Advances in Engineering Software 41 (2010) 1132–1147

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

# A comparison of networked approximators in parallel mode identification of a bioreactor

Mehmet Önder Efe

*Department of Electrical and Electronics Engineering, TOBB University of Economics and Technology, Sögütözü Cad. No. 43, TR-06560, Sögütözü, Ankara, Turkey*

ARTICLE INFO

*Article history:*
Received 5 April 2010
Received in revised form 7 June 2010
Accepted 13 July 2010
Available online 5 August 2010

*Keywords:*
Bioreactor
Identification
Multilayer perceptron
ANFIS
Support vector machine
Chemical process modeling

ABSTRACT

This paper presents a simulation based comparison of Multilayer Perceptron (MLP), Adaptive Neuro-Fuzzy Inference Systems (ANFIS) and Least Squares Support Vector Machines (LS-SVM) in parallel mode identification of a chemical process displaying several challenges. The paper provides a graphical analysis of the nonlinear behavior for the system under investigation, a case study of purely parallel identification scheme, the effects of noise in the training data on the prediction performance and the performance comparison of the standard approaches under limited amount of numerical data. The results have shown that the emulators utilizing the MLP structure are superior to the others in terms of predicting the system trajectories, locating the limit cycle, noise driven response and predicting the steady state conditions given only 582 pairs of training data. Furthermore, as opposed to others, with the MLP structure, these qualities disappear smoothly as the noise level is increased gradually.



## 1. Introduction

Networked approximators have become very popular and many engineering applications have enjoyed their powerful mapping capabilities. These capabilities are based on the selection of a structure and an associated optimization routine. Once the structure is chosen, many alternatives are available to refine its content but, constructing the structure is certainly a matter of the adopted paradigm. For example, machine imitated brain models can be built in the form of Neural Networks (NNs) if the massively interconnected structure was adopted, and in the form of Fuzzy Logic (FL) if the reasoning and inference were aimed, Jang et al. [16]. Not only because of their highly interconnected and layered structure, but also MLP structures are very well known for a rich set of applicable parameter tuning algorithms. The rationale behind utilizing such approximators is to model the associations and regularities that are difficult to figure out from the numerical data. This aspect of MLP makes it a good candidate for nonlinear identification problems like the one presented in this paper. Initialization of the MLP parameters, choosing the best number of hidden layers, the number of neurons in each layer, learning algorithm and its parameters stipulate a multivariable trial-and-error based optimization process and this abundance is a major drawback of MLP based strategies, Haykin [13]. Recent research studies focusing on the identification of nonlinear systems with the aid of MLP or its variants emphasize the importance of the topic. For example, in Yu and Li [37], the boundedness in the weight update mechanisms with backpropagation type networks are considered with an example chosen from Narendra and Parthasarathy [21]. Parallel identification with long-and short term memory equipped networks is studied in Lo and Bassu [20] and that with nonlinear-in-the-parameters type of a neural network is reported in Abdollahi et al. [1]. An intermediate processing layer is introduced in Patra and Kot [22] and this has resulted in better approximation performance in system identification. A recurrent network structure is studied in Wang and Chen [36] with some examples from Narendra and Parthasarathy [21]. Series-parallel identification scheme is utilized for a novel dynamic neural network architecture in Ren et al. [25] and closed loop control performance is assessed.

When the structural interpretability is taken into consideration, one prefers methods that admit human expertise in the form of comprehensible statements, such as *IF antecedent THEN consequent* rules, as exploited in fuzzy systems. The representation of knowledge utilizes the linguistic tools with a functionally layered structure indicating the fuzzifier, inference engine and defuzzifier of a FL system, Wang [34,35]. A special type of fuzzy systems used for modeling purposes is proposed by Takagi and Sugeno [29], where the defuzzification stage is composed of a linear function of input variables with an offset for the consequent part of each rule. Today, Adaptive Neuro-Fuzzy Inference Systems (ANFIS) proposed by Jang et al. [16] has a special position in this field. The merits and effectiveness of it in various applications are reported many times in the literature (see Jang et al. [16]; Hou et al. [14];

*E-mail address:* onderfe@etu.edu.tr

0965-9978/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2010.07.004

Refaat and Nahavandi [24] and the references therein) and we consider ANFIS based identification to position among the three foremost strategies of learning based approximation theory.

SVM technique introduced in the pioneering work of Vapnik [32] for classification and regression problems aims at minimizing a structural risk function, i.e. the upper bound of an appropriately defined generalization error. Not only are SVMs superior to conventional NNs because of this fact, but also the lack of multiple local minima in SVM learning and the lack of best configuration search effort make SVMs attractive to use instead of conventional NN structures, the training algorithms of which minimize the empirical risk over a set of training pairs instead of its bound (see Gunn [11] and Cristianini and Shawe-Taylor [6]) and as a result of this, a trained neural network may produce higher errors for another data set generated by the same process. However, regarding SVM learning, a significant amount of the computing power is reserved for the optimization algorithm that yields the best set of coefficients. The SVM setting displays sparseness properties leading to few support vectors to describe a desired map. LS-SVM setting, on the other hand, modifies the inequality constraints to equality constraints and similar tools of optimization applies together with the kernel trick. Similar to SVMs, LS-SVMs do not suffer from the problems caused by local minima and they offer solutions to convex problems. The major difference is that the SVM setting determines some support vectors that are generally less than the number of given data points , whereas the LS-SVM setting can preserve all of the given data as support vectors. In Espinoza et al. [10], identification of nonlinear ARX systems with partially linear LS-SVMs are considered and a very good treatment is presented in Suykens et al. [28]. In Zhang et al. [38], a nonlinear NARMAX model under series-parallel identification mode is considered with SVMs with particular emphasis on noise robustness. In Chen et al. [5], a thorough treatment of SVM use in chemistry is presented with a number of application examples.

Clearly the advantages peculiar to each one of MLP, ANFIS and SVM/LS-SVM approaches, make them appealing for identification purposes and the field of process engineering offers interesting modeling problems for comparison of these techniques. Chemical processes often display a complicated behavior due to the strong interdependencies between the variables involved. Although in some cases the process is described by a few state variables, the beauty of nonlinear interactions in the form of limit cycles, attractors and repellers can be created or destroyed under varying operating conditions. Ungar [31] defines a Bioreactor Benchmark Problem that excellently fits in the context. The state of the process is described by two dimensionless variables named the cell mass denoted by $c_1(t)$ and the amount of nutrients denoted by $c_2(t)$ (see Ungar [31]; Puskorius and Feldkamp [23] and Efe et al. [8]). The units for both variables are the normalized concentrations, i.e. grams within the reaction volume. The goal in the original problem is to maintain the cell mass at desired levels by altering the inflow rate at a rate equal to outflow rate keeping the reaction volume constant. Although Ungar [31] emphasizes the challenges associated with the control problem in detail, the considered plant is certainly a good candidate for benchmarking identification methods, the entity of which needs some degrees of intelligence. For instance, NNs, Fuzzy Inference Systems (FISs) or methods adapted from machine learning, such as SVMs are fairly in this class of identification tools to which some degrees of autonomy and fault tolerance are attributed through an appropriate optimization process. This process is called *learning* in the realm of intelligent systems and is the way of refining the information content of the entity being tuned. The motivation for using these methods is the fact that they can infer the structure from numerical data and each one of the considered methods enjoys most of the standard tuning schemes.

In the past, the process described in the bioreactor benchmark problem was studied several times. Efe et al. [8] consider this problem for developing a nonlinear control law forcing the process states to those of a first order linear one. A MLP structure is used to build the nonlinear function in the control law and the plant is forced to follow a reference model. Puskorius and Feldkamp [23] study this problem in the context of demonstrating the efficacy of a NN learning algorithm and consider the control problem about a setpoint in the stable region, another setpoint in the unstable region and a transition between these regions. Brengel and Seider [4] propose a multi step nonlinear controller based on predictive control theory and validate the performance of the closed loop control system on a variant of the model considered here. The authors emphasize the preferability of operating at highest possible cell mass solutions, which are desired to be reasonably away from the region of periodic oscillations. In Anderson and Miller [2], it is emphasized that the controller design for the bioreactor benchmark problem is a challenge due to the nonlinearity and a set of complicated regimes are caused by the nonlinear interactions in between the variables involved. Recent works focusing on the bioreactor benchmark problem consider the adaptive learning based identification in Zou et al. [39], ANFIS based identification in Savran [26] and hybrid control methods combining fuzzy systems and SVMs a in Serdar [15]. Sliding mode control based approaches are also tested on the bioreactor model, such as Efe [7] considers a multi-input multi-output version of the problem whereas Tokat [30] elaborates the switching line adaptation to meet the closed loop performance expectations. Clearly, the works mentioned above motivate us to position the merit and effectiveness of identification tools on such a problem displaying a diverse set of dynamical responses.

From a practitioner's point of view, a good emulator can help the process operators to take precautions timely and to manage the process inputs so that an optimum yield is observed safely. From this viewpoint, recognition of the process dynamics in detail becomes a way that can be followed through an optimal use of the available information, which is typically noisy and limited. Since the goal in this study is to assess to what extent the considered methods capture the essential properties of the system, initially we assume that the available numerical data are noiseless and no on-line refinement is performed after the model is built. The models are then studied with the noisy data sets to see how successful they are in capturing the essential behavior. The contribution of this paper is to unfold the mapping performances of well known three approaches comparatively. One should note that a common property of these methods is their nonlinear nature. In spite of the presence of many linear identification techniques (see Ljung [19]), our goal is to compare those displaying some degrees of intelligence and learning capability. The level of Mean Squared Error (MSE), model complexity measured by the propagation delay, realization accuracy and the ability of discovering the dynamics hidden in the numerical data, e.g. the limit cycle and zero output conditions, are the main categories of the presented comparison. The paper differentiates from the existing body of literature from the points of (i) comparing standard approaches with their standard optimization (learning or training) algorithms, (ii) making a performance comparison of models generated with noisy as well as noiseless training data, (iii) elaborating a purely parallel identification scheme yielding a standalone dynamic model exploiting few data and (iv) concluding mainly with the outstanding success of the models based on the classical MLP structure. In the past, mapping performance of NN structures are compared for series-parallel identification scheme (see for example Efe and Kaynak [9]; Hou et al. [14]; Zhang et al. [38]), where the plant states are fed to the model and one step ahead prediction is performed. Unlike this approach, as discussed in the sequel, in

the parallel mode identification, the trained approximator provides the derivatives of the state variables and after the integration of these values, the outputs are fed to the model itself and due to the approximation errors, the result is more likely to diverge or saturate after a while. For this reason, the adopted modeling strategy, i.e the model structure, its tuning scheme and the training data, contains key points that embody the essence of the contribution of this paper.

This paper is organized as follows. Section 2 introduces the bioreactor benchmark problem and analyzes its behavior. Sections 3–5 present the MLP, ANFIS and LS-SVM approaches in turn. Operating conditions, results and discussion are presented in Section 6. The concluding remarks are presented at the end of the paper (Section 7).

## 2. Bioreactor benchmark problem

The bioreactor is a tank in which the biological cells are mixed with nutrients and water as shown in Fig. 1. The cells and nutrients are in a dynamical interaction modeled by (1) and (2), where $c_1(t)$ denotes the cell mass while $c_2(t)$ stands for the nutrient amount. The process is continuously fed by pure water and the variable characterizing the inflow rate is denoted by $w(t)$, which is in liters per second. In order to maintain the reaction volume constant, the contents of the tank is removed at a rate equal to the inflow rate, $S(t)$, which is composed of a mixture of cells, nutrients and water, and its dimension is also liters per second.

$$\dot{c}_1(t) = -c_1(t)w(t) + c_1(t)(1-c_2(t))e^{\frac{c_2(t)}{\gamma}} \tag{1}$$

$$\dot{c}_2(t) = -c_2(t)w(t) + c_1(t)(1-c_2(t))e^{\frac{c_2(t)}{\gamma}}\frac{1+\beta}{1+\beta-c_2(t)} \tag{2}$$

The state variables are constrained by $\Omega$: = $0 \leqslant c_1(t), c_2(t) \leqslant 1$. In the model of the plant given above, the growth rate is characterized by the parameter $\beta = 0.02$ and the nutrient inhibition parameter is given by $\gamma = 0.48$.

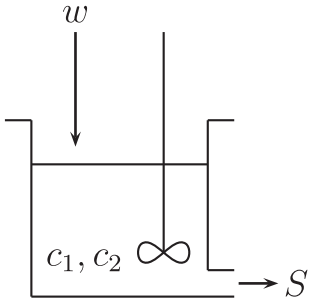


**Fig. 1.** Reaction tank with equal inflow and outflow rates.

In Fig. 2, several trajectories are shown for a set of initial conditions denoted by a circle. Each subplot depicts the evolution of the system at a constant inflow rate indicated on the top. Depending on the value of the inflow rate, the attractors change their locations or new attractors emerge. One visible one is a limit cycle which becomes apparent when $w = 1$. Particularly when $w = 0.829$, the system changes its qualitative behavior radically. Computing the equilibrium values corresponding to this inflow rate, one obtains $c_1 = 0.1331$ and $c_2 = 0.8226$. The eigenvalues of the linearized system of equations at this point stipulate that in the increasing direction of $c_2$, the system undergoes Hopf bifurcation at this operating point and turns into an unstable one displaying spontaneous oscillations due to the limit cycle. In this regime, cell mass varies in between 0.1219 and 0.1466 while the nutrient amount fluctuates in between 0.8242 and 0.8996. At the points of crossing the imaginary axis, the eigenvalues of the linearized model are approximately equal to $0 \pm j1.7543$, from which we infer that the self sustained oscillations are quite fast.

In the left subplot of Fig. 3, the limit cycle and the convergence of the neighboring trajectories are illustrated for $w = 1.2$. In fact, limit cycles can occur for all values of admissible inflow rates, i.e. $0 \leqslant w \leqslant 2$. According to Bendixson theorem (see Slotine and Li [27] and Khalil [18]), since the quantity

$$\begin{aligned}\Psi &:= \frac{\partial}{\partial c_1}\left(-c_1 w + c_1(1-c_2)e^{\frac{c_2}{\gamma}}\right) \\ &\quad + \frac{\partial}{\partial c_2}\left(-c_2 w + c_1(1-c_2)e^{\frac{c_2}{\gamma}}\frac{1+\beta}{1+\beta-c_2}\right) \\ &= -2w + \psi(c_1, c_2)\end{aligned} \tag{3}$$

does not vanish and does not change sign in $\Lambda \subseteq \Omega$, no limit cycles can exist entirely in $\Lambda$. For a given constant $w$, the curve of sign change for $\Psi$ is moved to the curve described by $\psi(c_1, c_2) = 2w$. Therefore, one should run the quantity $2w$ from 0 to 4 and determine where the sign change occurs. In the right subplot of Fig. 3, we illustrate the limit cycle shown on the left once again, and the regions where the limit cycles cannot lie entirely within are depicted as white regions, termed $\Lambda$ above, and the value of $2w$ is contoured for $2w$ equals to 0, 1, 2, 3 and 4. According to this result, we figure out that it is possible to have other limit cycle trajectories in the system dynamics and $\Lambda$ is a significantly wide subspace of $\Omega$. From a systems engineering point of view, this practically tells us that for a given sequence of inflow rate values, new attractors and/or repellers can be created or destroyed depending on the value of $w$ and a good model, which we call *emulator*, must be able to capture these properties satisfactorily.

Consider the process at the steady state, i.e. $\dot{c}_1 = 0$ and $\dot{c}_2 = 0$. This yields the steady state inflow rate $w_{ss} = (1-c_2)e^{\frac{c_2}{\gamma}}$ and with this feed stream we obtain $\dot{c}_2 = (1-c_2)e^{\frac{c_2}{\gamma}}(-c_2 + c_1 g)$, i.e. when

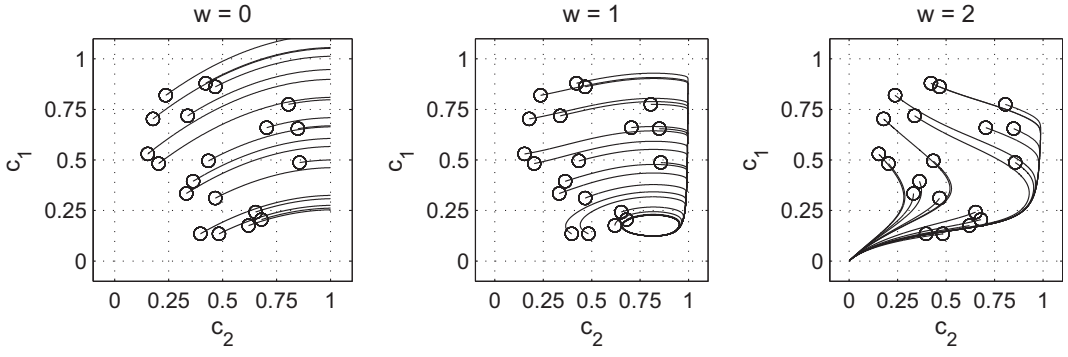


**Fig. 2.** The evolution of the process states for different initial conditions and at different inflow rates. The trajectories are for 60 s time.

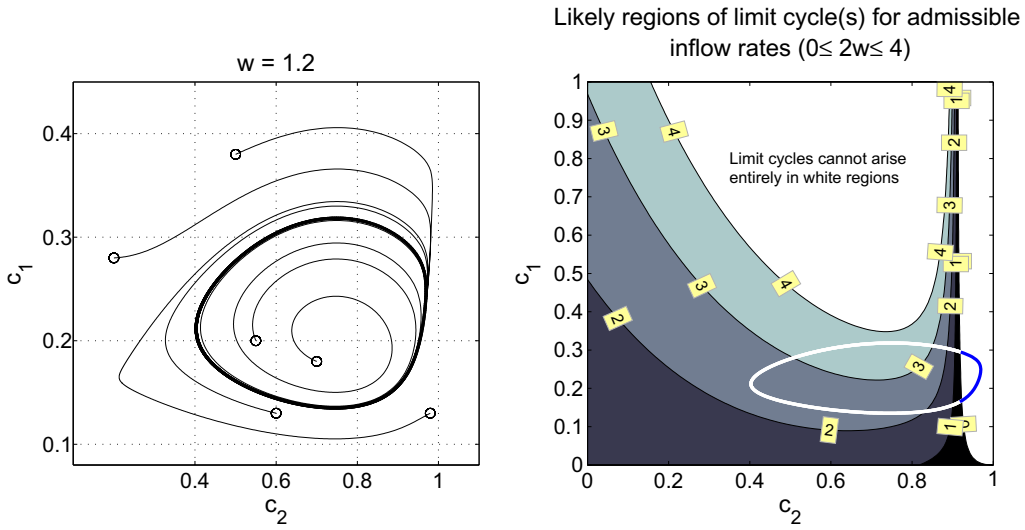


**Fig. 3.** Left: Limit cycle arising when $w = 1.2$. Right: The region where the limit cycles cannot arise entirely and the locus of the limit cycle shown in the left.

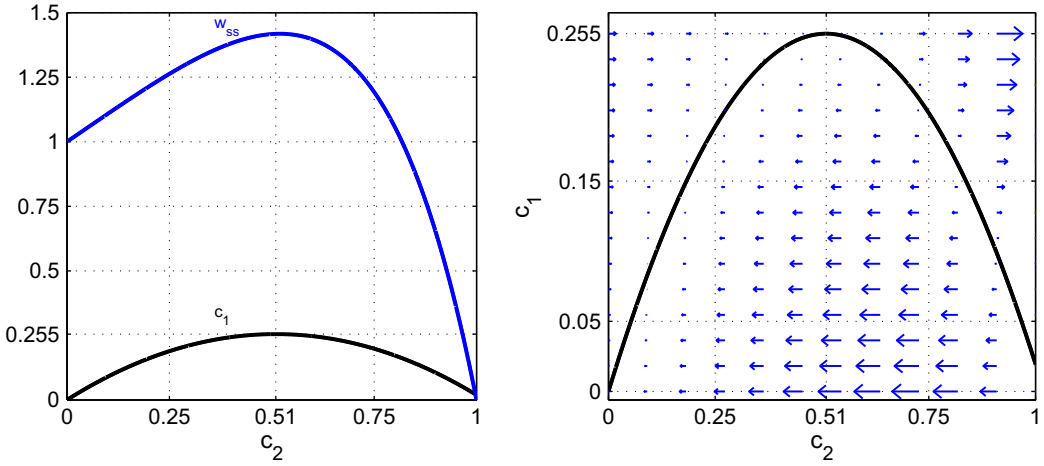


**Fig. 4.** Left: Equilibrium points. Right: The behavior with steady state inflow rate $w_{ss}$.

the steady state is reached $c_2$ is either 1 or a value that satisfies $g = c_2/c_1$ where $g(c_2(t)) := \frac{1+\beta}{1+\beta-c_2(t)}$. In the left subplot of Fig. 4, the parabolic (lower) curve depicts the solution obtained from $-c_2 + c_1 g = 0$. The same subplot also depicts the value of steady state inflow rate $w_{ss}$ along with the $c_2$-axis. This figure stipulates that when the steady state is reached, the cell mass cannot assume values larger than $\frac{1+\beta}{4} = 0.255$. This practical constraint was also highlighted in Puskorius and Feldkamp [23] and Efe et al. [8]. The right subplot of Fig. 4 depicts how an arbitrary value of $(c_1(t_0), c_2(t_0))$ moves under the inflow rate $w_{ss}$. Few comments can be made for the motion for $t > t_0$ with $w_{ss}$. If $c_1(t_0) > \frac{1+\beta}{4}$ then the final value of the state vector is $(c_1(\infty), c_2(\infty)) = (c_1(t_0), 1)$. If $c_1(t_0) < \frac{1+\beta}{4}$, then any initial condition that is below the curve defined by $-c_2 + c_1 g = 0$ moves in the horizontal direction and stops on the left segment of the curve $-c_2 + c_1 g = 0$, which is shown in the bottom right of Fig. 4. Alternatively, the initial conditions that are above the curve defined by $-c_2 + c_1 g = 0$ move right until an equilibrium is reached, i.e. those satisfying $c_2(t_0) < \frac{1+\beta}{2}$ converge the left segment of the curve $-c_2 + c_1 g = 0$, however, those with $c_2(t_0) > \frac{1+\beta}{2}$ stop at $(c_1(\infty), c_2(\infty)) = (c_1(t_0), 1)$. At the point $(c_1(t_0), c_2(t_0)) = (\frac{1+\beta}{4}, \frac{1+\beta}{2})$, the eigenvalues of the linearized system are both equal to zero. According to the behavior indicated by the vector field (flow), we conclude that this point is a half stable point, i.e. the trajectories close to this point but below $-c_2 + c_1 g = 0$ behave different from those above $-c_2 + c_1 g = 0$. This discussion with the flow illustrated in Fig. 4 clarifies the stability of the equilibrium states thoroughly.

Ungar [31] points out that although this system is not a completely realistic model of any bioreactor, as seen from the presented discussion, the system considered in this paper displays several challenges, which can be available in real time applications, highlighted also by Anderson and Miller [2] with a similar motivation. Due to the presented properties of the system, the model constitutes a good test bed for scrutinizing the capabilities of networked modeling techniques revealing a set of flexible nonlinearities in various forms.

## 3. Multilayer perceptron network

The history of MLP goes back to the works devoted to the understanding of brain activity based on neurons (nodes). An ordered structure of a set of neurons form a layer, and building layers in an organized fashion constitutes a MLP structure as shown in

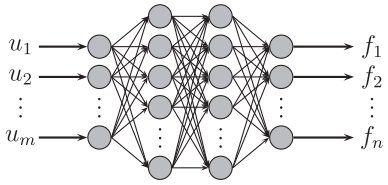


**Fig. 5.** Structure of a MLP.

Fig. 5. The figure depicts an input layer containing $m$ nodes, an output layer composed of $n$ nodes and two hidden layers, in which the number of nodes are to be determined by the designer.

As seen from the figure, the structure of MLP has many internal connections, called synaptic connections, which possess synaptic strengths. In other words, biologically resistive nature of such connections with a saturating response of a neuron is simply modeled as $\mathbf{h} = \Phi(\mathbf{s})$, where $\mathbf{s}$ denotes the weighted sum of incoming signals and $\Phi(\mathbf{s})$ stands for the neuronal activation function that is responsible for the saturated response arises when the argument $\mathbf{s}$ is large in magnitude. Clearly, the number of terms involved in computing the value of $\mathbf{s}$ for each neuron indicates the possible architectural redundancy in the representation of knowledge within a NN structure.

For learning with the MLP structure, consider the regression problem over the pairs

$$\mathscr{T} = \{(\mathbf{u}_1, \mathbf{d}_1), \ldots, (\mathbf{u}_{N_\mathscr{T}}, \mathbf{d}_{N_\mathscr{T}})\}, \qquad \mathbf{u}_i \in \mathbb{R}^m, \quad \mathbf{d}_i \in \mathbb{R}^n \tag{4}$$

Compactly, denote the number of layers excluding the input layer by $H$ and the $i$th layer output vector by $\mathbf{h}_i$, where $\mathbf{h}_i = \Phi(\mathbf{s}_i)$ and $\mathbf{s}_i$ is the vector of net sums computed as

$$\mathbf{s}_i = \mathbf{W}_i \mathbf{h}_{i-1} + \mathbf{B}_i, \quad i = 1, 2, \ldots, H \tag{5}$$

where $\mathbf{W}_i$ and $\mathbf{B}_i$ correspond to the weight and bias terms of the $i$th layer. It is clear that for a NN structure with two hidden layers containing hyperbolic tangent type activation functions, and a linear output layer, the successive computations through the network (one forward pass to compute the output) would be

$$\mathbf{h}_0 = \mathbf{u} \tag{6a}$$
$$\mathbf{s}_1 = \mathbf{W}_1 \mathbf{h}_0 + \mathbf{B}_1 \text{ and } \mathbf{h}_1 = \tanh(\mathbf{s}_1) \tag{6b}$$
$$\mathbf{s}_2 = \mathbf{W}_2 \mathbf{h}_1 + \mathbf{B}_2 \text{ and } \mathbf{h}_2 = \tanh(\mathbf{s}_2) \tag{6c}$$
$$\mathbf{s}_3 = \mathbf{W}_3 \mathbf{h}_2 + \mathbf{B}_3 \text{ and } \mathbf{f} = \mathbf{s}_3 \tag{6d}$$

where the weight and bias terms seen above have appropriate dimensions and the input–output relation would simply be $\mathbf{f} = \mathbf{W}_3 \tanh(\mathbf{W}_2 \tanh(\mathbf{W}_1 \mathbf{u} + \mathbf{B}_1) + \mathbf{B}_2) + \mathbf{B}_3$. Once a structure is built, the next issue is to adopt a suitable learning strategy. Although there are numerous alternatives for tuning the MLP weights and biases, Levenberg–Marquardt (LM) optimization technique is the one that is frequently used for its rapid convergence. The LM algorithm is an approximation to the Newton's method, and both of these methods have been designed to solve the nonlinear least squares problem (see Battiti [3] and Hagan and Menhaj [12]). Since the problem considered here is involved with offline training of a MLP structure, LM algorithm best fits in our problem settings. Briefly, vectorize the set of all adjustable parameters and denote this vector by $\omega$, which is a $P \times 1$ vector. At time $k$, a cost function, which is an empirical risk function, qualifying the performance of the interpolation can be given as in (7) and the LM update is formulated as in (8)

$$E(\omega_k) = \frac{1}{2} \sum_{i=1}^{N_\mathscr{T}} \|\mathbf{d}_i - \mathbf{f}(\mathbf{u}_i, \omega_k)\|^2 \tag{7}$$

$$\omega_{k+1} = \omega_k - \left(\alpha \mathbf{I} + \nabla^2_{\omega_k} E(\omega_k)\right)^{-1} \nabla_{\omega_k} E(\omega_k) \tag{8}$$

where $\nabla_{\omega_k} E(\omega_k)$ corresponds to the transpose of the Jacobian denoted by $\mathbf{J}$ and the Hessian $\nabla^2_{\omega_k} E(\omega_k)$ is approximated by $\mathbf{J}^{\mathrm{T}}\mathbf{J}$. $\alpha > 0$ is a user-defined scalar design parameter and $\mathbf{I}$ is an identity matrix of appropriate dimensions. It is important to note that, for small $\alpha$, (8) becomes the standard Gauss–Newton method, and for large $\alpha$, the tuning law becomes the standard Error Backpropagation (EBP) algorithm. Therefore, LM method establishes a good balance between EBP and Gauss–Newton strategies.

The pioneering work of Narendra and Parthasarathy [21] considers the MLP structure with the EBP algorithm for identification and control purposes through some abstract yet descriptive examples with an emphasis on slow convergence of the EBP technique which has later been resolved by the use of LM technique.

## 4. Learning and generalization with ANFIS

Fuzzy logic offers one natural way for representing knowledge that is similar to human reasoning. Partitioning the input space by the use of fuzzy membership functions, determining the local conclusions through rules and utilizing a flexible method of combining the localized information result in a highly interpretable and useful model that acts globally. ANFIS, in this respect, is one of the widely known architectures exploiting the power of connectionist structures while maintaining the verbal nature through membership functions and inference mechanisms, Hou et al. [14]; Refaat and Nahavandi [24]. In Fig. 6, general structure of a FIS is illustrated. The crisp inputs are fuzzified through the computation of membership functions. This practically maps the input space to a feature space characterized by fuzzy sets. In the

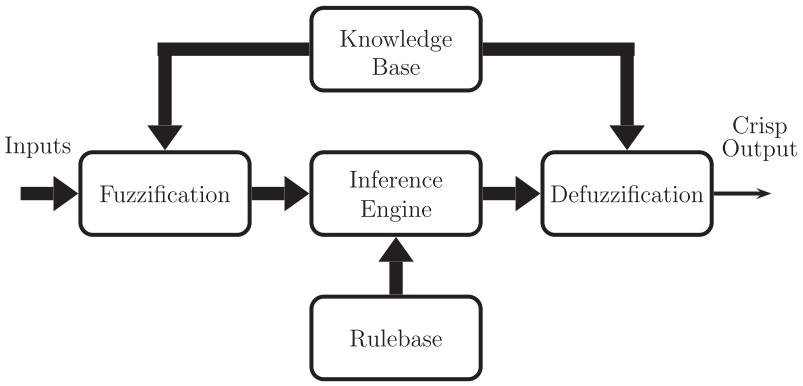


**Fig. 6.** Structure of a Fuzzy inference system.

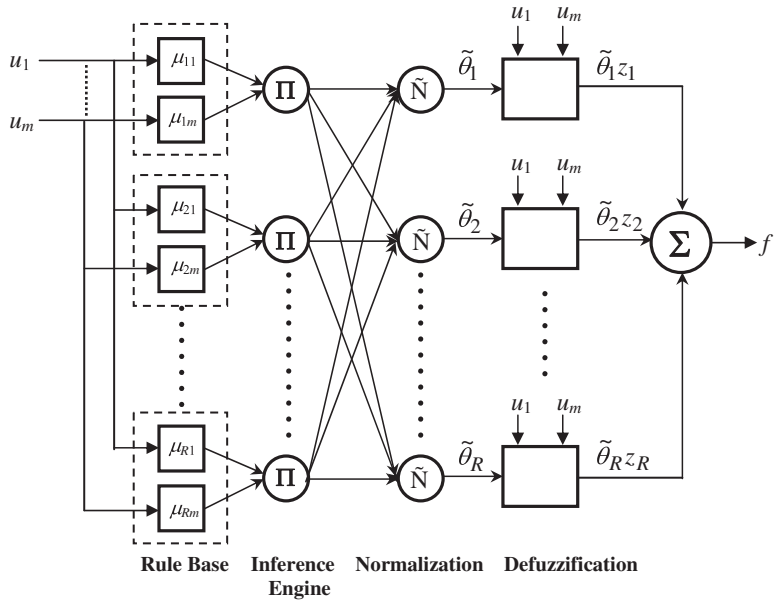


**Fig. 7.** Internal connectivity of ANFIS structure.

inference mechanism, computed membership values for each rule are converted into a firing strength that indicates the activation level of the rule. The parameters of the membership functions and auxiliary parameters are stored in the knowledge base, and a defuzzifier maps the output of the inference engine to a scalar output value, which is crisp.

In Fig. 7, the structure of ANFIS is depicted. As shown also on the figure, defining $\theta_i$, $\tilde{\theta}_i$ as the firing strength and normalized firing strength of $i$th rule, respectively, the input output relation of the ANFIS structure with the rulebase structure containing $R$ rules as below, product inference and first order Sugeno type defuzzifier is as given in (9a)–(9d) (see Takagi and Sugeno [29]). Note that $\mathbb{U}_i$, $\mathbb{V}_i$ and $\mathbb{W}_i$ stand for the fuzzy sets characterized by the membership functions, $y_i$ in the $i$th rule is the local conclusion suggested by the rule and, $n_1$, $n_2$ and $n_m$ correspond to the number of linguistic labels for the first, second and $m$th input variables, respectively

IF $u_1$ is $\mathbb{U}_1$ AND $u_2$ is $\mathbb{V}_1$ AND … AND $u_m$ is $\mathbb{W}_1$ THEN $y_1 = z_1$
IF $u_1$ is $\mathbb{U}_1$ AND $u_2$ is $\mathbb{V}_1$ AND … AND $u_m$ is $\mathbb{W}_2$ THEN $y_2 = z_2$
⋮ ⋮
IF $u_1$ is $\mathbb{U}_{n_1}$ AND $u_2$ is $\mathbb{V}_{n_2}$ AND … AND $u_m$ is $\mathbb{W}_{n_m}$ THEN $y_R = z_R$

$$\theta_i = \prod_{j=1}^{m} \mu_{ij}(u_j) \tag{9a}$$

$$\tilde{\theta}_i = \frac{\theta_i}{\sum_{k=1}^{R} \theta_k} \tag{9b}$$

$$z_i = \zeta_i + \sum_{j=1}^{m} \phi_{ij} u_j \tag{9c}$$

$$f = \sum_{i=1}^{R} \tilde{\theta}_i z_i \tag{9d}$$

In (9a) and (9c), $u_j$ corresponds to the $j$th entry of the input vector $\mathbf{u}$. According to (9d), it is seen that the ANFIS structure has single output. The training is achieved by adopting a hybrid tuning mechanism. Specifically, $\zeta_i$ and $\phi_{ij}$ are adjusted by Least Mean Squares (LMS) algorithm, while the other parameters are tuned by EBP method. It is emphasized in Jang [17]; Jang et al [16]; Refaat and Nahavandi [24] that such a tuning scheme reduces the dimensionality of the search space of EBP algorithm and speeds up the convergence.

## 5. Least Squares Support Vector Machines for approximation

Due to powerful regression and classification capabilities based on numerical observations, the use of LS-SVMs can be considered as a remedy among other alternatives such as FL, NNs or genetic algorithms, which generally suffer from the presence of multiple local minima, structure selection (hidden layer number or node/rule number, population size) problem and overfitting, Wang and Ye [33]. Similar to MLP and ANFIS, LS-SVMs map the input vector to a feature space, where the regression is performed much efficiently. Yet the major difference in LS-SVM learning is the minimization of an upper bound on a quantity instead of minimizing the quantity itself. The goal of the former is to minimize the structural risk function whereas the latter minimizes the empirical risk function. This practically makes it possible to generalize the information contained implicitly in the training set in some sense of optimality, Gunn [11]; Suykens et al. [28]. In this section, we summarize the LS-SVM structure utilized in the paper.

Consider the regression problem over the pairs

$$\mathscr{T} = \{(\mathbf{u}_1, d_1), \ldots, (\mathbf{u}_{N_\mathscr{T}}, d_{N_\mathscr{T}})\}, \qquad \mathbf{u}_i \in \mathbb{R}^m, \quad d_i \in \mathbb{R} \tag{10}$$

with a function

$$f(\mathbf{u}) = \mathbf{w}^{\mathrm{T}} \varphi(\mathbf{u}) + b \tag{11}$$

where $\mathbf{w}$ and $b$ denote the weight vector and the bias value, respectively. $\varphi(\cdot)$ stands for an implicitly defined, possibly a nonlinear map allowing the application of kernel trick wherever necessary. We define a quadratic loss function as in (12) to quantify the performance for the $i$th data pair,

$$L(d_i, f(\mathbf{u}_i)) = \frac{1}{2} e_i^2 \tag{12}$$

where $e_i := d_i - f(\mathbf{u}_i)$. Minimizing the structural risk given by (13) lets us obtain the best values of $\mathbf{w}_i$s causing least complexity represented by $\|\mathbf{w}\|^2$

$$R = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N_{\mathscr{T}}} L(d_i, f(\mathbf{u}_i)) \tag{13}$$

where $C$ is the regularization constant determining the relative importance of the terms contributing to $R$, Gunn [11]. According to (13), large $C$ results in better fit to the given data. The primal form of the optimization problem can be expressed compactly as

$$\min_{\mathbf{w},b,e} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}C\sum_{i=1}^{N_{\mathscr{T}}} e_i^2$$
$$\text{such that } d_i = \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{u}_i) + b + e_i, \quad i = 1, 2, \ldots, N_{\mathscr{T}} \tag{14}$$

The above described optimization problem can be converted into a linear system equations by exploiting the dual representation. Denoting the Lagrange multipliers by $\lambda$, the Lagrangian can be constructed as in (15), and the solution to this optimization problem is obtained at the saddle point of the Lagrangian, i.e. $\max_{\lambda} \min_{\mathbf{w},b,e} \mathscr{L}(\mathbf{w}, b, e, \lambda)$. The conditions of optimality are given in (16–19).

$$\mathscr{L}(\mathbf{w}, b, e, \lambda) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}C\sum_{i=1}^{N_{\mathscr{T}}} e_i^2 - \sum_{i=1}^{N_{\mathscr{T}}} \lambda_i(\mathbf{w}^{\mathrm{T}}\varphi(\mathbf{u}_i) + b + e_i - d_i) \tag{15}$$

$$\frac{\partial \mathscr{L}}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{N_{\mathscr{T}}} \lambda_i \varphi(\mathbf{u}_i) \tag{16}$$

$$\frac{\partial \mathscr{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N_{\mathscr{T}}} \lambda_i = 0 \tag{17}$$

$$\frac{\partial \mathscr{L}}{\partial e_i} = 0 \Rightarrow \lambda_i = Ce_i, \quad i = 1, 2, \ldots, N_{\mathscr{T}} \tag{18}$$

$$\frac{\partial \mathscr{L}}{\partial \lambda_i} = 0 \Rightarrow \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{u}_i) + b + e_i - d_i = 0, \quad i = 1, 2, \ldots, N_{\mathscr{T}} \tag{19}$$

The solution can be obtained by solving the $N_{\mathscr{T}} + 1$ equations simultaneously. These equations are

$$b + \frac{\lambda_k}{C} - d_k + \sum_{i=1}^{N_{\mathscr{T}}} \lambda_i \varphi(\mathbf{u}_i)^{\mathrm{T}}\varphi(\mathbf{u}_k) = 0, \quad k = 1, 2, \ldots, N_{\mathscr{T}} \tag{20}$$

and the condition in (17), where $\lambda$ and $b$ are the unknowns. The optimization problem can now enjoy the kernel trick letting us to write $K(\mathbf{u}_i, \mathbf{u}_k) := \varphi(\mathbf{u}_i)^{\mathrm{T}}\varphi(\mathbf{u}_k)$, with $K(\cdot,\cdot)$ being an appropriately defined kernel satisfying the Mercer conditions, Cristianini and Shawe-Taylor [6] and Suykens et al. [28].

It should be noted that the support vectors are the $\mathbf{u}_i$s for which the corresponding $\lambda_i$ is nonzero. However, in Suykens et al. [28], it is emphasized that the model obtained utilizing the method above suffers from the lack of sparseness. More explicitly, due to the condition in (18), a significant part of the Lagrange multipliers will not be exactly zero and as a natural consequence of this, most of the training data pairs will be contained as support vectors leading to a possible increase in the memory storage capacity.

## 6. Operating conditions, results and discussion

In this section, the three approaches discussed so far will be utilized to predict the derivatives of the state variables. Clearly, the training data will be provided through the use of (1) and (2), and the major difficulty of utilizing a predictor in the form of aforementioned structures is the integration of the approximation error together with the true values. Although one's expectation could be to observe a saturation in the long run, some models are seen to be able to recover and yield good predictions for the state variables. Towards this goal, in Fig. 8, the training structure for the aforementioned models is illustrated. In what follows, we summarize the procedure followed during the training data generation, the results of the simulations with a thorough discussion.

### 6.1. Training, checking and validation data sets

In the presented comparison work, three sets of data are generated randomly from the interval $(w, c_1, c_2) \in [0, 2] \times [0, 1] \times [0, 1]$ and the target values $(\dot{c}_1, \dot{c}_2)$ are computed using (1) and (2).

- Training data ($\mathscr{T}$): The numerical information available in the training data set is used when the parameters of the considered emulator are modified. The number of pairs contained in $\mathscr{T}$ is denoted by $N_{\mathscr{T}}$.
- Checking data ($\mathscr{C}$): This data set is used whenever the training algorithm enables terminating the training procedure at an optimal level of generalization. The number of pairs contained in $\mathscr{C}$ is denoted by $N_{\mathscr{C}}$. The pairs in this set are generated randomly and are not necessarily in temporal order.
- Validation data ($\mathscr{V}$): The information used to measure the representational capabilities of the trained emulator. The number of pairs contained in $\mathscr{V}$ is denoted by $N_{\mathscr{V}}$ and the pairs in this set are consecutive samples in time.

The emulators are constructed utilizing the noiseless (clean) data first and then the noisy observations are dwelt on to figure out the generalization performance under imprecise observations.

The emulator, which is based on one of the MLP, ANFIS or LS-SVM structures, aims to learn the function $\mathbf{f}^{\star}(w, \mathbf{c})$ described in (1) and (2), but in reality, it learns a function denoted by $\mathbf{f}(w, \mathbf{x})$. The vector of approximation errors, which is $2 \times 1$, is denoted by $\varepsilon(w, \mathbf{c}, \mathbf{x})$ that is defined as $\varepsilon(w, \mathbf{c}, \mathbf{x}) := \mathbf{f}^{\star}(w, \mathbf{c}) - \mathbf{f}(w, \mathbf{x})$. The output of the emulator is computed as

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{f}(\mathbf{u}(\tau))\mathrm{d}\tau \tag{21}$$

where $\mathbf{u}(t) := (w(t)\mathbf{x}(t)^{\mathrm{T}})^{\mathrm{T}}$. Clearly the system is in a parallel mode identification scheme and the approximation error contained in $\mathbf{f}(w, \mathbf{x})$ will be highly effective as it is integrated, this can be seen in the error equation given in (22)

$$\mathbf{e}(t) = \mathbf{e}(0) + \int_0^t \varepsilon(w(\tau), \mathbf{c}(\tau), \mathbf{x}(\tau))\mathrm{d}\tau \tag{22}$$

Without loss of generality, in our tests, we assume that the initial error ($\mathbf{e}(0)$) is zero. According to (22), the trajectories are expected to diverge from each other in a manner determined by the approximation error $\varepsilon$. This enables us to determine the best performing approximator and to assess the effect of different noise levels on the overall performance for each approach.

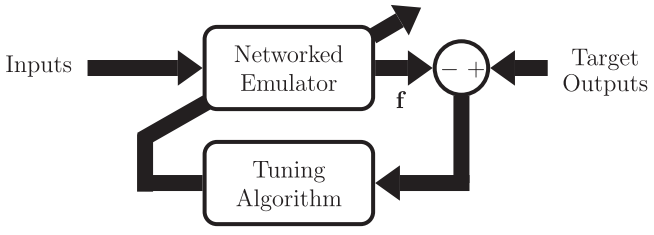


**Fig. 8.** Block diagram for obtaining the networked emulator.

### 6.2. Noise free case: numerical data provides $\dot{c}_1$ and $\dot{c}_2$ perfectly

Initial work to assess the performance under noiseless training data is comprised of a comparison of the propagation delay, MSE level achieved when the training is stopped, the value of checking MSE level and the MSE level for a given trajectory. The results of the performed experiments are summarized in Table 1, the propagation delay in milliseconds is denoted by $T_{\Delta}$, training MSE level achieved with the samples in $\mathscr{T}$ is denoted by $\mathscr{T}_{\text{MSE}}$, $\mathscr{C}_{\text{MSE}}$ corresponds to checking MSE level. Since the approximators predict the derivatives of the state variables, the quantities $\mathscr{T}_{\text{MSE}}$ and $\mathscr{C}_{\text{MSE}}$ are computed as given below

$$\mathscr{T}_{\text{MSE}} := \frac{1}{N_{\mathscr{T}}} \sum_{i=1}^{N_{\mathscr{T}}} \|\varepsilon_i\|^2 \tag{23a}$$

$$\mathscr{C}_{\text{MSE}} := \frac{1}{N_{\mathscr{C}}} \sum_{i=1}^{N_{\mathscr{C}}} \|\varepsilon_i\|^2 \tag{23b}$$

where $\varepsilon_i$ corresponds to the approximation error vector computed with the values of the $i$th data pair. $\mathscr{V}_{\text{MSE}}$, which is based on the emulator responses, stands for the performance index (validation MSE level) given by

$$\mathscr{V}_{\text{MSE}} := \frac{1}{T_f} \sum_{i=1}^{2} \int_0^{T_f} (c_i(t) - x_i(t))^2 \mathrm{d}t \tag{24}$$

where $T_f$ denotes the final time of the simulation for a single trial. We set $T_f = 100$ s indicating $N_{\mathscr{V}} = 10,000$ pairs in the validation data set as the discretization interval is 0.01 s.

In the simulations, we have used a MLP structure with two hidden layers having six neurons in the first and 24 neurons in the second hidden layer. Although using two hidden layers is not common and the theory stipulates that a MLP with enough number of hidden neurons is a universal approximator, cascaded nonlinear layers provide feature spaces through which a desired input output map can be established with neurons less in number than a single hidden layer MLP needs. The output layer is chosen as a linear one containing two neurons that provide the value of $\mathbf{f}(w, \mathbf{x})$. The tests carried out with MLP structure are repeated for ANFIS and LS-SVM based emulators with the settings summarized below. We have generated $N_{\mathscr{C}} = 500$ pairs for the checking data set and used this set wherever applicable.

Two ANFIS structures having the same structure are considered for the modeling goal. The output of the first structure is a prediction for $\dot{c}_1(t)$ while that of the second is a prediction for $\dot{c}_2(t)$. Two linguistic labels are utilized for the variables $w(t)$ and $c_1(t)$, and five labels are utilized for the variable $c_2(t)$. In total, 20 rules are built in the rule base of each ANFIS architecture and these numbers are set by an extensive trial-and-error process. Generalized bell shaped membership functions are utilized and two ANFIS structures having a total of 107 adjustable parameters in each is constructed. The adjustable parameters of the defuzzifier ($\zeta_i$s and $\phi_{ij}$s), are zero initially and the membership functions cover the input space uniformly.

Regarding the LS-SVM approach, which is a single output structure, two LS-SVMs are considered for derivative predictions. In this paper, we utilize the spline kernel defined as

$$K(\mathbf{l}, \mathbf{v}) := \prod_{i=1}^{m} k_i \tag{25}$$

where $\mathbf{l} = (l_1, l_2, \ldots, l_m)^{\mathrm{T}}$, $\mathbf{v} = (v_1, v_2, \ldots, v_m)^{\mathrm{T}}$, $\mathbf{k} = (k_1, k_2, \ldots, k_m)^{\mathrm{T}}$ and $k_i$ is computed as

$$k_i = 1 + l_i v_i + \frac{1}{2} l_i v_i \min(l_i, v_i) - \frac{1}{6} \min(l_i, v_i)^3 \tag{26}$$

where $m = 3$ in our application. The optimization process for both LS-SVM structures corresponds to obtaining the solution of the obtained system of equations. As emphasized earlier, in the resulting LS-SVM based predictors, almost every member of the training data set is kept as a support vector.

During the training phase, $c_i(t) \in \mathscr{C}$, and we obtain the results tabulated in Table 1, where it is clear that the smallest $\mathscr{T}_{\text{MSE}}$ value and the smallest $\mathscr{T}_{\Delta}$ values are obtained with the MLP structure. It is useful to note that MLP structure utilized in this study has 242 adjustable parameters whereas ANFIS structures have a total of 214 adjustable parameters. In terms of the number of adjustable parameters, MLP structure seems to be more complicated yet the computation time required for one forward pass through it is much smaller than that for ANFIS and this information is a highly useful evidence for those aiming at practicing these structures with real time hardware. Similarly, the MLP structure is structurally much simpler than the LS-SVM based emulators having 582 support vectors, which correspond to the entire training data set, in each. Although the data are the same for all experiments, the architectures are different and the hypergeometric features of the corresponding cost hypersurface for each approach is naturally different from each other. The results in Table 1 emphasizes that we encounter the best cost hypersurface, over which a convincingly small cost value, presumably the global minimum, can be detected very precisely with the MLP structure.

For the LS-SVM based emulators, the training procedure corresponds to the solution of a linear system of equations. Since this is not an iterative scheme, in Table 1, the checking error cell contains the term not applicable (n/a). Regarding the ANFIS and MLP structures, the training is stopped if the checking error denoted by $\mathscr{C}_{\text{MSE}}$ increases for five successive epoches. If no increase is observed, the training is continued approximately for 60 hours corresponding 1.5e+6 epochs. Since the approximators are tuned offline and the training is implemented only once, the training times are not considered as metrics of comparison. Having a different stopping criterion for each approach is another reason for omitting the training times, which continue almost 60 h particularly for ANFIS based models of noiseless or lightly noisy training data.

In order to see the performance of all three approaches in an ideal (noiseless) environment, we have chosen the trajectories shown in Fig. 9. The rationale behind this choice is to consider the full range of applicable inflow rates to see highest possible variations in the cell mass and the nutrient amount so that the prediction accuracies at different levels become visible. The validation error denoted by $\mathscr{V}_{\text{MSE}}$ (see (24)) is computed for each approach and the results are tabulated in Table 1. According to the obtained numerical values, the most promising predictions are obtained with the MLP approach, the state predictions ($\mathbf{x}(t)$) with which are almost indistinguishable from the desired process states denoted by $\mathbf{c}(t)$. Note that the number of patterns used during the training is $N_{\mathscr{T}} = 582$, which clearly indicate that the available information about the system is highly limited and the performance obtained under such a restriction is decent. According to Table 1, LS-SVM architectures have yielded the worst performance values in providing the derivatives of the state variables, and when integrated to obtain the model response, large errors are observed as can also be seen from the trajectories in Fig. 9.

**Table 1**
Comparison of the results with perfect observations.

| | MLP | ANFIS | LS-SVM |
|---|---|---|---|
| $T_{\Delta}$ (ms) | 0.1860 | 21.6062 | 369.4844 |
| $\mathscr{T}_{\text{MSE}}$ | 1.1352e−12 | 6.1312e−4 | 4.3303e−5 |
| $\mathscr{C}_{\text{MSE}}$ | 4.9249e−12 | 7.4743e−4 | n/a |
| $\mathscr{V}_{\text{MSE}}$ | 4.6976e−9 | 23.4826e−4 | 155.8974e−4 |

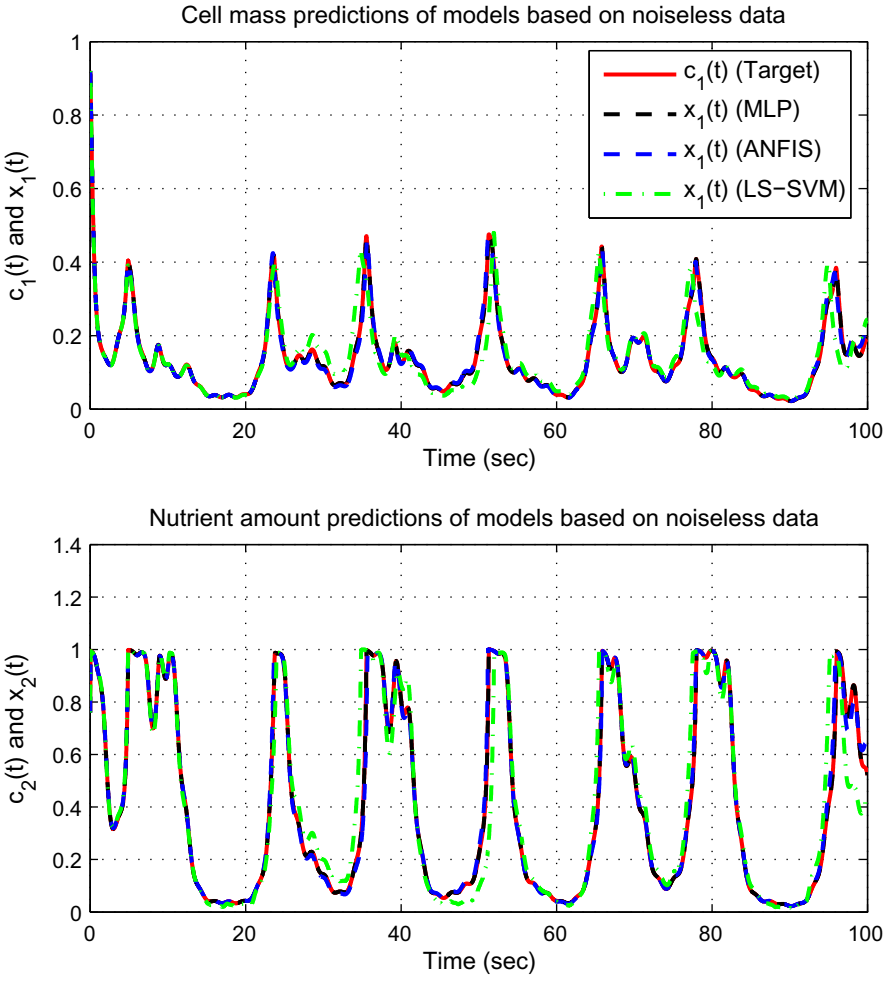


**Fig. 9.** The evolution of the process states for $c_1(0) = 0.92$, $c_2(0) = 0.76$, $T_f = 100$ s and $w(t) = 1 + 0.43\cos(2\pi 0.07t) + 0.23\sin(2\pi 0.17t) + 0.33\sin(2\pi 0.56t)$ and the predictions provided by each approach. The same legend for the top subplot is adopted for the second state variable ($c_2(t)$ and $x_2(t)$) in the bottom subplot.

One might ask whether the obtained results could be improved with alternative settings and selections, e.g. number of hidden layers, number of neurons in MLP, number of rules, type of membership functions in ANFIS, and the type of kernel and the type of cost function in LS-SVM. Based on the numerous tests towards the best performing approximator, the selections presented in the paper are the best ones observed throughout this research. The comparison considering the other qualities such as limit cycle prediction, effects of fast changing inputs and zero output conditions are elaborated in the next subsection.

### 6.3. Numerical data is noisy

In this part, we compare the three approaches in the following categories, and discuss the results including emulators obtained with noisy as well as the noiseless training data.

- Trajectory prediction, which is important in the cases where the process operator needs predictions of future performance if a preselected operating conditions are to be maintained.
- Limit cycle prediction, which is substantial as limit cycles can have dangerous consequences and precise positioning of them can let the process operator take precautions against getting trapped to an oscillatory regime.
- Prediction under high frequency excitations is the third category that has the goal of explaining the system behavior when fast changes in the system inputs are required. Particularly, a model giving reliable predictions would be a good guide for the process operator to command safely.
- Prediction of the zero output (steady state) conditions is the last category we focus on. A model predicting the steady state behavior accurately enables the process operator monitor any drifts safely before a possible instability arises.

#### 6.3.1. Trajectory prediction

Despite the superior performance of MLP structure with noiseless data (see the $\mathscr{V}_{\mathrm{MSE}}$ values in Table 1), the results seen in Fig. 9 stipulate that ANFIS based emulators have shown no visible deviation from the target values. For this reason, in the rest of the study, we eliminate the LS-SVM structure as the performance figures peculiar to it were poor compared to the others under both noiseless and noisy data, and present the comparison results focusing on MLP and ANFIS based structures for the remaining experiments. For this purpose, we have generated a set of noise sequences and additively corrupted the input and output patterns in the training data. In performing this, we have scaled the noise sequence to several percentages before corrupting the clean data in order to maintain the consistency of the experiments and to unfold how the performance is affected as the noise percentage is varied. First set of results are tabulated in Table 2, where we can compare $\mathscr{V}_{\mathrm{MSE}}$ values for the trajectories shown in Fig. 9 with the noiseless case, which is given in the first row of the table.

**Table 2**
Comparison of the results eith MLP and ANFIS structures operated under noisy data and noiseless data.

| Noise level (%) | $\mathscr{V}_{MSE}$ for MLP | $\mathscr{V}_{MSE}$ for ANFIS |
|---|---|---|
| 0 | 4.6976e−9 | 23.4826e−4 |
| 0.0001 | 0.0234e−4 | 23.4666e−4 |
| 0.001 | 0.1891e−4 | 23.3248e−4 |
| 0.01 | 0.9983e−4 | 22.0818e−4 |
| 0.1 | 33.0502e−4 | 36.2924e−4 |
| 1 | 333.6820e−4 | 59.9943e−4 |
| 10 | 836.8080e−4 | 1.4422e+12 |

Expectedly, $\mathscr{V}_{MSE}$ value increases as the noise level increases. This result practically stipulates that for each approach the global minimum of the noiseless case is perfectly hidden due to the addition of the noise signals. According to Table 2, we see that the performance of the MLP based emulator is smoothly degraded as the noise percentage increases, nevertheless, the values with small noise percentages are found to be tolerable. Regarding the ANFIS structures, except the results with 1% noisy data, the $\mathscr{V}_{MSE}$ values are higher than those for the MLP structure, which convincingly performs better than ANFIS. At 10% noise level, ANFIS approach does not yield a meaningful model whereas the MLP approach yields a model which performs poorly. The observations letting us build the results in Table 2 are illustrated in Figs. 10,11, where it is seen that the emulator responses are acceptable up to a certain level of noise and the performance of the emulators providing the derivatives is highly vulnerable to noise corrupting the training data.

*6.3.2. Limit cycle prediction*

Aside from the trajectories seen in Figs. 9–11 and Table 3, we compare the performance of the obtained emulators in predicting the limit cycle arising when $c_1 = 0.1331$ and $c_2 = 0.8226$ and $w = 0.829$ (see Section 2). The top row indicates the values obtained with the true system given in (1) and (2). Following three data rows present the results obtained with noiseless data set. Among the three emulators, closest values to the true ones are obtained with the MLP based structure. The third and fourth parts of the table list the values recorded through the use of noisy data sets. According to the table, MLP structures trained under varying levels of noise predict the limit cycle appropriately till the noise level reaches 0.01%. Beyond this value, the emulator still predicts a limit cycle yet the predicted locus is considerably away from the true one. This can be seen from the last three rows of the table. Regarding the ANFIS based structures, the predicted limit cycles are either visibly inside the true one or outside of it for all noise levels considered. The ANFIS model obtained with the 10% noisy training data, the emulator states converge a point (0.1082, 0.8498) indicating no limit cycle (No LC). On the other hand, the ANFIS models trained with very lightly corrupted training data, e.g. 0.001% and

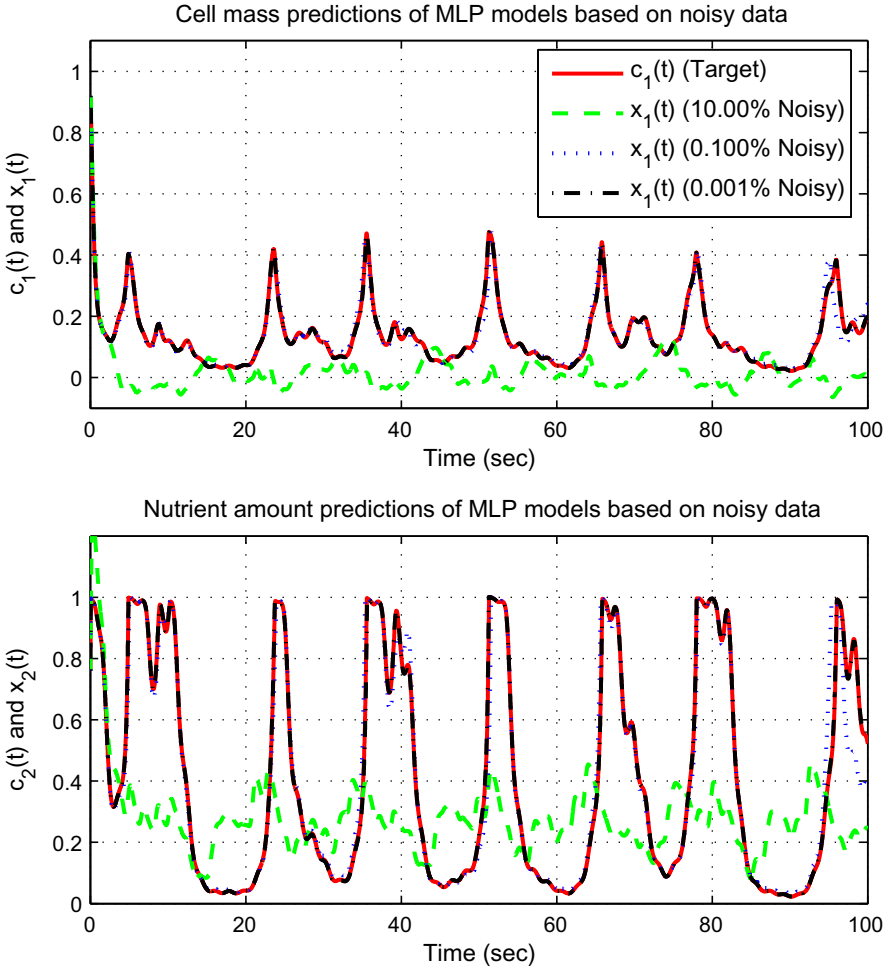


**Fig. 10.** The evolution of the process states for MLP, $c_1(0) = 0.92$, $c_2(0) = 0.76$, $T_f = 100$ s and $w(t) = 1 + 0.43\cos(2\pi\ 0.07t) + 0.23\sin(2\pi 0.17t) + 0.33\sin(2\pi 0.56t)$ and the predictions provided by each emulator trained with noisy data. The same legend for the top subplot is adopted for the second state variable ($c_2(t)$ and $x_2(t)$) in the bottom subplot.

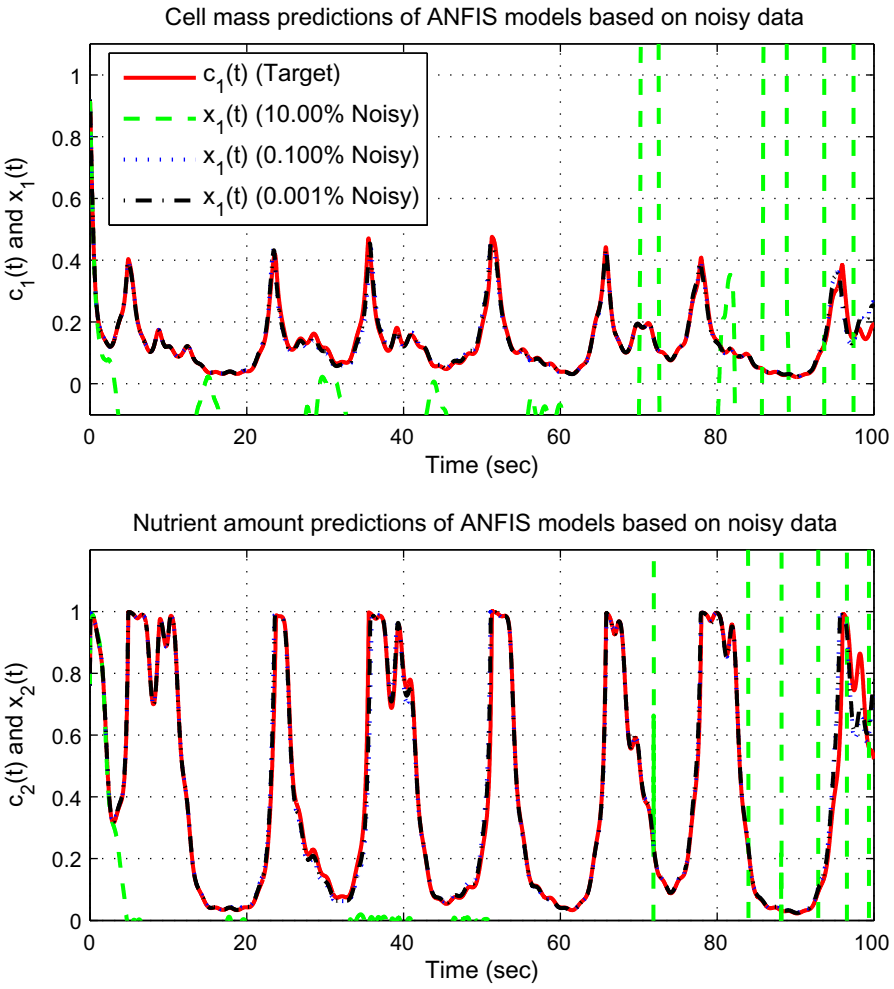


**Fig. 11.** The evolution of the process states for ANFIS, $c_1(0) = 0.92$, $c_2(0) = 0.76$, $T_f = 100$ s and $w(t) = 1 + 0.43\cos(2\pi 0.07t) + 0.23\sin(2\pi 0.17t) + 0.33\sin(2\pi 0.56t)$ and the predictions provided by each emulator trained with noisy data. The same legend for the top subplot is adopted for the second state variable ($c_2(t)$ and $x_2(t)$) in the bottom subplot.

**Table 3**
Performances in limit cycle prediction.

| Model | Period (s) | $\min c_1$ | $\max c_1$ | $\min c_2$ | $\max c_2$ | Noise (%) | $\mathscr{V}_{MSE}$ |
|---|---|---|---|---|---|---|---|
| True system | **3.59** | **0.1219** | **0.1466** | **0.8242** | **0.8996** | **0** | n/a |
| | | $\min x_1$ | $\max x_1$ | $\min x_2$ | $\max x_2$ | | |
| MLP | 3.59 | 0.1215 | 0.1473 | 0.8225 | 0.9012 | 0 | 9.0844e−11 |
| ANFIS | 3.58 | 0.1297 | 0.1367 | 0.8517 | 0.8732 | 0 | 6.1624e−4 |
| LS−SVM | 3.68 | 0.1307 | 0.1334 | 0.8684 | 0.8665 | 0 | 6.3679e−4 |
| MLP | 3.59 | 0.1215 | 0.1473 | 0.8225 | 0.9012 | 0.0001 | 3.6694e−7 |
| MLP | 3.60 | 0.1215 | 0.1473 | 0.8225 | 0.9011 | 0.001 | 1.6038e−5 |
| MLP | 3.60 | 0.1209 | 0.1481 | 0.8206 | 0.9033 | 0.01 | 6.8842e−5 |
| MLP | 3.59 | 0.1135 | 0.1661 | 0.7667 | 0.9325 | 0.1 | 43.4600e−4 |
| MLP | 3.97 | 0.1117 | 0.1835 | 0.7085 | 0.9560 | 1 | 70.3571e−4 |
| MLP | 4.70 | 0.0929 | 0.1679 | 0.7606 | 0.9412 | 10 | 37.4625e−4 |
| ANFIS | 3.58 | 0.1331288 | 0.1331291 | 0.8625447 | 0.8625455 | 0.0001 | 7.8957e−4 |
| ANFIS | 3.58 | 0.1331265 | 0.1331268 | 0.8625407 | 0.8625415 | 0.001 | 7.9067e−4 |
| ANFIS | 3.58 | 0.1298 | 0.1366 | 0.8520 | 0.8728 | 0.01 | 6.9397e−4 |
| ANFIS | 3.57 | 0.1294 | 0.1365 | 0.8512 | 0.8727 | 0.1 | 11.8912e−4 |
| ANFIS | 3.57 | 0.1170 | 0.1477 | 0.8137 | 0.9033 | 1 | 37.5421e−4 |
| ANFIS | No L.C. | 0.1082 | 0.1082 | 0.8498 | 0.8498 | 10 | n/a |

0.0001% cases, the trajectory converges a limit cycle which is so small that it can hardly be distinguished from a point in the state space, and these points are tabulated with higher numerical precision. In conjunction with the facts presented in Table 3 and Fig. 12, we illustrate the true limit cycle and the limit cycles predicted by MLP and ANFIS based structures trained under five different noise levels to visualize the degradation in the prediction performances. Clearly, the results tabulated in Table 3 and the trajectories seen in

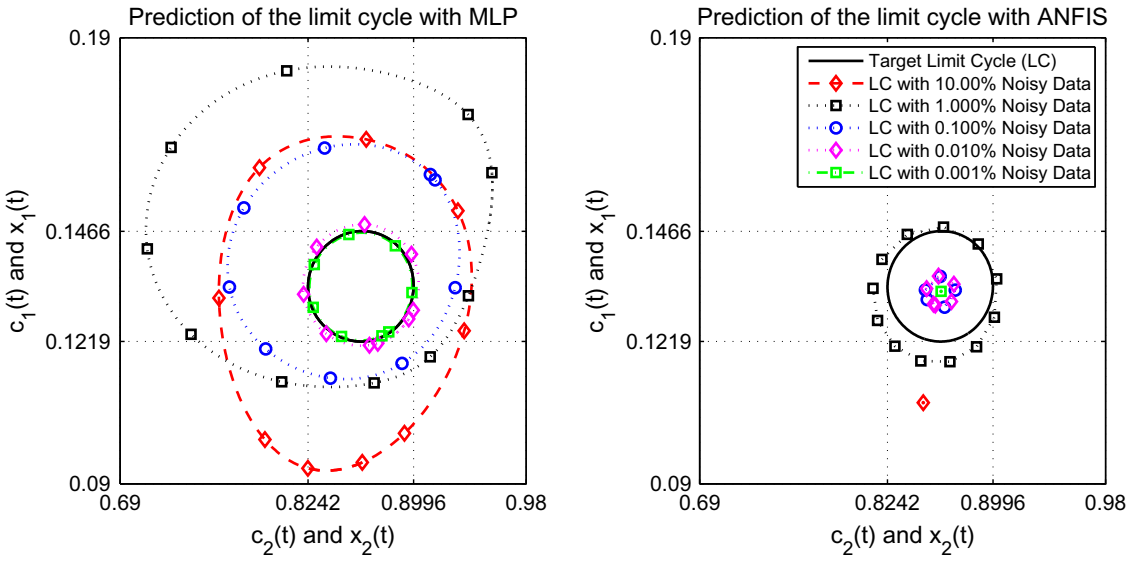


**Fig. 12.** A comparison of the predicted limit cycles with MLP structure (left) and ANFIS structure (right).

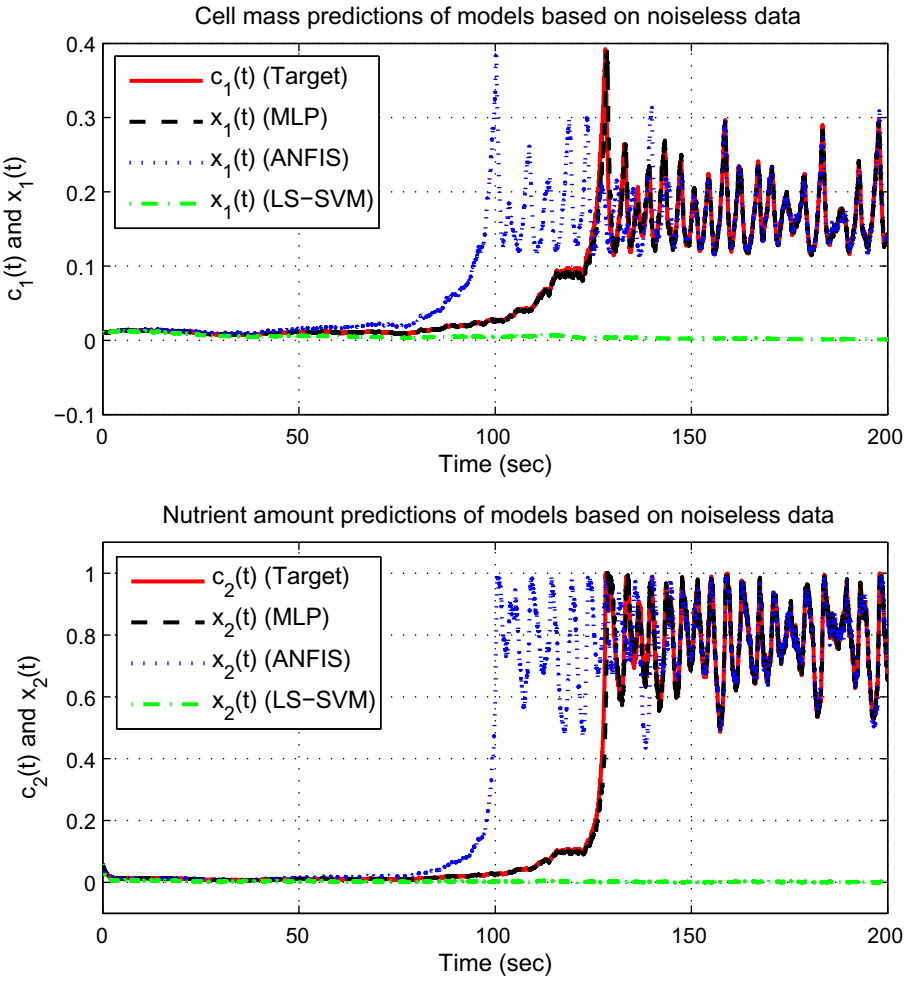


**Fig. 13.** Noise driven process response for $c_1(0) = 0.01$ and $c_2(0) = 0.06$. The training data used in generating the MLP, ANFIS and LS-SVM emulators are noiseless.

Fig. 12 emphasize that the emulator utilizing MLP has the capability of developing an almost true limit cycle at least in of the considered conditions and in another, a very close locus is detected.

*6.3.3. Prediction under high frequency excitations*

In this part, we compare the performance of the emulators when the inflow rate has high frequency components. In Fig. 13,

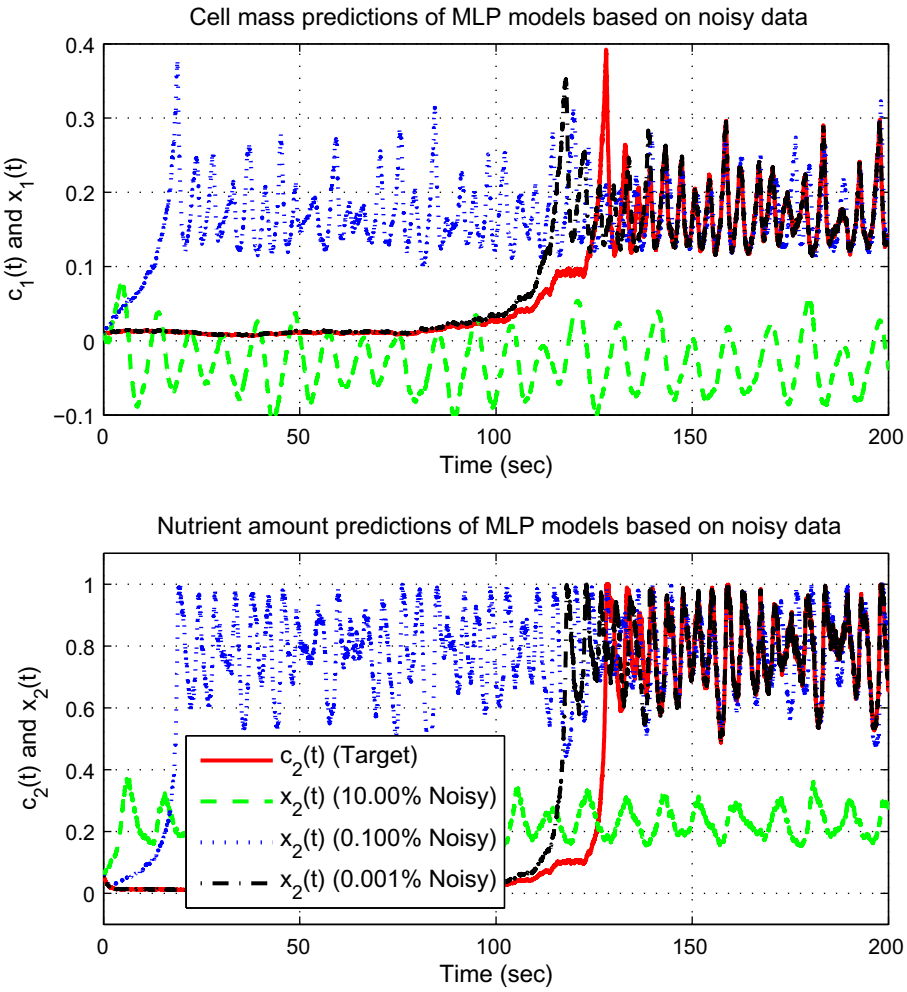


**Fig. 14.** Response of the MLP emulators driven by a noise sequence. The initial conditions are $c_1(0) = 0.01$ and $c_2(0) = 0.06$ and the training data used in generating the emulators are noisy.

we demonstrate the results with the noiseless data. The input, $w(t)$, to the emulator and to the true system is a noise sequence with mean and variance equal to unity.

Clearly, such a test is supposed to excite the high frequency components of the emulator dynamics in hand. According to the results shown in Fig. 13, LS-SVM fails totally whereas ANFIS predicts the increasing state values followed by an oscillatory trajectory to some extent. It is visible that the best predictions are provided again by the emulator built using MLP structure, whose predictions are accurately on the response of the true system.

In Figs. 14 and 15, we illustrate the results obtained with the MLP and ANFIS based emulators, whose predictors are trained with the noisy training data. For the clarity of the discussion, we consider the trajectories with three different noise levels. Looking at the two figures simultaneously, together with the $\mathscr{V}_{\mathrm{MSE}}$ values provided in Table 4, one sees that emulators utilizing MLP structure perform better than the ANFIS based ones in general. As the noise level is increased, the emulators start predicting the oscillatory regime earlier. Although for the 10% case, the tabulated value for the MLP seems to decrease, the response of the emulator is unacceptably away from the true response and no prediction of oscillations is observed in the later phases of the experiment. In brief, according to the tabulated results and the presented figures, this category also emphasizes the usefulness of emulators utilizing MLP structure in predicting the derivatives of the state variables.

*6.3.4. Prediction of the zero output (steady state) conditions*

Finally, we compare the reconstruction performance of the three approaches in predicting the steady state behavior. For this purpose, we apply the following input vector to each one of these emulators and concatenate the obtained results in Table 5

$$\mathbf{u}(x_2) = \begin{pmatrix} (1-x_2)e^{\frac{x_2}{7}} \\ \frac{x_2(1+\beta-x_2)}{1+\beta} \\ x_2 \end{pmatrix} \tag{27}$$

More explicitly, referring to the equilibrium plots in Fig. 4, when the states are on the parabolic curve shown in the figure, and when the steady state control input depicted in the left subplot together with the corresponding state values are applied, the emulator states should remain motionless. This situation can be simulated by applying the input vector in (27) to each one of the predictors. Practically, computing this input vector for a set obtained by running $x_2$ from zero to unity and evaluating the response of each approach would yield $\dot{x}_1$ and $\dot{x}_2$, which should remain close to zero. We have considered $\mathscr{N}_P = 10,000$ linearly distributed samples of $x_2 \in [0, 1]$, computed $\mathbf{f}(\mathbf{u}(x_2))$. Since $\mathbf{f}^{\star}(\mathbf{u}(c_2)) = \mathbf{0}$, we can evaluate a cost given by (28), where the squared approximation error value $\|\varepsilon_i\|^2$ is summed over the $\mathscr{N}_P$ samples lying on the parabola depicted in Fig. 4. The results of this computation are summarized in Table 5

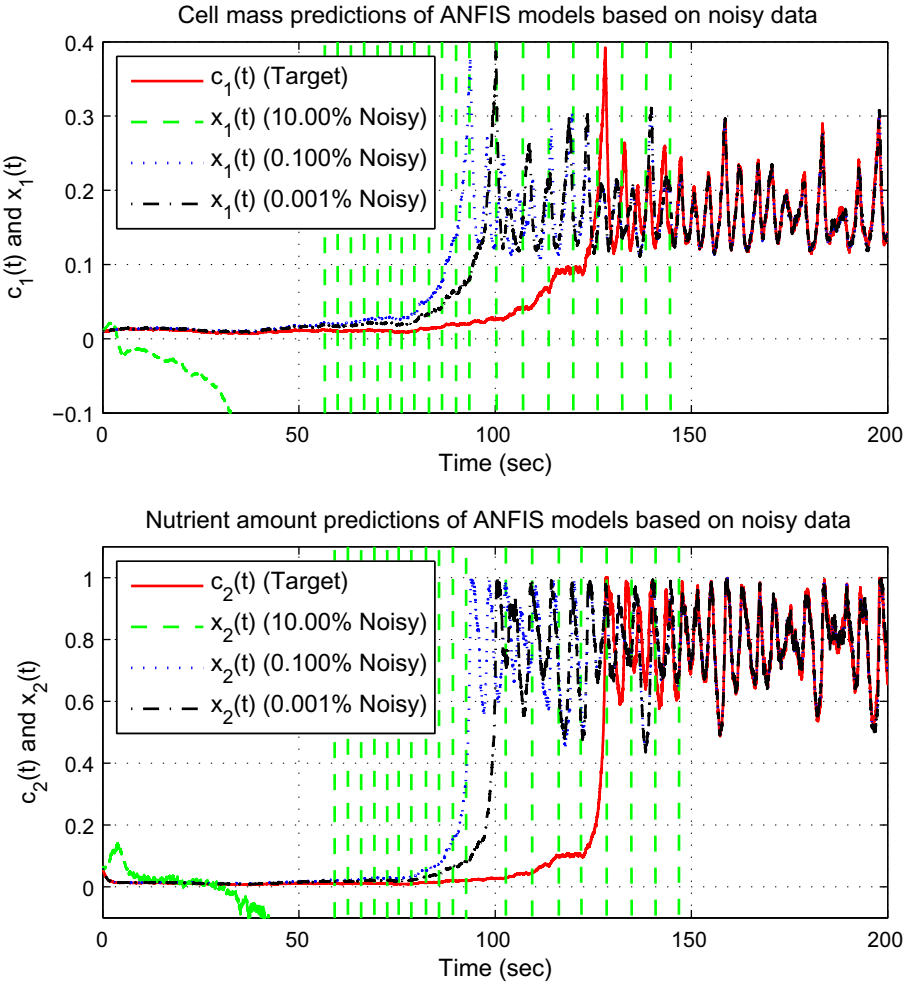


**Fig. 15.** Response of the ANFIS emulators driven by a noise sequence. The initial conditions are $c_1(0) = 0.01$ and $c_2(0) = 0.06$ and the training data used in generating the emulators are noisy.

$$\begin{aligned}\mathscr{P}_{\mathrm{MSE}} &= \frac{1}{\mathscr{N}_P}\sum_{i=1}^{\mathscr{N}_P}\|\mathbf{f}^{\star}(\mathbf{u}_i(c_2)) - \mathbf{f}(\mathbf{u}_i(x_2))\|^2 \\ &= \frac{1}{\mathscr{N}_P}\sum_{i=1}^{\mathscr{N}_P}\|\mathbf{f}(\mathbf{u}(x_2))\|^2 \\ &= \frac{1}{\mathscr{N}_P}\sum_{i=1}^{\mathscr{N}_P}\|\varepsilon_i\|^2 \end{aligned} \tag{28}$$

The value of $\mathscr{P}_{\mathrm{MSE}}$ for LS-SVM trained with the noiseless training data is 135284.3061e−4, which is found unacceptably large, and because of this, we arrange the table for a comparison of MLP and ANFIS based emulators.

According to the numerical results presented in Table 5, we infer that the MLP structure gives better predictions for the cases with lightly noisy training data experiments. A rough comparison of the results related to more noisy training data cases given in the last three rows of Table 5 may mislead the reader as they indicate ANFIS as better performing approach. However, the $\mathscr{P}_{\mathrm{MSE}}$ values in those rows are so high that an emulator yielding such values of $\mathscr{P}_{\mathrm{MSE}}$ is very likely to diverge from the steady state conditions quickly. At 0.01% noise level, ANFIS seems to perform better yet the $\mathscr{P}_{\mathrm{MSE}}$ values are reasonably close to each other. Therefore, the results associated to the MLP based emulator are more coherently those staying close to the steady state conditions.

It is important to point out that processing time or the propagation delay for every approach is a good indicator when considered

**Table 4**
Comparison of the results with MLP and ANFIS based emulators driven by noisy inflow rate, $w(t)$.

| Noise level (%) | $\mathscr{V}_{\mathrm{MSE}}$ for MLP | $\mathscr{V}_{\mathrm{MSE}}$ for ANFIS |
|---|---|---|
| 0 | 13.5214 | 1504.6656 |
| 0.0001 | 356.2774 | 1504.8376 |
| 0.001 | 508.0423 | 1505.3025 |
| 0.01 | 1501.9921 | 1583.2775 |
| 0.1 | 6583.5232 | 1894.8841 |
| 1 | 6154.2249 | 6859.5187 |
| 10 | 3303.1744 | 2.4494e+84 |

**Table 5**
Performances in reconstructing the steady state (zero output) conditions.

| Noise level (%) | $\mathscr{P}_{\mathrm{MSE}}$ for MLP | $\mathscr{P}_{\mathrm{MSE}}$ for ANFIS |
|---|---|---|
| 0 | 3.9197e−8 | 3.9875e−4 |
| 0.0001 | 0.1498e−4 | 3.9807e−4 |
| 0.001 | 1.0960e−4 | 3.9346e−4 |
| 0.01 | 25.2373e−4 | 4.8484e−4 |
| 0.1 | 3329.2052e−4 | 143.7285e−4 |
| 1 | 59581.0290e−4 | 16367.2135e−4 |
| 10 | 1098345.9237e−4 | 1628358.9758e−4 |

as a candidate for a real time application. For the methods studied here, presented MLP configuration has the processing time of 0.29 ms. The value for ANFIS is equal to 12.6 ms and that for LS-SVM is equal to 220.7 ms. The values have been obtained under Matlab/Simulink environment that runs on a Pentium IV platform. According to the observations, MLP structure outperforms the other alternatives also in terms of the processing time. LS-SVM displays a long processing time making it a candidate only for applications having costly hardware.

### 6.4. An overall assessment of the obtained results

In this section, several features and responses of the bioreactor benchmark problem are studied. These features include the prediction of a given trajectory spanning the whole range of the state and the input variables, the prediction of a limit cycle and the match of the responses under high frequency excitations and the capturing of the steady state conditions. Each one of these items have carefully been studied and the performance obtained in each one of them is quantified for comparison considering the clean and noisy training data.

Since the modeling process is subject to very limited number of training data pairs, and the desired form of the model is an emulator identifying the plant in parallel mode, approximation error in the learning stage gains a crucial importance as it determines the temporal accuracy of the mapping.

Three well known architectures are chosen for comparison, namely, MLP, ANFIS and LS-SVM are considered. This is deliberate as the standing point in the emergence of each one of these approaches gives reference to the others with an emphasis of the advantage introduced by each specific approach. Historically, since MLP architecture is the one discovered first, the fuzzy information processing and support vector optimization have been postulated as alternatives. However, the findings in this paper stipulate the outstanding capabilities of the MLP structure. That is to say, the operational simplicity, accuracy and robustness against noise in the training data are the most prominent features that could not be observed with its rivals originated from different philosophic viewpoints. The MLP structure successfully predicts the system behavior with noiseless training data and retains its merits to a certain amount of noise corruption in the training data set. Showing such good figures of performance under the presence of very limited number of training data is a remarkable feature that is persistently maintained by MLP structure.

## 7. Concluding remarks

Chronologically, the research on intelligent systems has started with neural models of brain activity and the approaches like fuzzy logic and support vector machines have come into the picture with the possibilities of processing linguistic descriptions for the former and optimization theory for the latter. This paper unfolds the outstanding performance of MLP structure, which is the oldest one of these approaches, in parallel mode identification of a nonlinear benchmark system. This aspect of the current paper is its major contribution standing on a comparison of modeling a biochemical process.

Due to a rich set of responses emerging under different operating conditions, the bioreactor benchmark problem is a prime example to study the effectiveness of identification tools exploiting numerical data and displaying some degrees of autonomy and intelligence. This paper considers three approaches and their standard tuning schemes to develop an emulator for the process. The parallel mode identification is considered with noiseless and noisy training data. The results have shown that the emulators based on the MLP structure are able to provide accurate predictions for a diverse set of operating conditions. Although we make no claim that the deduced results will necessarily extend to other problems, it is seen that the studied problem is a good test bed for distinguishing the learning and generalization performances of the considered MLP, ANFIS and LS-SVM structures, and MLP approach is found to be the most powerful alternative among its rivals.

Despite the numerous different configurations and parameter settings tested for each approach, it is observed that the MLP structure with the given configuration is able to achieve the global minimum of the cost function within the limits permitted by the raw - and possibly noisy-information. When the issues of noise and the limited training data are taken into consideration, it becomes clear that the MLP structure displays the most desired characteristics. As opposed to the results from the experiments exploiting the ANFIS structures, another remarkable property that is inferred from the experiments carried out is that the performance is degraded smoothly in MLP based emulators as the noise percentage is increased smoothly.

In brief, this paper compares three widely used approaches with associated tuning or optimization schemes. According to the results obtained, the MLP structure is found to be the most successful one in representing the dynamic properties available in the system under investigation, e.g. predicting the limit cycle, predicting trajectories, predicting the behavior at the steady state and predicting the behavior with inputs having fast fluctuations. All such experiments with noiseless and noisy training data sets point the use of the MLP structure for the benchmark problem considered here.

## Acknowledgment

The author gratefully acknowledges the facilities of TOBB ETÜ library.

## References

[1] Abdollahi F, Talebi HA, Patel RV. Stable identification of nonlinear systems using neural networks: theory and experiments. IEEE/ASME Trans Mech 2006;11(4):488–95.
[2] Anderson CW, Miller III WT. Challenging control problems. In: Miller III WT, Sutton RS, Werbos PJ, editors. Neural networks for control. MIT Press; 1990. p. 475–510.
[3] Battiti R. First-second-order methods for learning: between steepest descent and Newtons method. Neural Comput 1992;4:141–66.
[4] Brengel DD, Seider WD. Multistep nonlinear predictive controller. Ind Eng Chem Res 1989;28:1812–22.
[5] Chen N, Lu W, Yang J, Li G. Support vector machine in chemistry. World Scientific; 2004.
[6] Cristianini N, Shawe-Taylor J. An introduction to support vector machines and other Kernel-based learning methods. Cambridge University Press; 2000.
[7] Efe MÖ. MIMO variable structure controller design for a bioreactor benchmark process. ISA Trans 2007;46(4):459–69.
[8] Efe MÖ, Abadoglu E, Kaynak O. A novel analysis and design of a neural network assisted nonlinear controller for a bioreactor. Int J Robust Nonlin Contr 1999;9(11):799–815.
[9] Efe MÖ, Kaynak O. A comparative study of soft computing methodologies in identification of robotic manipulators. Robot Auton Syst 2000;30(3):221–30.
[10] Espinoza M, Suykens JAK, De Moor B. Kernel based partially linear models and system identification. IEEE Trans Autom Contr 2005;50(10):1602–6.
[11] Gunn SR. Support vector machines for classification and regression. ISIS technical report. United Kingdom: University of Southampton; 1998.
[12] Hagan MT, Menhaj MB. Training feedforward networks with the Marquardt algorithm. IEEE Trans Neural Networks 1994;5(6):989–93.
[13] Haykin S. Neural networks. New Jersey: Macmillan College Printing Company; 1994.
[14] Hou Z, Shen Q, Li H. Nonlinear system identification based on ANFIS. In: Proceedings of the 2003 international conference on neural networks and signal processing, China, December 14–17, 2003. p. 510–2.
[15] İplikçi S. Support vector machines based neuro-fuzzy control of nonlinear systems. Neurocomputing 2010;73(10-12):2097–107.
[16] Jang J-SR, Sun C-T, Mizutani E. Neuro-fuzzy and soft computing. PTR Prentice-Hall; 1997.
[17] Jang J-SR. ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cyb 1993;23(3):665–85.
[18] Khalil H. Nonlinear systems. New Jersey: Prentice-Hall; 2001.

[19] Ljung L. System identification: theory for the user. PTR Prentice-Hall; 1998.
[20] Lo JT, Bassu D. Adaptive parallel identification of dynamical systems by adaptive recurrent neural networks. In: Proceedings of the international joint conference on neural networks (IJCNN'03), Portland, Oregon, vol. 2, July 20–23, 2003. p. 914–8.
[21] Narendra KS, Parthasarathy K. Identification and control of dynamical systems using neural networks. IEEE Trans Neural Networks 1990;1(1):4–27.
[22] Patra JC, Kot AC. Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks. IEEE Trans Syst Man Cyb B 2002;32(4):505–11.
[23] Puskorius GV, Feldkamp LA. Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks. IEEE Trans Neural Networks 1994;5(2):279–97.
[24] Refaat S, Nahavandi S. Nonlinear identification of pneumatic servo-drive. Int J Model Simulat 2006;26(1):11–6.
[25] Ren XM, Rad AB, Chan PT, Lo WL. Identification and control of continuous-time nonlinear systems via dynamic neural networks. IEEE Trans Ind Electron 2003;50(3):478–86.
[26] Savran A. An adaptive recurrent fuzzy system for nonlinear identification. Appl Soft Comput 2007;7(2):593–600.
[27] Slotine J-JE, Li W. Applied nonlinear control. New Jersey: Prentice-Hall; 1991.
[28] Suykens JAK, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J. Least squares support vector machines. Singapore: World Scientific Publishing; 2002.
[29] Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cyb 1985;15(1):116–32.
[30] Tokat S. Sliding mode controlled bioreactor using a time-varying sliding surface. Trans Inst Measur Contr 2009;31(5):435–56.
[31] Ungar LH. A bioreactor benchmark for adaptive-network based process control. In: Miller III WT, Sutton RS, Werbos PJ, editors. Neural networks for control. MIT Press; 1990. p. 387–402.
[32] Vapnik. The nature of statistical learning theory. New York: Springer-Verlag; 1995.
[33] Wang X-D, Ye M-Y. Nonlinear dynamic system identification using least squares support vector machine regression. In: Proceedings of the 3rd international conference on machine learning and cybernetics, Shanghai, August 26–29, 2004. p. 941–5.
[34] Wang L-X. Adaptive fuzzy systems and control, design and stability analysis. PTR Prentice-Hall; 1994.
[35] Wang L-X. A course in fuzzy systems and control. PTR Prentice-Hall; 1997.
[36] Wang J-S, Chen Y-P. A fully automated recurrent neural network for unknown dynamic system identification and control. IEEE Trans Circuits Syst I 2006;53(6):1363–72.
[37] Yu W, Li X. Discrete-time neuro identification without robust modification. IEE Proc 2003;150(3):311–6.
[38] Zhang HR, Wang XD, Zhang CJ, Cai XS. Robust identification of non-linear dynamic systems using support vector machine. IEE Proc 2006;153(3):125–9.
[39] Zou Z, Yu D, Feng W, Yu L, Guo N. An intelligent neural networks system for adaptive learning and prediction of a bioreactor benchmark process. Chin J Chem Eng 2008;16(1):62–6.