

# A Comparison of ANFIS, MLP and SVM in Identification of Chemical Processes

Mehmet Önder Efe

**Abstract**—This paper presents a comparison of Adaptive Neuro Fuzzy Inference Systems (ANFIS), Multilayer Perceptron (MLP) and Support Vector Machines (SVMs) in identification of a chemical process displaying a rich set of dynamical responses under different operating conditions. The methods considered are selected carefully as they are the foremost approaches exploiting the linguistic representations in ANFIS, connectionist representations in MLP and machine learning based on structural risk minimization in SVM. The comparison metrics are the computational complexity measured by the propagation delay, realization performance and design simplicity. It is seen that SVM algorithm performs better in terms of providing an accurate fit to the desired dynamics but a very close performance result can also be obtained with ANFIS with significantly lower computational cost. Performance with MLP is comparably lower than the other two algorithms yet MLP structure has the lowest computational complexity.

## I. INTRODUCTION

Choosing the best approximator for a given experimental data is an active research field in engineering, with the name *identification*. Although the linear theory offers many tools and approaches (See [1]), problems involving the investigation of nonlinear systems have motivated the scientists to develop alternative forms of representing the knowledge compactly, (See [2], [3]). One major goal in all cases was to maintain the generality and one of the well known strategies is Adaptive Neuro Fuzzy Inference Systems exploiting the power of verbal descriptions through membership functions and inference engines equipped with a rulebase. Another is Multilayer Perceptron structure of artificial neural networks (NNs), which store the desired form of knowledge within a massively interconnected layered structure. The third is a Support Vector Machine exploiting the structural risk minimization principle instead of empirical risk minimization as in the cases of ANFIS and MLP. This new approach enables the minimization of the upper bound of a risk function thereby enabling the achievement of optimum regression functions.

This paper is organized as follows. The second section introduces the bioreactor benchmark problem and analyzes its behavior. The following three sections present the SVM, MLP and ANFIS approaches in turn. Operating conditions, results and a discussion are presented in the sixth section. The concluding remarks are given at the end of this paper.

## II. BIOREACTOR BENCHMARK PROBLEM

The bioreactor is a tank in which the biological cells are mixed with nutrients and water. 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  $w(t)$ . In order to maintain the reaction volume constant, the chemical content of the tank is removed at a rate equal to the inflow rate.

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

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

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

In Fig. 1, 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 while new attractors emerge, e.g. for  $w = 0$  no attractors are observed, for  $w = 1$  a limit cycle which is common to all starting points occurs, and for  $w = 2$  a single attractor is created. When  $w = 0.8290$ , the eigenvalues of the linearized model are approximately equal to  $0 \pm j1.7543$ , i.e. the system undergoes Hopf bifurcation and displays 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 0.8243 and 0.8996.

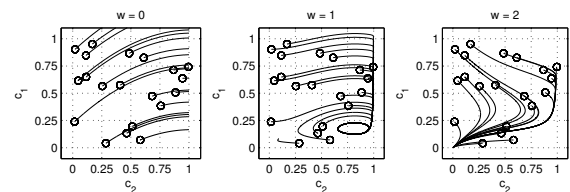


Fig. 1. The evolution of the process state for different initial conditions and at different inflow rates. The trajectories are for 6 seconds time.

M.Ö. Efe is with TOBB Economics and Technology University, Department of Electrical and Electronics Engineering, TR-06560, Söğütözü Cad., No 43, Ankara, Turkey. onderefe@etu.edu.tr

In [4], Ungar 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 highlighted also in [5], [6], [7] with a similar motivation. Due to the presented properties of the system, the model constitutes a good candidate for scrutinizing the merits and effectiveness of networked modeling techniques revealing a set of flexible nonlinearities in various forms.

### III. A SUMMARY OF SVMs FOR APPROXIMATION

Due to powerful regression and classification capabilities based on numerical observations, the use of SVMs is a remedy among other alternatives such as fuzzy logic, 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, [8]. A common property of MLP, ANFIS and SVMs is to map the input vector to a feature space, where the regression can be performed much efficiently. Yet the major difference in SVM based learning is the minimization of an upper bound of a cost function instead of minimizing the cost itself. The goal of the former is to achieve an optimum value of the structural risk function whereas the latter terminates at the minimum of the empirical risk function. This practically makes it possible to generalize the information contained implicitly in the training set in some sense of optimality, [9]. Consider the regression problem over the pairs  $\mathcal{T} = \{(\mathbf{u}_1, d_1), \dots, (\mathbf{u}_{N_T}, d_{N_T})\}$ ,  $\mathbf{u}_i \in \mathbb{R}^m$ ,  $d_i \in \mathbb{R}$  with a function

$$f(\mathbf{u}) = \langle \mathbf{w}, \mathbf{u} \rangle + b \quad (3)$$

where  $\mathbf{w}$  and  $b$  denote the weight vector and the bias value, respectively. In above,  $\langle \cdot, \cdot \rangle$  stands for an appropriately defined operator, which is an inner product for linear regression and a kernel for nonlinear regression. Defining a quadratic loss function as in (4) quantifies the performance for the  $i$ th pair,

$$L(d_i, f(\mathbf{u}_i)) = (d_i - f(\mathbf{u}_i))^2. \quad (4)$$

and minimizing the structural risk given by (5) lets us obtain the best values of  $\mathbf{w}_i$ s causing least complexity represented by  $\|\mathbf{w}\|^2$ ;

$$R = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N_T} L(d_i, f(\mathbf{u}_i))^2, \quad (5)$$

where  $C$  is a parameter determining the relative importance of the terms contributing to  $R$ , [9]. The primal form of the optimization problem can be expressed compactly as

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N_T} (\xi_i^2 + \hat{\xi}_i^2) \text{ such that } \begin{cases} f(\mathbf{u}_j) - d_j \leq \xi_j \\ d_j - f(\mathbf{u}_j) \leq \hat{\xi}_j \\ \xi_j, \hat{\xi}_j \geq 0 \end{cases} \quad (6)$$

where  $j = 1, 2, \dots, N_T$ ,  $\xi_j$  and  $\hat{\xi}_j$  are slack variables penalizing the deviations from the target output. The problem described above can be converted into a convex quadratic

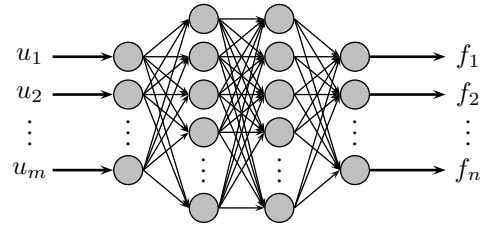


Fig. 2. Structure of a MLP

optimization problem by writing the dual representation (See [3] for details). The solution can be obtained by introducing the Lagrange multipliers ( $\lambda$ ) and performing the following minimization for  $\lambda \in \mathbb{R}^{N_T}$ ;

$$\min_{\lambda} \frac{1}{2} \sum_{i=1}^{N_T} \sum_{j=1}^{N_T} \lambda_i \lambda_j \langle \mathbf{u}_i, \mathbf{u}_j \rangle - \sum_{i=1}^{N_T} \lambda_i d_i + \frac{1}{2C} \sum_{i=1}^{N_T} \lambda_i^2, \quad (7)$$

with constraint  $\sum_{i=1}^{N_T} \lambda_i = 0$ . It should be noted that the support vectors are the  $\mathbf{u}_i$ s for which the corresponding  $\lambda_i$  is nonzero. The result of the minimization lets us obtain

$$\mathbf{w}^* = \sum_{i=1}^{N_T} \lambda_i \mathbf{u}_i \quad (8a)$$

$$b^* = \frac{1}{N_T} \sum_{i=1}^{N_T} \left( y_i - \sum_{j=1}^{N_T} \langle \mathbf{u}_i, \mathbf{u}_j \rangle \lambda_j \right) \quad (8b)$$

which are to be used in (3). The nonlinear regression problem is to replace the operator  $\langle \cdot, \cdot \rangle$  in (3) with a kernel function satisfying the Mercer conditions, [3]. In this paper, we utilize the spline kernel defined as  $\langle \mathbf{u}, \mathbf{v} \rangle := \prod_{i=1}^m k_i$ , where  $\mathbf{u} = (u_1, u_2, \dots, u_m)^T$ ,  $\mathbf{v} = (v_1, v_2, \dots, v_m)^T$ ,  $\mathbf{k} = (k_1, k_2, \dots, k_m)^T$  and  $k_i$  is computed as  $k_i = 1 + u_i v_i + \frac{1}{2} u_i v_i \min(u_i, v_i) - \frac{1}{6} \min(u_i, v_i)^3$ .

### IV. A SUMMARY OF MULTILAYER PERCEPTRON

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 as shown in Fig. 2. 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 the MLP has many internal connections, called synaptic connections, which possess synaptic strengths. In other words, the biologically resistive nature of such connections with a saturating response of a neuron is simply modeled as  $h = \Phi(s)$ , where  $s$  denotes the weighted sum of incoming signals and  $\Phi(\cdot)$  stands for the neuronal activation function that is responsible for the saturated response for the large values of the absolute value of its argument. Clearly, the number of terms involved in

computing the value of  $s$  for each neuron indicates the possible architectural redundancy within which the desired form of knowledge (mapping) is stored. Compactly, denote the number of layers excluding the input layer by  $K$  and the output vector of the  $i$ th layer by  $h_i$ , where  $h_i = \Phi(s_i)$  and  $s_i$  is the vector of net sums computed as

$$s_i = W_i h_{i-1} + B_i, \quad i = 1, 2, \dots, K \quad (9)$$

where  $W_i$  and  $B_i$  correspond to the weight and bias terms of the  $i$ th layer. It is clear that for a NN structure with single hidden layer 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

$$h_0 = \mathbf{u} \quad (10a)$$

$$s_1 = W_1 h_0 + B_1 \quad (10b)$$

$$h_1 = \tanh(s_1) \quad (10c)$$

$$s_2 = W_2 h_1 + B_2 \quad (10d)$$

$$\mathbf{f} = s_2 \quad (10e)$$

where the weight and bias terms seen above have appropriate dimensions and the input-output relation would simply be  $\mathbf{f} = W_2 \tanh(W_1 \mathbf{u} + B_1) + B_2$ . Once the structure is fixed, 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 and accurate results. 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 [10], [11]). Since the problem considered here requires 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  $\sigma$ , which is a  $P \times 1$  vector. At discrete time  $k$ , the cost function, which is an empirical risk function, qualifying the performance of the interpolation can be computed as in (11) and the LM update is formulated in (12).

$$E(\sigma_k) = \frac{1}{2} \sum_{i=1}^{N_T} \sum_{j=1}^n (d_{ij} - f_j(\mathbf{u}_i, \sigma_k))^2 \quad (11)$$

$$\sigma_{k+1} = \sigma_k - (\alpha I + \nabla_{\sigma_k}^2 E(\sigma_k))^{-1} \nabla_{\sigma_k} E(\sigma_k) \quad (12)$$

where  $\alpha > 0$  is a user-defined scalar design parameter and  $I$  is the identity matrix. It is important to note that, for large  $\alpha$ , (12) becomes Gauss-Newton method, and for small  $\alpha$ , the tuning law becomes the standard Gradient Descent (GD). Therefore, LM method establishes a good balance between GD and Gauss-Newton methods.

The pioneering work of [12] considers the MLP structure with the GD algorithm for identification and control purposes through some abstract yet descriptive examples with an

emphasis on slow convergence of the GD technique which has later been resolved by the use of LM technique.

## V. A SUMMARY OF 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 denoted by  $\mu$ , 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. In a fuzzy system structure, 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 inference engine, 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 (if any) are stored in the knowledge base, and a defuzzifier maps the input vector  $\mathbf{u}$  to a scalar output value, which is crisp.

Specifically, defining  $w_i$ ,  $\tilde{w}_i$  as the firing strength and the normalized firing strength of  $i$ th rule, respectively, the input output relation of the ANFIS structure with the rulebase structure containing  $R$  rules as given below, can be expressed as in (13a), where product inference and first order Sugeno type defuzzifier are the standard settings of ANFIS architecture (See [2], [13]). 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  $n_u$ ,  $n_v$  and  $n_w$  correspond to the number of linguistic labels for the first, second and  $m$ th input variables respectively.

$$\text{IF } u_1 \text{ is } \mathbb{U}_i \ \& \ u_2 \text{ is } \mathbb{V}_i \ \& \ \dots \ \& \ u_m \text{ is } \mathbb{W}_i \ \text{THEN } y_i = z_i$$

$$w_i = \prod_{j=1}^m \mu_{i,j}(u_j) \quad (13a)$$

$$\tilde{w}_i = \frac{w_i}{\sum_{k=1}^R w_k} \quad (13b)$$

$$z_i = b_i + \sum_{j=1}^m \phi_{i,j} u_j \quad (13c)$$

$$f = \sum_{i=1}^R \tilde{w}_i z_i \quad (13d)$$

In (13a),  $u_j$  corresponds to the  $j$ th entry of the input vector  $\mathbf{u}$ . The training is achieved by adopting a hybrid tuning mechanism. Specifically,  $b_i$  and  $\phi_{i,j}$  are adjusted by Least Mean Squares (LMS) algorithm, while the other parameters are tuned by GD method. It is emphasized in [2], [14] that such a tuning scheme reduces the dimensionality of the search space of GD algorithm and speeds up the convergence.

## VI. OPERATING CONDITIONS, RESULTS AND DISCUSSION

Let the output of the model and the plant be  $\mathbf{x}(t) = (x_1(t) \ x_2(t))^T$  and  $\mathbf{c}(t) = (c_1(t) \ c_2(t))^T$ , respectively. In the series-parallel connectivity, the emulator receives the plant state,  $\mathbf{c}(t)$ , as well as the external input  $w(t)$ . The response of the model is the response after a sampling period, say  $\Delta t$ , i.e.

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{c}(t), w(t)) = \mathbf{c}(t + \Delta t) + \varepsilon(t) \quad (14)$$

where  $\varepsilon(t)$  is the approximation error and the input vector of the emulator is  $\mathbf{u}(t) := (w(t) \ \mathbf{c}(t)^T)^T$ . The emulator is forced to learn the map  $\mathbf{x}(t) = \mathbf{c}(t + \Delta t)$  yet the approximation error denoted by  $\varepsilon(t)$  determines the accuracy of realization.

In the training stage, randomly generated samples are utilized. As suggested by [4], the derivative term in the continuous time plant dynamics is discretized with first order Euler approximation with a discretization interval of 0.01 sec. Three sets of data are generated from the interval  $(w_k, c_{1k}, c_{2k}) \in [0, 2] \times [0, 1] \times [0, 1]$  and the target values  $(c_{1k+1}, c_{2k+1})$  are computed using (1)-(2).

- Training Data ( $\mathcal{T}$ ): 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 is  $N_{\mathcal{T}} = 1002$ .
- Checking Data ( $\mathcal{C}$ ): This data set is used whenever the training algorithm enables terminating the training procedure at the best level of generalization. The number of pairs contained is  $N_{\mathcal{C}} = 200$ .
- Validation Data ( $\mathcal{V}$ ): The information used to visualize after the training stops. The number of pairs contained is  $N_{\mathcal{V}}$ . The samples in the set  $\mathcal{V}$  are those constructing the trajectories shown in Figure 1. Although the samples in  $\mathcal{C}$  are enough to consider, we elaborate trajectories that contain continuous evolution to see to what extent the emulator performs.

One important point in the comparison work we carry out here is the assumption that the model in (1)-(2) is nominal and the data are noiseless. Since the generalization ability of each approach is assessed by utilizing a checking data set, working with noise free data sets do not constitute a difficulty for drawing performance conclusions. This is a deliberate choice to unfold the performance of the widely used approximators under ideal conditions. Due to the space limit, we do not consider the issues of noise in the data, which would of course entail a thorough analysis of responses under different signal to noise ratios.

The results of the performed experiments are summarized in Tables I-III, where  $T_T$  denotes the total training time in seconds,  $T_{\Delta}$  stands for the propagation delay in milliseconds,  $\mathcal{T}_{\text{MSE}}$  corresponds to training MSE level,  $\mathcal{C}_{\text{MSE}}$  corresponds to checking MSE level, and  $\mathcal{V}_{\text{MSE}}$  stands for the performance index given by

$$\mathcal{V}_{\text{MSE}} := \frac{1}{N} \sum_{p=1}^N \sum_{i=1}^2 \frac{1}{T_f} \int_0^{T_f} (c_i^p(t) - x_i^p(t))^2 dt \quad (15)$$

In above, a superscript  $p$  stands for indexing the individual trials and  $T_f$  denotes the final time of the simulation for a single trial.

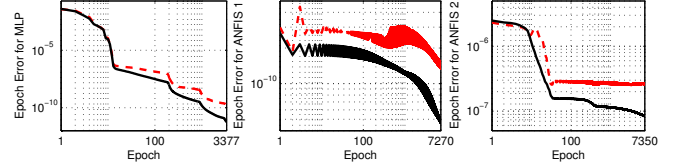


Fig. 3. MSE plots for MLP and ANFIS structures.

The training of SVM structures is continued until the optimal solution is found. The training of the MLP structure is terminated when the error with the checking data increases for 5 successive epochs. For the ANFIS structure, we have observed that the checking error is decreasing with the training error and decided to stop the training after a predetermined period of time, which is kept close to that recorded in the training of the SVM structure. By this means, we aim to compare the algorithms whose training last almost equal periods of time. The time evolution of MSE values computed for training data (solid) and checking data (dashed) are shown in Fig. 3. The final values of training times are tabulated in the first rows of Table I-III.

TABLE I  
RESULTS OBTAINED WITH ANFIS

	ANFIS <sub>1</sub> and ANFIS <sub>2</sub>
$T_T$	2369.7 and 2370.5
$T_{\Delta}$	3.6796
$\mathcal{T}_{\text{MSE}}$	4.6602e-11 and 8.3445e-8
$\mathcal{C}_{\text{MSE}}$	1.3432e-10 and 2.5312e-11
$\mathcal{V}_{\text{MSE}}$	1.1402e-7

TABLE II  
RESULTS OBTAINED WITH SVM

	SVM <sub>1</sub> and SVM <sub>2</sub>
$T_T$	2400.1 and 2341.4
$T_{\Delta}$	244.6677
$\mathcal{T}_{\text{MSE}}$	1.0078e-19 and 3.9455e-16
$\mathcal{C}_{\text{MSE}}$	3.0235e-11 and 4.5366e-11
$\mathcal{V}_{\text{MSE}}$	1.1376e-7

According to the tables, one sees that MLP realizes the dynamics in (1)-(2) within a single structure yet ANFIS and SVM approaches individually realize the state predictions as they have single output.

TABLE III  
RESULTS OBTAINED WITH MLP

	MLP
$T_T$	3843.8
$T_\Delta$	0.0309
$\mathcal{T}_{\text{MSE}}$	5.7276e-12
$\mathcal{C}_{\text{MSE}}$	2.0507e-10
$\mathcal{V}_{\text{MSE}}$	6.8370e-6

Triangular membership functions are utilized with ANFIS architectures, two linguistic labels are utilized for  $w(t)$  and  $c_1(t)$  while 5 labels are reserved for  $c_2(t)$  in both ANFIS structures as in (13a). First order Sugeno type defuzzification is exploited and a hybrid learning strategy is utilized. The membership functions are tuned with GD algorithm while the coefficients of the defuzzifier polynomials are tuned with Least Mean Squares (LMS) algorithm (See [2] and [13] for details).

Regarding the SVM structures, with a quadratic loss function and spline kernel as discussed in Section 3, we have obtained the structural complexity values  $\|\mathbf{w}_1\|^2 = 1.015475$  and  $\|\mathbf{w}_2\|^2 = 34.380741$ . Besides, it is observed that the entire set of training pairs in  $\mathcal{T}$  are contained in the set of support vectors. Quadratic programming tools of Matlab<sup>®</sup> is utilized to solve the optimization problem and for both SVM structures, the condition  $\sum_{i=1}^{N_T} \lambda_i = 0$  is individually met precisely. The parameter  $C$  is taken as infinity for both SVM structures to increase the importance of regression precision in the optimization problem.

A comparison in terms of training times stipulate that a designer can quickly obtain a prototype with MLP structure. The MLP emulator considered here has two hidden layers having 24 neurons in the first hidden layer, and 12 neurons in the second hidden layer. The neuronal activation functions in the hidden layers are hyperbolic tangent, and those in the output layer are linear. These settings highlight a standard MLP structure and the smallest propagation delay measured as  $T_\Delta = 0.0309$  msec. is with MLP structure and this fact emphasizes the operational simplicity of this approach.

Considering the training and checking MSE levels for all three approaches, it is seen that SVM reaches the smallest values in terms of  $\mathcal{T}_{\text{MSE}}$ . For a fair comparison of all three approaches, summing  $\mathcal{T}_{\text{MSE}}$  and  $\mathcal{C}_{\text{MSE}}$  values for ANFIS<sub>1</sub> and ANFIS<sub>2</sub> structures yield a total value of  $\mathcal{T}_{\text{MSE}} = 8.3492\text{e-}8$  and  $\mathcal{C}_{\text{MSE}} = 1.5963\text{e-}10$ . Performing the same computations for SVM structures result in  $\mathcal{T}_{\text{MSE}} = 3.9465\text{e-}16$  and  $\mathcal{C}_{\text{MSE}} = 7.5601\text{e-}11$ . With these values, noting that the total  $\mathcal{C}_{\text{MSE}}$  values are almost at the same order, it is understood that in terms of realization performance measured over the training pairs, the best approximation results are obtained with SVM emulators, then with MLP emulator. In spite of the availability of other kernels for SVMs, e.g. radial basis functions as one of the popular ones, we tried all such alternatives yet the best results have

been obtained with the spline kernel. Though we make no generalization, the application we consider here is best handled by the use of spline kernel.

A good measure of performance is  $\mathcal{V}_{\text{MSE}}$ , which is given in (15). We compute this quantity for the trajectories shown in Fig. 1, and obtain a single quantity by summing up the values computed for every subplot. Basically, this suggests  $\mathcal{N} = 60$  trials in total and the trajectories contain the full diversity of the regimes. These settings imply the consideration of  $N_\gamma = 360,000$  pairs during the validation phase. According to the results seen in the three tables, SVM approach is seen to be more accurate than ANFIS and MLP but the SVM approach is computationally more demanding than the others. In addition to this, the result with ANFIS structures is slightly higher than that with SVM yet the values are very close to each other. This result shows that one should choose ANFIS structure as its precision is as high as SVM with affordable propagation delay indicating low operational complexity.

A last point that should be stressed here is the way these emulators are obtained. For each approach, we have carried out many experiments and chosen the ones displaying best characteristics, and performed the comparison among the most successful candidates of every approach. Clearly, one can develop simpler emulators for prototyping but a comparison of almost equally time consuming emulator design efforts stipulate dramatically different results in terms of practical applicability and realization accuracy as discussed above.

## VII. CONCLUSIONS

Due to a rich set of responses emerging under different operating conditions, the bioreactor benchmark problem is a prime example for studying the merit and effectiveness of identification tools based on numerical data. This paper considers three approaches and their widely used tuning schemes to develop a one-step ahead predictor for the benchmark process. Series-parallel identification scheme is used and no noise is considered in assessing the performance of the approaches.

In brief, MLP approach is found to be simple and practical for data based modeling applications yet the SVM approach, despite its computational burden, approximates the presented data very accurately and the same precision can also be obtained in the trajectories seen in Fig. 1. According to the results obtained, a good balance in the computational cost versus precision scale is realized by the ANFIS architecture.

The contribution of this paper is to present a fair comparison among three foremost approximation strategies utilized for the identification of a plant displaying several challenges. The paper unfolds the design tradeoffs and aims at guiding the practitioners by highlighting key aspects of each approach. Future research in this subject has the goal of unfolding the facts arise when the considered data and the operating environment contain uncertainties.

## VIII. ACKNOWLEDGMENTS

The computations of this work were done on the facility provided through the project from TÜBİTAK 1001 Programme, Grant No 107E137, and by TOBB ETÜ BAP Program (Contract No: ETÜ BAP 2006/04).

The author gratefully acknowledges the facilities of TOBB ETÜ library.

## REFERENCES

- [1] Ljung, L., *System Identification: Theory for the User*, Prentice Hall PTR.
- [2] Jang, J.-S.R., C.-T. Sun, E. Mizutani, (1997), *Neuro-Fuzzy and Soft Computing*, PTR Prentice-Hall.
- [3] Cristianini, N. and J. Shawe-Taylor (2000), *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press.
- [4] Ungar, L.H. (1990), "A Bioreactor Benchmark for Adaptive-Network Based Process Control," *Neural Networks for Control*, W.T. Miller III, R. S. Sutton, P.J. Werbos, Eds, MIT Press, pp.387-402.
- [5] Anderson, C. W. and W. T. Miller III (1990), "Challenging Control Problems," *Neural Networks for Control*, W.T. Miller III, R. S. Sutton, P. J. Werbos, Eds, MIT Press, pp.475-510.
- [6] Efe, M.Ö., E. Abadoglu, and O. Kaynak (1999), "A Novel Analysis and Design of a Neural Network Assisted Nonlinear Controller for a Bioreactor," *International Journal of Robust and Nonlinear Control*, v.9, no.11, pp.799-815.
- [7] Puskorius, G.V., L.A. Feldkamp (1990), "Neurocontrol of Nonlinear Dynamical Systems with Kalman Filter Trained Recurrent Networks," *IEEE Transactions on Neural Networks*, v.5, n.2, pp.279-297.
- [8] Wang, X.-D. and M.-Y. Ye, (2004), "Nonlinear Dynamic System Identification Using Least Squares Support Vector Machine Regression," *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics*, August 26-29, Shanghai, China, pp.941-945.
- [9] Gunn, S.R. (1998), "Support Vector Machines for Classification and Regression," *ISIS Technical Report*, Univ. of Southampton, United Kingdom.
- [10] Hagan, M.T. and M.B. Menhaj (1994), "Training Feedforward Networks with the Marquardt Algorithm," *IEEE Transactions on Neural Networks*, v.5, no.6, pp.989-993.
- [11] Battiti R. (1992), "First-second-order methods for learning: between steepest descent and Newtons method," *Neural Computation*, v.4, pp.141166.
- [12] Narendra, K.S. and K. Parthasarathy (1990) "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, v.1, No. 1, pp. 4-27.
- [13] Takagi, T., and M. Sugeno (1985), "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Transactions on Systems, Man and Cybernetics*, v.15, n.1, pp.116-132.
- [14] Jang, J.-S.R. (1993) "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Transactions on Systems, Man and Cybernetics*, v.23, no.3, pp.665-685.