the discrete time domain can be described as [1]:

$$\mathrm{y}(k+1) = g\big(y, K(u,y), \beta\big)\Big|_{\substack{y=y(k)\\ u=u_1(k)}}$$

$$= \mathrm{y}(k) + \beta \cdot \left\{ \mathrm{u}_2(k) - \sqrt{\mathrm{K}(\mathrm{u}_1(k), \mathrm{y}(k)) \cdot \ln\left[\frac{\mathrm{y}(k)}{\mathrm{P}_0}\right]} \right\} \qquad (13)$$

where $\beta = (39.7\Delta t \cdot R \cdot T)/V_h$ and

y	pressure (Pa)
R	gas constant = 8.31434 (J/mol/°K)
T	temperature = 333 (K)
V_h	head space of the vessel = 0.015 (m^3)
u_1	valve position (% closed)
u_2	incoming air flow = 3.75 × 10^{-4} (m^3/s)
P_0	atmospheric pressure = 1.013 × 10^5 (Pa)
K	lumped friction coefficient of the outlet (m^6/s^2)
Δt	sample interval = 5 (s)

The lumped friction coefficient of the outlet (K), referred to as friction parameter, is an unknown nonlinear function of u_1 and y; hence, we have a nonlinear and non-affine plant with structural uncertainty as shown in Equation (13). Figure 2 shows a test control signal (valve position) along with the plant output (pressure), which will be used for training purposes.

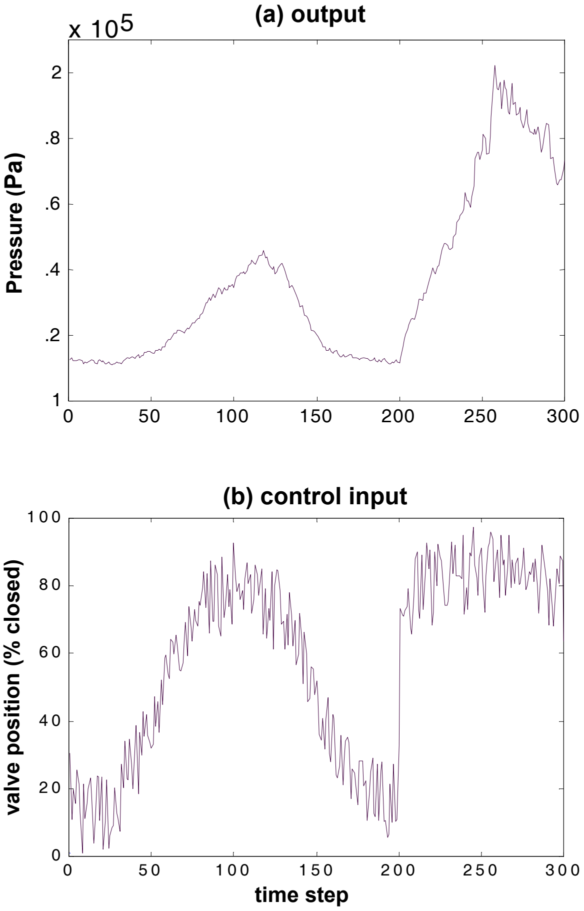

Figure 2. Simulation results of the process

Affine Gray-box Modeling and Control

Previous works such as [10] show successful application of exact/approximate FL approaches to black-box neural models of the plant. Here, we would like to extend FL to gray-box neural models of the plant via affine neural networks, hoping that the resulting control schemes inherit the benefits of serial gray-box models mentioned in Section 1.

In the black-box modeling schemes, we did not utilize our partial knowledge about the dynamical equations governing the plant for selecting the structure of the model even though the unknown part of the plant in this case is only the friction function.

In gray-box modeling, introduced in [1] , the structure of the model is selected such that the neural network models only the unknown part of the plant equations. Since this model utilizes the known parts of the plant dynamics, derived from physical laws governing the plant, it is referred to as a first-principles-based gray-box model.

It is obvious that the gray-box model is superior to the black-box one when it can be generated. Moreover, since the identification effort is only on the unknown part of the plant, the accuracy of the trained gray-box model will increase, while the required training time will decrease in comparison with the black-box model. Furthermore, the validity domain of a gray-box model is more than that of a black-box one, because a gray-box model provides a greater degree of freedom for the unknown function in the plant, constructed by the neural network. These functions are not explicitly trained in a black-box model, and if they undergo some variations, the trained black-box model may produce larger errors. Moreover, if some measurable parameters in the known part of the plant change after the training, the gray-box model is still valid using the new values of the measurable parameters, while the black-box model requires new training to incorporate the resulting changes in the overall input/output map.

Since the fermentor model is invertible with respect to friction, we can extract the data required for modeling the friction function by rewriting Equation (13) as:

$$\mathrm{K}(\mathrm{u}_1(k),\mathrm{y}(k)) = \frac{\left\{\mathrm{u}_2(k) - \left[\dfrac{\mathrm{y}(k+1)-\mathrm{y}(k)}{\beta}\right]\right\}^2}{\ln\left[\dfrac{\mathrm{y}(k)}{\mathrm{P}_0}\right]} \qquad (14)$$

Using the I/O data of Figure 2 we can extract the output data of function $K(u_1,y)$ from the above equation. We can now train suitable neural models to approximate the friction function. We develop and train an affine gray-box model, and then we apply FL techniques to it. The affine model has two inputs ($u_1(k)$, $y(k)$) and one output $\hat{K}(k)$. The following equation shows the general structure of the gray-box model in OSA mode:

$$\begin{aligned}\hat{\mathrm{y}}(\mathrm{k}+1) &= \mathrm{F}(\mathrm{u}_1(\mathrm{k}),\mathrm{y}(\mathrm{k})) \\ &= y(k) + \beta\cdot\left\{\mathrm{u}_2(k) - \sqrt{\left|\hat{\mathrm{K}}(\mathrm{u}_1(k),\mathrm{y}(k))\right|\cdot\ln\left[\dfrac{\mathrm{y}(k)}{\mathrm{P}_0}\right]}\right\}\end{aligned} \qquad (15)$$

where $\hat{K}(u_1(k),y(k))$ is obtained from the trained affine neural network. Due to some scarce data that may occasionally drive $\hat{K}$ to negative values, its absolute value is used to keep a nonnegative argument in the square root function. In FR mode, $y(k)$ is replaced by $\hat{y}(k)$ in the right hand side of Equation (15) and in the second argument of $\hat{K}$.

The affine neural model of the unknown part of the process has the following structur

$$\hat{K}_n(k) = f(y_n(k)) + u_{1n}(k)\cdot g(y_n(k)) \qquad (16)$$

where $\mathrm{y}_\mathrm{n}(k) = 10^{-5}\mathrm{y}(k)$, $\hat{\mathrm{K}}_\mathrm{n}(\mathrm{k}) = 10^6\hat{\mathrm{K}}(k)$, and $\mathrm{u}_{1\mathrm{n}}(\mathrm{k}) = 0.02\,\mathrm{u}_1(\mathrm{k})$. In the above equation, $f(\cdot)$ and $g(\cdot)$ are the outputs of two separate neural networks. Both networks are single-input single-output two-layer perceptron with three hyperbolic tangent neurons in the hidden layer and one linear neuron in the output layer. Figure 3 shows the structure of the neural network that generates $f(y_n)$. The other neural network has the same topology. Function $\hat{K}_n$ is the normalized friction and

$y_n(k)$ is the normalized pressure. Because the target data derived from equation (14) is of order 10^{-6}(m^6/s^2), the input data *y(k)* is of order 10^5 (Pa), and the input data u_1 varies between 0 and 100, we have chosen the above normalizations as a simple data pre- and post-processing for improving the data-fitting capability of the network. The structure of affine model of friction is shown in Figure 4.

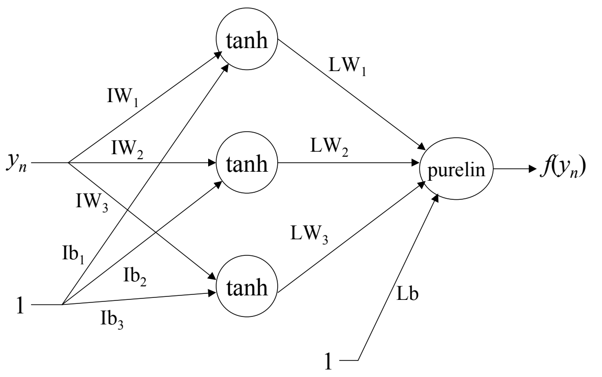

Figure 3. Structure of $f(y_n(k))$ in affine neural model

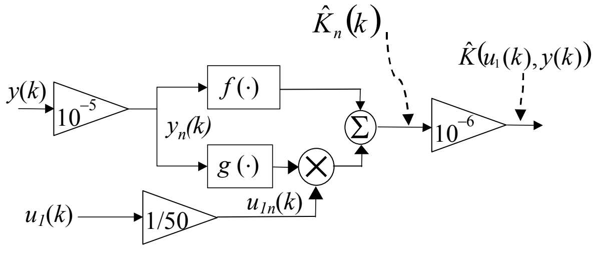

Figure 4. Overall structure of the affine friction neural model

With the training data shown in Figure 2, the overall model was trained by an adaptive L.M. algorithm. However, this configuration requires that both feed-forward networks be trained simultaneously. Figure 5 shows the relative estimation error of the trained affine model, evaluated by the training data. It can be seen that the relative error is below 0.05%.

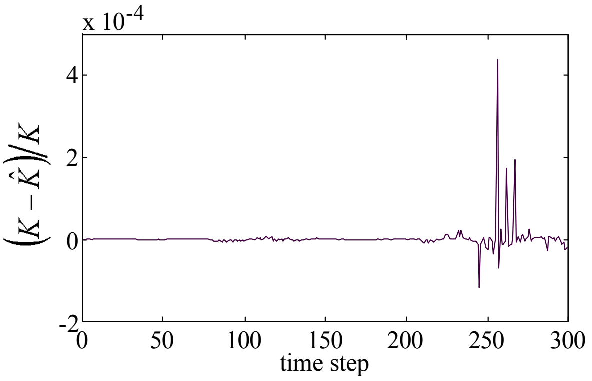

Figure 5. Relative error of the affine neural model

Notice that in the affine gray-box model, depicted in Figure 6,the control input u_1 does not appear linearly in the right hand side of Equation (15) which describes the model structure. Thus, calling the second model 'affine' was only due to the shape of an internal part of the model. In other words, by affine gray-box we mean the gray-box model whose internal neural block forms an affine model for the unknown part of the plant.

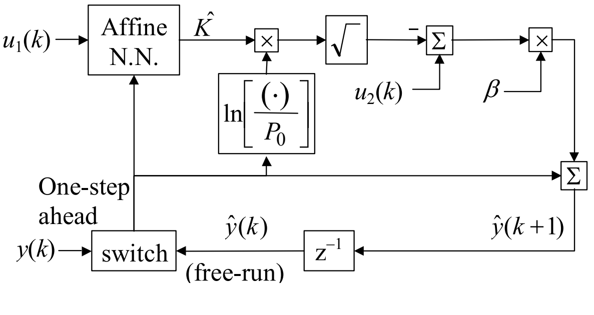

Figure 6. Gray-box model of the fermentor

Now we apply FL procedure to the affine gray-box model described by Equation (15) whose neural model is given in Equation (16) and is depicted in Figure 4. Although control input $u_1(k)$ does not appear linearly in the right hand side of Equation (15) ,it is possible to extract $u_1(k)$ from the equation of the affine gray-box model. This is because $u_1(k)$ appears linearly in the output of the neural friction model given in Equation (16).

We would like to find a control law that forces the model output error to follow a desired stable dynamics described by the following equation:

$$e(k+1)-0.75e(k)=0 \qquad (17)$$

Using Equations (17) and (15) the following equation is obtained in OSA mode:

$$\sqrt{\left|\hat{K}(k)\right|\cdot\ln\left[\frac{y(k)}{P_0}\right]}=u_2-\frac{0.75e(k)+r(k+1)-y(k)}{\beta} \qquad (18)$$

Because the right hand side of the Equation (18) must be nonnegative, we pass it through a saturator defined as:

$$\mathrm{sat}_{0,\infty}(x)=\begin{cases}0 & ,x<0\\ x & ,x\geq 0\end{cases} \qquad (19)$$

Thus, Equation (18) turns to:

$$\sqrt{\left|\hat{\mathrm{K}}(k)\right|\cdot\ln\left[\frac{\mathrm{y}(k)}{\mathrm{P}_0}\right]}=\mathrm{sat}_{0,\infty}\left[\mathrm{u}_2-\frac{0.75\mathrm{e}(k)+\mathrm{r}(k+1)-\mathrm{y}(k)}{\beta}\right] \qquad (20)$$

Since $\hat{K}$ has been trained to approximate a positive function, it is positive for almost all data in the region under consideration ($1<(y/P_0)<2$ and $0<u_1<100$). Thus, we must select only the positive solution for $\hat{K}$ from the above equation to extract $u_1(k)$ from it. Using Equations (16) and (20) and limiting the control signal to interval [0 100], the following control law is obtained:

$$\mathrm{u}_1(\mathrm{k})=\mathrm{sat}_{0,100}\left(\frac{50}{\mathrm{g}(\mathrm{y}_\mathrm{n}(\mathrm{k}))}\cdot\left\{\frac{10^6 s^2(k)}{\ln\frac{\mathrm{y}(k)}{\mathrm{P}_0}}-\mathrm{f}(\mathrm{y}_\mathrm{n}(\mathrm{k}))\right\}\right) \qquad (21)$$

Where

$$\mathrm{s}(\mathrm{k})=\mathrm{sat}_{0,\infty}\left(\mathrm{u}_2-\frac{0.75\mathrm{e}(k)+\mathrm{r}(k+1)-\mathrm{y}(k)}{\beta}\right) \qquad (22)$$

$$y_n(k)=10^{-5}\cdot y(k) \qquad (23)$$

In FR mode, *y(k)* is replaced by $\hat{y}(\mathrm{k})$ in the right hand sides of Equations (21), (22), and (23).

Simulation Results: Results of applying the above control laws with a sinusoidal reference are shown in Figure 7 for both FR and OSA modes.

Notice that during the time that the control signal remains within its limits, the affine gray-box scheme exactly linearizes the model dynamics and $\hat{y}(k)$ converges to *r(k)*.

In Figure 8, results of applying the above FL scheme with different initial conditions of plant and the gray-box model (in FR mode) for a saw-tooth reference signal are shown. The control scheme has provided good tracking even when the initial state of the model is not close to that of the plant. As we can see in Figure 8, when the control law is obtained from model output in FR mode, it would require some time for the model output to converge to the plant output (within an acceptable error). This time interval is not under our control and may be large or small depending on the initial conditions and the open-loop stable nature of the plant; however, after that period, the plant output approximately tracks the reference signal along with the model output whose tracking speed is under control via AFL.

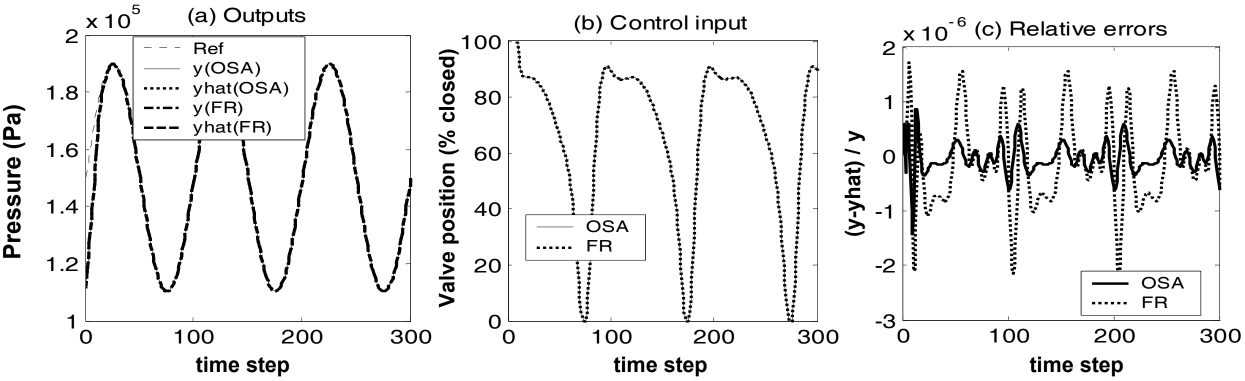

Figure 7. Simulation results of FL via affine gray-box model with sinusoidal reference

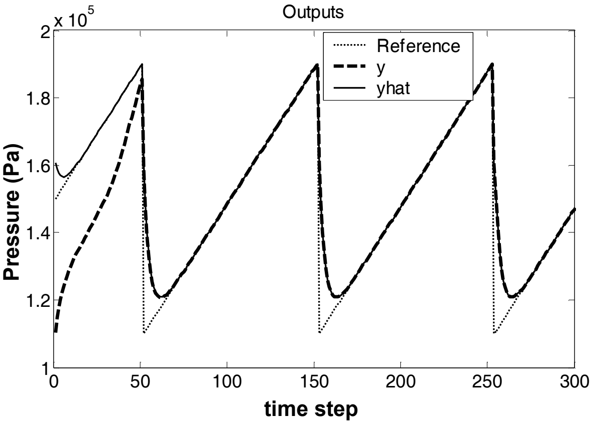

Figure 8. Results of FL via affine gray-box model with different initial conditions in FR mode

Black-Box Affine Neural Modeling and Control

One approach for applying F.L. to this plant, which has been described in section 4.3of [10] , is to generate an affine black-box neural model of the plant, and then, to apply a control law to the physical plant so as to exactly linearize the affine neural model. To compare our result with that of the black-box model, we select the following black-box affine neural model for the plant with the same topology that is employed in [10]:

$$y_n(k) = 10^{-5} y(k) \qquad (24)$$

$$\hat{y}_n(k+1) = f_n(y_n(k), y_n(k-1), y_n(k-2)) + g_n(y_n(k), y_n(k-1), y_n(k-2)\cdot u_1(k) \qquad (25)$$

$$\hat{y}(k+1) = 10^{5}\hat{y}_n(k+1) \qquad (26)$$

where $f_n(\cdot)$ is modeled by a two-layer perceptron with three inputs $y_n(k)$, $y_n(k\text{-}1)$ and $y_n(k\text{-}2)$, one hyperbolic tangent neuron in its hidden layer, and a linear neuron in its output layer. Function $g_n(\cdot)$ is realized by another neural network with the same architecture as the network $f_n(\cdot)$ [10]. Due to their interconnections via equation (25), the off-line training of the networks $g_n(\cdot)$ and $f_n(\cdot)$ should be performed simultaneously. Letting the desired tracking error dynamic be in the form of equation (17), the feedback linearizing control law for the black-box neural model in one-step-ahead mode is:

$$u_1(k) =$$
$$\mathrm{sat}_{0,100}\left(\frac{10^{-5}[r(k+1)+0.75e(k)]-f_n(y_n(k),y_n(k-1),y_n(k-2))}{g_n(y_n(k),y_n(k-1),y_n(k-2))}\right) \tag{27}$$

Where

$$y_n(k)=10^{-5}\cdot y(k) \qquad \& \qquad e(k)=\hat{y}(k)-r(k) \tag{28}$$

Clearly, in FR mode $y(k)$ and $y_n(k)$ must be replaced with $\hat{y}(k)$ and $\hat{y}_n(k)$ in the above equations respectively. Simulations with the sinusoidal reference signal and initial conditions similar to those of the previous section were performed for one-step-ahead and free-run modes, and the resulting relative errors are shown in Figure 9. Comparing these results with those of Figure 7(c) reveals that the affine gray-box FL scheme yielded much smaller tracking errors than that of the affine black-box FL scheme.

Gray-Box and Black-Box Methods in Extrapolation Mode:

We can now examine performances of affine gray-box and black-box neural linearization methods when some measurable parameters of the plant such as input flow rate u_2 or the head space of the vessel do not correspond to those of the training data, i.e., u2=3.75× 10^{-4} (m^3/s) and V_h = 0.015 (m^3). For instance, we reduce headspace V_h by 60% (which may happen as a result of distillation) and apply feedback linearizing control schemes using the previously trained models. Figure 10 shows the plant and the reference output for the control schemes using the affine black-box and gray-box models with a square-wave reference in OSA mode. As we can see, the controller utilized with the black-box model fails to track the desired output so much and exhibits dither. On the other hand, the gray-box based controller still provides good tracking under this much of headspace variation. This is because parameter V_h is in the known part of the plant and thus its measurement is available and its variation in the form of β is directly reflected in the FL control law (see Equation (21)). This is while in the black-box-model-based controller; V_h (or β) does not explicitly show up in the control law, and the neural models that have been trained with the previous value of V_h exhibit a considerable amount of error under this variation. A remedy would be to use online adaptive black-box modeling to track such variations in the plant parameters. However, the adaptive black-box FL controllers have a much slower response than the non-adaptive gray-box FL controllers since, in the latter, the fresh measurements of the parameters in the known/measurable part of the plant are directly used in the control law.

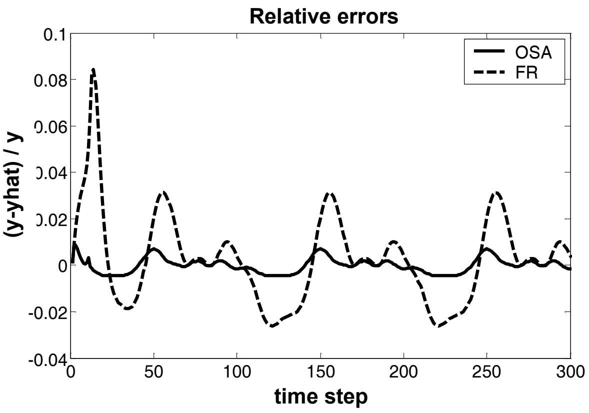

Figure 9. Relative output errors of FL via affine black-box model with sinusoidal reference

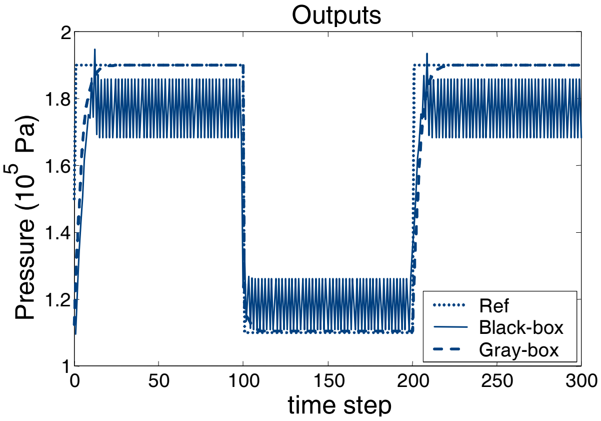

Figure 10. Results of the black-box and the affine gray-box FL controls with reduced V_h (OSA)

Discussions and Conclusions

Exact feedback linearization (FL) methods for nonlinear dynamical systems can provide desirable global performance, but they require exact knowledge of the plant dynamic equations. Thus, when there are structural and/or parametric uncertainties in the plant dynamics, it is not possible to apply FL.

The methods discussed in this paper extend application of FL techniques to nonlinear uncertain plants that can be modeled by neural networks. When there is not any stnctural knowledge about the dynamics of the plant and only suitable I/O data is available, black-box neural networks may be used to model the plant in an affine form, and then, FL can easily be applied to them. However, the validity domain of a black-box model is highly dependent on the domain of the training data.

When precise knowledge about some parts of an uncertain plant exists, it may be possible to use this knowledge in forming a partial structure of the model and to concentrate the modeling capability of the neural network on the unknown part of the plant. This would result in a serial gray-box neural model. Similar to the black-box scheme, it is better to select an affine form for the structure of the gray-box neural model in order to directly apply FL scheme to it.

The structure invertibility must be examined before training the serial gray-box model. If the model is invertible, it may help to obtain a more accurate affine gray-box model.

The serial gray-box models have further versatility and their domains of validity are larger than those of the black-box models. On the other hand, black-box models can be obtained for larger groups of plants. Because in general, using plant input/output data for training a neural network to model an unknown part of a plant is not a routine procedure while training a neural network to model the entire plant is relatively straightforward.

In gray-box modeling, if the measurable parameters in the known part of the plant undergo some variations, they do not affect the performance of the FL controller since their variations are directly reflected in the FL control law. On the other hand, in black-box modeling, such parameters do not explicitly appear in control law, and the neural models that have been trained with the previous value of those parameters exhibit a considerable number of errors after such variations. These variations in the known part of the plant alter the input/output mapping of the entire plant, and consequently, invalidate the black-box model. Using an online adaptive black-box FL controller to track parameter variations would yield a much slower response than the non-adaptive gray-box FL controller. This is because, in the latter, the fresh measurements of the parameters in the known/measurable part of the plant are directly used in the control law.

In the proposed control scheme, both OSA and FR modes may be adopted. Naturally, OSA mode yields better results, but it requires the exact measurement of the plant output in real time. However, when this measurement is not available or is highly corrupted by noise or disturbance, FR mode can be used for open-loop stable plants.

References

1. Vidyasagar, M., Nonlinear Systems Analysis, Prentice-Hall,Englewood Cliffs ,NJ (1993).
2. Slotine, J-J. E., and Li, W., Applied Nonlinear Control, Prentice-Hall, Englewood Cliffs,NJ (1991).
3. Henson, M.A., and Seborg, D.E., "Input-Output Linearization of General Nonlinear Processes," *AIChE Journal*, **36**, 1753 (1990).
4. Henson, M.A., and Seborg, D.E., "Critique of Exact Linearization Strategies for Process Control," *Journal of Process Control*, **1**, 122 (1991).
5. Kravaris, C., and Kantor, J.C., "Geometric Methods for Nonlinear Process Control. Part 1: Background," *Ind. and Eng. Chemistry Res.*, **29**, 2295 (1990a).
6. Kravaris, C., and Kantor, J.C., "Geometric Methods for Nonlinear Process Control. Part 2: Controller Synthesis," *Ind. and Eng. Chemistry Res.*, **29**, 2310 (1990b).
7. Cybenko, G., "Approximation by Superposition of Sigmoidal Function," *Math. Control, Signals Systems*, **2**, 303 (1989).
8. Widrow, B., *et. al.*, "Neural Networks: Application in Industry, Business and Science," *Comm. of the ACM*, **37**, 93 (1994).
9. Nikolaou, M., and Hanagandi, V., "Control of Nonlinear Dynamical Systems Modeled by Recurrent Neural Networks," *AIChE Journal*, **39**, 1890 (1993).
10. Braake, H.A.B. te, Van Can, E.J.L., J.M.A. Scherpen, and H.B. Verbruggen, "Control of Nonlinear Chemical Processes using Neural Models and Feedback Linearization," *Computers Chem. Eng.*, **22** (7-8), 1113 (1998).
11. Van Can, H.J.L., *et. al.*, "Strategy for Dynamic Process Modeling Based on Neural Network in Macroscopic Balances," *AIChE Journal*, **42** (12), 3403 (1996).
12. Psichogios, D.C., and Ungar, L.H., "A Hybrid Neural Network First Principles Approach to Process Modeling," *AIChE Journal*, **38**, 1499 (1992).
13. Schubert, J., *et. al.*, "Bioprocess Optimization and Control: Application of Hybrid Modeling," *Journal of Biotechnology*, **35**, 51 (1994).
14. Thompson, M. L., "Modeling Chemical Processes using Prior Knowledge and Neural Networks," *AIChE Journal*, **40**, 1328 (1994).
15. Wright, R.A., and Kravaris, C., "Practical Development of a Nonlinear Controller for an Industrial-Scale Batch Polymerization Reactor," *Proceedings of the American Control Conference*, Albuquerque, New Mexico, pp. 724 728 (1997).
16. Cô te, M., "Dynamic Modeling of the Activated Sludge Process: Improving Prediction using Neural Network," *Wat. Res.*, **29**, 995 (1995).
17. Su, H.T., *et. al.*, "Integrating Neural Networks with First Principles Models for Dynamic Modeling," *IFAC Symp. on Dynamics and Control of Chemical Reactors*, p.327 (1992).
18. Tulleken, H.J.A., "Gray-Box Modelling and Identification using Physical Knowledge and Bayesian Techniques," *Automatica*, **29**, 285 (1993).
19. Johansen, T. A., "Identification of Non-Linear Systems using Empirical Data and *a Priori* Knowledge-an Optimization Approach," *Automatica*, **32**, 337 (1996).
20. Abonyi, J., "Incorporating Prior Knowledge in Fuzzy Model Identification," *International Journal of Systems Science*, **31** (5), 657 (2000).
21. Abonyi, J., *et. al.*, "Fuzzy Modeling with Multivariate Membership Functions: Gray-Box Identification and Control Design," IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, **31** (5), 755 (2001).