IMPLEMENTATION OF RECURRENT MULTI-MODELS FOR SYSTEM IDENTIFICATION

Lamine Thiaw, Kurosh Madani
Laboratoire Image, Signal et Systmes Intelligents (LISSI / EA 3956) IUT de Snart, Universit Paris XII
Av. Pierre Point, F-77127 Lieusaint, France

Rachid Malti
Laboratoire Automatique Productique et Signal University Bordeaux 1 351, Cours de la Libration, 33405 Talence Cedex, France

Gustave Sow
LER, Ecole Suprieure Polytechnique de Dakar, Universit Cheikh Anta Diop, BP 5085, Dakar Fan, Senegal

Keywords: System identification, non-linear systems, multi-model, recurrent models.

Abstract: Multi-modeling is a recent tool proposed for modeling complex nonlinear systems by the use of a combination of relatively simple set of local models. Due to their simplicity, linear local models are mainly used in such structures. In this work, multi-models having polynomial local models are described and applied in system identification. Estimation of model's parameters is carried out using least squares algorithms which reduce considerably computation time as compared to iterative algorithms. The proposed methodology is applied to recurrent models implementation. NARMAX and NOE multi-models are implemented and compared to their corresponding neural network implementations. Obtained results show that the proposed recurrent multi-model architectures have many advantages over neural network models.

1 INTRODUCTION

Identification of nonlinear systems is an important task for many real world applications such as process behavior analysis, control, prediction, etc. In the last years, several classes of models have been developed, among which Artificial Neural Networks (ANN) and multi-models (also known as operating regime approach), for non linear system identification.

ANN are widely used for dynamical nonlinear system modeling (Cheng et al., 1997; Konur and Okatan, 2004; Vartak et al., 2005). Such implementations like Time Delay Neural Network (TDNN) (Corradini and Cohen, 2002; Konur and Okatan, 2004), Jordan Network (Jordan, 1986), Elman Network (Elman, 1999) are very suitable for time series applications but they suffer of some limitations which restrict their use (Huang et al., 2005; Tomasz and Jacek, 1997). Several papers have been dedicated to the enhancement of neural networks for recurrent models identification (Bielikova, 2005; Huang et al., 2006). In (Huang et al., 2006) for example, a Multi-Context Recurrent Neural Network (MCRN) is studied and its performances are compared with those of the Elman Network and Elman Tower Network. Even though the proposed MCRN allows to achieve good performances, the main drawback remains its complexity due to the number of parameters induced by the context layer (Huang et al., 2006) which has weighting connections with both hidden and output layers.

The main difficulty encountered in recurrent neural networks is parameters estimation complexity. The parameters estimation is mostly performed using the gradient descent method (Backpropagation Through Time algorithm, Real Time Recurrent Learning algorithms, etc.) which cannot guarantee convergence to global minimum. On the other hand, the related algorithm's performance is very sensitive to the learning rate parameter which determines the convergence rate and the stability of the algorithm.

Multi-models have recently been proposed in numerous papers (Boukhris et al., 2000; Vernieuwe et al., 2004; Li et al., 2004) for modeling and control of nonlinear systems. For such systems, it is gen-

Thiaw L., Madani K., Malti R. and Sow G. (2007).
IMPLEMENTATION OF RECURRENT MULTI-MODELS FOR SYSTEM IDENTIFICATION.
In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics*, pages 314-321
DOI: 10.5220/0001615503140321

erally difficult to find a single analytical relationship describing system's behavior in its whole operating range. The system's complexity can be considerably reduced if system's operating range is divided into different regions where local behavior could be described with relatively simple models. The system's behavior is approximated by the weighted contribution of a set of local models. The difficulty encountered in this approach is the splitting of the system's operating range into convenient regions. For that purpose, various techniques have been studied among which grid partitioning, decision tree partitioning, fuzzy clustering based partitioning (see (Vernieuwe et al., 2004; Murray-Smith and Johansen, 1997)). Fuzzy clustering based partitioning enables to gather those data that may have some "similarities", facilitating system's local behavior handling. The main difficulty is the number of clusters needed to determine the multi-model's architecture. A method is presented here to bypass this difficulty.

Parameter estimation of recurrent multi-models is much simpler as compared to recurrent neural networks. We present in this work a multi-model implementation of recurrent models with polynomial local models. The proposed structure is applied to NARMAX and NOE models. The main advantage of such structure is that it allows to adjust the complexity of local models to the detriment of global one and vice-versa. Parameters are estimated using least squares algorithms, avoiding time consuming calculations and local minima.

The paper is structured as follows: in section 2 an overview of models identification principle is presented. Section 3 describes the general principle of multi-models using polynomial local models. The implementation of recurrent multi-model is presented in section 4. Results and discussions are presented in section 5.

2 OVERVIEW OF NON LINEAR MODELS

"Black box" models are very suited for complex systems representation (Sjoberg et al., 1995). Identification of such models consists of determining the mathematical relationship linking system's outputs (or its states) to its inputs from experimental data. In general, model describing system's behavior[1] can be expressed as:

$$y(t+h) = F_0\big(\underline{u}(t), \underline{\tilde{y}}(t)\big) + e(t+h) \qquad (1)$$

where :
$y(t+h)$ is the unknown system output at time instant $t+h$;
t is the current time instant and h is the prediction step;
$F_0(\cdot)$ is an unknown deterministic nonlinear function describing the system (the true model);
$\underline{u}(t)$ is a column vector which components are system's inputs at time t and at previous time instants;
$\underline{\tilde{y}}(t)$ is a column vector which components are obtained from system's output at time t and at previous time instants. It can be built from measured output data, estimated output data, prediction errors, or simulation errors;
$e(t+h)$ is an error term at time $t+h$.

The identification task consists of determining the function $F(\cdot)$ which is the best approximation of $F_0(\cdot)$ and estimating the system's output $\hat{y}$:

$$\hat{y}(t+h) = F\big(\underline{u}(t), \underline{\tilde{y}}(t), \theta\big) = F\Big(\underline{\varphi}(t), \theta\Big) \qquad (2)$$

where :
$\underline{\varphi}(t) = [\underline{u}(t)^T, \underline{\tilde{y}}(t)^T]^T$ is the regression vector obtained by the concatenation of the elements of vectors $\underline{u}(t)$ and $\underline{\tilde{y}}(t)$; and θ is a parameter vector to be estimated.
If $\underline{\tilde{y}}(t)$ in (2) depends on model's output or model's states, then the model (2) is said to be recurrent. Recurrent models have the ability to take into account system's dynamics. On the other hand, data collected from a process are usually noisy due to the sensors or the influence of external factors. Recurrent models allow to obtain unbiased parameters estimation. Various model classes have been established for modeling dynamical systems in presence of various noise configurations. Model classes differ by the composition of their regression vector. Since the exact model class is frequently unknown various classes are usually tested and the best one is chosen. In this work we focus on recurrent models called Nonlinear AutoRegressive Moving Average with eXogenous inputs (NARMAX) and Nonlinear Output Error (NOE) models. These classes of models are widely used because of their ability to capture nonlinear behaviors.

NARMAX model is a very powerful tool for modeling and prediction of dynamical systems (Gao and Foss, 2005; Johansen and Er, 1993; Yang et al., 2005). It is well suited for modeling systems using noisy outputs and noisy states. It generalizes the Nonlinear AutoRegressive with eXogenous inputs (NARX) model. Its regression vector is composed of the past inputs

[1] Multi-input and single-output (MISO) systems are considered here for ease of understanding. Results can be generalized to multi-input and multi-output systems.

u_k, the past measured outputs y_s, and the past prediction errors (difference between measured and predicted outputs) e. The output of the NARMAX model is given by:

$$\begin{aligned} y(t+1) \quad = \quad & F\big(u_1(t-d_{u_1}+1),\ldots, \\ & \ldots \\ & u_k(t-d_{u_k}+1),\ldots,u_k(t-d_{u_k}-n_{u_k}+2) \\ & \ldots \\ & y_s(t-d_{y_s}+1)\ldots,y_s(t-d_{y_s}-n_{y_s}+2), \\ & e(t-d_e+1),\ldots,e(t-d_e-n_e+2)\big) \\ & +e(t+1) \end{aligned} \tag{3}$$

where:
d_{u_k}, d_{y_s} and d_e are inputs, output, and error delays respectively;
n_{u_k}, n_{y_s} and n_e are inputs, output, and error orders respectively;
The prediction step in this representation corresponds to:

$$h = \min(d_{u_k}, d_{y_s}, d_e) \tag{4}$$

A NOE model is suited for system's simulation because it does not require measured outputs (Palma and Magni, 2004). The corresponding regression vector is composed of past inputs u_k and past simulated outputs $\hat{y}_u$. The output of the NOE model is given by:

$$\begin{aligned} y(t+1) \quad = \quad & F\big(u_1(t-d_{u_1}+1),\ldots, \\ & \ldots \\ & u_k(t-d_{u_k}+1),\ldots,u_k(t-d_{u_k}-n_{u_k}+2) \\ & \ldots \\ & \hat{y}_u(t-d_{\hat{y}_u}+1),\ldots,\hat{y}_u(t-d_{\hat{y}_u}-n_{\hat{y}_u}+2)\big) \\ & +e(t+1) \end{aligned} \tag{5}$$

Identification of recurrent models such as NARMAX or NOE models is a difficult task because some of the regressors have to be computed at each time step. The parameter estimation must then be carried out recursively.

3 MULTI-MODEL'S PRINCIPLE

Multi-models were first proposed by Johansen and Foss in 1992 (Johansen and Er, 1992). A multi-model is a system representation composed by a set of local models each of which is valid in a well defined feature space corresponding to a part of global system's behavior. The local validity of a model is specified by an activation function which tends to 1 in the feature space and tends to zero outside. The whole system's behavior can then be described by the combination of all local models outputs. Figure 1 presents the basic architecture of a multi-model. The relation (2) can then be expressed as:

$$\hat{y}(t+h) = \sum_{i=1}^{M} \omega_i\big(\underline{\xi}(t)\big)\ f_i\Big(\underline{\varphi}(t), \theta_i\Big) \tag{6}$$

where:
M is the number of local models;
$\omega_i(\cdot)$ is the activation degree of local model $f_i(\cdot)$, with :

$$\omega_i\big(\underline{\xi}(t)\big) \in [0,\ 1]\ , \quad \sum_{i=1}^{M} \omega_i\big(\underline{\xi}(t)\big) = 1 \quad \forall t$$

$\underline{\xi}(t)$ is the vector of indexing variables (variables whereby system's feature space is divided into subspaces (Orjuela et al., 2006));
θ_i is a parameter vector characterizing the local model $f_i(\cdot)$;
$f_i\big(\underline{\varphi}(t), \theta_i\big) = \hat{y}_i(t+h)$ is the predicted output of the ith local model.

In (6), the prediction step h may take any discrete value. It can also be specified by an appropriate choice of the time delays d_u, d_y and d_e of regressors in $\underline{\varphi}(t)$ (see equation (4)). So without loss of generality, we will assume that $h = 1$.

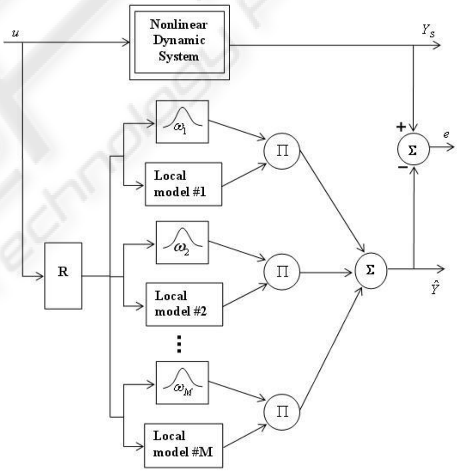

Figure 1: Basic architecture of a multi-model. Bloc R is a set of time delay operators combined with a linear or nonlinear transformation and used for the regression vector construction; Y_s is the measured system output.

Activation degrees of local models can be defined in a deterministic way using membership functions like gaussian functions, sigmoidal functions, etc. They can also be defined fuzzily using a fuzzy clustering of the system's feature space. This latter solution seems to be more natural as it allows to gather data which may have some "similarities". The main difficulty is the determination of the number of clusters. The proposed implementation combines architectural (number of local models or clusters) and parametrical identification. The number of clusters is successively incremented and the parameters are estimated at each step. The incrementation of the number of clusters is stopped when Akaike Information Criterion (see section §5) starts deteriorating.

The "fuzzy-c-means" algorithm (Bezdec, 1973) is implemented here because of its simplicity. This algorithm consists of maximizing the intra-cluster similarities and minimizing the inter-cluster similarities. The corresponding objective function is defined as:

$$J(c_1,c_2,\ldots,c_M)=\sum_{i=1}^{M}\sum_{t=1}^{N}\mu_{it}^m\, d_{it}^2 \qquad (7)$$

where:
$d_{it}=\|\underline{\varphi}(t)-\underline{c_i}\|$ denotes the distance between the observation $\underline{\varphi}(t)$ ($t=1,\ldots,N$, N - number of observations) and the center $\underline{c_i}$ of the ith cluster ($i=1,\ldots,M$, M - number of clusters or local models);
$\mu_{it}=\frac{1}{\sum_{k=1}^{M}(\frac{d_{it}}{d_{kt}})^{2/(m-1)}}$ represents membership degree of the observation $\underline{\varphi}(t)$ in the cluster i and stands for the local model's activation degree for that observation: $\mu_{it}=\omega_i\big(\underline{\xi}(t)\big)$;
$\underline{c_i}=\frac{\sum_{t=1}^{N}\mu_{it}^m}{\sum_{t=1}^{N}\mu_{it}^m\,\underline{\varphi}(t)}$ is the center of the ith cluster;
$m\geq 1$ is the "fuzzy exponent" and represents the overlapping shape between clusters (generally, $m=2$).

Local models may be of any structural type. As suggested in (Johansen and Er, 1993), local models may be defined as the first p terms of the Taylor's series expansion of the true (unknown) model $F_0(\cdot)$ about a point located in the local model's feature space. Affine local models ($p=1$) are mostly used because of their simplicity. This multi-model structure is very close to Takagi-Sugeno one. For complex systems, the number of linear local models may be very important because of the simplicity of their structure. We propose in this work polynomial local models with $p\geq 1$ which enable to enhance the handling of local nonlinearities, reducing then the number of models. We use a nonlinear transformation of the regression vector:

$$\underline{\varphi_p}(t)=g_p\big(\underline{\varphi}(t)\big)$$

where $g_p(\cdot)$ is a nonlinear transformation producing the new regression vector $\underline{\varphi_p}(t)$ which components are the products of elements of $\underline{\varphi}(t)$ at orders 1 to p. $\underline{\varphi_p}(t)$ can be easily obtained from the following procedure:
Let

$$\underline{\varphi}(t)=[\varphi_1 \quad \varphi_2 \quad \cdots \quad \varphi_{n_\varphi}]^T \qquad (8)$$

where n_φ is the dimension of $\underline{\varphi}(t)$.
Let us consider the following row vectors:

$$\begin{aligned}
V_{1,1} &= [\varphi_1 \quad \varphi_2 \quad \cdots \quad \varphi_{n_\varphi}]\\
V_{1,2} &= [\varphi_2 \quad \cdots \quad \varphi_{n_\varphi}]\\
&\ldots\\
V_{1,n_\varphi} &= [\varphi_{n_\varphi}]\\
V_{2,1} &= [\varphi_1 V_{1,1} \quad \varphi_2\, V_{1,2} \quad \cdots \quad \varphi_{n_\varphi} V_{1,n_\varphi}]\\
V_{2,2} &= [\varphi_2\, V_{1,2} \quad \cdots \quad \varphi_{n_\varphi} V_{1,n_\varphi}]\\
&\ldots\\
V_{2,n_\varphi} &= [\varphi_{n_\varphi} V_{1,n_\varphi}]\\
&\ldots\\
V_{p-1,1} &= [\varphi_1 V_{p-2,1} \quad \varphi_2\, V_{p-2,2} \quad \cdots \quad \varphi_{n_\varphi} V_{p-2,n_\varphi}]\\
V_{p-1,2} &= [\varphi_2\, V_{p-2,2} \quad \cdots \quad \varphi_{n_\varphi} V_{p-2,n_\varphi}]\\
&\ldots\\
V_{p-1,n_\varphi} &= [\varphi_{n_\varphi} V_{p-2,n_\varphi}]\\
V_{p,1} &= [\varphi_1 V_{p-1,1} \quad \varphi_2\, V_{p-1,2} \quad \cdots \quad \varphi_{n_\varphi} V_{p-1,n_\varphi}]
\end{aligned}$$

$\underline{\varphi_p}(t)$ is then obtained from the relation:

$$\underline{\varphi_p}(t)=[V_{1,1} \quad V_{2,1} \quad \cdots \quad V_{p,1}]^T \qquad (9)$$

For example if $\underline{\varphi}(t)=[\varphi_1 \quad \varphi_2 \quad \varphi_3]^T$ and $p=2$, then relation (9) gives:

$$\underline{\varphi_2}(t)=[\varphi_1 \;\; \varphi_2 \;\; \varphi_3 \;\; \varphi_1^2 \;\; \varphi_1\varphi_2 \;\; \varphi_1\varphi_3 \;\; \varphi_2^2 \;\; \varphi_2\varphi_3 \;\; \varphi_3^2]^T$$

The number of parameters n_{φ_p} of $\underline{\varphi_p}(t)$ may be very important if the size of $\underline{\varphi}(t)$ is important or if the order p is high.

For notation simplicity we will replace $V_{k,1}$ by V_k. Local models can then be expressed by the relation:

$$f_i\big(\underline{\varphi}(t),\theta_i\big)=\sum_{k=1}^{n_{\varphi_p}}V_k\theta_{i_k}+\theta_{i_0} \qquad (10)$$

where:
θ_{i_k} ($k=0\cdots n_{\varphi_p}$ and $i=1\cdots M$) are real constants;
$\theta_i=[\theta_{i_0} \quad \theta_{i_1} \quad \cdots \quad \theta_{i_{n_{\varphi_p}}}]^T$ parameters vector of the ith local model.
The main advantage of such a representation is that local models are nonlinear whereas they are linear with respect to parameters. This structure considerably simplifies parameter estimation (see §4). Equation (6) can be rewritten as:

$$\hat{y}(t+1)=\Phi(t)^T\;\theta \qquad (11)$$

where:
$\Phi(t)=\Big[\omega_1\big(\underline{\xi}(t)\big)\underline{\phi_e}(t)^T\cdots\omega_M\big(\underline{\xi}(t)\big)\underline{\phi_e}(t)^T\Big]^T$ is the global weighted regression vector;
$\underline{\phi_e}(t)=\big[\underline{\varphi_p}(t)^T\;1\big]^T$ is the extended regression vector;
$\theta=\big[\theta_1^T\ldots\theta_i^T\ldots\theta_M^T\big]^T$ is a concatenation of all local models parameters vectors;

Estimating θ can be carried out by using a global learning criteria J which consists of minimizing the error between system's output and multi-model's output:

$$J = \frac{1}{2}\sum_{t=1}^{N}\left(y_s(t) - \hat{y}(t)\right)^2 = \sum_{t=1}^{N}\left[\varepsilon(t)\right]^2 \qquad (12)$$

For non-recurrent multi-models with polynomial local models, J is linear with respect to the multi-model's parameters vector. J is minimized analytically using Least-Squares method. Multi-model parameters are then computed using the expression:

$$\hat{\theta} = (\Phi_g^T\,\Phi_g)^{-1}(\Phi_g^T\,Y_s) \qquad (13)$$

where:
$\hat{\theta}$ is the estimation of θ;
$\Phi_g = \left[\Phi(t)\right]_{t=1}^{t=N}$ is global weighted regression matrix of all observations;
$Y_s = \left[y_s(t)\right]_{t=1}^{t=N}$ is the vector of output values of all observations;

For recurrent multi-models, parameters are estimated by a parametrical adaptation algorithm using at each time step the values of $\Phi(t)$, $y_s(t)$ and $\omega_i[\underline{\xi}(t)]$ as presented in the next section.

4 MULTI-MODEL'S IMPLEMENTATION OF RECURRENT MODELS

Parameter estimation in recurrent neural network models is carried out iteratively using gradient based algorithm. Convergence towards global minimum is not guaranteed and convergence rate might be high. As it will be stated here, for the proposed recurrent multi-model (RMM), parameters are estimated using recursive least squares. Hence, the criterion J in relation (12) is computed up to time step k according to:

$$J(k) = \frac{1}{2}\sum_{t=1}^{k}\left[\varepsilon(t)\right]^2 = \frac{1}{2}\sum_{t=1}^{k}\left(y_s(t) - \Phi^T(t-1)\,\theta_k\right)^2 \qquad (14)$$

with θ_k the value of θ evaluated up to time instant k. The minimization of this criterion leads to:

$$\theta_k = \left[\sum_{t=1}^{k}\Phi(t-1)\,\Phi^T(t-1)\right]^{-1}\sum_{t=1}^{k}y_s(t)\Phi(t-1) \qquad (15)$$

Relation (15) can be written in a recursive form. Assuming

$$A_k = \left[\sum_{t=1}^{k}\Phi(t-1)\,\Phi^T(t-1)\right]^{-1} \qquad (16)$$

then

$$\theta_k = A_k\sum_{t=1}^{k}y_s(t)\Phi(t-1) \qquad (17)$$

$$\theta_{k+1} = A_{k+1}\sum_{t=1}^{k+1}y_s(t)\Phi(t-1) \qquad (18)$$

The sum in the right hand side of (18) can be transformed after some manipulations to:

$$\sum_{t=1}^{k+1}y_s(t)\Phi(t-1) = A_{k+1}^{-1}\theta_k + \Phi(k)\,\widetilde{\varepsilon}(k+1) \qquad (19)$$

where:
$\widetilde{\varepsilon}(k+1) = y_s(k+1) - \Phi^T(k)\,\theta_k$ is the *a priori* prediction error (the error at time instant $k+1$ evaluated with parameters computed up to time instant k). Putting (19) in (18) leeds to a recursive expression of θ:

$$\theta_{k+1} = \theta_k + A_{k+1}\,\Phi(k)\,\widetilde{\varepsilon}(k+1) \qquad (20)$$

A_{k+1} can also be computed recursively. From (16) one can write :

$$\left[A_{k+1}\right]^{-1} = \left[A_k\right]^{-1} + \Phi(k)\,\Phi^T(k) \qquad (21)$$

Applying matrix inversion lemma to relation (21), A_{k+1} is computed recursively:

$$A_{k+1} = A_k - \frac{A_k\,\Phi(k)\,\Phi^T(k)\,A_k}{1+\Phi^T(k)\,A_k\,\Phi(k)} \qquad (22)$$

So, the parameters vector θ is updated recursively at each time step using relations (22) and (20). This learning algorithm is used for the identification of NARMAX and NOE structures based on the RMM architectures (see figures 2 and 3).

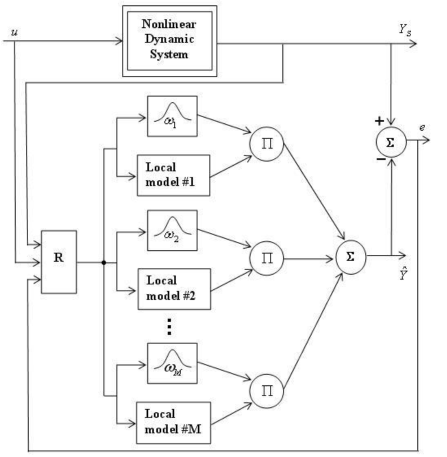

Figure 2: Recurrent multi-model implementation of a NARMAX model.

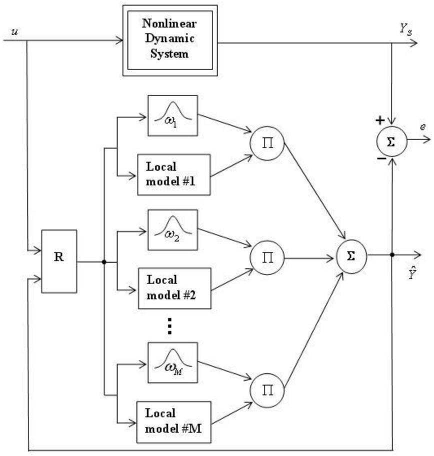

Figure 3: Recurrent multi-model implementation of a NOE model.

5 RESULTS AND DISCUSSION

To validate the proposed RMM architecture, two non linear systems are used. The first one is a simulated system which data are generated from a NARX model (Gasso, 2000). The second one is Box-Jenkins gas furnace benchmark (Box and Jenkins, 1970). For comparison purposes, we have implemented recurrent Multi-Layer Perceptron (MLP) with one hidden layer for NARMAX and NOE models, both trained with the Backpropagation Through Time (BPTT) algorithm (Werbos, 1990). To enhance the speed of learning with the BBTT algorithm, the learning rate is adapted so that it takes high values when the learning error decreases fastly and take small values when it decreases slowly.

Performances of recurrent multi-models with given order p of polynomial local models (RMM_p) are evaluated. Akaike Information Criterion (AIC) is used for model's parsimony estimation (least error with minimum parameters):

$$AIC = N \ln J + 2n_\theta \tag{23}$$

where n_θ denotes the number of model's parameters. Root Mean Square Error criterion (RMSE) is also used for performance evaluation in learning ($RMSE_L$) and validation ($RMSE_V$) phases. The architecture of models (Arch) specifies the number of local models in multi-models case or the number of hidden neurons in MLP case. Computation time (CT) during which models parameters are determined is used for algorithms convergence speed evaluation.

5.1 Example 1: Narx Dynamic Process

The following system is simulated in a noisy context and then identified using recurrent multi-model and recurrent MLP.

$$\begin{aligned} y_s(t) = \; & \frac{y_s(t-1)\left[0.5u_1(t-1) - 0.3u_2(t-1)\right]}{1 + y_s^2(t-1)} \\ & + 0.3u_1^2(t-1) - 0.5u_2^2(t-1) \end{aligned} \tag{24}$$

Exogenous input signals $u_1(\cdot)$ and $u_2(\cdot)$ are chosen to be pulses of random magnitude (in interval $[0,1]$) and different widths; the output signal is then corrupted by a white noise e issued from a normal distribution. The signal to noise ratio equals 14 dB. The obtained noisy output y_{s_n} (see figure 4) is expressed by:

$$y_{s_n}(t) = y_s(t) + e(t) \tag{25}$$

NOE model class is the most suitable one to identify this kind of system (Dreyfus, 2002). Both NOE RMM and NOE MLP models are implemented. The following regression vector is used:

$$\varphi(t) = [u(t-1) \; \hat{y}(t-1)]^T$$

where $\hat{y}$ is the estimated model output. The vector of indexing variables is:

$$\xi(t) = \left[u_1(t) \; u_2(t)\right]^T$$

Table 1 shows obtained results for NOE RMMs and a NOE MLP structures. System and NOE RMM_1 outputs are plot on validation data in figure 5. The obtained results show that multi-model structures have performances equivalent to MLP structures. However, their computation time is much lower.

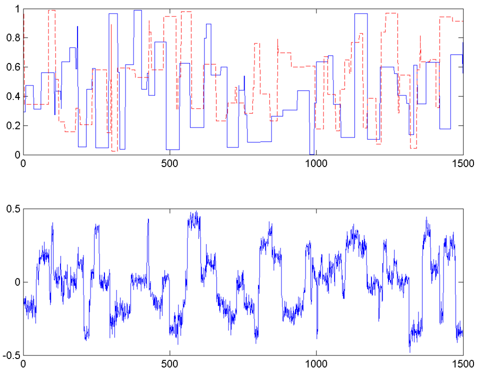

Figure 4: Inputs and output indentification data.

Table 1: NOE model for the nonlinear dynamic process: results for NOE RMM and NOE MLP structures.

Model	Arch.	AIC	CT(s)	$RMSE_L$	$RMSE_V$
RMM_1	7	-9595	8	0.040	0.010
RMM_2	3	-9617	6	0.040	0.007
MLP	3	-9587	199	0.040	0.006

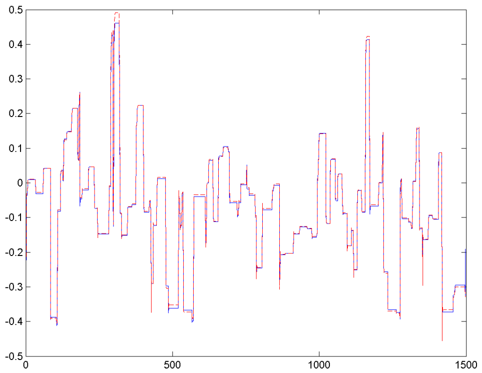

Figure 5: NOE RMM_1 output (dotted line) and noise free system output on validation data.

5.2 Example 2: Box-Jenkins Gas Furnace Benchmark

In this benchmark, data set are obtained from a combustion process of methane-air mixture. The process input is the methane gas flow into the furnace and the output is CO_2 concentration in the outlet gas (Box and Jenkins, 1970). System inputs and outputs are presented in figure 6. We have implemented and compared a NARMAX MLP and a NARMAX RMM structures based on the described methodologies. The following regression vector is used:

$$\begin{aligned}\underline{\varphi}(t) &= [u(t-1)\ u(t-2)\ u(t-3)\\ &\quad y_s(t-1)\ y_s(t-2)\ y_s(t-3)\ e(t-1)]^T\end{aligned}$$

The vector of indexing variables is:

$$\underline{\xi}(t) = \left[u(t)\ y_s(t)\right]^T$$

The results are presented in table 2. The NARMAX RMM has the best parsimony and gives best performances on validation data, with a very low computation time compared to the NARMAX MLP. It can be seen that high polynomial orders reduces the number of local models. Figure 7 shows process and NARMAX RMM_1 outputs on validation data.

Table 2: NARMAX model for Box-Jenkins gas furnace data: results for Multi-model and MLP structures.

Model	Arch.	AIC	CT(s)	$RMSE_L$	$RMSE_V$
RMM_1	6	-692	3	0.12	0.55
RMM_2	2	-637	2	0.13	0.63
MLP	3	-622	35	0.17	0.58

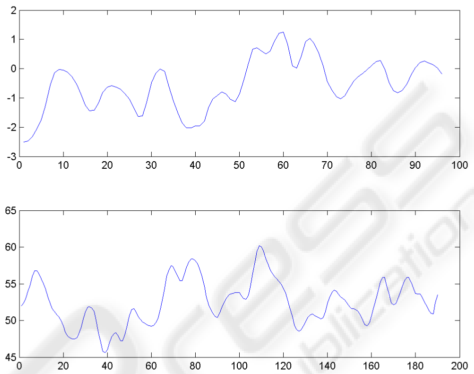

Figure 6: Process input and output on identification data.

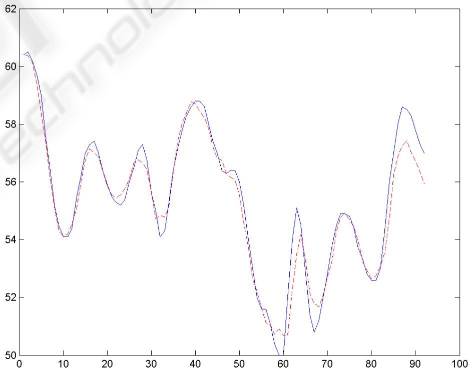

Figure 7: Process and NARMAX RMM_1 (dotted line) outputs on validation data.

6 CONCLUSION

In this work, a new recurrent multi-model structure with polynomial local models is proposed. The advantage of using polynomial local models is a better handling of local nonlinearities and reducing henceforth the number of local models. The proposed structure is used to implement NARMAX and NOE models.

Identification task is carried out very simply and obtained results show that the proposed recurrent multi-model has many advantages over recurrent

MLP model, among which the reduction of computation time. This is due to the way the parameters are estimated: least squares formula in the former model and iterative algorithm in the latter.

The perspective of this study is the implementation of the proposed structures for model predictive control in industrial processes.

REFERENCES

Bezdec, J. (1973). *Fuzzy mathematics in pattern classification*. PhD thesis, Applied Math. Center, Cornell University Ithaca.

Bielikova, M. (2005). Recurrent neural network training with the extended kalman filter. *IIT. SRC*, pages 57–64.

Boukhris, A., Mourot, G., and Ragot, J. (2000). Nonlinear dynamic system identification: a multiple-model approach. *Int. J. of control*, 72(7/8):591–604.

Box, G. and Jenkins, G. (1970). Time series analysis, forecasting and control. *San Francisco, Holden Day*, pages 532–533.

Cheng, Y., Karjala, T., and Himmelblau, D. (1997). Closed loop nonlinear process identification using internal recurrent nets. *Neural Networks*, 10(3):573–586.

Corradini, A. and Cohen, P. (2002). Multimodal speech-gesture interface for hands-free painting on virtual paper using partial recurrent neural networks for gesture recognition. *in Proc. of the Int'l Joint Conf. on Neural Networks (IJCNN'02)*, 3:2293–2298.

Dreyfus, G. (2002). *Rseaux de Neurones - Mthodologie et applications*. Eyrolles.

Elman, J. (1999). Finding structure in time. *Cognitive Science*, 14(2):179–211.

Gao, Y. and Foss, A. (2005). Narmax time series model prediction: feed forward and recurrent fuzzy neural network approaches. *Fuzzy Sets and Sytems*, 150:331–350.

Gasso, K. (2000). *Identification de systmes dynamiques non linaires: approche multi - modle*. thse de doctorat de l'INPL.

Huang, B., Rashid, T., and Kechadi, M.-T. (2005). A recurrent neural network recognizer for online recognition of handwritten symbols. *ICEIS*, 2:27–34.

Huang, B., Rashid, T., and Kechadi, M.-T. (2006). Multi-context recurrent neural network for time series applications. *International Journal of Computational Intelligence*, 3(1):45–54.

Johansen, T. and Er, M. (1992). Nonlinear local model representation for adaptive systems. In *Proc. of the IEEE Conf. on Intelligent Control and Instrumentation*, volume 2, pages 677–682, Singapore.

Johansen, T. and Er, M. (1993). Constructing narmax using armax. *Int. Journal of Control*, 58(5):1125–1153.

Jordan, M. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of IASTED International Conference of the Cognitive Science Society. (Reprinted in IEEE Tutorials Series, New York: IEEE Publishing Services, 1990)*, pages 531–546, Englewood Cliffs, NJ: Erlbaum.

Konur, U. and Okatan, A. (2004). Time series prediction using recurrent neural network architectures and time delay neural networks. *ENFORMATIKA*, pages 1305–1313.

Li, N., Li, S. Y., and Xi, Y. G. (2004). Multi-model predictive control based on the takagi-sugeno fuzzy models: a case study. *Information Sciences*, 165:247–263.

Murray-Smith, R. and Johansen, T. (1997). *Multiple Model Approaches to Modeling and Control*. Taylor and Francis Publishers.

Orjuela, R., Maquin, D., and Ragot, J. (2006). Identification des systmes non linaires par une approche multimodle tats dcoupls. *Journes Identification et Modlisation Exprimentale JIME'2006 - 16 et 17 novembre - Poitiers*.

Palma, F. D. and Magni, L. (2004). A multimodel structure for model predictive control. *Annual Reviews in Control*, 28:47–52.

Sjoberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P., Hjalmarsson, H., and Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica 31*, 31(12):1691–1724.

Tomasz, J. and Jacek, M. (1997). Neural networks tool for stellar light prediction. In *Proc. of the IEEE Aerospace Conference*, volume 3, pages 415–422, Snowmass, Colorado, USA.

Vartak, A., Georgiopoulos, M., and Anagnostopoulos, G. (2005). On-line gauss-newton-based learning for fully recurrent neural networks. *Nonlinear Analysis*, 63:867–876.

Vernieuwe, H., Georgieva, O., Baets, B., Pauwels, V., Verhoest, N., and Troch, F. (2004). Comparison of data-driven takagi-sugeno models of rainfall-discharge dynamics. *Journal of Hydrology*, XX:1–14.

Werbos, P. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Yang, W. Z., L.H., Y., and L., C. (2005). Narmax model representation and its application to damage detection for multi-layer composites. *Composite Structures*, 68:109–117.