computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

A COMPARATIVE STUDY BETWEEN A MULTI-MODELS BASED APPROACH AND AN ARTIFICIAL NEURAL NETWORK BASED TECHNIQUE FOR NONLINEAR SYSTEMS IDENTIFICATION

Lamine Thiaw, Mariusz Rybnik, Rachid Malti, Abdennasser Chebira, Kurosh Madani

Intelligence in Instrumentation and Systems Laboratory (I2S/JE2353 Lab.)
PARIS XII University, Senart-Fontainebleau Institute of Technology,
Bât.A, Av. Pierre Point, F-77127 Lieusaint, France,
{madani ; malti ; chebira}@univ-paris12.fr

Abstract: *Recently, a number of works propose multi-model based approaches to model non-linearsystems. Such approach could also been seen as some "specific" approach, inspired from ANN operation mode, where each neurone, represented by one of the local models, realizes some higher level transfer function. In this paper we present two different neural based approaches to such multiple models concept: one issued from conventional structure and the other based on self-organizing dynamic architecture. A comparative study between a multi-model based architecture and an ANN based one, in the frame of nonlinear system identification is reported.*

Keywords: *Multi-Model Structure, Neural Networks, Self-organizing Architecture, Multi-model generator, Identification, Nonlinear System, Comparative Study.*

1. INTRODUCTION

Identification of non-linear systems is an important task for model based control, system design, simulation, prediction and fault diagnosis. The identification task involves two essential steps: structure selection and parameter estimation. These two steps are linked and generally have to be performed in order to achieve the best compromise between error minimization and the total number of parameters in the final model. In real world applications (situations), strong linearity and large number of related parameters make the realization of those steps challenging, and so, the identification task difficult. A large variety of structures to take into account systems non-linearity have already been proposed among which, Wiener and Hammerstein type models [1], Volterra series [2], Fuzzy logic based models [3], [4] and especially in last decades, neural network based approaches, which applications are numerous in dynamical system modeling [6].

Recently, a number of works propose multi-model based approaches to avoid difficulties (modelling complexity) related to non-linearity ([7], [8] [9] and [12]). In a general way, in such approaches, a set of models, corresponding to a set of operating ranges contributes to identify the whole system. In other words, such approach could appear as some kind of weighted contribution of a set of models approximating the whole system's behaviour. Thus, it could also be seen as some "specific" approach, inspired from ANN operation mode, where each neurone (here represented by one of the local models) realizes some higher level transfer function. Concerning the learning process (taking into account the analogy with learning in ANN), it be performed on the level of contribution rate of each model to the global response, called also "activation degree" associated to each of participant models in such multi-model architecture.

In this paper we present a comparative study between a conventional multi-model architecture and an ANN based multi-model structure, in the frame of nonlinear system identification. We present first the principle of Multi-Model approach. Then, we will present the application of such multi-model based architecture to the non linear system identification. In section 4, we will introduce a self-organizing neural multi-model generator that we called T-DTS (Tree-like Divide To Simplify) paradigm [11]. In section 5 we will briefly describe a software tool, which has been designed and implemented to make possible multi-model based structures study through graphical user interface. The section 6 will present, on the basis of an ARMAX based non linear system, a comparative study between neural based identifier and a conventional multi-model structure performing the identification task. Finally, section 7 will conclude this paper.

2. GENERAL PRINCIPLE OF MULTI-MODEL APPROACH

The use of multi-model for identifying complex systems is due to the fact that it is not always possible to find an analytical relation which describes the system's behavior in its whole operating range. The complexity of the system can be considerably reduced if the operating range of the system is divided into different regions in which its local behavior could be described with relatively simple functions. This approach is at the base of multi-model theory.

A multi-model is composed of several models where each is valid in a well defined shape (interval) which corresponds to a part of the operation range of the system. The local validity of a model in a well defined interval is specified by using functions with limited supports witch tend to significantly increase the contribution of the local models in that zone and tend to decrease it elsewhere. The combination of all local models allows describing the whole system's behaviour.

Let us consider a non-linear dynamical system described by the general equation, expressed by relation (1), where F(.) – global unknown model, $\varphi(t)$ – regression vector. The associated multi-model $f_i(\varphi(t))$, composed by M local models and their weights $\rho_i(\varphi(t),\beta_i)$, is defined by the weighted average expressed in the relation (2), where $\rho_i(\varphi(t)\beta_i) \geq 0$ (for all i), $\sum_{j=1}^{M} \rho_i(\varphi(t),\beta_i) > 0$ (for all $\varphi(t)$) and β_i is a parameter related to the validity function ρ_i.

$$y(t) = F[\varphi(t)] \tag{1}$$

$$y(t) = \frac{\sum_{i=1}^{M} \rho_i(\varphi(t),\beta i) f_i(\varphi(t))}{\sum_{j=1}^{M} \rho_j(\varphi(t),\beta_j)} \tag{2}$$

Defining the "activation degree" of the i-th local model conformably to the relation (3), the general relation for multi-model architecture could then be written according to the relation (4), where ω (.) is the "activation degree" of the i-th local model.

$$\omega_i(\varphi(t),\beta) = \frac{\rho_i(\varphi(t),\beta_i)}{\sum_{j=1}^{M} \rho_j(\varphi(t),\beta_j)} \tag{3}$$

$$y(t) = \sum_{i=1}^{M} \omega_i(\varphi(t),\beta) f_i(\varphi(t)) \tag{4}$$

The general concept of Multi-modeling could be seen as some kind of artificial neural network structure, where the neurons activation functions are those corresponding to the local models transfer functions. The appropriated choice of the "activation degree" parameters could be seen here as some learning procedure. Generally the proposed multi-model structures use conventional (polynomial) local models. In this paper, we propose to use ANN based local models. Moreover, we present a dynamic multi-model able to self-organize the operating range clusters (intervals) and to adapt local models to the related operating range.

3. SYSTEM IDENTIFICATION AND MULT-MODELS CONCEPT

System identification with multi-models can be done in two steps: structural identification and parametric identification. Structural identification allows the specification of the operating zones and the structure of each local model. The specification of the operating zones consists of determining the characteristic space "Z" and its decomposition into different zones, each of which have a validity function. There are three different techniques for the characteristic space decomposition: grid partition, decision tree partition and classification. In our study, we use the first two techniques. In the grid partition technique, the support of each characteristic variable z_j is divided into many individual partitions p_j. The number of generated local models is given by relation (5), where n_z – number of characteristic variables and p_j – j-th individual partitions corresponding to the individual variable z_j.

$$M = \prod_{j=1}^{nz} p_j \tag{5}$$

The decision tree partition method allows to reduce the number of local models which can be very large in the case of grid partition. In each step of the decomposition, a new zone is added so that the number of local models is increased by one. The decomposition can be orthogonal [10]. The specification of the structure of each local model consists of determining the relation which describe the model in its validity domain.

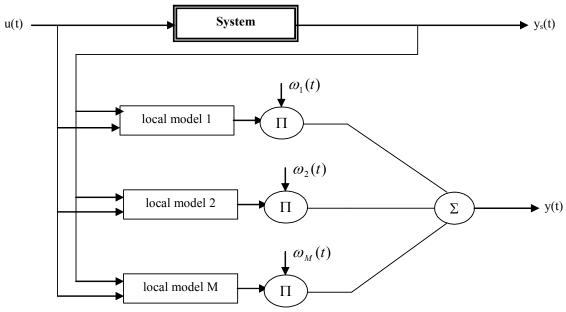


Fig. 1 – Bloc diagram of equation error based multi-model.

Depending of the regression vector construction, one can have three types of multi-model: parallel-series multi-model (equation error based multi-model – EE, figure 1), parallel multi-models with disconnected local models (output error based multi-models – OE, figure 2) and parallel multi-models with fully disconnected local models. In our study, we use the EE and OE multi-models.

The parametric one consist of the estimation of the local models parameters starting from the measurements on the system. The parameter estimations result from the minimization of the error between the system output and the multi-model output. One can plan to minimize the error between the system output and the local models outputs (local learning) or to minimize the error between the system output and the multi-model output (global learning). In the local learning case, a quadratic criterion is defined for each local model (locally) by the expression (6), where N– number of measurements.

$$J_i = \frac{1}{N} \cdot \sum_{t=1}^{N} \omega_i(t) \cdot \left(y_i(t) - y_s(t)\right)^2 \tag{6}$$

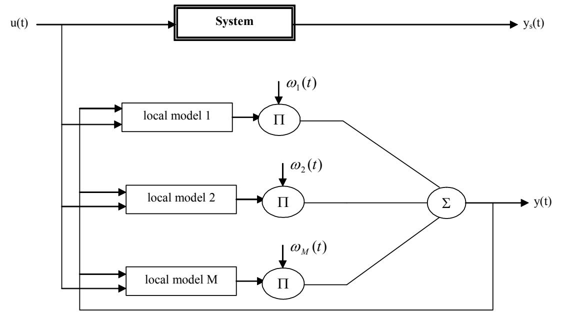


Fig. 2. Bloc diagram of output error based multi-model.

The local learning criterion for the multimodel is obtained by the relation (7) and in the global learning case, the criterion is computed by the relation (8).

$$J_L = \frac{1}{N} \cdot \sum_{i=1}^{M} \sum_{t=1}^{N} \omega_i(t) \cdot \left(y_i(t) - y_s(t)\right)^2 \tag{7}$$

$$J_g = \frac{1}{N} \cdot \sum_{t=1}^{N} \left(y(t) - y_s(t)\right)^2 \tag{8}$$

According to the criterion nature J, one distinguishes two types of optimization:

- Linear optimization: if the criterion J is quadratic compared to the parameters θ of the local models, one can use the least square method to set the parameters. In general the regression vector contains some delayed outputs which make the criterion not quadratic. But by replacing these delayed outputs by the corresponding measurements, the criterion becomes quadratic.
- Non - linear optimization: if the model is non-linear, the parameters can't be computed analytically and iterative techniques of optimisation are to be used. The estimation $\theta(k+1)$ of θ at the k+1'th iteration is obtained by the relation (9).

$$\hat{\theta}(k+1) = \hat{\theta}(k) - \eta_k \cdot H_k^{-1} \cdot G_k \tag{9}$$

where $\hat{\theta}(k)$ – estimation of θ obtained at the k'th iteration, η_k – "relaxation" control parameter, G_k – estimation of the gradient of the criterion at the iteration k, H_k – matrix which modifies the search direction. Its choice defines the non-linear optimization method. We use the Levenberg – Marquardt method for which H_k is the regularised Hessien matrix expressed by relation (10),

$$H_k = Ha_k + \lambda_k \cdot I \tag{10}$$

where Ha_k – the approximated Hessian, λ_k – coefficient of regularization, I – identity matrix.

4. NEURAL NETWORK BASED MULTI-MODELS APPROACHES

4.1 MULTI-LAYER PERCEPTRON (MLP) BASED MULTI-MODEL

In this kind of multi-models, the local model associated to each region of the system's feature space is a neural network. Mostly, a Multi Layer Perceptron (MLP) is considered. Each local network will learn a specified interval of the output. This principle is illustrated in the figure 3. That supposes availability of a learning database including samples of system's output with corresponding inputs.

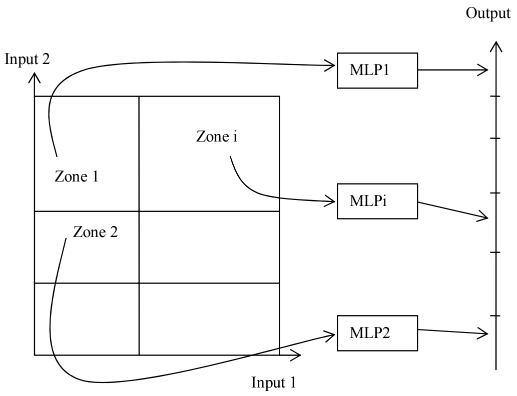


Fig. 3: Principles of ANN local models based multi-model with MLP-like neural models.

Supposing that for each operation interval i (i-th sub-feature-space) two sets I_i and O_i, containing input and output data, respectively, are available, the pair (I_i, O_i) will be used to train the corresponding local neural network (in our case an MLP-like neural network, with back propagation learning rule). During the operation phase (after learning), each unlearned input will be associated to the corresponding sub-feature-space and the related MLP based neural local model will generate the system's output. Contrary to the conventional multi-model, where the output is the weighted sum of the local model outputs, here only one local model will be activated generating the appropriated system's output.

Taking into account the above-mentioned specificities related to non linear system identification task, and based on multi-modeling concept, described in previous section, one of the key points on which one can act is the complexity reduction at both problem solution and processing levels. An issue could be the model's complexity reduction by splitting of a complex problem into a set of simpler problems: multi-modeling of a complex behavior. Another promising approach is to take advantage of processing hybridization.

4.2 SELF-ORGANIZING NEURAL BASED MULTI-MODEL GENERATOR

In biological structures the information processing complexity is reduced, but also enhanced, by natural self organization of nervous system and brain in dedicated (specialized) modules. An example of such specialization is the bird's auditory and motor pathways involved in the song recognition and the production (figure 4). So, multi-modeling could approach such structures if some adaptive self-organizing procedure involves in its structure construction.

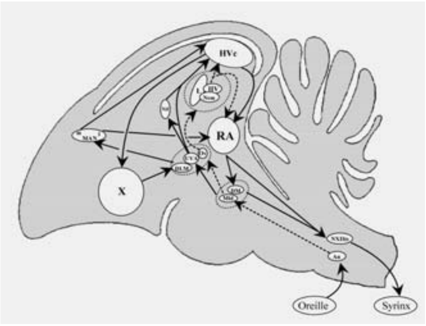


Fig.4: A schematic view of the left side of the bird's brain showing the auditory and motor pathways.

On this basis and inspired from animal brain structure, we have designed an ANN based data driven treelike Multiple Model generator, that we called T-DTS (Treelike Divide To Simplify).

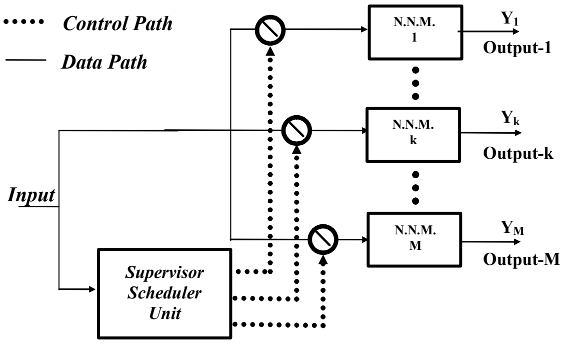


Fig.5: Bloc diagram of T-DTS Operation phase

T-DTS (Treelike Divide To Simplify) is a data driven neural networks based Multiple Processing (multiple model) structure. It is able to reduce complexity on both data and processing chain levels. T-DTS construct a treelike evolutionary neural architecture automatically, where nodes are ***Decision Units (DU)*** and leafs correspond to processing units called ***Neural Network Models (NNM)*** [11], as presented in figure 5. The structure decomposes complex task into several simpler ones to ease the processing. Each NN model covers a small distinct area of feature space. All together, they cover the whole feature space.

T-DTS operating modes:

The T-DTS operates according to two main operational modes. The first one, called "learning phase", concerns the T-DTS training and structuring. The second one, called "generalization phase", involves with unlearned data processing.

T-DTS learning phase:

The T-DTS learning operating mode is an important phase during which T-DTS builds decomposition structure and trains NNM models. In the learning phase, T-DTS algorithm constructs a neural tree structure dynamically and without operator's intervention. A given learning dataset is decomposed in a recursive way using Decomposition Units (i.e. unsupervised Neural Networks). To obtain decomposition decision many techniques can be used, including *classification complexity estimation* [11]. In this case, the decomposition decisions (*to-split-or-not-split*) are based on the splitting threshold, which is a parameter representing required degree of decomposition. We have used a data dispersion rule founded on a standard deviation based indicator (MaxStd), representing maximum tolerated data dispersion in a given subset.

After decomposition tree was built, learning database is decomposed into a set of learning sub-databases (see figure 5). The sub-databases are then used to train a corresponding set of NN Models. Each NN model covers a small area of feature space. Ensemble of Neural Network Models covers (model) the system behavior region-by-region in the problem's feature space. In this way, a complex problem is decomposed recursively into a set of simpler sub-problems: the initial feature space is divided into *M* sub-spaces.

T-DTS generalization phase:

In generalization phase, each incoming data pattern is assigned to a process by the most appropriate NN model. It can be done by interacting with decomposition tree structure or by evaluating the similarity with sub-databases prototypes (to find the most similar cases). Then, the most appropriated NN Model is authorized (activated) to process that pattern. The output is gathered from individual NN models to create output matrix for the whole generalization database. Let: Ψ(t) be the input ($\Psi(t) \in \Re^{n_\Psi}$), a n_Ψ-Dimensional vector and $Y_k(t) \in \Re^{n_Y}$ be the k-th ($k \in \{1, \cdots, M\}$) model's output vector of dimension n_Y; $F_k(.) : \Re^{n_\Psi} \to \Re^{n_Y}$ be the k-th NNM's transfer function; then the DU output $S(\Psi(t), p, \xi) \in B^M$ (with $B = \{0,1\}$) could be expressed by relation (12).

$$S(\Psi(t), p, \xi) = (s_1 \quad \cdots \quad s_k \quad \cdots \quad s_M)^T \qquad (12)$$

$$\text{with} \quad \begin{bmatrix} s_k = 1 & \textit{if } p = p_k \textit{ and } \xi = \xi_k \\ s_k = 0 & \textit{else} \end{bmatrix}$$

where p and ξ represent some parameters or conditions. Depending to S(Ψ, p, ξ) the processing of an unlearned input data will be performed by the selected NNM according to the relation (13).

$$Y(\Psi, t) = Y_k(t) = F_k(\Psi(t)) \qquad (13)$$

5. A SOFTWARE TOOL DESIGN AND IMPLEMENTATION

We use an object oriented approach for the creation in Matlab a class named "multimod" which represents a multi-model.

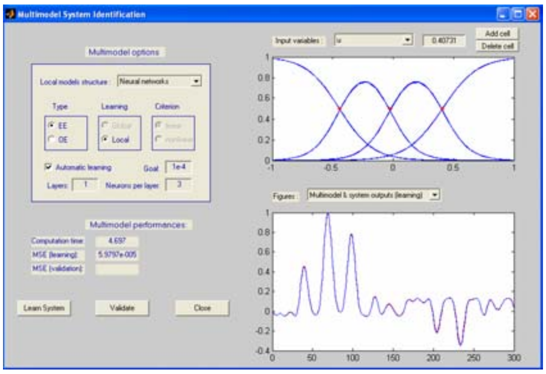
The main attributes of the "multimod" class are:

- The configuration parameters witch allow to specify the multi-model type to generate, (EE or OE), the learning mode (local or global) and the type of optimization (linear or non-linear).
- The parameters for the local models and the validity functions;
- The parameters characterizing the performances of the multi-model.

The main member functions of the class multi-mod are:

- multimod : constructor of the class;
- display : for displaying objects of the class ;
- AddLocalModels : to add local models ;
- DivideLocalModel : to divide local models into two different models ;
- DelLocalModel : to delete a local model ;
- AddGrid : to add a grid in the partitions ;
- DelGrid : to delete a grid ;
- Activate : to generate activation functions ;
- TrainMultimod : to train the multi-model with a system data ;
- ValidateSystem : to simulate the multi-model with not learned inputs ;
- OutMultiModel : to evaluate the output of the multi-model ;
- set and get methods which allow to access some data members of the class from outside ;

We simplified the use of the "multimod" class by creating a graphic interface allowing: the multi-model options setting (type, learning mode, optimization, decomposition), an interactive creation of characteristic variables partitions (decomposition and grid partition) and graphical results display.



Multimodel System Identification

Multimodel options

Local models structure : Neural networks

Type: ● EE ○ OE

Learning: ○ Global ● Local

Criterion: ● linear ○ nonlinear

☑ Automatic learning   Goal: 1e-4

Layers: 1   Neurons per layer: 3

Multimodel performances

Computation time: 4.697

MSE (learning): 5.9797e-005

MSE (validation):

Input variables : u   0.40731

Figures : Multimodel & system outputs (learning)

Fig. 6: A graphical user interface for multi-model system identification.

6. VALIDATION AND EXPERIMENTAL RESULTS

To compare the conventional and neural based multi-model performances we have constructed a benchmark based on a dynamic non linear system described by the following equations:

$$u(t) = 0.7\sin\left(\frac{2\pi t}{300}\right) + 0.3\sin\left(\frac{2\pi t}{30}\right) \qquad (14)$$

$$y(t) = 0.18y(t-1) + 0.3y(t-2) + 0.6u^3(t) + 0.18u^2(t) - 0.2u(t) \qquad (15)$$

The data for validation are obtained from the relation:

$$v(t) = 0.7\sin\left(\frac{2\pi t}{300}\right) \qquad (16)$$

The comparison criterions are: the estimation mean square quadratic errors (MSE) in learning and generalization and the processing delay. We have considered an EE multi-model with a global learning, a linear criterion and a decision tree partition. The conventional multi-model has been based on the splitting the initial complex problem into a set of linear simple models. Concerning the neural based multi-model identifier, we have considered two cases. One of them splits the initial model into a set of Multi-Layer Perceptrons (MLP) based models and the other one implements the T-DTS self-organizing concept.

Concerning the MLP neural networks, they are generated dynamically, starting from one hidden layer with one neuron and one output layer with one neuron too. The number of neurons in a hidden layer is then progressively increased. If adding neurons in hidden layer leads to the fixed global MSE, then no additional layer is generated. But if the performance of the network does not improve in a significant way or decrease, a new hidden layer is added. The MLP's number of neurons as well as its number of layers are increased until it reach the fixed goal.

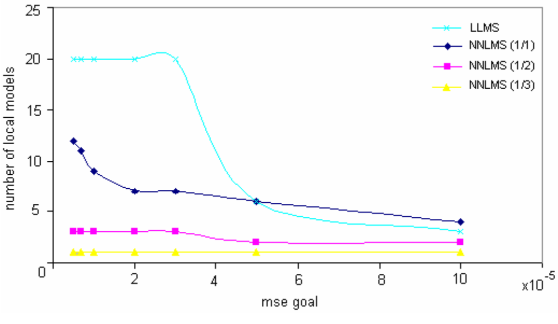


Fig. 7: Evolution of number of local models for different multi-models versus MSE goal.

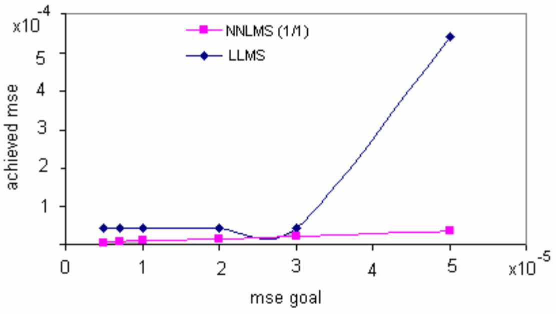


Fig. 8: Achieved MSE for linear local models based multi-model and neural based local models one.

We repeatedly fix a goal which is the maximum value of the MSE, train the models, evaluate the durations of the learning, and the reached MSE in training and in generalization. Results are presented in the figures 7 to 9. One can note that computing time of the neural network based multi-model is less than that of the multi-model (figure 9) whereas both models have practically the same MSE in training.

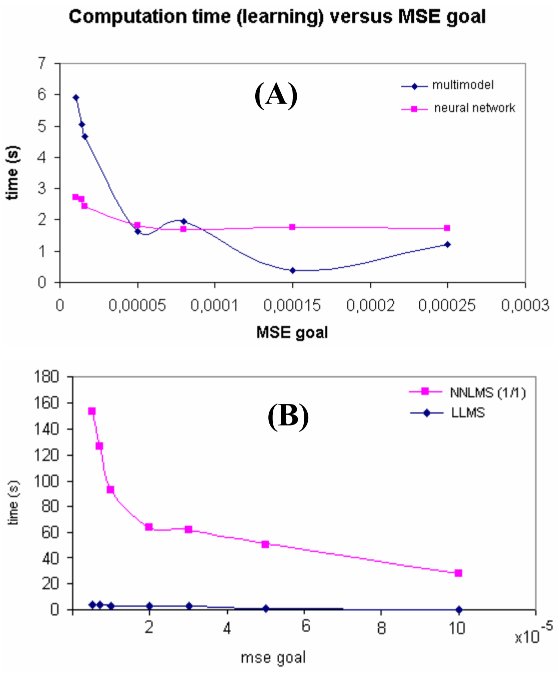


Fig. 9: Learning computing time comparison between a conventional and MLP based multi-models: learning time in the case of a one neuron hidden layer local model (A); average learning time (B).

Concerning the T-DTS based multi-model generator, results lead to the treelike structure represented in figure 10. The obtained multi-model includes 3 DU at the node level and 4 NNM at the leaf level. That means that the initial problem's feature space has been spitted into four sub-spaces (models). Firstly, two sub-spaces "1a" and "1b" have been identified. The algorithm achieves a first model (NNM 1) but decomposes the second subspace ("1b") into two sub-spaces "2a" and "2b". In the same way, the T-DTS based multi-model generator associates a neural model to the "2a" feature sub-space (NNM 2) and performs a new splitting for the subspace "2b" generating two new models.

Figures 11 and 12 give respectively, a graphic representation of learned profiles in each obtained feature sub-spaces and learning errors (local models convergence). Figure 13 compares the T-DTS issued model's output and the original one when the unlearned signal ($s^g(t)$) is used as input. As it could be remarked from that figure, the built model, identifying the system, is a faithful model: the difference between the system's output and the T-DTS based estimated output is very low (reaching a square mean error lees than 10^{-6}), which proves the T-DTS paradigm efficiency. Reducing initial database complexity drops NNM training time: only few epochs (recursion) are needed to reach a10^{-6} mean square error (50 epochs in the worst case).

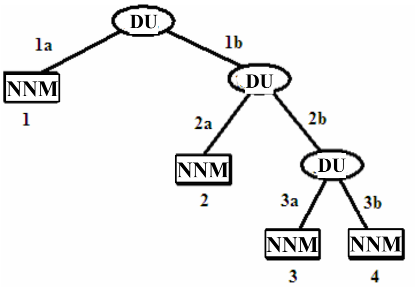


Fig. 10: Splitting pathway obtained for benchmark system identification example and number of models built by T-DTS.

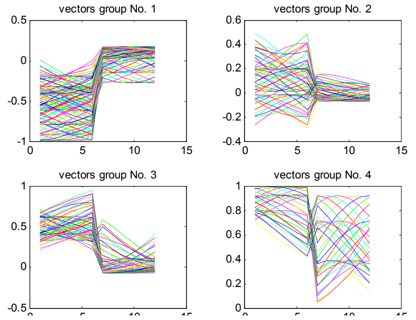


Fig. 11: Learning Sub-databases and corresponding patterns obtained after splitting.

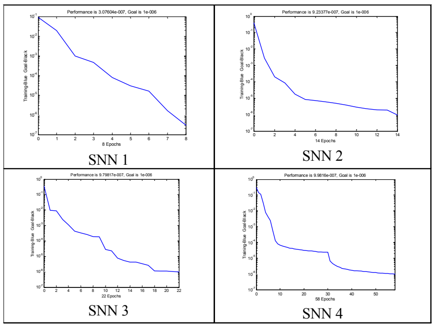


Fig. 12: Learning error Sub-databases and corresponding patterns obtained after splitting.

7. CONCLUSION

In this paper, we have presented a comparative study between conventional multi-model structure and a new approach of multi-modeling using local neural models. This new type of multi-model presents several advantages comparing to conventional structures (where multi-modeling is performed by some weighted contribution of local models). These advantages come directly from neural nature of local models.

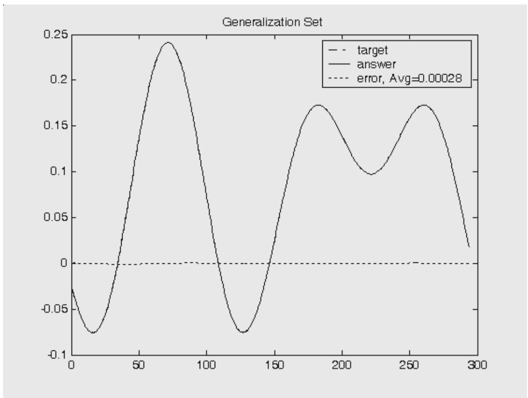


Fig. 13: Comparison between original process output and the T-DTS based identifier prediction.

The first significant advantage due to local models neural nature is the "local adaptivity" allowing possibility to adapt local models to the related feature sub-space's complexity: local models are constructed by "learning" the corresponding feature sub-space. Moreover, learning based local model construction offers attractive features to avoid the difficult problem of "appropriated models" selection. The second advantage is local models number reduction. En fact, the "universal approximation" capability of ANN, obtained by adjusting the number of hidden layer and the number of neurons in those layers, makes possible to consider feature sub-spaces with higher complexity (nonlinearity). However, some compromise is needed to avoid situations where NN local models complexity may affect execution delay performances of the whole multi-model structure.

If conventional Multi-models with linear local model structure can be very simply implemented, this advantage may not be ensured in the case of conventional Multi-models with other kinds of local models. In these cases, some compromises should also be obtained regarding execution time and local models implementation complexity.

Actually, we are working on two directions. Firstly, we are working on enhancement of the software tool making possible to realize comparison between neural local models based structure and other types of conventional multi-models. Secondly, we are working on compromise criteria to design adaptive splitting strategies ensuring local models implementation simplicity, multi-model's global performances and acceptable execution delay.

8. REFERENCES

[1] N. Wiener, Non linear Problems in Random Theory, *Technology Press MIT*, and *John Wiley, New York* (1958).

[2] M. Schetzen, The Voltera and Wiener Theories of Nonlinear Systems, *John Wiley, New York,* (1980)

[3] L. Zadeh, Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, *IEEE Trans. On Systems, Man and Cybernetics 3*, pp. 28-44

[4] T. Takagi and M. Sugeno, Fuzzy identification of systems and its application to modeling and control. *IEEE Trans. on Systems Man and Cyberneticc, Vol. 15*, (1985). pp. 116-132

[5] Boukhris, A, Mourot G. and Ragot J. (2000). Nonlinear dynamic system identification: a multiple-model approach. Int. J. of control, Vol. 72, N°7/8, pp. 591-604

[6] K.S. Narendra, and K. Parthasarath., Identification and control of dynamical systems using neural networks, *IEEE Trans. Neural Networks, Vol. 1, No. 1,* (1990).

[7] O. Nelles, On the identification with neural networks as series-parallel and parallel models, *Int. Conf. on Artificial Neural Networks (ICANN'95), Paris, France*, (1995).

[8] *Multiple Model Approaches to Modeling and Control, edited by R. Murray-Smith and T.A. Johansen, Taylor & Francis Publishers*, (1997), ISBN 0-7484-0595-X.

[9] M. Mayoubi, M. Schafer, S. Sinsel, Dynamic Neural Units for Non-linear Dynamic Systems Identification, *LNCS Vol. 930, Springer Verlag*, (1995), pp.1045-1051.

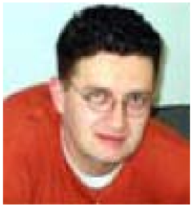
[10] S. Ernst, Hinging hyper-plane trees for approximation and identification, *37th IEEE Conf. on Decision and Control, Tampa, Florida, USA*, (1998)

[11] K. Madani, M. Rybnik, A. Chebira, Data Driven Multiple Neural Network Models Generator Based on a Tree-like Scheduler, *LNCS series*, *Edited by: J. Mira, A. Prieto - Springer Verlag* (2003), ISBN 3-540-40210-1, pp. 382-389

[12] K. Madani, M. Rybnik, A. Chebira, Non Linear Process Identification Using a Neural Network Based Multiple Models Generator, *LNCS series*, *Edited by: J. Mira, A. Prieto - Springer Verlag*, (2003). ISBN 3-540-40211-X, pp. 647-654.

Lamine Thiaw *received his Master Science degree in Energy and Process Control Engineering from PARIS XII – Val de Marne University, Creteil, France, in 1996. Since October 2003 he works as Ph.D. student in Intelligence in Instrumentation and Systems Laboratory of PARIS XII – Val de Marne University. His research work concerns multi-modeling and non linear process identification.*

Mariusz Rybnik *received his Master of Computer Science degree from Bialystok Technical University, Bialystok, Poland, in 2001. Since October 2001 he works as Ph.D. student in Intelligence in Instrumentation and Systems Laboratory of PARIS XII – Val de Marne University. His research work deals with self-organizing neural network based multi-modeling and hybrid neural based information processing systems.*



Rachid Malti *received his PhD degree in Automatic Control in 1999 from Institut National Polytechnique de Lorraine (INPL), Nancy, France, in 1999. Since September 2001 he works as Paofessor Assistant at Senart Institute of Technology of PARIS XII – Val de Marne University. He is a staff researcher in Intelligence in Instrumentation and Systems Laboratory of this University. His current research works deal with self-organizing neural network based multi-model and hybrid neural based information processing systems.*

Abdennasser Chebira *received his Ph.D. degree in Electrical Engineering and Computer Sciences from University PARIS XI (PARIS-SUD), Orsay, France, in 1994. Since September 1994 he works as Paofessor Assistant at Senart Institute of Technology of PARIS XII – Val de Marne University. He is a staff researcher in Intelligence in Instrumentation and Systems Laboratory of this University. His current research works concern self-organizing neural network based multi-modeling, hybrid neural based information processing systems, Neural based data fusion and complexity estimation.*

Prof. ***Kurosh Madani*** *received his Ph.D. degree in Electrical Engineering and Computer Sciences from University PARIS XI (PARIS-SUD), Orsay, France, in 1990. In 1995, he received the DHDR senior research doctorate degree from University PARIS XII – Val de Marne. Since 1998 he works as Chair Professor in Electrical*

Engineering of Senart Institute of Technology of University PARIS XII – Val de Marne. He is head of Intelligence in Instrumentation and Systems Laboratory of PARIS XII – Val de Marne University. His current research interests include large ANN structures behavior modeling and implementation, hybrid neural based information processing systems and their software and hardware implementations, design and implementation of real-time neuro-control and neural based fault detection and diagnosis systems. Since 1996 he is a permanent member (elected Academician) of International Informatization Academy. In 1997, he was also elected as Academician of International Academy of Technological Cybernetics.