Recurrent Neural Kalman Filter Identification and Indirect Adaptive Control of a Continuous Stirred Tank Bioprocess

Ieroham S. Baruch[1] Carlos R. Mariaca-Gaspar[1]

1 Department of Automatic Control, CINVESTAV-IPN,
Ave. IPN No 2508, Col. Zacatenco, A.P. 14-740,
07360 Mexico City, MEXICO
Email: {baruch,cmariaca}@ctrl.cinvestav.mx

Abstract— *The aim of this paper is to propose a new Kalman Filter Recurrent Neural Network (KFRNN) topology and a recursive Levenberg-Marquardt (L-M) algorithm of its learning capable to estimate states and parameters of a highly nonlinear Continuous Stirred Tank Bioreactor (CSTR) in noisy environment. The estimated parameters and states obtained by the proposed KFRNN identifier are used to design an indirect adaptive sliding mode control scheme. The obtained simulation results of the real-time neural identification and control of a CSTR model, taken from the literature, exhibited fast convergence, noise filtering, and low mean squared error of reference tracking. A 20 runs comparative validation experiment in noisy environment is also done. It gives some priority of the L-M learning over the BP one.*

Keywords— Backpropagation learning, continuous stirred tank bioreactor, indirect adaptive sliding mode control, Kalman filter recurrent neural network, Levenberg-Marquardt learning.

1 Introduction

The Recent advances in understanding of the working principles of artificial neural networks has given a tremendous boost to identification, prediction and control tools of nonlinear systems, [1], [2]. The main network property namely the ability to approximate complex non-linear relationships without prior knowledge of the model structure makes them a very attractive alternative to the classical modeling and control techniques, [3]. It is particularly useful in applications where the complexity of the data or tasks makes the design of such a functions by hand impractical. Among several possible network architectures the ones most widely used are the Feedforward NN (FFNN) and Recurrent NN (RNN). In a FFNN the signals are transmitted only in one direction, starting from the input layer, subsequently through the hidden layers to the output layer, which requires applying a tap delayed global feedbacks and a tap delayed inputs to achieve a Nonlinear Autoregressive Moving Average (NARMAX) neural dynamic plant model. A RNN has local feedback connections to some of the previous layers. Such a structure is suitable alternative to the FFNN when the task is to model dynamical systems. Its main advantage is the reduced complexity of the network structure. However, the analysis of state of the art in the area of classical RNN-based modelling and control has also shown some of their inherent limitations as follows: 1. The RNN input vector consists of a number of past system inputs and outputs and there is not a systematic way to define the optimal number of past values, and usually, the method of trials and errors is performed, [2]. 2. The RNN model is naturally formulated as a discrete model with fixed sampling period, therefore, if the sampling period is changed, the neural network has to be trained again; 3. It is assumed that the plant order is known, which represents a quite strong modelling assumption in general, [1]; 4. The managing of noisy input/output plant data required to augment the filtering capabilities of the identification RNNs, [4]. Driven by these limitations, a new Kalman Filter Recurrent Neural Network (KFRNN) topology and the recursive Backpropagation (BP) type learning algorithm in vector-matrix form was derived and its convergence was studied, [5]. But the recursive BP algorithm, applied for KFRNN learning, is a gradient descent first order learning algorithm which not permits to augment the precision and to accelerate the learning. So, the aim of the paper is to use a second order learning algorithm for the KFRNN like the Levenberg-Marquardt (L-M) algorithm is, [4, 6]. The KFRNN with L-M learning will be applied for Continuous Stirred Tank Reactor (CSTR) model identification, [7]. In [8, 9] a comparative study of linear, nonlinear and neural-network-based adaptive controllers for a CSTR is done. The papers proposed to use the neuro-fuzzy and adaptive nonlinear control systems design, applying FFNNs (multilayer perceptron and radial basis functions NN). The proposed control gives a good adaptation to the nonlinear plants dynamics, better with respect to the other methods of control, but the used FFNNs have a great complexity, and the plant order has to be known. The application of KFRNNs, [5], together with the recursive L-M, [4], could avoid these problems improving the learning and the precision of the plant states and parameters estimation. Here the obtained from the KFRNN identifier parameters and states will be used to design an indirect adaptive sliding mode control scheme, [5].

2 Topology and learning of the KFRNN

2.1 KFRNN Topology

A Kalman Filter Recurrent Neural Network model and its learning algorithm of dynamic Backpropagation-type, together with the explanatory figures and stability proofs, are described in [5]. The KFRNN topology (see Fig. 1) is described by the following vector-matrix equations:

ISBN: 978-989-95079-6-8

$$X(k+1) = A_1 X(k) + BU(K) - Dy(k) \quad (1)$$

$$Z(k) = \mathrm{G}[X(k)] \quad (2)$$

$$V_1(k) = CZ(k) \quad (3)$$

$$V(k+1) = A_2 V(k) + V_1(k) \quad (4)$$

$$Y(k) = \mathrm{F}[V(k)] \quad (5)$$

$$A_1 = block\text{-}diag\,(A_{1,i})\,;\ |A_{1,i}| < 1$$
$$A_2 = block\text{-}diag\,(A_{2,i})\,;\ |A_{2,i}| < 1 \quad (6)$$

Where: Y, X, and U are, respectively, output, state and input vectors with dimensions l, n, m; A_1 and A_2 are (nxn) and (lxl) block-diagonal local feedback weight matrices; $A_{1,i}$ and $A_{2,i}$ are i-th diagonal block of A_1 and A_2 with (nxn) and (lxl) dimensions, respectively. Equation (6) represented the local stability conditions, imposed on all blocks of A_1 and A_2; B and C are (nxm) and (lxn) input and output weight matrices; D is a (nxl) global output feedback weight matrix; G(.), F(.) are vector-valued sigmoid or hyperbolic tangent-activation functions, Z, V_1, V are vector variables with corresponding dimensions; the integer k is a discrete-time variable.

2.2 Backpropagation learning of the KFRNN

The general BP learning algorithm is given by the following equation:

$$W(k+1) = W(k) + \eta \Delta W(k) + \alpha \Delta W(k-1) \quad (7)$$

Where: W is the weight matrix, being modified (A_1, A_2, B, C, D); ΔW is the weight matrix correction (ΔA_1, ΔA_2, ΔB, ΔC, ΔD); η and α are learning rate parameters. Applying the diagrammatic method, [10], and using the block- diagram of the KFRNN topology (see Fig. 1), we could design an error predictive adjoint KFRNN (see Fig. 2). Following this adjoint KFRNN block diagram we could obtain the next matrix KFRNN weight updates:

$$\Delta C(k) = E_1(k) Z^T(k) \quad (8)$$

$$\Delta A_2(k) = E_1(k) V^T(k) \quad (9)$$

$$E_1(k) = F'[Y(k)]E(k);\ E(k) = Y_p(k) - Y(k) \quad (10)$$

$$\Delta B(k) = E_3(k) U^T(k) \quad (11)$$

$$\Delta A_1(k) = E_3(k) X^T(k) \quad (12)$$

$$\Delta D(k) = E_3(k) Y^T(k) \quad (13)$$

$$E_3(k) = G'[Z(k)]E_2(k);\ E_2(k) = C^T E_1(k) \quad (14)$$

$$\Delta vA_1(k) = E_3(k) \oplus X(k);\ \Delta vA_2(k) = E_1(k) \oplus V(k) \quad (15)$$

Where: ΔA_1, ΔA_2, ΔB, ΔC, ΔD are weight corrections of the of the learned matrices A_1, A_2, B, C, D, respectively; E is an error vector of the output KFRNN layer, where Yp is a desired target vector and Y is a KFRNN output vector, both with dimensions l; X is a state vector, and E1, E2, E3 are error vectors, shown on Fig. 2; F'(.), G'(.) are diagonal

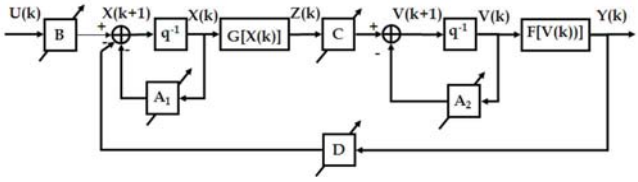

Figure 1: Block-diagram of the KFRNN topology.

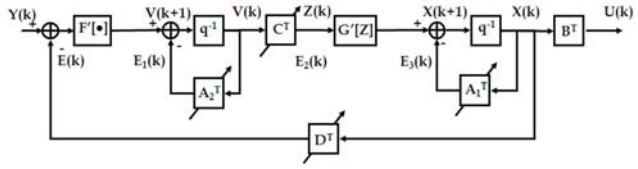

Figure 2: Block-diagram of the adjoint KFRNN topology.

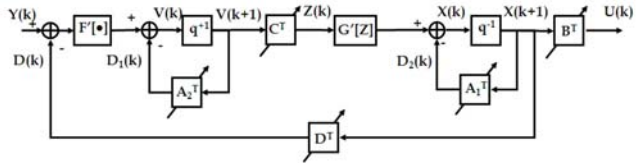

Figure 3: Block-diagram of the adjoint KFRNN used for the L-M algorithm.

Jacobean matrices with appropriate dimensions. The diagonal elements of these matrices are derivatives of the activation functions. The equations (12), (9) represented the weight update of the feedback weight matrices A_1, A_2, as full (nxn), (lxl) matrices. The equation (15) gives the learning solution as an element by element vector product when these matrices are diagonal vA_1, vA_2 which is our case.

2.3 Recursive Levenberg-Marquardt KFRNN learning

The general recursive L-M algorithm of learning, [4, 6], is given by the following equations:

$$W(k+1) = W(k) + P(k)\nabla Y[W(k)]E[W(k)] \quad (16)$$

$$Y[W(k)] = g[W(k), U(k)] \quad (17)$$

$$E^2[W(k)] = \{Y_p(k) - g[W(k), U(k)]\}^2 \quad (18)$$

$$DY[W(k)] = \frac{\partial}{\partial W} g[W, U(k)]\Big|_{W=W(k)} \quad (19)$$

Where: W is a general weight matrix (A_1, A_2, B, C, D), under modification; P is the covariance matrix of the weights estimates, being updated; DY is a nw-dimensional gradient vector; Y is the KFRNN output vector which depended on the updated weights and the input; E is an error vector; Yp is the plant output vector which is in fact the target vector. Using the same KFRNN adjoint block diagram (see Fig.2), we could obtain the values of the gradients DY for each updated weight, propagating the value D(k) = I through it and following the block diagram of Fig. 3, we could apply equation (19) for each element of the weight matrices (A_1, A_2, B, C, D) to be updated. The corresponding gradient components are as follows:

$$DY[C_{ij}(k)] = D_{1,i}(k) Z_j(k) \quad (20)$$

$$DY[A_{2ij}(k)] = D_{1,i}(k) V_j(k) \quad (21)$$

ISBN: 978-989-95079-6-8

$$D_{1,i}(k) = F_i'[Y_i(k)] \tag{22}$$

$$DY[A_{1ij}(k)] = D_{2,i}(k) X_j(k) \tag{23}$$

$$DY[B_{ij}(k)] = D_{2,i}(k) U_j(k) \tag{24}$$

$$DY[D_{ij}(k)] = D_{2,i}(k) Y_j(k) \tag{25}$$

$$D_{2,i}(k) = G_i'[Z_i(k)] C_i D_{1,i}(k) \tag{26}$$

So the Jacobean matrix could be formed as:

$$DY[W(k)] = [DY(C_{ij}(k)), DY(A_{2ij}(k)), DY(B_{ij}(k)) \\ DY(A_{1ij}(k)), DY(D_{ij}(k))] \tag{27}$$

The P(k) matrix is computed recursively by the equation:

$$P(k) = \alpha^{-1}(k)\{P(k-1) - \\ -P(k-1)\Omega[W(k)] S^{-1}[W(k)] \Omega^T[W(k)] P(k-1)\} \tag{28}$$

Where: the S(.), and Ω(.) matrices are given as follows:

$$S[W(k)] = \alpha(k)\Lambda(k) + \Omega^T[W(k)] P(k-1)\Omega[W(k)] \tag{29}$$

$$\Omega^T[W(k)] = \begin{bmatrix} & \nabla Y^T[W(k)] & \\ 0 \quad \cdots & 1 & \cdots \quad 0 \end{bmatrix}; \\ \Lambda(k)^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \rho \end{bmatrix}; 10^{-4} \le \rho \le 10^{-6}; \\ 0.97 \le \alpha(k) \le 1; 10^3 \le P(0) \le 10^6 \tag{30}$$

The matrix Ω(.) has dimension (nwx2), where the second row has only one unity element (the others are zero). The position of that element is computed by:

$$i = k \bmod (nw) + 1; k > nw \tag{31}$$

Next the given up topology and learning are applied for CSTR system identification and control.

3 Indirect Adaptive Control Scheme (Sliding Mode Control)

The indirect adaptive control using the RTNN as plant identifier has been described in, [11]. Later the proposed indirect control has been derived as a Sliding Mode Control (SMC) and applied for control of unknown hydrocarbon biodegradation processes, [12]. The block diagram of the indirect adaptive control scheme is shown on Fig. 4. It contains identification and state estimation KF RNN and a sliding mode controller. The stable nonlinear plant is identified by a KFRNN model with topology, given by equations (1)-(6) learned by the stable BP-learning algorithm, given by equations (7)-(15), or using the second order LM-learning algorithm, given by equations (16)-(31). The simplification and linearization of the neural identifier equations (1)-(2), omitting the DY(.) term, leads to the next

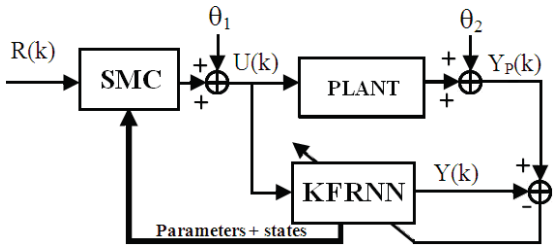

Figure 4: Block diagram of the closed-loop system containing KF RNN identifier and a SMC.

local linear plant model, extracted from the complete KF RNN model:

$$X(k+1) = A_1 X(k) + BU(k) \tag{32}$$

$$Z(k) = HX(k); \; H = CG'(Z) \tag{33}$$

Where G'(.) is the derivative of the activation function and L = M, is supposed.

In [13], the sliding surface is defined with respect to the state variables, and the SMC objective is to move the states form an arbitrary space position to the sliding surface in finite time. In [14], the sliding surface is also defined with respect to the states but the states of the SISO systems are obtained from the plant outputs by differentiation. In [15], the sliding surface definition and the control objectives are the same. The equivalent control systems design is done with respect to the plant output, but the reachability of the stable output control depended on the plant structure. In [12], the sliding surface is derived directly with respect to the plant outputs which facilitated the equivalent SMC systems design. Let us define the following SS as an output tracking error function:

$$S(k+1) = E(k+1) + \sum_{i=1}^{P} \gamma_i E(k-i+1); \; |\gamma| < 1 \tag{34}$$

Where: S(.) is the Sliding Surface Error Function (SSEF) defined with respect to the plant output; E(.) is the systems output tracking error; γ_i are parameters of the desired stable SSEF; p is the order of the SSEF. The tracking error and their iterate are defined as:

$$E(k) = R(k) - Z(k); E(k+1) = R(k+1) - Z(k+1) \tag{35}$$

Where R(k), Z(k) are L-dimensional reference and output vectors of the local linear plant model. The objective of the sliding mode control systems design is to find a control action which maintains the systems error on the sliding surface which assure that the output tracking error reaches zero in P steps, where P < N. So, the control objective is fulfilled if:

$$S(k+1) = 0 \tag{36}$$

ISBN: 978-989-95079-6-8

Now, let us to iterate (33) and to substitute (32) in it so to obtain the input/output local plant model, which yields:

$$Z\left(k+1\right) = FX\left(k+1\right) = F\left[AX\left(k\right) + BU\left(k\right)\right] \quad (37)$$

From (34), (35), and (36) it is easy to obtain:

$$R\left(k+1\right) - Z\left(k+1\right) + \sum_{i=1}^{P} \gamma_i E\left(k-i+1\right) = 0 \quad (38)$$

The substitution of (37) in (38) gives:

$$R\left(k+1\right) - FAX\left(k\right) + FBU\left(k\right) - \sum_{i=1}^{P} \gamma_i E\left(k-i+1\right) \quad (39)$$

As the local approximation plant model (32), (33), is controllable, observable and stable (see [5]), the matrix A_1 is diagonal, and L = M, then the matrix product (HB), representing the plant model static gain, is nonsingular, and the plant states X(k) are smooth non-increasing functions. Now, from (39) it is easy to obtain the equivalent control capable to lead the system to the sliding surface which yields:

$$U_{eq}\left(k\right) = \left(FB\right)^{-1}\left[-FAX\left(k\right) + R\left(k+1\right) + \sum_{i=1}^{P} \gamma_i E\left(k-i+1\right)\right] \quad (40)$$

Following [13], the SMC avoiding chattering is taken using a saturation function instead of sign one. So the SMC takes the form:

$$U^*(k) = \begin{cases} U_{eq}\left(k\right) & \text{if } \left\|U_{eq}\left(k\right)\right\| < U_0 \\ -U_0 U_{eq}\left(k\right) / \left\|U_{eq}\left(k\right)\right\| & \text{if } \left\|U_{eq}\left(k\right)\right\| \geq U_0 \end{cases} \quad (41)$$

The SMC substituted the multi-input multi-output coupled high order dynamics of the linearized plant with desired decoupled low order one.

4 Description of the CSTR bioprocess

The CSTR model given in [16, 17] was chosen as a realistic example for application of the KFRNN and the SMC for solution of system identification and control problems. The CSTR is described by the following continuous time, nonlinear, system of ordinary differential equations:

$$\frac{dC_A(t)}{dt} = \frac{Q}{V}\left(C_{Af} - C_A(t)\right) - k_0 C_A(t) \exp\left(-\frac{E}{RT(t)}\right) \quad (42)$$

$$\frac{dT(t)}{dt} = \frac{Q}{V}(T_f - T(t)) + \frac{(-\Delta H)C_A(t)}{\rho C_p}\exp\left(\frac{E}{RT(t)}\right) + \frac{\rho_c C_{pc}}{\rho C_p V} Q_c(t)\left[1 - \exp\left(\frac{-hA}{Q_c(t)\rho_c C_{pc}}\right)\right](t_{ef} - T(t)) \quad (43)$$

It suffices to know that within the CSTR two chemicals are mixed and react to produce a product compound A at a concentration CA(t), with the temperature of the mixture being T(t). The reaction is exothermic and producing heat which slows down the reaction. By introducing a coolant flow-rate Qc(t), the temperature can be varied and hence the product concentration controlled. CAf is the inlet feed concentration, Q is the process flow-rate, Tf and Tef are the inlet feed and coolant temperatures, respectively, all of which are assumed constant at nominal values. Likewise k0, E/R, V, ΔH, ρ, Cpc, Cp and ρc are thermodynamic and chemical constant relating to this particular problem. Numerical values for the parameters and nominal operating conditions of this model are given in Table 1.

The quantities Qc0, T0, and CA0 shown in Table 1 are steady values for steady operating point in the CSTR. The objective is to control the product compound A, manipulating the variable $Q_c(t)$. The operating values of the variables are obtained from the papers [16], [17]. The authors of these papers examined the performances of neural networks and of fuzzy-neural networks based control systems, respectively.

5 Simulation Results

The proposed indirect adaptive SMC is applied for the given up CSTR plant. The Fig. 5 and Fig. 7 showed the comparison between the reference and the plant output signals applying the BP and L-M algorithms of learning for the KFRNN, respectively. Detailed comparative graphical simulation results of CSTR KFRNN plant identification by means of the BP and the L-M learning are given in Fig.6 and Fig.8. A 10% white noise is added to the plant inputs and outputs and the behavior of the plant identification and control has been studied accumulating some statistics of the final MSE% (ξ_{av}) for BP and L-M learning, which results are given in Table 3 and Table 4 for 20 runs. The mean average cost for all runs (ε) of control and the standard deviation (σ) with respect to the mean value and the deviation (Δ) are presented in Table 2 for BP and L-M algorithms of learning.

Table 1: CSTR parameters and operating conditions

$Q = 100 \quad (L/\min)$	$E/R = 9.95\times10^3 \quad (K)$
$C_{Af} = 1.0 \quad (mol/L)$	$-\Delta H = 2\times10^5 \quad (cal/mol)$
$T_f = 350 \quad (K)$	$\rho\cdot\rho_c = 1000 \quad (g/L)$
$T_{cf} = 350 \quad (K)$	$C_p C_{pc} = 1 \quad (cal/gK)$
$V = 100 \quad (L)$	$Q_{c0} = 103.41 \quad (L/\min)$
$hA = 7\times10^5 \quad (cal/\min K)$	$T_0 = 440.2 \quad (K)$
$k_0 = 7.2\times10^{10} \quad (1/\min)$	$C_{A0} = 0.0836 \quad (mol/L)$

Table 2: Standard deviation and mean average values using BP and L-M algorithms of learning.

BP Algorithm	LM Algorithm
ε = 1.0997	ε =1.0370
σ = 0.0295	σ =0.0232

ISBN: 978-989-95079-6-8

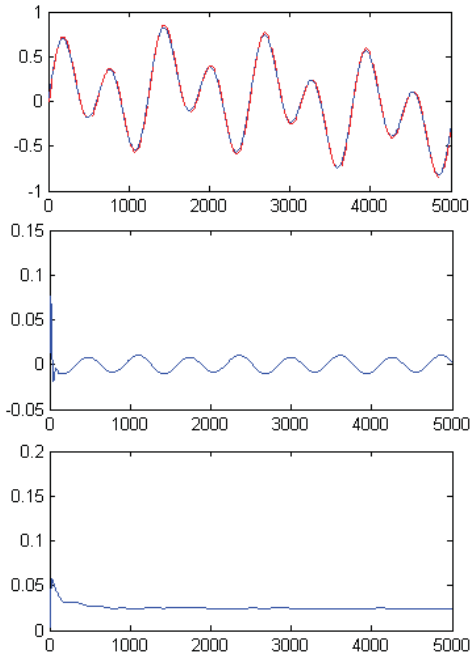

Figure 5: Detailed graphical simulation results of CSTR plant Sliding Mode Indirect Control using BP KFRTNN learning a) comparison between the plant output and the reference signal; b) control signal; c) MSE% of control.

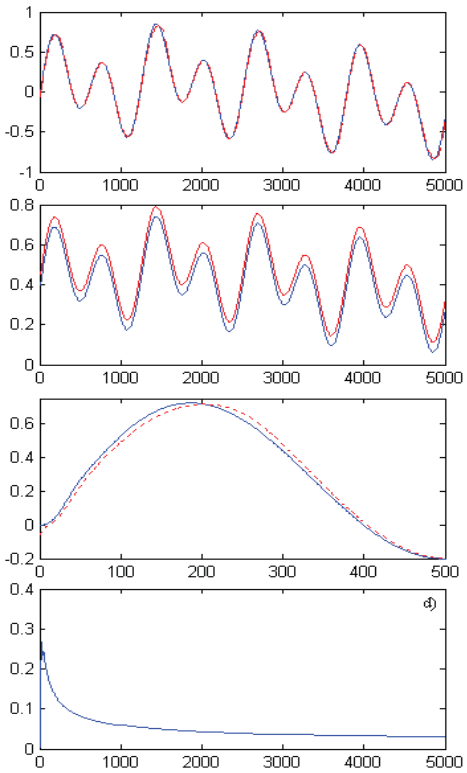

Figure 6: Graphical results of identification using BP KFRNN learning. a) Comparison of the plant output (continuous line) and KFRNN output (pointed line); b) state variables; c) comparison of the plant output (continuous line) and KFRNN output (pointed line) in the first instants; d) MSE% of identification

The values are computed using the following formulas:

$$\varepsilon = \frac{1}{n}\sum_{k=1}^{n}\xi_{av_k}, \quad \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Delta_i^2}, \quad \Delta = \xi_{av} - \varepsilon \tag{44}$$

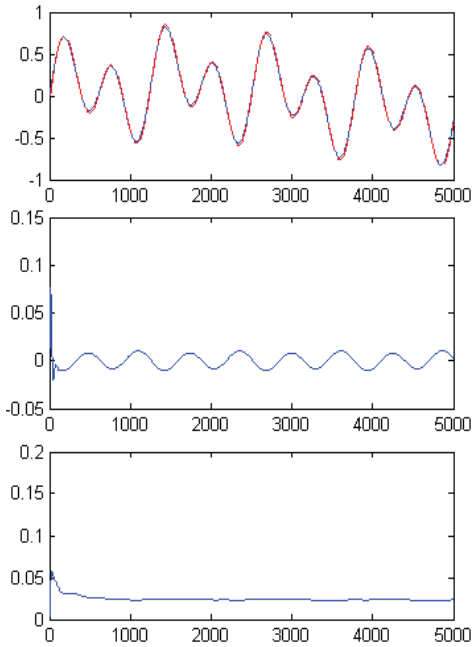

Figure 7: Detailed graphical simulation results of CSTR plant Sliding Mode Indirect Control using L-M KFRTNN learning a) comparison between the plant output and the reference signal; b) control signal; c) MSE% of control.

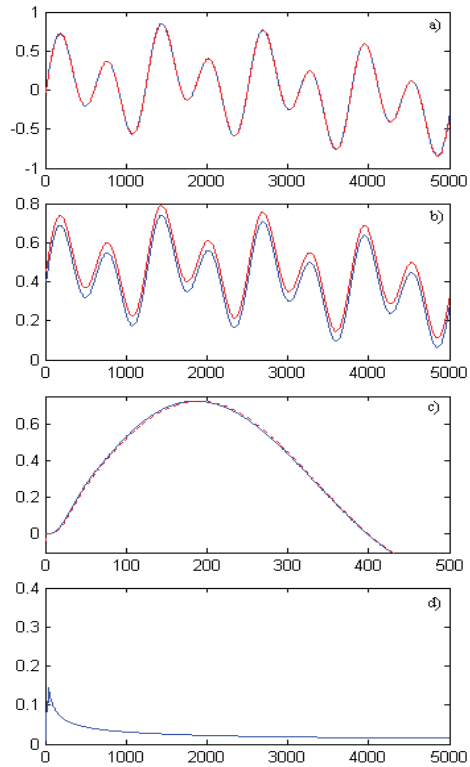

Figure 8: Graphical results of identification using L-M KFRNN learning. a) Comparison of the plant output (continuous line) and KFRNN output (pointed line); b) state variables; c) comparison of the plant output (continuous line) and KFRNN output (pointed line) in the first instants; d) MSE% of identification

ISBN: 978-989-95079-6-8

Table 3: Final MSE% of control for 20 runs of control program using the Backpropagation algorithm.

No.	**1**	**2**	**3**	**4**	**5**
MSE%	1.1096	1.1530	1.1331	1.0660	1.1375
No.	**6**	**7**	**8**	**9**	**10**
MSE%	1.0701	1.1171	1.1202	1.1311	1.0899
No.	**11**	**12**	**13**	**14**	**15**
MSE%	1.0964	1.1007	1.1139	1.0685	1.0502
No.	**16**	**17**	**18**	**19**	**20**
MSE%	1.1330	1.1039	1.0724	1.0609	1.0665

Table 4: Final MSE% of control for 20 runs of control program using the Levenberg-Marquardt algorithm.

No.	**1**	**2**	**3**	**4**	**5**
MSE%	1.0296	1.0462	1.0472	1.0670	1.0134
No.	**6**	**7**	**8**	**9**	**10**
MSE%	1.0078	1.0396	1.0678	1.0017	1.0609
No.	**11**	**12**	**13**	**14**	**15**
MSE%	1.0019	1.0364	1.0135	1.0501	1.0175
No.	**16**	**17**	**18**	**19**	**20**
MSE%	1.0654	1.0096	1.03652	1.0627	1.0660

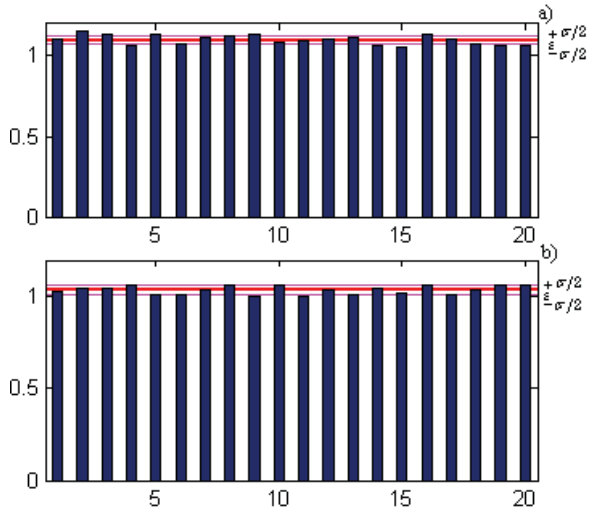

Figure 9: Comparison between the final MSE% for 20 runs of control program: a) using BP algorithm, b) using L-M algorithm.

6 Conclusions

The paper proposed a new KFRNN model for systems identification and states estimation of nonlinear plants. The KFRNN is learned by the second order recursive learning algorithm of Levenberg-Marquardt. The estimated parameters and states of the RNN model are used for indirect (SM) adaptive trajectory tracking control systems design. The applicability of the proposed neural control system is confirmed by simulation results with a CSTR plant. The results showed good convergence of both L-M and BP learning algorithms. The L-M algorithm of learning is more precise (see Table 2) but more complex then the BP one.

Acknowledgment

The Ph.D. student Carlos-Roman Mariaca-Gaspar is thankful to CONACYT, Mexico for the fellowship received during his studies at CINVESTAV-IPN, Mexico.

References

[1] K.S. Narendra, and K. Parthasarathy, Identification and Control of Dynamic Systems Using Neural Networks, *IEEE Transactions on Neural Networks,* 1(1): 4 -27, 1990.

[2] K.J. Hunt, D. Sbarbaro, R. Zbikowski, and P.J. Gawthrop, Neural Network for Control Systems - A Survey, *Automatica*, 28(6): 1083-1112, 1992.

[3] S.A. Pao, S.M. Phillips, and D.J. Sobajic, Neural Net Computing and Intelligent Control Systems, *International Journal of Control*, 56: 263-289, 1992.

[4] I. S. Baruch, S. Escalante, and C. R. Mariaca-Gaspar, Identification, Filtering and Control of Nonlinear Plants by Recurrent Neural Networks Using First and Second Order Algorithms of Learning, *Dynamics of Continuous, Discrete and Impulsive Systems, International Journal, Series A: Mathematical Analysis, Special Issue on Advances in Neural Networks-Theory and Applications*, Watam Press, Canada, ISSN 1201-3390, 14: 512-521, 2007.

[5] I.S. Baruch, C.R. Mariaca-Gaspar, and J. Barrera-Cortes, Recurrent Neural Network Identification and Adaptive Neural Control of Hydrocarbon Biodegradation Processes, *In: Recurrent Neural Networks, Xiaolin Hu, and P. Balasubramaniam (Eds.)*, ISBN 978-3-902613-28-8, I-Tech Education and Publishing KG, Vienna, Austria, Chapter 4: 61-88, 2008.

[6] L.S. Ngia, and J. Sjöberg, Efficient Training of Neural Nets for Nonlinear Adaptive Filtering Using a Recursive Levenberg-Marquardt Algorithm, *IEEE Transactions on Signal Processing*, 48: 1915-1927, 2000.

[7] L. D. Schmidt, *The Engineering of Chemical Reactions,* New York: Oxford University Press, ISBN 0-19-510588-5, 1998.

[8] Shi-Rong Liu, and Jin-Shou Yu, Robust Control Based on Neuro Fuzzy Systems for a Continuous Stirred Tank Reactor, *In: Proc. of the First International Conference on Machine Learning and Cybernetics,* Beijing, 4-5 November 2002, 1483-1488, 2002.

[9] T. Zhang, and M. Guay. Adaptive Nonlinear Control of Continuously Stirred Tank Reactor Systems, *In: Proc. of the American Control Conference, Arlington*, 25-27 June 2001, 1274-1279, 2001.

[10] E. Wan, and F. Beaufays, Diagrammatic Method for Deriving and Relating Temporal Neural Networks Algorithms, *Neural Computations*, 8: 182-201, 1996.

[11] I.S. Baruch, J.M.Flores, F. Thomas, and R. Garrido, Adaptive Neural Control of Nonlinear Systems, *In: Artificial Neural Networks-ICANN 2001, LNCS 2130, G. Dorffner, H. Bischof, K. Hornik (Eds.)*, ISBN 3-540-42486-5, Springer, Berlin, 930-936, 2001.

[12] I.S. Baruch, C.R. Mariaca-Gaspar, I. Cruz-Vega, and J. Barrera-Cortes, Sliding Mode Control of a Hydrocarbon Degradation in Biopile System Using Recurrent Neural Network Model, *In: MICAI 2007: Advances in Artificial Intelligence, LNAI 4827, A. Gelbukh, A.F. Kuri-Morales (Eds.)*, ISBN-10 3-540-76630-8, Springer, Berlin Heidelberg New York, 1184-1194, 2001.

[13] K. D. Young, V.I. Utkin, and U. Ozguner, A Control Engineer's Guide to Sliding Mode Control, *IEEE Transactions on Control Systems Technology*, 7 (3): 328-342, 1999.

[14] A. Levent, Higher Order Sliding Modes, Differentiation and Output Feedback Control, *International Journal of Control, Special Issue Dedicated to Vadim Utkin on the Occasion of his 65th Birthday (Guest editor: Leonid M. Fridman),* ISSN 0020-7179, 76(9/10): 924-941, 15 June-10 July 2003.

[15] C. Eduards, S.K. Spurgeon, and R.G. Hebden, On the Design of Sliding Mode Output Feedback Controllers, *International Journal of Control, Special Issue Dedicated to Vadim Utkin on the Occasion of his 65th Birthday (Guest editor: Leonid M. Fridman),* ISSN 0020-7179, 76(9/10): 893-905, 15 June-10 July 2003.

[16] G. Lightbody, and G. W. Irwin, Nonlinear Control Structures Based on Embedded Neural Systems Models, *IEEE Transactions on Neural Networks*, 8: 553-557, 1997.

[17] A. Aoyama, F. J. Doyle, and V. Venkatasubramanian, Control-Affine Fuzzy Neural Networks Approach for Nonlinear Process Control, *Journal of Process Control*, 8: 375-386, 1995.