SELF-ORGANIZING LOCAL MODELS FOR NONLINEAR SYSTEMS MODELLING

Dimitrios Moshou, Bart De Ketelaere, Josse De Baerdemaeker, Herman Ramon,
Department of Agro-Engineering and Economics, Laboratory of Mechanical Engineering, K.U. Leuven, Kardinaal Mercierlaan 92, 3001 Heverlee, BELGIUM
Phone: +32-16-321478, Fax: +32-16-321994,
email:{dimitrios.moshou, bart.deketelaere, josse.debaerdemaeker, herman.ramon}@agr.kuleuven.ac.be

ABSTRACT: A new method for on-line system identification based on the Self-Organizing Map is presented. The standard Self-Organizing Map (SOM) is extended with Local Models. To every node in the SOM along with the input weight two output weights are assigned: one that stores the output part of an input-output pair and one that stores the local gradient matrix (Jacobian) that is calculated from the training pairs. A training algorithm for the Jacobian matrices is derived. The method is tested in system identification of two nonlinear systems: a highly nonlinear system and a flexible agricultural spray boom.

KEYWORDS: Neural Networks, Self-Organizing Systems, Identification Algorithms, Nonlinear Systems, Agriculture, Machinery

INTRODUCTION

Many systems exhibit uncertain, time-variable or nonlinear behaviour. Because of this there is a need for flexible identification methods that can identify on-line the behaviour of a system for use in Adaptive or Optimal Control schemes. Such flexibility is inherent in methods like Neural Networks.

The Self-Organizing Map commonly referred to as SOM (Kohonen, 1982) is a neural network (NN) that converts complex, nonlinear statistical relationships between high-dimensional data into simple geometric relationships.

However the original SOM is known to perform poorly on regression problems because of the discrete representation of the data. Also it cannot by itself represent input-output relationships. By extending the SOM with output weights that store the output part of a mapping together with local gradient information, a first order expansion around the representative output can provide the original algorithm with the ability to approximate continuous relationships. Such a network is generally called an LLM (Local Linear Map) network. LLM networks have been introduced earlier (Ritter, 1991; Ritter, *et al*., 1992). Local Linear Map Networks have already been applied in time-series prediction (Martinetz, *et al*., 1993) and in non-linear prediction of particle trajectories (Walter, *et al*., 1990).

In the current paper for first time an LLM network is used for system identification. A novel training algorithm is presented for updating the parameters of this network and a proof is given for the updating equation. Then, this training algorithm is successfully applied in the on-line system identification of a highly nonlinear system frequently used as a benchmark for identification algorithms and an agricultural structure: a flexible spray-boom that is used for pesticide application. The flexible spray boom that has been used as the second identification example shows a quite nonlinear behaviour due to its complex structure.

EXTENDED SOM WITH LLMs

The Self-Organizing Map (Kohonen, 1995) is a neural network (NN) that maps signals ($\mathbf{x}$) from a high-dimensional space to a one- or two-dimensional discrete lattice of neuron units ($\mathbf{s}$). Each neuron stores a weight ($\mathbf{w_s}$). The map preserves topological relationships between inputs in a way that neighbouring inputs in the input space are mapped to neighbouring neurons in the map space. When extended with output weights ($\mathbf{y_s}$) it can actually learn in a supervised way the mapping $\mathbf{y}=\mathbf{f}(\mathbf{x})$. By utilising Local Linear Mappings, the approximation accuracy is increased because gradient information of the actual function is used to produce a first-order (Taylor) expansion around the representative output weight leading to the approximation

$$\mathbf{y}_{net}=\mathbf{y_s}+\mathbf{A_s}(\mathbf{x}-\mathbf{w_s}) \tag{1}$$

where $\mathbf{A_s}$ denotes an approximation of the local gradient of function $\mathbf{f}(\mathbf{x})$ around $\mathbf{w_s}$. The learning algorithm for the input and output weights is derived from the original Kohonen algorithm (Kohonen, 1982) and has as:

$$\Delta\, \mathbf{w_s}^{(in)} = \varepsilon h(\mathbf{x}-\mathbf{w_s}^{(in)}) \tag{2}$$

$$\Delta\, \mathbf{w_s}^{(out)} = \varepsilon' h' (\mathbf{x}-\mathbf{w_s}^{(out)}) \tag{3}$$

where ε, ε' and h, h′ are the learning rates and the neighborhood kernels respectively. Here with $\mathbf{w_s}^{(out)}$ the output weight $\mathbf{y_s}$ is denoted. The neighbourhood kernels used have the form of a Gaussian like:

$$h = \exp(-\|\mathbf{x}-\mathbf{w_s}\|^2/\sigma^2) \tag{4}$$

where $\|\cdot\|$ denotes the Euclidean norm and σ denotes the variance of the Gaussian distribution.
The learning algorithm for the Jacobian matrices ($\mathbf{A_s}$) is derived from Kohonen's learning rule.
The updating of the output weights is performed also on the enhanced approximation that uses the first-order expansion (1). This leads to the updating equation:

$$\Delta\, \mathbf{y_{net}} = \varepsilon'' h''(\mathbf{y}-\mathbf{y_{net}}) \tag{5}$$

By assuming that $\mathbf{y_s}$ has already been updated, so for this purpose is kept constant, the first order approximation (1), substituted in (5) gives:

$$\mathbf{A_s}^{(new)}(\mathbf{x}-\mathbf{w_s}^{(in)}) = \mathbf{A_s}^{(old)}(\mathbf{x}-\mathbf{w_s}^{(in)})+\varepsilon'' h'' (\mathbf{y}-\mathbf{y_{net}}) \tag{6}$$

By using properties of the pseudoinverses of vectors the updating scheme for the Jacobian matrices becomes:

$$\Delta\, \mathbf{A_s} = \varepsilon'' h'' (\mathbf{y}-\mathbf{y_{net}})(\mathbf{x}-\mathbf{w_s})^{+} \tag{7}$$

where the symbol $^{+}$ denotes the pseudoinverse and ε'', h″ are the learning rate and the neighborhood kernel respectively. It must be noted that (7) is a least squares approximation of (6). Hence the learning algorithm for the Jacobian matrices becomes:

$$\Delta\, \mathbf{A_s} = \varepsilon'' h'' (\mathbf{y}-\mathbf{y_{net}})(\mathbf{x}-\mathbf{w_s})^{T}/\ \|\mathbf{x}-\mathbf{w_s}\|^2 \tag{8}$$

It must be noted that from relation (8) the updates of the Jacobian matrices are proportional to the outer product of the error vector ($\mathbf{y}-\mathbf{y_{net}}$) and the normalised version of ($\mathbf{x}-\mathbf{w_s}$). In the following applications, equations (2), (3) and (8) are used as the updating equations and the network estimate is calculated from (1). This learning algorithm for LLM SOMs can be applied to all regression problems.

SYSTEM IDENTIFICATION

The learning algorithm of section 2 is applied in on-line system identification. System identification with neural networks (NN), based on system inputs and outputs, can be performed using a parallel or a series-parallel structure. In both structures the input of the network are vectors of previous input and output values of the system determined through a sliding time window of a certain length. The output of the network is the next step output of the system. In a parallel structure, the outputs of the network itself are fed back to the network through a tapped delay line (TDL) together with the excitation inputs of the system that are also fed back to the network through a second tapped delay line. In a series-parallel structure, the outputs of the real system instead of the outputs of the network are fed back to the network through a tapped delay line (TDL) like in figure (1). The Series-Parallel structure is the most commonly used structure in neural network based system identification.

The use of SOMs to cluster concatenated sequential data has already been attempted by Kangas (1990). In the present paper is shown that the expansion around the representative output weights provides the SOM algorithm with the ability to provide a quite accurate smooth estimate of the output compared with the discrete representation provided by the original algorithm.
The input data form vectors of a high dimension that are mapped on a two-dimensional Self-Organizing Map. At the same time the measured outputs are learned together with the Jacobian matrices ($\mathbf{A}_s$).

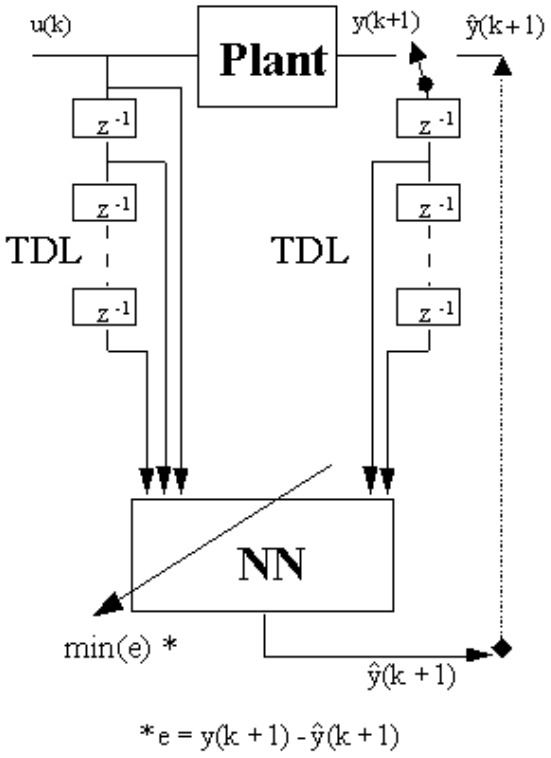

Figure 1: Series-Parallel Identification using a Neural Network (NN).

NONLINEAR SYSTEM IDENTIFICATION

The nonlinear system chosen in this example comes from (Narendra, *et al*., 1990) and has the form:

$$y(k+1) = \frac{y(k)}{1+y(k)^2} + u(k)^3 \qquad (9)$$

The nonlinearity of this system is considered as ‘hard’. Several sizes were used for the moving time windows but the best approximation was given when the size of the tapped delay lines matched the structure of the system which means that the best results were obtained with a size of one delay for both the previous inputs and outputs. Consequently one can say that the system was parametrized by the network in the form:

$$y_{net} = \hat{y}(k+1) = NN(y(k),u(k)) \tag{10}$$

where the hat (^) denotes the network's estimate for the system output at time-step t=k+1 as predicted from the network's inputs at step t=k.

The best results were obtained with a 10x10 network that was trained up to 5000 epochs. The training signal was chosen as random in the interval [-2,2]. This interval was chosen because the cube nonlinearity of the system is more evident in this interval than if the input is taken from [-1,1]. The initial and final settings of the learning parameters, namely the learning rates ε, and the widths σ of the gaussian kernels h for the updating equations were chosen to be $\varepsilon_I = \varepsilon_I' = \varepsilon_I'' = 0.9$ (initial), $\varepsilon_f = \varepsilon_f' = \varepsilon_f'' = 0.05$ (final), $\sigma_I = \sigma_I' = \sigma_I'' = 0.3$x(number of units in one dimension of the map) (initial), $\sigma_f = \sigma_f' = \sigma_f'' = 0.01$x(number of units in one dimension of the map) (final). The values of these parameters were chosen to decrease exponentially with time between the initial and the final value.

This network resulted in an RMSE (root mean square error) of 0.24 which, given the range and the variance of the outputs, translates in an NRMSE (normalized RMSE) of 0.08 at the end of the training session. The normalized RMSE is calculated from the RMSE when it is divided by the standard deviation of the actual system output. That means that if the average of a function is predicted the NRMSE tends to 1 and for perfect prediction it tends to 0.

The NRMSE is calculated from the following equation:

$$NRMSE = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y-\hat{y})^2}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y-\bar{y})^2}} \tag{11}$$

In this equation N denotes the number of samples, y denotes the measured value, $\hat{y}$ denotes the neural network estimate, and $\bar{y}$ the mean value of the N samples. The evolution of the NRMSE during learning for the highly nonlinear system given by relation (9) is shown in figure (3). Further testing with another random signal lying in the same interval [-2,2] and up to 500 epochs resulted in an NRMSE of around 0.09 while without the LLMs (SOM only) one obtains an NRMSE of about 0.38. Just for comparison linear time-domain identification by means of ARMAX models (na=0…5, nb=1…5, nk=0...2) gives the lowest NRMSE of 0.41 which means that it gives a bad estimate.

A way of visualising the spatial structure of the representative vectors that the Self-Organizing Map has stored is by plotting these vectors in the case that they are also two-dimensional like the SOM itself. In the case of a higher dimension of the input data the geometrical relations of the representative vectors are difficult to visualise. However in this case the weights (or representative vectors) are two-dimensional, so plotting them can provide us with useful information. By plotting the weights (figure 2), the dependence of the map formation on the cube of the input signal as given from equation (9), is clearly visible. Also the relative amplitudes of the learned gradients that are projected on the map approximate correctly the real gradient which is of second order.
Part of the testing session (figure 4) shows clearly the benefit of the LLMs. The prediction that uses LLMs is hardly discriminable from the actual system output while the plain SOM approximation clearly deviates from the desired values.

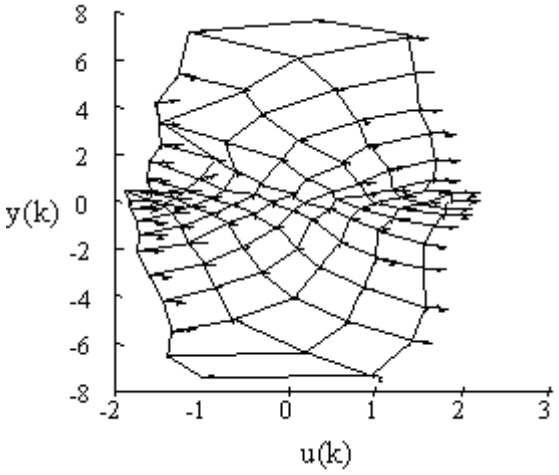

Figure 2: The learned gradients projected on the SOM.

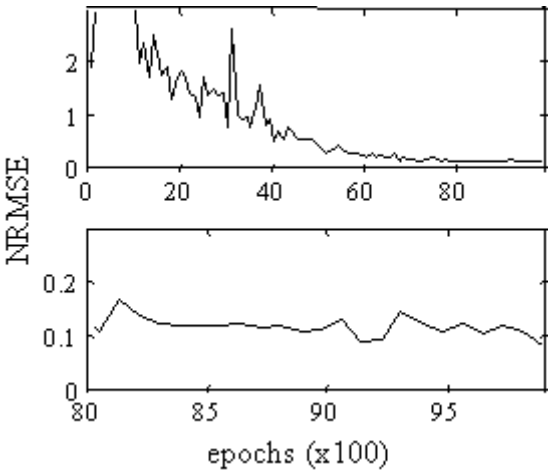

Figure 3: First subplot: The NRMSE for the highly nonlinear system during training. Second subplot: The NRMSE at the end of training (final value = 0.08).

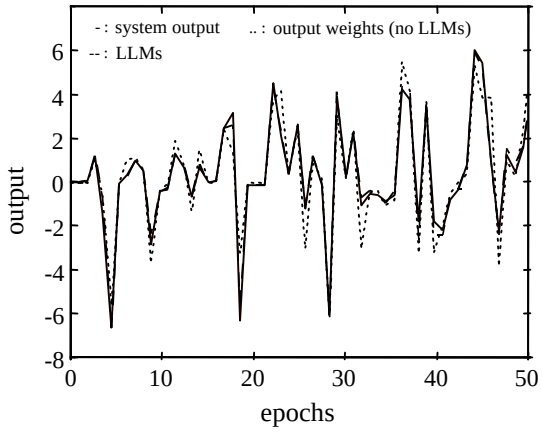

Figure 4: Part of the testing session that shows LLMs vs system output.

SPRAY BOOM IDENTIFICATION

The second system that has been used for testing the above mentioned identification method was a flexible spray boom (figure 5) with a total length of 12 metres from tip to tip. Only the left arm is shown in the figure (the structure

is symmetric around the center). Flexible spray booms are used for pesticide application in Agriculture. They are attached to tractors and are driven over fields in order to apply the pesticides. Because very frequently they have lengths between 12 m and 40 m, they have to be made from very lightweight material. The combination of material and structure gives a very low weight with respect to the size of the boom. The problem that arises then is the flexible behaviour that becomes dominant in the horizontal direction when driven on rough fields. Due to backlash and friction in the joints and torsion in the links the spray boom exhibits a quite complex nonlinear behaviour.
A similar vibration problem is experienced with flexible space structures because they have to be very lightweight but also very robust to disturbances.
A hydraulic cylinder (not shown) excited the boom in the horizontal direction near to the connection joint (hinge) at the center of the boom. The rotations of the boom were constrained by the use of a fixture, so that only translational modes were excited. As output of the system, the acceleration measured at one of the boom tips was selected. A monoaxial accelerometer was used to measure the acceleration in the horizontal direction. The excitation used, was a multisine shaped voltage, taking values in the interval [-3,3] Volt and with frequencies between 0.3 and 5 Hz. The multisine is a superposition of harmonically related sinusoids (Schoukens and Pintelon, 1991). An antialiasing filter with a cutt-off frequency of 10 Hz was used. The sampling frequency was selected equal to 100 Hz.
The coherence between the input and output signals is an indicator of linearizability of the relation between the two signals and is defined by the following equation:

$$g^2 = \frac{S_{xy}{}^2}{S_{xx} S_{yy}} \tag{12}$$

In equation (12) by S_{xy} the cross-spectral density of signals x and y is denoted, while, when the two signals are identical, we have an autospectral density. The coherence between the hinge and tip accelerations is plotted in figure (6). From figure (6) it is evident that the coherence between the two accelerations is low, so a linear model will give a very bad fit.
A 5x5 LLM SOM trained for 5000 epochs gave excellent results for the hinge to tip acceleration identification. A TDL of length 5 was used for the input and output measurements. Therefore the input to the LLM SOM had a dimension of 10. As output the tip acceleration of the next time-step was used.
For all the training sessions, the initial and final values of the learning parameters (learning rate and neighborhood radius) were chosen similar to that of the example with the nonlinear system of subsection 3.1 (decreasing exponentially with time). The NRMSE during training is plotted in figure (7).
Testing on a noise signal for 2500 epochs with amplitudes from the same interval and with the same frequency content gave an NRMSE of 0.09, a value which was almost the same with that near the end of the training session (0.07). The actual output data against the network's estimate is given in figure (8).

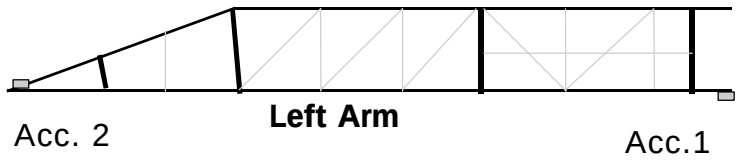

Figure 5: Configuration for spray-boom measurements (Acc. 1 denotes the tip accelerometer and Acc. 2 the hinge accelerometer).

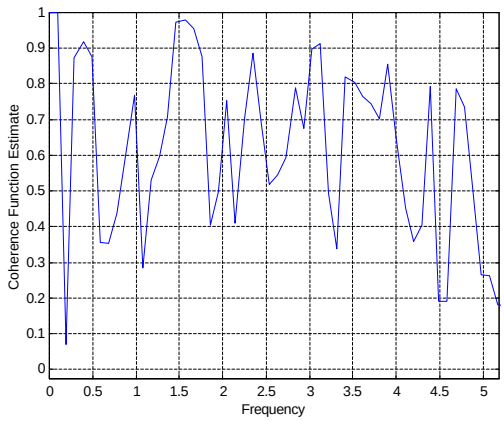

Figure 6: The Coherence between the hinge and tip accelerations for the flexible spray boom (frequency in Hz).

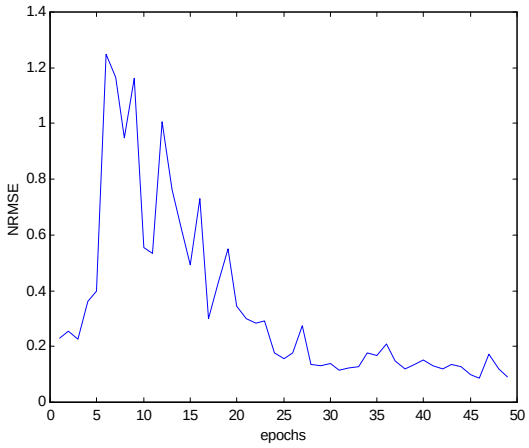

Figure 7: The NRMSE during training (every 100 epochs).

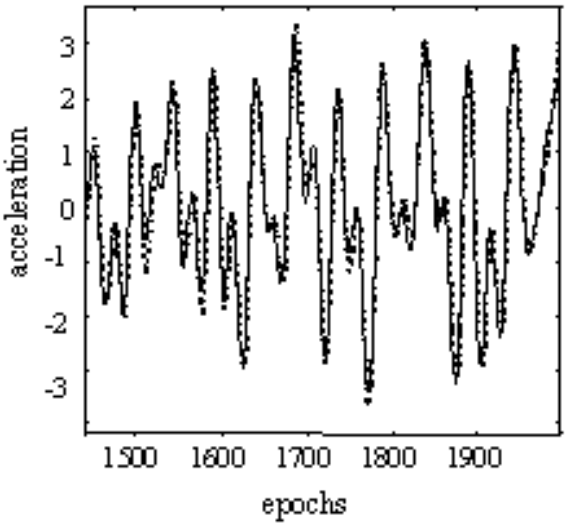

Figure 8: Estimated boom tip acceleration (dotted) vs measured (continuous).

CONCLUSIONS

A new neural network method for system identification has been presented. The main advantage of this method is the local representation of the data accompanied by a local updating algorithm. Local updating algorithms assure much faster convergence than global updating algorithms like for example the backpropagation algorithm for training multilayer perceptrons (MLPs). The method has been tested in system identification of a highly nonlinear system and an agricultural machine with promising results. However the LLM SOM by using the novel learning algorithm that is proposed in section 2 can be used for the solution of general regression problems. This method can be used equally well for on-line system identification of linear and nonlinear systems or systems with changing parameters because it uses Local Linear Mappings as proven from the results of this paper. Furthermore the gradient information that is stored in the learned Jacobian matrices can be used for several other interpolation schemes (e.g. the construction of fuzzy membership functions).

REFERENCES

Kangas, Jari, 1990, "Time-Delayed Self-Organizing Maps". Proceedings of the IEEE IJCNN-90 Conference, S. Diego, CA, vol. 2, pp. 331 - 336.

Kohonen, Teuvo, 1982, "Self-Organized Formation of Topologically Correct Feature Maps", Biological Cybernetics, vol. 43, pp. 59 - 69.

Kohonen, Teuvo, 1995, "Self-Organizing Maps", Springer Series in Information Sciences.

Martinetz, Thomas, Ritter, Helge, Schulten, Klaus, 1990, "Three-Dimensional Neural Net for Learning Visuomotor Coordination of a Robot Arm", IEEE Trans.on Neural Networks, vol. 1, pp. 131 - 136.

Martinetz, Thomas, S. Berkovich, Schulten, Klaus, 1993, "Neural-Gas Network for Vector Quantization and its Application to Time-Series Prediction", IEEE Trans.on Neural Networks, vol. 4, pp. 558 - 568.

Narendra, Kumpati, K. Parthasarathy, 1990, "Identification and Control of Dynamical Systems using Neural Networks" IEEE Transactions on Neural Networks, vol. 1, pp. 4 - 27.

Ritter Helge, Martinetz, Thomas, Schulten, Klaus, 1992, "Neural Computation and Self-Organizing Maps: An Introduction", Adisson-Wesley, New York.

Schoukens, Johan, Pintelon, Ric, 1991, "Identification of linear systems, A practical guideline to accurate modeling", Pergamon Press.

Walter, Jörg, Ritter, Helge, Schulten, Klaus, 1990, "Non-linear Prediction with Self-Organizing Maps", Proceedings of the IEEE IJCNN-90 Conference, S. Diego, CA, vol. 3, pp. 589 - 594.