System modelling and analysis in optimal control using evolutionary artificial neural networks

M. Di Giamberardino, M. Annunziato, S. Pizzuti
ENEA – Energy, New technologies and Enviroment Agency
'Casaccia' Research Center
Via Anguillarese 301
00060, Rome, Italy
Phone: +39-06-30484411, Fax: +39-06-30484811
{mauro.digiamberardino,mauro.annunziato,stefano.pizzuti}@casaccia.enea.it

Abstract

Artificial Neural Networks (ANNs) methods have proven to be powerful tools in modelling nonlinear processes. In this paper it is presented an analysis of the problem of modelling a process in presence of model uncertainties, focusing the attention on its use regarding the optimization of signal control. It will be shown how the topology of the neural network affects the performance model and it will be presented a procedure to select the best parameters to accomplish a typical control optimization problem. Starting from a particular case study, the attention will be focused on the analysis of the process inertia variations and on the changes in the prediction horizon. As training algorithm we will use an algorithm based on an Artificial Life (ALife) environment.

1 Introduction

In the past years the classical approach in modelling a dynamical system was based on the study of the chemical-physical properties of the process itself with the aim to build an appropriate mathematical representation, establishing a first principles model. This kind of methodology fails when the real system is characterized by non linear dynamics with high dependence on initial conditions, in non stationary environments and chaotic behavior [Prasad *et al.*, 2003].

For this reasons in the last decades there was a raised interest in the *black box* methodologies, in which the mathematical approach is replaced by an empirical study based on input-output signals. In this framework the theory of Artificial Neural Networks (ANNs), which are proved to be powerful instruments to solve complex modelling problems for nonlinear systems [Nikravesh *et al.*, 1996], is inserted. A detailed comprehensive foundation on neural networks can be found in [Haykin, 1999]. Two classes of neural networks which have received considerable attention in the research area are:

- multilayer feed forward neural networks (FF-ANN);
- recurrent neural networks (RNN).

From a system point of view, the first one represents static non linear maps, while the second one is interpreted like a nonlinear dynamic feedback system [Elman , 1990].

In a FF-ANN the inputs to any layer consist only of outputs of the preceding layers, while in the RNN, in addiction to these connections, outputs of the following layers are feeded back, with an appropriate time delay, as inputs of the preceding layers [Tsoi *et al.*, 1997]. Most ANN applications in dynamic modelling have involved the use of feed forward networks to identify a model in autoregressive form, in which the inputs of the network are the past values of the system inputs-outputs, and the network output corresponds to the current system output; these values identify a time window of the system inputs and outputs to be chosen wide enough to capture the essentials of the plant dynamics [Narendra *et al.*, 1990]. On the other hand, recurrent networks, due to their internal structure, are suited to provide a state-space description of the system, but their use is limited by the additional degree of difficulty involved in the training phase and by stability problems due to the presence of the feedback loops [Eaton *et al.*, 1994].

In modern systems theory, a typical optimal control problem consists of finding the optimal regulation to apply to the modelled process that minimize/maximize some certain criteria of optimality (i.e. optimal tracking, min/max cost function, etc..). In the case in which the optimization routine needs a model of the process this dual necessity becomes clear:

- the model has to be as close as possible to the modeled process, especially for the transient regimes;

- the inputs of the model have to be as close as possible to the inputs of the modeled process, without the presence of further external data.

As shown in figure 1, let's suppose that at the step *t* the optimizer gives the neural model optimal regulations. At the next step *t+1* the FF-ANN produces a prediction/estimation of the process output to be used by the performance evaluator and by the optimization routine. The performance parameters are then used by the optimizer to understand how good are the optimal regulations previously suggested, and then find the next optimal value. If the built model used for optimization has as input not only the regulations of the process, it will be not so clear which is the contribution of the optimal regulations and that of extra added inputs. This is the case of a model constructed using a FF-ANN with inputs characterized also by the past values of the output variables. If the modelled output receives only the regulations suggested by the optimality routine then the performance parameters will give exact information on the regulations goodness.

In this way, while in a pure modelling problem the choice between these two approaches depends on the particular studying case, when the ANN model to be constructed is used inside an optimization routine, we have to be careful on the class of neural network.

In the following paragraphs it is shown how the topology of the neural network affects the performance of the model, and it is presented a procedure to determine the best configuration of the network suitable for control signal optimization. The shown results are obtained with a training algorithm based on evolutionary algorithms.

2 Evolutionary algorithms

Evolutionary algorithms (EAs) are global, parallel, and powerful search and optimization methodologies based on the principles of natural selection. They permit a remarkable level of flexibility with regard to performance assessment and design specification [Fleming *et al.*, 2002].

EAs work by means of a population of hypothetical solutions; each individual of the population represents a particular solution to the problem, expressed in some form of genetic coding. Each individual is characterized by its state, its genotype and by the fitness value, which is a parameter of the goodness of the carried solution. This index determines also how successful is the individual at propagating its genes to the following generations: better solutions are assigned higher fitness values than those performing poorly. Despite the simple concepts involved, evolutionary algorithms may become quite complicated, because of the lack of a rigorous mathematical analysis and of many variations proposed to the basis concepts. In our study, the artificial life approach is used to develop a new training algorithm to train the Artificial Neural Networks.

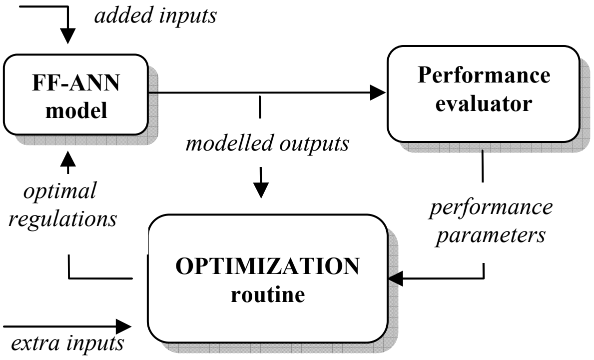

Figure 1. **FF-ANN model used in optimization**

2.1 EAs to Artificial Neural Networks

Evolutionary Artificial Neural Networks (EANNs) have been developed in these last years to improve the performances of the created model in presence of noises, disturbances and other modelling uncertainties [Annunziato *et al.*, 2002]. Indeed, in such cases it seems to be necessary to update the model to the new variations and to make possible an on-line modelling task: one approach to reach the goal consist in combining the neural networks modelisation capabilities with the adaptation property of evolutionary algorithms. In an off-line contest EAs may be very useful in tasks like training connection weights, network design topology, or both of them [Annunziato *et al.*, 2003]. Their usefulness must be principally regarded in higher robustness, poor probability to get stuck in local minima and major simplicity of the resulting topology network with respect to those obtained with classical training algorithms [Balakrishnan *et al.*, 1995].

In our study the artificial life approach is used to build up and train the optimal network to model the process. Each individual represents a FF-ANN, in competition with each other to optimize the *fitness* function (P_{fit}), which indicates how good the neural network model is, and which is expressed by this simple mathematical equation:

$$P_{fit} = 1 - E_{rmse}, \tag{1}$$

where E_{rmse} is the normalized mean square error, described by the following:

$$E_{rmse} = \sqrt{\frac{\frac{1}{2}\sum_{i=1}^{M}(y(i) - \bar{y}(i))^2}{M}}, \tag{2}$$

with M, the dataset dimension; $y(\cdot) \in [0,1]^M \subset \Re^M$, the normalized modeled process output; $\bar{y}(\cdot) \in [0,1]^M \subset \Re^M$, the neural model output.

The EA [Annunziato *et al.*, 2002b] utilized is characterized by three principal characteristics: *meeting*, *reproduction*, and *competition.* At every iteration there is a meeting probability among individuals depending on the density of the population and after that it may happen:

- *bisexual reproduction:* two new individuals are added to the population through crossover;
- *monosexual reproduction*: an individual clone itself and mutates, then it is added to the population;
- *competition*: two individuals meet and fight for survival, then the loser is expelled from the population;

In this way the final population is characterized by a chaotic dynamics, in which the number of individuals and their growing rate tends to stabilize towards an appropriate final region: that's why we call this algorithm *chaotic population*.

3 EANN model: a case study

The EANN methodology is therefore used to find a model of a dynamical process suitable for the previously described control optimization task.

For this purpose the study was developed starting from a well know mathematical process, described by the following linear dynamical system:

$$\begin{cases} \dot{x}(t) = A(\alpha)x(t) + Br(t) + B_d d(t) \\ y(t) = Cx(t) \end{cases} \tag{3}$$

with $A(\alpha), B, C \in \Re^{2x2}, B_d \in \Re^{2x1}, x(\cdot) \in \Re^{2x1}$ the system state vector, $y(\cdot) \in \Re^{2x1}$ the output vector, $r(\cdot) \in \Re^{2x1}$ the regulation vector. The couples $[A, B]$ and $[C, A]$ are, respectively, controllable and observable, the parameters $\alpha \in \Re$ and $d(\cdot) \in \Re$ are used to vary the inertia and the disturbance of the process (α acts on state matrix $A(\alpha)$ in a way that it changes the system modes).

Then we let vary the parameters identifying the inertia and the disturbance actions, in parallel with the variation of the EANN topology, in order to find out the best configuration and to analyze their influence on the process modelling.

The process is simulated by applying typical regulation signals taken from a real plant and the corresponding dynamics are then utilized to construct the training and testing data set.

Network topology

The network we used is an FF-ANN, trained with the EA described before, which works in parallel with the process. Some topology characteristics (hidden neurons and layers, activation functions) have been defined using off-line procedures based on instruments of classical data analysis (correlation, variance, etc…), as well as the number of inputs of the EANN in addiction to the process regulation (as shown in figure 2).

To properly model the system dynamics, especially transients and rapid signal variation, the feed forward approach needs to learn more about the state of the process, receiving on its inputs appropriate values of its past history [Nikravesh *et al.*, 1996]. For this reason the network topology is increased by δ input neurons which receive the regulations and the derivatives of various order of the process outputs to model. The input vector is so defined:

$$\left[r(t-h) \quad y(t-h) \quad y(t-h-\tau) \quad \cdots \quad y(t-h-(\delta-1)\tau\right], \tag{4}$$

where $h \in \mathrm{N}^+$ (*prediction horizon*) represents the future temporal starting point to predict the process output; $\tau \in \mathrm{N}^+$ (*time delay*) is the appropriate temporal step to use (i.e. the order of derivatives); $\delta \in \mathrm{N}^+$ (*time history*) identifies the minimum number of elements of the past signal which is adequate to characterize the state of the process.

The attention has been therefore focused on studying the effects of these parameters on the modelling properties of the EANN. The chaotic data analysis [Abarbanel, 1996] helps us to define the right value of the time delay and the time history, leaving the h parameter varying to fit the best solution.

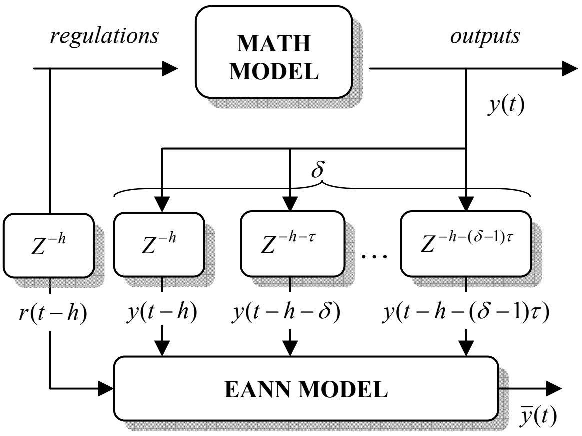

Figure 2. **Inputs to EANN model**

Figure 3 shows how the use of extra inputs permits to increment the modelling performance in presence of dynamics; the reported results are obtained using a prediction horizon equals to one, a time delay of two steps, and a time history of dimension three.

Variation of inertia parameter

To test the EANN modelling capabilities, we consider modelling uncertainties and disturbance actions. To satisfy this aim, the α parameter of eq. (3) is varied through its stability range in such a way that it emulates some process dynamics variations. In fact, its

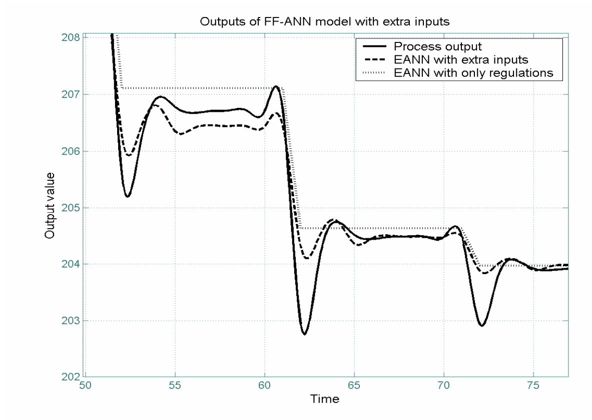

Figure 3. **Comparison of different EANN topologies**

variations correspond to same variations of system modes, which may be used as inertia indicators. Assuming the process affected by step-size regulations and taking as reference the time (calculated as percentage of time step) that the modeled signal takes to achieve its regime state, it is possible to study the EANN modelling capabilities when inertia parameter is varying. Many simulations have been carried out taking in consideration four different cases of inertia percentage (30%, 50%, 90%, and 150% of step size time), studying the EANN having as inputs only regulations, regulations plus dynamics, and only dynamics. Three different runs for each case have been done in order to produce statistical relevant results.

In figure 4 it is shown a graph summarizing the results, which may be summed into the following considerations:

- using only regulations as network inputs, the EANN modelling capabilities strongly decreases as inertia increases. In fact, when inertia grows up the past state of the process takes a lot of importance, increasing the dynamical effects not easily understood by the defined network;
- using dynamical extra inputs, the modelling performance is increased. However it makes the network less sensible to inertia variations, as shown by the asymptotic curve in figure 4;
- for low values of inertia it seems better to use an EANN with no extra dynamical inputs, decreasing in this way the computational time.

In figure 4 it is also shown the variation of RMSE as inertia increases when the EANN is trained using only dynamical inputs without information on regulations. In fact, it is important to understand which is the weight of the last ones with respect to the first ones in order to build a good model. Looking at the curves it seems clear that the past states become preponderant with respect to the current regulations as the inertia of the process increases. Under the optimization point of view this aspect assumes great importance leaving room for further analysis.

4 EANN modelling for optimization

The exposed considerations left out the influence of the prediction horizon (*h* term) (eq. 4) in performance modelling. In the situation where $h = 1$ we are supposing to predict the process state at the immediately next time step. This supposition becomes in reality a restriction when the intent is to study the EANN modelling capabilities in presence of inertia variation in the context of control optimization.

Therefore it is analyzed the performance of the EANN when varying the prediction horizon in eq. 4 in an appropriate temporal range ($h \in [1,20] \subset \mathrm{N}^+$). The aim is to perform a methodology able to assure the best configuration of the network in presence of a specific inertia value and for an adequate choice of the *h* term. The simulation results are reported in figures 5, 6, and 7, in which it is underlined the minimum value of the RMSE index (eq. 2) to be used to build the best model for the control optimization task for a specific value of inertia. In figure 5 it is also shown the comparison of the RMSE value between the EANN modelling with or without extra dynamical inputs. The reported graphs show the following important results:

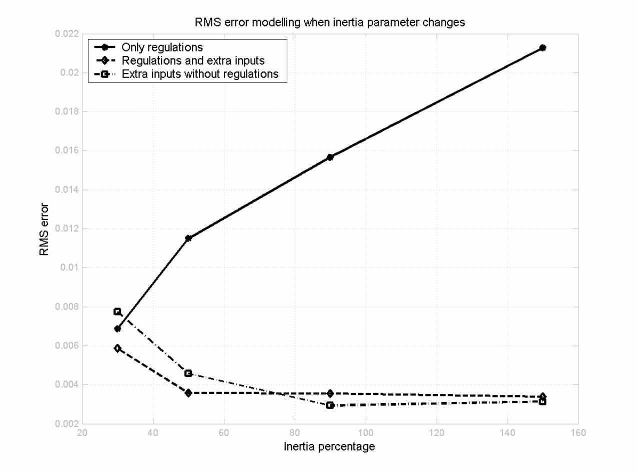

Figure 4. **RMSE with inertia changes**

- when varying the prediction horizon, using an evolutionary FF-ANN with only regulations as network inputs, the modelling error has a minimum which shifts forward along the time horizon axes as the inertia value increases (see figure 6). This fact stresses the EANN dependence on the *h* term, moreover it gives us the instruments to build an algorithm which let us find the optimal value of the horizon, supposing known the inertia value, or the process percentage inertia, with a simple iterative procedure described later;
- using dynamical extra inputs, the RMS error in eq. (2) grows up as the prediction horizon increases

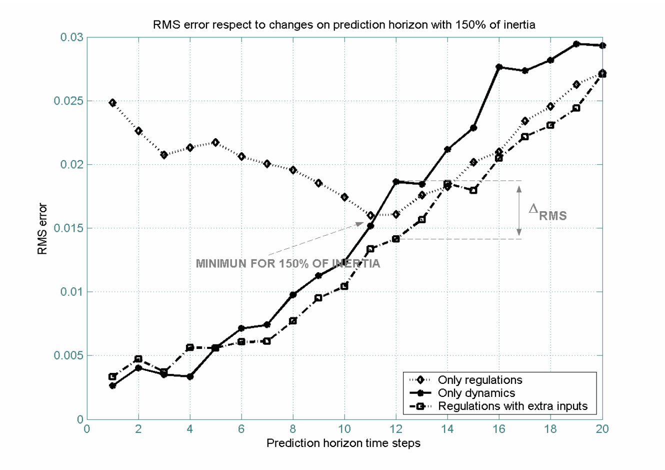

Figure 5. Δ_{RMS} **in case of 150% of inertia**

and the inertia parameter decreases (see figure 7). This is understandable because for low inertia values the process stands a lot of time in its steady state regime and the use of past terms of the signals as inputs may confuse the EANN and produce a lower modelling performance (these terms have a low correlation with the present process state);

- from the optimization point of view, in which EANN is used to give the optimal set point to a controller, it is important to make the right choice of h in eq. (4) as the best compromise between the lowest value of RMSE (to assure the best modelling) and the high influence of regulations. It is not possible to choose the network with the lowest RMSE in eq (2) (usually the one with extra dynamical inputs and low value of h), since it is clearly poorly sensible to changes in regulations. Looking at figure 5 it is possible to point out that, when the EANN with only regulations has a minimum in its RMSE value, there is an error gap, identified with Δ_{RMS}, between the neural model with only dynamics and the one with dynamics plus regulations. Near this value the neural network is mainly affected by the influence of regulation signals. In this range of values there's the appropriate choice of the optimal value of the prediction horizon.

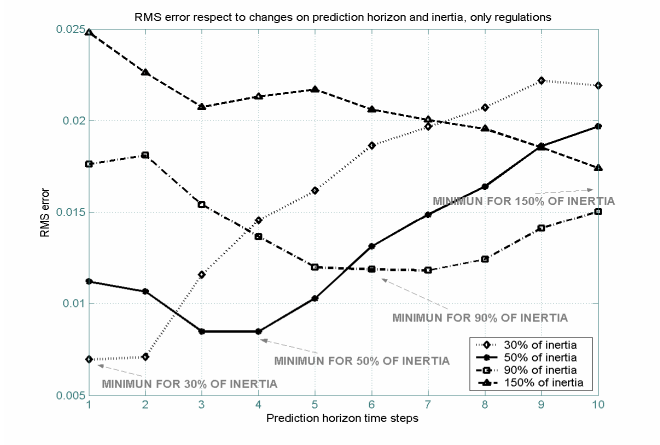

Figure 6. **RMSE minimum with only regulations**

All these results underline that for low inertia values it is more desirable to feed the EANN with only the regulations of the process. On the other way, as the inertia increases, it might be useful to try reducing the RMS error in eq. (2) and choosing a neural network with extra dynamical inputs (being careful to choose the working area which guarantees the highest value for Δ_{RMS} gap).

5 Finding inertia via EANN

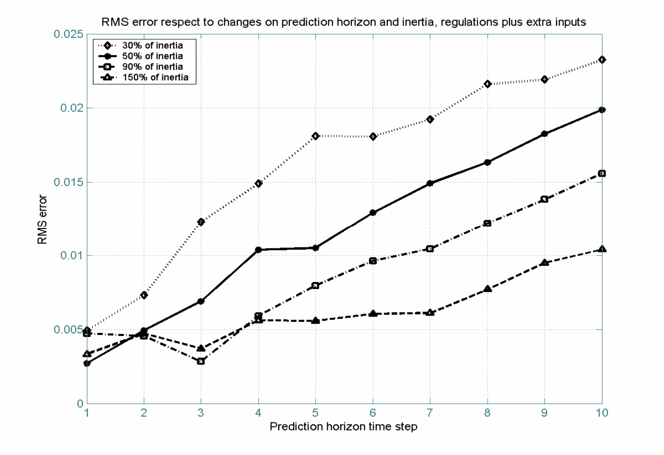

Figure 7. **RMSE variations with extra inputs**

As pointed out in the previous section, the use of the prediction horizon permits to identify an algorithm to find a first inertia estimation of an unknown process. Let's define:

- T_g, the permanence average time of the signal on the reference step signal;
- T_i, the permanence average time of the signal in transient regime;
- $h_{\min}$, the prediction horizon value in presence of the first RMSE minimum;
- $\gamma = f(T_g, T_i)$, the tuning parameter depending on the process in study;

The algorithm uses a FF-ANN with only regulations as inputs, trained with a training procedure in parallel with the variation of the prediction horizon. It stops when the term $\boldsymbol{h}_{\mathbf{min}}$ is reached (see figure 8). Then it is possible to have a first process estimation by the following formula:

$$T_i = \frac{T_g \cdot h_{\min} + \gamma}{10} \tag{5}$$

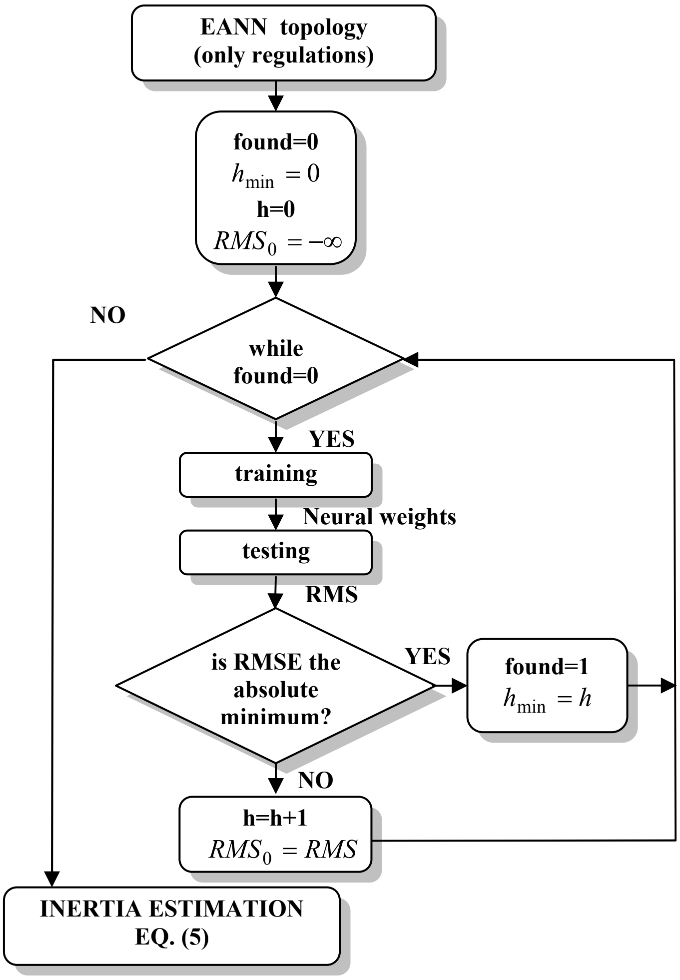

Figure 8. **Inertia estimation algorithm**

6 Conclusion

In this paper it is shown the ability of EANN to model a dynamical system in transient regimes. When the modelling capabilities of the EANN are used in a control optimization task, the presence of extra inputs in addition to regulations may pauperize the optimal routine. From this point of view, the main achievement is the procedure which permits to find the best neural network in control optimization problems in order to assure the highest regulations influence. Moreover, it is analysed the influence of the process inertia and the prediction horizon providing the optimal one which gives the best neural network modelling capabilities.

Finally, it is shown how this approach permits to find, via a simple iterative algorithm, a first estimation of the modelled process.

Future work will concern the study of robustness of the described procedure when the system is affected by noise on the measured variables.

References

[Abarbanel, 1996] H.D.I. Abarbanel. Analysis of observed chaotic data. ISBN 0-387-94523-7, Springer-Verlang.

[Annunziato *et al.*, 2002] M. Annunziato, M. Lucchetti, S. Pizzuti. Adaptive systems and evolutionary neural networks: a survey. In *Proc. EUNITE02, 2002,* Albufeira, Portugal.

[Annunziato *et al.*, 2002b] M. Annunziato, S. Pizzuti. Adaptive Parameterization of Evolutionary Algorithms and Chaotic Population Dynamics. *Int. J. of Knowledge-Based Int. Eng. Sys.*, 170-177, 2002, ISSN: 1327-2314.

[Annunziato *et al.*, 2003] M. Annunziato, I. Bertini, M. Lucchetti, S. Pizzuti. Evolving weights and transfer functions in feed forward neural networks. In *Proc. EUNITE03, 2003,* Oulu, Finland.

[Balakrishnan *et al.*, 1995] K. Balakrishnan, V. Honovar. Evolutionary design of neural architectures – a preliminary taxonomy and guide to literature. Technical report CS-TR 95-01, 1995, Iowa State University, Ames, IA.

[Eaton *et al.*, 1994] J.W. Eaton, J.B. Rawlings, L.H. Ungar. Stability of neural net based model predictive control. *Neurocomputing, Vol. 15,* 1997, pp. 183-223.

[Elman, 1990] J. L. Elman. Finding structure in time. *Cognitive science, Vol.14, 1990*, pp. 179-211.

[Fleming *et al.*, 2002] P.J. Fleming, R.C. Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Control Eng. Pr., Vol. 10,* 2002, pp. 1223-1241.

[Haykin, 1999] S.Haykin. Neural Networks, a comprehensive foundation. *Prentice Hall. New Jersey*, 2nd edition, 1999. ISBN 0-13-273350-1.

[Narendra *et al.*, 1990] K.S. Narendra, K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. Neur. Netw., Vol. 1, No. 1,* 1990, pp. 4-27.

[Nikravesh *et al.*, 1996] M. Nikravesh, A.E. Farell, T.G. Stanford. Model identification of non linear time variant processes via artificial neural network. *Computes and Chemical Engineering, Vol. 20, No. 11,* 1996, pp. 1277-1290.

[Prasad *et al.*, 2003] V. Prasad, B.W. Bequette. Non linear system identification and model reduction using artificial neural networks. *Computes and Chemical Engineering, Vol. 27,* 2003, pp. 1741-1754.

[Tsoi *et al.*, 1997] A.C. Tsoi, A. Back. Discrete time recurrent neural network architectures. *Neurocomputing, Vol. 15,* 1997, pp. 183-223.