ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 115-120

# Dynamic functional – link neural networks genetically evolved applied to system identification

Teodor Marcu* and Birgit Köppen-Seliger

University of Duisburg-Essen, Institute of Automatic Control (AKS)
Bismarckstrasse 81 (BB), D-47057 Duisburg, Germany
*Phone: +49-203-3794293; Fax: +49-203-3792928;
e-mail: {t.marcu, bks}@uni-duisburg.de

Abstract: The contribution concerns the design of a generalised functional-link neural network with internal dynamics and its applicability to system identification by means of multi-input single output non-linear models of auto-regressive with exogenous inputs' type. An evolutionary search of genetic type and multi-objective optimisation in the Pareto-sense is used to determine the optimal architecture of that dynamic network. The minimised objectives characterise the accuracy of the network and its complexity. Two case studies are included, referring to the identification of an evaporator from a sugar factory, and of a hydraulic looper from a hot rolling mill plant.
Keywords: non-linear system identification; dynamic functional-link network; multi-objective optimisation; genetic algorithms; industrial processes.

## 1. Introduction

Artificial Neural Networks (ANNs) are commonly used as a data-based technique in performing non-linear system identification of complex processes. For this purpose, models with adequate memory are required. Therefore, the ANNs have to be provided with dynamic elements and appropriate learning methods [4]. A first approach refers to neural networks with external dynamics, e.g. static ANNs equipped with tapped delay lines [4,8], the latter increasing the dimensions of the ANN's input space. A better approach is achieved by providing the ANN with internal dynamics [4]. This kind of network processes multi-inputs and does not require past values of process measurements as current inputs.
The Functional-Link Neural Network (FLNN) has been developed as an alternative architecture to the well-known Multi-Layer Perceptron (MLP) net with application to both function approximation and pattern recognition [9]. The main advantage of the FLNN is a reduced computational cost in the training stage, while maintaining the approximation performance of the MLP network. The FLNN with external dynamic elements has been generally used to perform system identification [9]. The present contribution presents the introduction of dynamic elements within the FLNN structure, in a generalised manner, along with the formulation of its design as a problem of multi-objective optimisation. The latter is solved by using genetic algorithms. Experimental results illustrate the efficiency of presented approach.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 115-120

## 2. Generalised dynamic functional-link neural network

The Generalised Dynamic Functional-Link Neural Network (GDFLNN) has a feed-forward architecture (Fig.1) with a number of non-linear enhancement hidden nodes, referred to as functional links. The suggested GDFLNN is characterised by a variable set of functional links. The initial inputs of the net $u_n$, $n = 1,...,N$, are functionally expanded by a sub-set of orthogonal trigonometric functions to constitute the actual inputs of the non-linear neuron, $v_m$, $m = 1,...,N+M$, given by the following set:

$$\{u_n, \{i_{u_n} \cdot \cos(j_n \cdot \pi \cdot u_n)\}, \{i_{u_n} \cdot \sin(k_n \cdot \pi \cdot u_n)\}\}\ ;\ j_n = 1,...,S_n^{\cos};\ k_n = 1,...,S_n^{\sin}$$

where, for each initial input $u_n$, $i_{u_n}$ indicates the presence (value of 1) or absence (value of 0) of the functional expansion, and $S_n^{\cos}$ and $S_n^{\sin}$ are the orders of functional expansion corresponding to the cosine and sine terms, respectively, varying from 1 to a pre-specified maximum order of $S_{\max}$. At least one initial input of the net has to be subject of the non-linear enhancement. A fixed order of functional expansion has been considered for all net inputs in previous approaches [9].

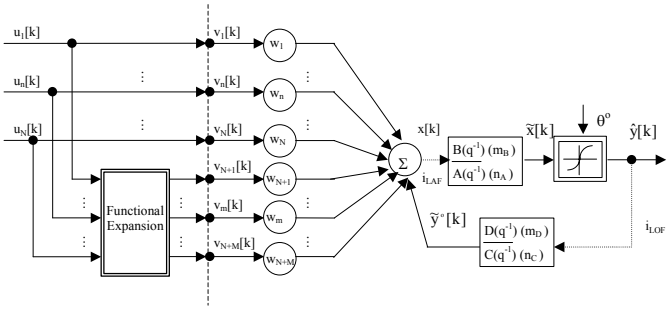


Figure 1: The structure of the generalised dynamic functional-link neural network: $N$ inputs, $M$ functional expansion terms, local activation feedback ($i_{\mathrm{LAF}}$=1), local output feedback ($i_{\mathrm{LOF}}$=1), one non-linear activation unit (of hyperbolic tangent type) and one output; $q^{-1}$ stands for the linear operator of time shifting; A,B,C,D are polynomials.

The GDFLNN integrates conveniently an Auto-Regressive Moving Average (ARMA) filter that can be placed (Fig.1): either before the non-linear activation unit of the neuron, the resulting network being a DFLNN with Local Activation Feedback (LAF); or on the back connection from the network output to the neuron's input, the resulting network being a DFLNN with Local Output Feedback (LOF); or on both places. If none of the internal dynamic structures is considered, $i_{\mathrm{LAF}} = 0$ and $i_{\mathrm{LOF}} = 0$, then the static FLNN [9] results.

The parameters defining the architecture of GDFLNN are given by the following sets:

$$\{i_{u_n}; S_n^{\cos}, S_n^{\sin}\}_{n=1,...,N};\quad \{i_{\mathrm{LAF}}; m_{\mathrm{B}}, n_{\mathrm{A}}\};\quad \{i_{\mathrm{LOF}}; m_{\mathrm{D}}, n_{\mathrm{C}}\}$$

where for the LAF filter $m_{\mathrm{B}}$ represents the numerator order and $n_{\mathrm{A}}$ denotes the denominator order, and for the LOF filter $m_{\mathrm{D}}$ denotes the numerator order and $n_{\mathrm{C}}$ represents the denominator order. These architecture's parameters are determined either by a rather difficult trial-and-error process in a pre-defined space, or by means

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 115-120

of an optimisation technique as described in the next section of the paper.
For a given architecture, the parameters of the GDFLNN are the connection weights, the coefficients of filter(s), and the bias term. These parameters are determined with an extended, Dynamic Back-Propagation (DBP) algorithm [6]. The latter minimises the sum of squared errors between the data which have to be approximated and the approximating values provided by the GDFLNN. The DBP is integrated into the optimisation search, when the potential solutions are evaluated for their fitness.
The design of the GDFLNN is based on three sets of representative process data as follows [2,8]: a *training data set* used for model identification; a *data set for validation* used to select the best identified model(s); a *data set for* model *testing*. To characterise the resulted model accuracy and validity, a post-training analysis is applied to the response of identified models [2]. A resulted correlation coefficient $R_c$ (R-value), between the network outputs and the approximated values, close to 1 indicates that there is a good fit of those data.

## 3. Genetic evolving of GDFLNN

The specification of the optimal GDFLNN architecture and related parameters can be done by a hierarchical Genetic Algorithm (GA) [5], as it is explained in the following. Evolutionary algorithms of genetic type are stochastic search and optimisation methods principally based on computational models of fundamental processes, such as selection, recombination and mutation [1,3,5]. An algorithm of this type begins with a set (population) of parameters' estimates (genes), called individuals (chromosomes) appropriately encoded. Each one is evaluated for its fitness in solving a given optimisation task. At each iteration (algorithm time-step), the most fit individuals are allowed to mate and bear offspring.

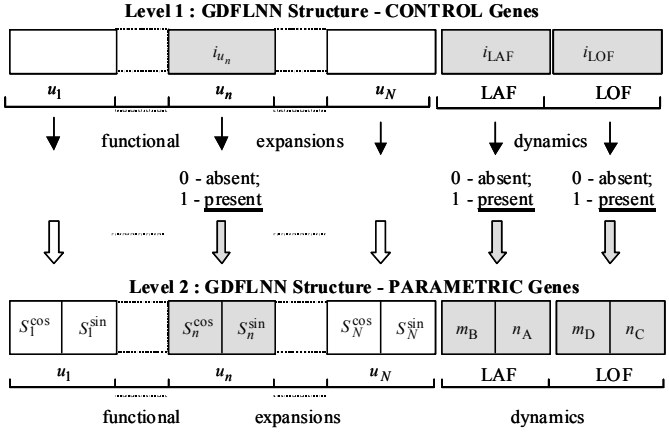


Figure 2: Hierarchical formulation of chromosome for the design of the generalised dynamic functional-link neural network.

Fig. 2 presents the hierarchical structure of the chromosome that is used for the design of GDFLNN. The highest level 1 controls the activation (the value of '1' is assigned) or deactivation (the value of '0' is assigned) of the functional expansion for each

network's initial input, and of the dynamic elements. The involved parameters are binary coded. The lowest (second) level contains the parameters of the functional expansion and of the dynamic internal structures. These parameters are coded as integer values. During the genetic search, the inactive parametric genes remain within the chromosome for possible use in latter generations. The standard genetic operations of recombination and mutation are applied independently to each level of genes. The search space is defined by assigning certain maximal values for the orders of functional expansion, $S_{\max}$, and of ARMA filter(s), $m_{\max}$ and $n_{\max}$, respectively.
A multi-objective approach is adopted for the design of an optimal GDFLNN. Two categories of objectives are considered for minimisation, characterising the approximation accuracy of the network and its complexity: $O_1$ = the sum of squared errors that characterise a certain architecture of GDFLNN and its training data set; $O_2$ = the sum of squared errors that characterise a certain architecture of GDFLNN and its validation data set; $O_{2+n} = S_n^{\cos} + S_n^{\sin}; n = 1,..., N$ = the number of active functional expansions corresponding to each net's input; $O_{N+3} = m_B + n_A$ = the number of coefficients of active LAF filter; $O_{N+4} = m_D + n_C$ = the number of coefficients of active LOF filter. The following priority assignment is considered: objectives $O_1$ and $O_2$ have the same priority of high level; the remaining objectives have the same priority of low level.
Multi-objective Optimisation based on GA (MOGA) seeks to optimise the components of a vector-valued cost function. The solution is not a single point, but a family of points known as the Pareto-optimal set [3,5]. Each point in that surface is optimal in the sense that no improvement can be achieved in one component of the objectives' vector without degradation in at least one of the remaining components. For the genetic evolving of the GDFLNN, the general MOGA procedure described in [3,5] has been adopted and applied as it is detailed in [6].

## 4. Applications

To implement the presented neural approach, the following were used and extended: the Neural Network Toolbox [2], and the standard GA Toolbox [1]. For the applications described in the sequel, the search space for the genetic procedure was defined by setting the following parameters: $S_{\max}$=5, and $m_{\max}$=$n_{\max}$=3. The evolutionary search was carried out for 30 generations, with 30 individuals in a population. During the genetic search, the GDFLNNs were trained (batch mode) for 500 epochs. The best architecture of GDFLNN found at exit was trained again for 3000 epochs. The models described in the following have as inputs different process inputs sampled at the current time *k*, and different process outputs sampled at previous time *k*-1. No other past process measurements are necessary, as the GDFLNN has internal dynamics represented by the ARMA filter(s).
**Identification of an evaporator.** In a first case study, real data from a sugar factory [11] were considered. The investigated process refers to the heater and the first section of an Evaporation Station (ES). The results related to the identification of the "evaporator" sub-system of ES are presented in the following. The considered inputs of the process are: $u_{P,1}$ – the steam flow to the input of ES; $u_{P,2}$ – the steam

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 115-120

temperature at the input of ES; $u_{P,3}$ – the juice temperature after heater. The modelled output is $y_P$ – the juice temperature after section 1 of ES. The data stored during one month, every $T_S$ = 10 seconds, were used to develop a neural model based on GDFLNN: $y_P[k] = f(u_{P,1}[k], u_{P,2}[k], u_{P,3}[k], y_P[k\text{-}1])$, where $f$ denotes the mapping performed by the net and $k$ denotes the normalised sampling time.

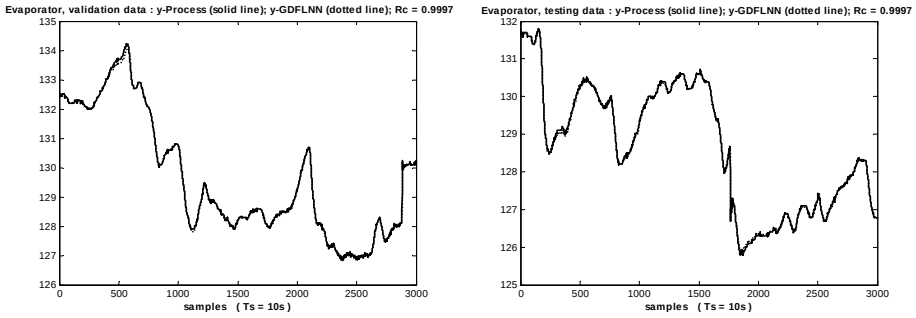


Figure 3: Evaporator, results of system identification with the GDFLNN, expanded data sets of 3000 rows: validation data (left figure) and testing data (right figure).

To design the model, a spectral analysis was performed. Based on this, a low-pass filtering by means of appropriate discrete-time Butterworth filters of 4$^{th}$ order was applied to reduce the noise and the amount of data used in the network learning. The training data, containing 3000 rows of measurements, were decimated using each 10$^{th}$ sampled value. The best results of identification obtained with the GDFLNN are presented in Fig.3, corresponding to validation and testing data sets from another month of plant exploitation. As exemplification, the resulted optimal architecture of GDFLNN is characterised by the following parameters: $S^{\cos}_{u_{P,1}} = S^{\sin}_{u_{P,1}} = 1$; $S^{\cos}_{u_{P,3}} = 1$, $S^{\sin}_{u_{P,3}} = 4$; $S^{\cos}_{y_P} = 5$, $S^{\sin}_{y_P} = 4$; $m_B$=2, $n_A$=3; $m_D$=1, $n_C$=3.

**Identification of a hydraulic looper.** In a second case study, real data from a hydraulic looper in a hot rolling mill [10] were considered.

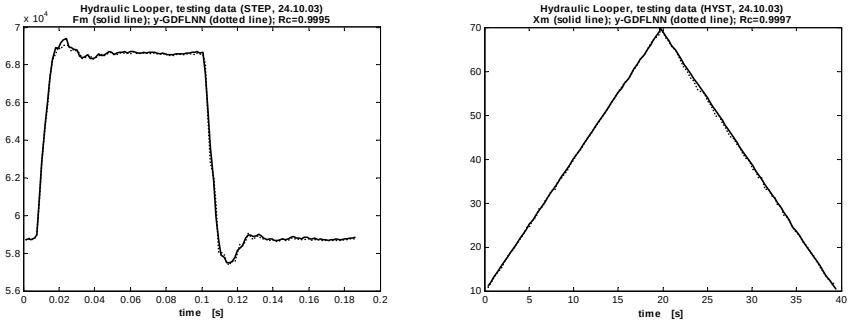


Figure 4: Hydraulic looper, results of system identification with the GDFLNN (training data: October 2002), testing data from October 2003: measured force in step response (force control) test (left figure), and angular position in force hysteresis (position control) test (right figure).

The following measured data were used, as provided by a monitoring system in

performing certain tests on the process: $u_{P,1}$ – reference force; $u_{P,2}$ – servo-valve reference position; $y_{P,1}$ – measured force in the hydraulic cylinder; $y_{P,2}$ – measured angular position of the looper.
The GDFLNN was used to identify different models, among which the following ones: $y_{P,1}[k] = f_1( u_{P,1}[k], u_{P,2}[k], y_{P,1}[k\text{-}1] )$; $y_{P,2}[k] = f_2( y_{P,1}[k\text{-}1], y_{P,2}[k\text{-}1] )$, where $f_i$, $i$=1,2 denote the mappings realised by the neural networks and $k$ represent the normalised sampling time. Fig.4 presents the obtained results of system identification, demonstrating that the developed models $f_1$ (left figure) and $f_2$ (right figure) have a validity of one year with respect to the processed data.

## 5. Conclusion

The used genetic procedure represents a semi-automatic method of selecting the appropriate network architecture for the task of non-linear system identification. The user must only assign certain parameters for the genetic search. This has been found to be easier than manually selecting the network architecture. Current research is done in investigating the application of the suggested approach to fault detection and isolation of technical processes [7].
**Acknowledgement.** The authors would like to acknowledge the support for this work by the EU-IST-2000-30009 project MAGIC, http://magic.uni-duisburg.de.

## References

1. A.J. Chiepperfield, P.J. Fleming, H. Pohlheim, C.M. Fonseca: *Genetic Algorithm Toolbox for Use with MATLAB*. University of Sheffield, UK, (1996).
2. H. Demuth, M. Beale: *Neural Network Toolbox*. The MathWorks Inc., Natick, MA, (2002).
3. C.M. Fonseca: *Multiobjective Genetic Algorithms with Application to Control Engineering Problems*. Ph.D. Thesis, University of Sheffield, UK, (1995).
4. R. Isermann, S. Ernst, O. Nelles: "Identification with Dynamic Neural Networks". Prep. *IFAC Symposium SYSID*, Fukuoka, Japan, Vol.3, pp. 997-1022, (1997).
5. K.F. Man, K.S. Tang, S. Kwong: *Genetic Algorithms: Concepts and Designs*. Springer-Verlag, London, (1999).
6. T. Marcu, P.M. Frank: "Process Fault Detection and Isolation Using Dynamic Multilayer Perceptrons Genetically Evolved". *Int. Journal of Differential Equations and Dynamical Systems*, **10**, 1-2, pp. 169-200, (2002).
7. T. Marcu, B. Köppen-Seliger, P.M. Frank, S.X. Ding: "Dynamic Functional-Link Neural Networks Genetically Evolved Applied to Fault Diagnosis". CD-ROM Proc. *EUCA European Control Conference*, Cambridge, UK, session Fault Diagnosis 4, (2003).
8. M. Nørgaard, O. Ravn, N.K. Poulsen, L.K. Hansen: *Neural Networks for Modelling and Control of Dynamic Systems*. Springer-Verlag, London, (2000).
9. J.C. Patra, R.N. Pal, B.N. Chatterji, G. Panda: "Identification of Non-linear Dynamic Systems Using Functional-Link Artificial Neural Networks". *IEEE Trans. on Systems, Man, and Cybernetics – part B: Cybernetics*, **29**, 2, pp. 254-262, (1999).
10. R. Stücher, M. Tuschhoff, S. Brombach: "Hydraulic Looper Simulations and Measurements for 'MAGIC' Benchmark Tests". Proc. *IAR-ICD/IFATIS/MAGIC Workshop on Advanced Control and Diagnosis*, Duisburg, Germany, pp. 107-112, (2003).
11. M. Syfert. (Organiser): "Research on Quantitative and Qualitative FDI Methods based on Data from Lublin Sugar Factory" (Invited Session). Prep. *IFAC Symp. SAFEPROCESS*, Budapest, Hungary, Vol.1, pp. 331-363, (2000).