Modular Neural Networks for Non-Linearity Recovering by the Haar Approximation

(revised version)

Zygmunt Hasiewicz

Institute of Engineering Cybernetics
Wrocław University of Technology
Wrocław, Poland

Acknowledgements: The author wishes to thank the reviewers for their helpful comments and suggestions. He also thanks M.Sc. P. Sliwinski for his assistance in preparing the numerical examples.

Address for correspondence: Zygmunt Hasiewicz, Institute of Engineering Cybernetics, Wrocław University of Technology, Janiszewskiego 11/17, 50-372 Wrocław, Poland; Tel: (48 71) 320 32 77; Fax: (48 71) 321 26 77; e-mail: zhas@ict.pwr.wroc.pl

Running title: *Modular Neural Networks for Non-Linearity Recovering*

Modular Neural Networks for Non-Linearity Recovering by the Haar Approximation

Abstract: *The paper deals with the design of a composite neural system for recovering non-linear characteristics from random input-output measurement data. It is assumed that non-linearity output measurements are corrupted by an additive zero-mean white random noise and that the input excitation is an i.i.d. random sequence with an arbitrary (and unknown) probability density function. A class of modular networks is developed. The class is based on the Haar approximation of functions with piecewise constant functions on a refinable grid and consists of the networks composed of perceptron-like modules connected in parallel. The networks provide a local mean value estimators of functions. The relationship between complexity and accuracy of modular networks is analysed. It is shown that under mild conditions on the non-linearities and input probability density functions the networks yield pointwise consistent estimates of non-linear characteristics, provided that complexity of the networks grows appropriately with the number of training data. Efficiency of the networks is examined and the asymptotic rate of convergence of the network estimates is established. Specifically, local ability of the networks to recover non-linear characteristics in dependence on local smoothness of the underlying non-linear function and the input probability density is discussed. Optimum complexity selection rules, guaranteeing the best performance of the networks, are given. Illustrative simulation examples are provided.*

Keywords: Modular networks, Non-linearity recovering, Haar approximation.

1. INTRODUCTION

From among many applications of neural networks, the problem of recovering non-linear characteristics of physical phenomena from the measurement data points has attracted much attention in the last decade. This interest is caused by the fact that the necessity of finding an exact or approximate model of a physical system occurs frequently in many engineering applications (e.g., in robotics, signal processing, automatic control) and that an *on-line* solution to the problem is often desired, i.e. truly fast modeling tools are required. Moreover, in many practical situations a prior knowledge of a possibly non-linear characteristic is poor and no well-grounded hyphotesis concerning its functional form can be formulated. Very often the true characteristic is known only at some sample points, recorded in an identification experiment, but the need is to derive the original function (or at least a satisfactory approximation to it) at all points, also on an unseen set of inputs. Thus a kind of generalization is required. Due to the widely recognized approximation and generalization capabilities of neural networks as well as their fast operation, a rational approach for solving such problems is to apply an analog artificial neural network.

Following the seminal results of Cybenko (1989), Hornik *et al.* (1989), Park and Sandberg (1991) and Barron (1993), among others, an extensive research has been done towards employing neural networks for recovering of non-linear characteristics and various kinds of architectures have been proposed and investigated. Most existing neural architectures can be placed into one of three categories:

1. Multilayer Perceptrons (MLP) and sigmoidal networks (Cybenko, 1989; Hornik *et al.*, 1989, 1990; Ito, 1991; Hornik, 1991; Barron, 1993, 1994, for instance);
2. Radial Basis Function (RBF) networks (Bishop, 1991; Park and Sandberg, 1991, 1993; Leonard *et al*., 1992; Elayanar and Shin, 1994; Chen and Chen, 1995; Gorinevsky, 1995; Krzyżak *et al*., 1996, for example); or
3. Wavelet networks (Zhang and Benveniste, 1992; Pati and Krishnaprasad, 1993; Delyon *et al*., 1995; Zhang *et al*., 1995; Zhang, 1997 and the references cited therein).

An early approach (first group of networks) relies on step or sigmoidal activation functions and leads to rather complicated multilayer structures. The sigmoidal networks use quite explicit parametric representation of functions, where parametric function models are built up as a linear (weighted) combination of sigmoids, with adjustable activation parameters (scale and translation factors). Both weights and activation parameters need training which leads to highly non-linear parametric optimization tasks. The popular method of backpropagation can take a large number of iterations to converge and can converge to local minima instead of finding the global minimum of the approximation error. Though a number of modifications of backpropagation and many new training algorithms have been proposed to overcome this problem (see, e.g., Azmi and Liou, 1993;

Verma, 1997), fast and guaranteed training of sigmoidal networks is still the open question (Jones, 1997).

An alternative strategy, using radial basis functions (second class of networks), results in simpler (one hidden layer) and more flexible architectures with better approximation capability, which has been demonstrated in a number of papers (see, for instance, Jackson, 1988; Sandberg, 1991; Powell, 1992 and the papers cited above). Unlike the former, the RBF networks provide linear-in-the-parameters approximations of functions (after prior establishing the basis function centroids and widths) and consequently linear optimization techniques (e.g. linear least squares; Chen *et al*., 1991) can be implemented for training, which significantly simplifies training procedures. However, performance of the RBF networks critically depends on the selection of the centroids and widths of RBF's and the latter is in turn a rather delicate and non-trivial problem (see Xu *et al*., 1994; Krzyżak *et al*. 1996 and the references therein). Despite many efforts (see the cited papers), simple and efficient means for training the RBF networks are still searched for (e.g. Kaminski and Strumillo, 1997).

Recently, good localization (zoom-in) properties and parsimony of wavelet representations, recognized within the multiresolution and wavelet theory (see e.g. the fundamental monographs by Chui, 1992*a, b*; Daubechies, 1992 or Walter, 1994), resulted in the development of the wavelet neural networks, particularly recommended for exploring fine details in highly non-linear characteristics. Although the wavelet networks have been introduced as a new tool for approximation of non-linear functions (Zhang and Benveniste, 1992), they repeat some of the standard disadvantages of sigmoidal networks - with the difference that a sigmoid activation function with tunable parameters is replaced with a mother wavelet with adjustable scale and shift (translation) factors. In particular, determination of the wavelet network parameters (i.e. synaptic weights, scales and translations of the wavelet activation functions) needs solution of also highly non-linear parametric optimization tasks, which are rather complicated even if the problem can be reduced to the convex optimization (Pati and Krishnaprasad, 1993). In spite of the fact that specialized techniques, considerably reducing complexity of the parameter search, have been developed (e.g. Monte Carlo approach in Delyon *et al*., 1995 or least squares algorithms in Pati and Krishnaprasad, 1993; Zhang *et al*., 1995; Zhang, 1997), the wavelet networks generally suffer from lacking of fast training routines and training of such nets is still a hard problem.

The power of wavelet neural networks is typically attributed to their ability to approximate closely more general, irregular, non-linear characteristics in localized regions (see the references above). Thus the local, pointwise, efficiency of the wavelet networks - touching the very essence of the wavelet bases - should be of particular interest. Unfortunately, till now only the global approximation properties (in the sense of L^2 and *sup*-norm error) of the wavelet networks have been investigated and the associated global approximation rates have been established (Delyon *et al*., 1995; Zhang *et al*., 1995).

In this paper, we propose and analyse a class of networks for non-linearity recovering, belonging to the intersection of the first and third of the categories distinguished above. The proposed networks are perceptron-based architectures originated from the Haar wavelet analysis. This simple formal ancestor yields the networks of composite modular structure, where the problem of achieving of a high resolution of training data (in order to follow fine local details in the run of a target non-linearity) is decomposed into a number of less demanding tasks of achieving lower resolution ability by the component modules (subnetworks). These modules (to some extent of an arbitrary complexity and 'precision') are connected in parallel which results in a simple, flexible, and easily expandable structure of the whole network. Such a structure, as composed of standard units, can be attractive from the viewpoint of hardware realization, the more so as the building blocks possess simple perceptron-like set-up (with step activation function). Training of the proposed networks is an easy, one-pass, process which does not involve any parametric optimization techniques. The problem of implementation of such modular networks to discover non-linear characteristics from the set of input-output training data is in the paper considered in a stochastic framework. We assume that output measurements are blurred by an additive zero mean white random noise (similarly as it was in Delyon *et al*., 1995 and Zhang, 1997) and that the non-linear characteristic is driven by a random i.i.d. input sequence possessing a probability density function. In contrast with Delyon *et al*. (1995) and Zhang *et al*. (1995) we do not require the input data to be uniformly distributed. We show that under moderate requirements concerning the unknown non-linearities and input probability density functions our networks successfully recover non-linear characteristics, i.e. yield their pointwise consistent estimates, provided that complexity of the networks (data resolution ability) grows in an appropriate manner with the number of training data. In the main part of the paper the considerations refer to the memoryless observation model (static system). In remarks, we shortly discuss in parallel the non-linearity recovering problem for the Hammerstein system, where the non-linearity output is transformed by a linear output dynamics before measurement. This is because such systems occur in many important applications in various areas such as biocybernetics (Hunter and Korenberg, 1986), automatic control (Vörös, 1999) or industrial engineering (Eskinat *et al*., 1991), among others.

The paper outline is as follows. In Section 2, the problem of non-linearity recovering is stated and the underlying assumptions are collected. Then, in Section 3, the Haar approximation of functions, providing the theoretical background for operation of the networks, is briefly reviewed. The basic neural architecture for recovering non-linear characteristics is presented in Section 4 and motivation, training algorithm and reference of the network outcome to the Haar approximation are there described. We also examine in this section accuracy of the network estimate and the relationship between complexity of the network, the number of training data and the corresponding approximation and estimation errors (bias and variance of the network estimate).

As a result of these studies, we give conditions for the weak pointwise consistency of the network issue. The conditions are distribution-free, i.e., do not rely on any specific probability distribution of the input sequence. A class of modular networks is introduced in Section 5. In Section 6, we consider efficiency of modular networks and establish the asymptotic rate of convergence of the network estimate to the target non-linearity in dependence on local smoothness of both the recovered non-linearity and the input probability density function. It is shown that the asymptotic rate of convergence can be optimized by proper selection of the network complexity. General guidelines for selecting the size of modular networks, for large and moderate number of training data, are given in Section 7. Section 8 presents the results of computer simulations. Conclusions in Section 9 complete the paper.

2. NON-LINEARITY RECOVERING PROBLEM

We consider the problem of recovering a non-linear characteristic $R(x)$ from the empirical input-output measurement (training) data $\{(x_k, y_k); k = 1, 2, \ldots, N\}$ in a stochastic environment. Basically, we focus on the standard task, where the scalar input-output observations (x_k, y_k) are generated according to the equation

$$y_k = R(x_k) + z_k \tag{2.1}$$

The following assumptions are imposed on the problem:

Assumption 1: The input process $\{x_k; k = \ldots, -1, 0, 1, 2, \ldots\}$ is a sequence of independent and identically distributed (*i.i.d.*) random variables with finite variance, and there exists a probability density of x_k, denoted by $f(x)$.

Assumption 2: The output noise $\{z_k; k = \ldots, -1, 0, 1, 2, \ldots\}$ is a stationary white random process with zero mean, $Ez_k = 0$, and finite variance, $\operatorname{var} z_k = \sigma_z^2 < \infty$.

Assumption 3: Processes $\{x_k\}$ and $\{z_k\}$ are mutually independent.

It is generally assumed that $R(x)$ and $f(x)$ are completely unknown. The crucial point to notice is that, under Assumptions 1-3, we have

$$R(x) = E\{y_k \mid x_k = x\} \tag{2.2}$$

i.e. that the unknown non-linear characteristic $R(x)$ is a regression function (hence denoted by $R(x)$) and therefore it can be potentially recovered, at the input point x, as a mean value of the output y given x, provided that $R(x) < \infty$. The subscript k in x and y in equation (2.2), though not necessary because of stationarity of the processes $\{x_k\}$ and $\{y_k\}$, is used to emphasize the link with equation (2.1). In the sequel, we additionally assume that:

Assumption 4: The non-linearity $R(x)$ is to be recovered for $x \in [0,1)$.

Assumption 5: For $x \in [0,1)$ it holds

$$|R(x)| \leq M_R \tag{2.3}$$

$$\delta \leq f(x) \leq M_f \tag{2.4}$$

where M_R, M_f, $\delta > 0$ are some finite constants.

Assumption 4 reflects, in the standardized formulation, a rather conventional situation where the non-linear characteristic is needed to be known only in some bounded region, $[a,b)$ say. The left-hand side of (2.4) in Assumption 5 says that the input density does not vanish in the domain of interest. We emphasize that the existence of finite constants M_R, M_f and δ in (2.3) and (2.4) is the only demand imposed on the non-linearity $R(x)$ and the input probability density $f(x)$ in our problem. In particular, the input density $f(x)$ can be an arbitrary probability density function, satisfying condition (2.4). As mentioned in Section 1, this is in contrast with more conventional approaches which assume that the input process $\{x_k\}$ is uniformly distributed, see Delyon *et al*. (1995) and Zhang *et al*. (1995).

In the above problem, in order to recover $R(x)$ its values have to be computed (estimated) from the sample data $\{(x_k,y_k)\}$ for each point $x \in [0,1)$, i.e. also generalized to samples which are not present in the data set. Therefore, we shall apply a neural network solution, with intrinsic generalization capability.

Remark 1. The observation model in (2.1) refers to recovering a non-linear characteristic $R(x)$ of a static (memoryless) system (Fig. 1a). In parallel, we shall shortly discuss the situation where the non-linearity output is passed, before measurement, through an unknown discrete-time linear output dynamics, described by the impulse response $\{\lambda_p; p = 0, 1, \ldots\}$. This leads to the so-called Hammerstein system as in Fig. 1b (e.g. Bendat, 1990), where the data generation process is governed by the equation

$$y_k = \bar{R}(x_k) + \bar{z}_k \tag{2.5}$$

where

$$\bar{R}(x) = cR(x) + d \tag{2.6a}$$

$$c = \lambda_0, \quad d = ER(x) \sum_{p=1}^{\infty} \lambda_p \tag{2.6b}$$

and

$$\bar{z}_k = \sum_{p=1}^{\infty} \lambda_p [R(x_{k-p}) - ER(x)] + z_k \tag{2.7}$$

In such a case, apart from Assumptions 1-5, we assume that the unknown non-linearity $R(x)$ is such that $ER^2(x) < \infty$ and that the unknown linear output dynamical filter is asymptotically stable, i.e. $\sum_{p=0}^{\infty} |\lambda_p| < \infty$. Under these conditions, along with Assumptions 1-3, $\{y_k\}$ is a correlated second-order stationary process (Hannan and Deistler, 1988). We easily see from (2.5) and (2.7) that

$$E\{y_k | x_k = x\} = \bar{R}(x) \tag{2.8}$$

i.e. the expectation as in (2.2) equals at present scaled and shifted non-linearity in (2.6a)-(2.6b), for each input point x. This was first observed by Greblicki and Pawlak (1986) (see also Greblicki and Pawlak, 1987, 1994 and Greblicki, 1989).

3. THE HAAR APPROXIMATION

For completeness and ease of reference we present here the basic facts from the Haar wavelet approximation theory, relevant to our considerations. Detailed treatment of this theory can be found, e.g., in Daubechies (1992), Ogden (1997) or Mallat (1998).

Let

$$\varphi(x) = I_{[0,1)}(x) = 1(x) - 1(x-1) \tag{3.1}$$

where $I_{[a,b)}(x)$ is the indicator function of [a,b) and $1(x)$ is the perceptron (step) activation function, i.e. we have $\phi(x) = 1$ if $x \in [0,1)$ and 0 otherwise. Assume that $m \geq 0$ is an integer and consider the functions

$$\varphi_{mn}(x) = 2^{m/2}\,\varphi(2^m x - n) = 2^{m/2}\, I_{[\frac{n}{2^m}, \frac{n+1}{2^m})}(x), \qquad n = 0,1,\ldots,2^m-1 \tag{3.2}$$

which are the scaled and translated versions of $\phi(x)$. Clearly, $\phi_{00}(x) = \phi(x)$. We have

$$\operatorname{supp} \varphi_{mn}(x) = [\frac{n}{2^m}, \frac{n+1}{2^m}) \tag{3.3a}$$

and for each fixed scale factor $m \geq 0$ it holds

$$\operatorname{supp} \varphi_{mn}(x) \cap \operatorname{supp} \varphi_{mn'}(x) = \varnothing \quad \text{for} \quad n \neq n' \tag{3.3b}$$

$$\bigcup_{n=0}^{2^m-1} \operatorname{supp} \varphi_{mn}(x) = [0,1) \tag{3.3c}$$

This means that various functions $\phi_{mn}(x)$ of the same scale m do not overlap and their supports form together a partition of the interval [0,1). This way, for every scale $m \geq 0$, the supports of $\phi_{mn}(x)$ for $n = 0,1,\ldots,2^m-1$ cover jointly [0,1) and quantize it onto the 'portions' $[n/2^m, (n+1)/2^m)$, each of the length $1/2^m$, determining thereby the partition (resolution) grid. It is obvious that the functions $\phi_{mn}(x)$, $n = 0,1,\ldots,2^m-1$, in (3.2) are square integrable over the interval [0,1) and form an orthonormal system in $L^2[0,1)$, the space of all square integrable functions on [0,1). Let V_m be the subspace of $L^2[0,1)$ spanned by the functions in (3.2):

$$V_m = \operatorname{span}\{\varphi_{mn}(x);\ n = 0,1,\ldots,2^m-1\} \subset L^2[0,1) \tag{3.4}$$

This space is the set of all functions in $L^2[0,1)$ which are piecewise constant on the intervals $[n/2^m, (n+1)/2^m)$, $n = 0,1,\ldots,2^m-1$, of the length $1/2^m$. Clearly $V_m \subset V_{m+1}$ since functions constant on intervals of size $1/2^m$ are also constant on intervals of size $1/2^{m+1}$. For $m = 0,1,2,\ldots$ this yields a nested chain of subspaces of $L^2[0,1)$:

$$V_0 \subset V_1 \subset \cdots \subset V_m \subset V_{m+1} \subset \cdots \subset L^2[0,1) \tag{3.5}$$

The wavelet literature refers to the function $\phi(x)$ in (3.1) as the Haar scaling function (or the Haar father wavelet function) and to the functions $\{\phi_{mn}(x)\}$ in (3.2) as the Haar father wavelet basis functions. The spaces V_m as in (3.4) (associated with the scaling function in (3.1)) are called the Haar approximation spaces (with the resolutions $1/2^m$) and the sequence in (3.5) is called the Haar multiscale approximation of $L^2[0,1)$ (see the references cited above).

Any function $F(x) \in L^2[0,1)$ (in particular, such that $\sup_{x \in [0,1)} |F(x)| \le M_F < \infty$) can be approximated in the Haar space $V_m \subset L^2[0,1)$ by using

$$F(x;m) = \sum_{n=0}^{2^m-1} \alpha_{mn} \varphi_{mn}(x) = 2^{m/2} \sum_{n=0}^{2^m-1} \alpha_{mn} I_{[\frac{n}{2^m}, \frac{n+1}{2^m})}(x) \tag{3.6}$$

where

$$\alpha_{mn} = \int_0^1 F(x) \varphi_{mn}(x)\, dx = 2^{m/2} \int_{n/2^m}^{(n+1)/2^m} F(x)\, dx \tag{3.7}$$

and the approximator $F(x;m)$ is the orthogonal projection of $F(x)$ onto V_m. For $x \in [n/2^m,(n+1)/2^m)$ where $0 \le n \le 2^m - 1$, we simply obtain

$$F(x;m) = 2^{m/2} \alpha_{mn} = 2^m \int_{n/2^m}^{(n+1)/2^m} F(x)\, dx \tag{3.8}$$

i.e. approximation of $F(x)$ in the Haar approximation space V_m equals, for each point $x \in [n/2^m,(n+1)/2^m)$, a *local mean value* of the function $F(x)$ in the interval $[n/2^m,(n+1)/2^m)$. The approximation in (3.6)-(3.8) is the best piecewise constant 'model' of the function $F(x)$ in the 'model' set V_m ((3.4)) i.e. the closest to $F(x)$ function in $L^2[0,1)$, piecewise constant on intervals of size $1/2^m$. The Haar basis functions in (3.2) are good building blocks for approximation of functions. Asymptotically, for $m \to \infty$, the following pointwise convergence takes place:

$$\lim_{m \to \infty} F(x;m) = F(x) \tag{3.9}$$

at each point x where $F(x)$ is a continuous function. This follows as a simple conclusion from Theorem 2.1 in Kelly *et al.* (1994).

4. BASIC NETWORK

The motivation of the network structure presented in this section follows from the fact that the theoretical mean value (expectation) in equation (2.2) (respectively (2.8)) can be estimated by the empirical (sample) mean computed from the output observations y_k for x_k lying in a neighbourhood of x. Such an intuitive idea is commonly used in non-parametric estimation of functions (see e.g. Prakasa Rao, 1983; Eubank, 1988 or Härdle, 1990). On the other hand, computation of a local mean value of a function (in a neighbourhood of a given point x) is the usual procedure for the Haar approximation of functions (see equation (3.8)). In this particular framework the size of the

neighbourhood $[n/2^m,(n+1)/2^m)$ around x is standardized to $1/2^m$ and can be conveniently controlled by changing the scale (resolution level) m.

Since the non-linearity $R(x)$ is to be recovered for $x\in[0,1)$ (Assumption 4), we assume that m, the scale factor of the Haar approximation, is $m \geq 0$ (Section 3) and propose the network shown in Fig. 2.

4.1. Network Architecture

The net in Fig. 2 is a feedforward three layer structure with the following allocation of tasks.

The first layer, consisting of 2^m+1 perceptron-type neurons with standard 0-1 activation function, splits the interval $[0,1)$ into 2^m non-overlapping regions $[n/2^m,(n+1)/2^m)$, $n = 0,1,...,2^m-1$, each of the length $1/2^m$. This way a set of basis functions as in (3.2) is modelled (up to the constant $2^{m/2}$) and the resolution grid is established.

The second layer, composed of 2^m linear neurons - each with two synapses, constant weights ('1' and '- 1') and binary $\{0,1\}$ output value, classifies the actual value of the network input x into adequate bin $[n/2^m,(n+1)/2^m)$ distinguished in the first layer. This is accomplished by ascribing '1' to the output of the neuron associated with the pair of neighbouring perceptrons in the first layer with reverse activation, and '0' to the outputs of all remaining neurons in the layer. Consequently, the active path to the neurons in the third layer is determined.

The third layer, composed of 2 adaline neurons - each with 2^m synapses and 2^m synaptic weights ($a_{mn,N}$ and $b_{mn,N}$, respectively) which need training, produces the network estimate. Adaptive training of the weights, made on the basis of input-output measurements $\{(x_k,y_k)\}_{k=1}^N$ (cf. Section 2), is a recursive one-pass process accomplished according to the equations

$$a_{mn,N} = a_{mn,N-1} + y_N I_{[0,1)}(u_N), \tag{4.1a}$$

$$b_{mn,N} = b_{mn,N-1} + I_{[0,1)}(u_N), \tag{4.1b}$$

where $u_N = 2^m x_N - n$, with the initial conditions $a_{mn,0} = 0$ and $b_{mn,0} = 0$. Training procedure is illustrated in Fig. 2 with dashed line (L-position of the switch corresponds to the learning phase and I-position to the implementation phase of the network). Since for each $x\in[0,1)$ all synapses of adaline neurons are inactive (driven by '0' from the second layer) except one (excited by '1'), these neurons factually operate as if they have only one synapse (instead of 2^m) and move in correspondence to the active output in the second layer (instead of being fixed). If the network input x is assigned by the second layer to the bin $[n/2^m,(n+1)/2^m)$ then the outcome of adaline neurons is $a_{mn,N}$ and $b_{mn,N}$.

4.2. Network Estimate

After presenting N patterns, for $x\in[n/2^m,(n+1)/2^m)$ the network yields the estimate (cf. (4.1a)-(4.1b) and Fig. 2)

$$R_N(x;m) = \frac{g_N(x;m)}{f_N(x;m)} \tag{4.2}$$

where

$$g_N(x;m) = \left(\frac{2^m}{N}\right)\sum_{k=1}^{N} y_k I_{[\frac{n}{2^m},\frac{n+1}{2^m})}(x_k) \; ; \quad f_N(x;m) = \left(\frac{2^m}{N}\right)\sum_{k=1}^{N} I_{[\frac{n}{2^m},\frac{n+1}{2^m})}(x_k) \tag{4.3}$$

Remark 2. For the sake of simplicity, the weights $2^m/N$ are omitted in the architecture in Fig. 2 as inessential for operation of the network, however they will be used in further analysis.

Equivalently, for $x\in[n/2^m, (n+1)/2^m)$ we obtain

$$R_N(x;m) = \frac{a_{mn,N}}{b_{mn,N}} \tag{4.4}$$

where

$$a_{mn,N} = \sum_{\{k:\, x_k \in [\frac{n}{2^m},\frac{n+1}{2^m})\}} y_k \; ; \qquad b_{mn,N} = \#\{\, x_k \in [\frac{n}{2^m},\frac{n+1}{2^m})\,\} \tag{4.5}$$

are synaptic weights established due to the training algorithm in (4.1a)-(4.1b). We see that $R_N(x;m)$ is the local sample mean value of the output measurements y_k for x_k falling in the interval $[n/2^m,(n+1)/2^m)$. Hence, for x in $[0,1)$ we obtain from the network a piecewise constant (stepwise) estimation of non-linearity.

For $x\notin[0,1)$, when both adaline outputs equal '0', the network output is set as '0'.

Remark 3. As a by-product the net provides the estimate of the probability density function $f(x)$ of the input x (cf. (4.3) and (4.5)):

$$f_N(x;m) = \frac{2^m}{N} b_{mn,N} = \frac{2^m}{N} \#\{\, x_k \in [\frac{n}{2^m},\frac{n+1}{2^m})\,\}$$

for $x \in [n/2^m,(n+1)/2^m)$. This is the histogram-type estimate of a density (e.g. Scott, 1992; Chapter 3) and is obtained as the weighted output (with the weight $2^m/N$) of the second adaline neuron (Fig. 2).

4.3. Reference to the Haar Approximation

For the observation model in (2.1), let us denote $g(x) = R(x)f(x)$. In view of equations (3.7) and (3.8) in Section 3, we recognize that for $x\in[n/2^m, (n+1)/2^m)$ the functions

$$g(x;m) = 2^{m/2}\,\alpha_{mn} \quad \text{and} \quad f(x;m) = 2^{m/2}\,\beta_{mn} \tag{4.6}$$

where

$$\alpha_{mn} = 2^{m/2}\int_{n/2^m}^{(n+1)/2^m} g(x)\,dx \qquad \text{and} \qquad \beta_{mn} = 2^{m/2}\int_{n/2^m}^{(n+1)/2^m} f(x)\,dx \tag{4.7}$$

are approximations (local mean values) of the functions $g(x)$ and $f(x)$ in the Haar approximation

space V_m as in (3.4). We note that due to Assumption 5 in Section 2, $g(x)$ and $f(x)$ are square integrable functions over the interval [0,1): $\int_0^1 g^2(x)\,dx < \infty$, $\int_0^1 f^2(x)\,dx < \infty$. Simultaneously, owing to the definition of the Haar basis functions $\phi_{mn}(x)$ in (3.2) and to equation (3.7), we ascertain that the coefficients α_{mn} and β_{mn} are factually equal to

$$\alpha_{mn} = \int_{-\infty}^{\infty} g(x)\,\varphi_{mn}(x)\,dx = E\{y\,\varphi_{mn}(x)\} \tag{4.8}$$

$$\beta_{mn} = \int_{-\infty}^{\infty} f(x)\,\varphi_{mn}(x)\,dx = E\{\varphi_{mn}(x)\} \tag{4.9}$$

i.e. are simple expectations. The latter conclusion follows immediately from equation (2.1), Assumptions 1-3 in Section 2 and the fact that $f(x)$ is a probability density function. For the Haar basis functions in (3.2) the usual sample mean estimates of the theoretical expectations in (4.8) and (4.9) (computed from random observations $\{(x_k, y_k);\, k = 1, 2, \ldots, N\}$) take the form

$$\alpha_{mn,N} = \frac{1}{N}\sum_{k=1}^{N} y_k\,\varphi_{mn}(x_k) = \frac{2^{m/2}}{N}\,a_{mn,N}\;; \qquad \beta_{mn,N} = \frac{1}{N}\sum_{k=1}^{N} \varphi_{mn}(x_k) = \frac{2^{m/2}}{N}\,b_{mn,N}$$

where $a_{mn,N}$ and $b_{mn,N}$ are as in (4.5). Therefore, $g_N(x;m)$ and $f_N(x;m)$ in (4.3) are in fact empirical estimations of the Haar approximators $g(x;m)$ and $f(x;m)$ in (4.6). In this sense they are the Haar estimators (i.e. empirical Haar smoothers) of the functions $g(x)$ and $f(x)$ in the Haar space V_m, for $x \in [n/2^m, (n+1)/2^m)$.

Remark 4. Denoting $g(x) = \bar{R}(x)f(x)$ for the measurement model in (2.5)-(2.7), the same conclusion can be drawn for the Hammerstein system (see Remark 1 in Section 2).

4.4. Accuracy and Asymptotic Behaviour

Consider the pointwise mean square errors of the estimators $g_N(x;m)$ and $f_N(x;m)$. As is well-known, the errors decompose into bias and variance error parts:

$$E\left[g_N(x;m) - g(x)\right]^2 = \text{var}\{g_N(x;m)\} + \text{bias}^2\{g_N(x;m)\} \tag{4.10}$$

$$E\left[f_N(x;m) - f(x)\right]^2 = \text{var}\{f_N(x;m)\} + \text{bias}^2\{f_N(x;m)\} \tag{4.11}$$

For the observation model in (2.1) and $x \in [n/2^m, (n+1)/2^m)$ we get after straightforward calculation that (see (A.2)-(A.3) and (A.7)-(A.8) in Appendix A)

$$\text{bias}\{g_N(x;m)\} = 2^m \int_{n/2^m}^{(n+1)/2^m} g(x)\,dx - g(x) \tag{4.12}$$

$$\text{bias}\{f_N(x;m)\} = 2^m \int_{n/2^m}^{(n+1)/2^m} f(x)\,dx - f(x) \tag{4.13}$$

and

$$\text{var}\{g_N(x;m)\} \leq \left(\frac{2^m}{N}\right)(M_R^2 + \sigma_z^2)M_f \tag{4.14}$$

$$\text{var}\{f_N(x;m)\} \leq \left(\frac{2^m}{N}\right)M_f \tag{4.15}$$

where M_R, $M_f > 0$ are the bounds in (2.3) and (2.4) in Assumption 5 and σ_z^2 is the variance of the output noise (Assumption 2). Owing to (4.6) and (4.7) we see that

$$\text{bias}\{g_N(x;m)\} = g(x;m) - g(x)$$

$$\text{bias}\{f_N(x;m)\} = f(x;m) - f(x)$$

i.e. the bias errors are pointwise approximation errors associated with the Haar approximation method. In turn, the variance errors are estimation errors due to the randomness of the observations (see Appendix A2). Specifically, the variance of $g_N(x;m)$ and $f_N(x;m)$ is the mean square estimation error of the approximator $g(x;m)$ and $f(x;m)$ by means of $g_N(x;m)$ and $f_N(x;m)$ (cf. (4.6)-(4.7) and (A.1)-(A.3) in Appendix A).

Remark 5. Since (see (3.8) and (3.9) in Section 3)

$$\text{bias}\{g_N(x;m)\} \to 0 \quad \text{as} \quad m \to \infty$$

$$\text{bias}\{f_N(x;m)\} \to 0 \quad \text{as} \quad m \to \infty$$

for every input point x at which g and f are continuous functions and simultaneously, for fixed number N of data, the variance bounds in (4.14) and (4.15) grow with growing m, the expressions in (4.12)-(4.13) and (4.14)-(4.15) reflect a fundamental phenomenon, common to all smoothing methods, that the two kinds of errors have antagonistic behaviour. As a function of smoothing bandwidth inverse (in our case $(1/2^m)^{-1} = 2^m$), the bias decreases and the variance increases (see e.g. Prakasa Rao, 1983; Eubank, 1988; Härdle, 1990; Scott, 1992). Such a behaviour is intuitively clear. In our case, the former follows immediately from the nature of the bias error. In turn, the latter is explained by the fact that with increasing scale m each of the smoothing regions $[n/2^m, (n+1)/2^m)$ in the resolution grid becomes narrower and hence, for fixed N, smaller number of data is used for estimation (smoothing) of a function. Thereby the estimate becomes rougher. Note that for m tending to infinity the estimates $g_N(x;m)$ and $f_N(x;m)$ in (4.3) can in theory approach arbitrarily large values. However, for each fixed scale m the variance bounds decrease if the number N of training data is increased. Hence, for each m the variance errors $\text{var}\{g_N(x;m)\}$ and $\text{var}\{f_N(x;m)\}$ can be made arbitrarily small by growing the number of training data, and asymptotically

$$\text{var}\{g_N(x;m)\} \to 0 \quad \text{as} \quad N \to \infty$$

$$\text{var}\{f_N(x;m)\} \to 0 \quad \text{as} \quad N \to \infty$$

In view of (4.12)-(4.15) and Remark 5 it is apparent that if the scale (resolution level) factor

m (and hence the size of the network in Fig. 2) is chosen as a function of the number *N* of training data, i.e. $m = m(N)$, and selected in such a way that

$$m(N) \to \infty \quad \text{as} \quad N \to \infty \tag{4.16}$$

and

$$2^{m(N)} / N \to 0 \quad \text{as} \quad N \to \infty \tag{4.17}$$

then (cf. (4.10), (4.11) and (4.12)-(4.15))

$$E\left[g_N(x;m) - g(x) \right]^2 \to 0 \tag{4.18}$$

$$E\left[f_N(x;m) - f(x) \right]^2 \to 0 \tag{4.19}$$

as $N \to \infty$. The convergence takes place for each network input x in [0,1) at which $g(x)$, respectively $f(x)$, is a continuous function (compare (3.8)-(3.9)). This results in the following.

THEOREM 1. Let Assumptions 1-5 of Section 2 hold. If the scale factor *m* of the net in Fig. 2 depends on *N*, $m = m(N)$, and is selected according to (4.16)-(4.17) then

$$R_N(x;m) \to R(x) \quad \textit{in probability} \quad \text{as} \quad N \to \infty \tag{4.20}$$

for every network input $x \in [0,1)$ where $R(x)$ and $f(x)$ are continuous functions.

PROOF. The conclusion follows directly from the mean square convergence of $g_N(x;m)$ and $f_N(x;m)$ in (4.18) and (4.19), the fractional form of the network estimate $R_N(x;m)$ ((4.2)) and the fact that $g(x) = R(x)f(x)$. Note that by Assumption 5, $f(x) > 0$ for each $x \in [0,1)$.

Remark 6. Similar conclusion is obtained for the Hammerstein system. In such a case the bias expressions for $g_N(x;m)$ and $f_N(x;m)$ are the same as in (4.12) and (4.13) with the only difference that $g(x) = \bar{R}(x)f(x)$, where $\bar{R}(x) = cR(x) + d$ (cf. Remark 1 and Appendix B1). In turn, the bound for the variance error of $f_N(x;m)$ is identical with (4.15) and for the variance of $g_N(x;m)$ we obtain the bound (see (B.7) in Appendix B2)

$$\begin{aligned} \text{var}\{ g_N(x;m) \} \le & \left(\frac{2^m}{N} \right) (\bar{M}_R^2 + \bar{\sigma}_z^2) M_f + \\ & + \left(\frac{1}{N} \right) (2\tilde{\sigma}_R^2 |\alpha_N| + 4 M_R \bar{M}_R |\beta_N|) M_f^2 \end{aligned}$$

where $\bar{M}_R$, $\bar{\sigma}_z^2$, $\tilde{\sigma}_R^2$, α_N, β_N are appropriate constants defined in Appendix B2, α_N and β_N dependent on *N*. If $\lambda_p = 0$ for $p \ge 1$ and $\lambda_0 = 1$, i.e. the Hammerstein system is reduced to the memoryless element and consequently (see (2.6a)-(2.6b) in Remark 1 and (B.2), (B.3), (B.4) and definition of $\bar{M}_R$, $\bar{\sigma}_z^2$ in Appendix B2)

$$\bar{R}(x) = R(x), \quad \tilde{\sigma}_z^2 = 0, \quad \alpha_N = 0, \quad \beta_N = 0$$

and $\bar{M}_R = M_R$, $\bar{\sigma}_z^2 = \sigma_z^2$, we get the bound in (4.14). Since for asymptotically stable output filter (see Remark 1),

$$\sum_{k=1}^{\infty} \sum_{p=1}^{\infty} |\lambda_p \lambda_{p+k}| \le C_1 < \infty, \qquad \sum_{k=1}^{\infty} |\lambda_k| \le C_2 < \infty$$

where C_1, C_2 are some positive constants, and hence $|\alpha_N| \le C_1$, $|\beta_N| \le C_2$ for each N (see (B.3) and (B.4) in Appendix B2), we obtain asymptotically (for $m = m(N)$ as in (4.16)-(4.17) and a large enough N)

$$\text{var}\{g_N(x;m)\} \le \left(\frac{2^m}{N}\right)(\bar{M}_R^2 + \bar{\sigma}_z^2)M_f$$

which is similar to (4.14). This, along with the similarities indicated above, yields under assumptions of Theorem 1 the convergence as in (4.20), however with respect to the function $\bar{R}(x) = cR(x) + d$, i.e.

$$R_N(x;m) \to \bar{R}(x) \quad \textit{in probability} \quad \text{as} \quad N \to \infty$$

The above shows that from the viewpoint of the network outcome, recovering of the non-linearity in the Hammerstein system (measurement model (2.5)-(2.7)) is asymptotically equivalent to solving the standard task for the memoryless system (observation model (2.1)) with the artificial characteristic $R(x) = \bar{R}(x)$, the non-linearity bound $M_R = \bar{M}_R$ and the output noise variance $\sigma_z^2 = \bar{\sigma}_z^2$.

The theorem says that the network in Fig. 2 can provide an arbitrarily precise estimate of non-linearity, however to this end the scale factor m must grow as in (4.16)-(4.17), i.e. the network must expand. This can be inconvenient from the technical viewpoint.

5. MODULAR NETWORKS

We shall introduce a class of modular networks for recovering non-linear characteristics. The basic building block for these networks will be the net in Fig. 2. In this class an arbitrarily high resolution of data (scale m) will be achieved by applying a suitable number of neural modules, instead of expanding the basic network.

For the fixed scale factor $m = m_0$ the net in Fig. 2 works with the resolution $1/2^{m_0}$ on the interval [0,1) (i.e. it does not differentiate the inputs $x \in [0,1)$ which belong to the same subinterval of the length $1/2^{m_0}$ in the resolution grid; Section 4.1) and yields, after training, the estimate $R_N(x;m_0)$ as in (4.2)-(4.3) or '0' for $x \notin [0,1)$. Similar network, denoted further by $C_0(m_0)$ and treated as the neural module, can be used for recovering non-linearity with higher resolution $1/2^{m_0+m_1}$, $m_1 \ge 1$, (i.e. with better precision) on an interval $[a/2^{m_1},(a+1)/2^{m_1})$ of smaller length $1/2^{m_1}$, where a is a constant. To this end, it suffices to pre-scale and translate the inputs $x \in [a/2^{m_1},(a+1)/2^{m_1})$ (i.e. the appropriate training data x_k and next the current inputs x) to the form $2^{m_1}x-a$ and then to apply such converted quantities to the module $C_0(m_0)$. As an immediate consequence of this, in order to obtain the resolution $1/2^{m_0+m_1}$ on the interval [0,1) it is enough to use a set of 2^{m_1} modules $C_0(m_0)$, associated with the subintervals $[n_1/2^{m_1}, (n_1+1)/2^{m_1})$ of the length $1/2^{m_1}$ and operating respectively on the pre-scaled and shifted inputs $2^{m_1}x-n_1$, where $n_1 = 0,1,...,2^{m_1}-1$. This leads to the composite modular network structure shown in Fig. 3a, where the

component units $C_0(m_0)$ are connected in parallel and work autonomously. After presenting N training patterns $\{(x_k,y_k)\}$ such a network yields as a product the estimate $R_N(x;m_0+m_1)$ for $x\in[0,1)$ and '0' otherwise.

Let us denote the net in Fig. 3a (operating with the resolution $1/2^{m_0+m_1}$) by $C_1(m_0+m_1)$:

$$C_1(m_0+m_1) = \left\{ 2^{m_1} \parallel \text{modules } C_0(m_0) \right\}$$

Proceeding in the same way as above, but considering $C_1(m_0+m_1)$ as the building block, we can build the modular network $C_2(m_0+m_1+m_2)$ as in Fig. 3b. This network, composed of 2^{m_2} parallel blocks $C_1(m_0+m_1)$, ensures the resolution $1/2^{m_0+m_1+m_2}$ of x-data from the interval [0,1). Each of the component elements operates (in the training and implementation phase) on the pre-scaled and shifted inputs $2^{m_2}x-n_2$, $n_2 = 0,1,...,2^{m_2}-1$, and is related to the adequate subinterval $[n_2/2^{m_2}, (n_2+1)/2^{m_2})$ of the length $1/2^{m_2}$. Schematically,

$$C_2(m_0+m_1+m_2) = \left\{ 2^{m_2} \parallel \text{modules } C_1(m_0+m_1) \right\}$$

After training the network produces the estimate $R_N(x;m_0+m_1+m_2)$ of the form as in (4.2)-(4.3) for $x\in[0,1)$ and '0' for $x\notin[0,1)$.

Generally, using as the building block in the i-th step of the 'recursive' assembling procedure the modular network $C_{i-1}(m_0+m_1+...+m_{i-1})$ composed in step i-1, one can build up the modular network $C_i(m_0+m_1+...+m_i)$ shown in Fig. 3c:

$$C_i(m_0+m_1+...+m_i) = \left\{ 2^{m_i} \parallel \text{modules } C_{i-1}(m_0+m_1+...+m_{i-1}) \right\}$$

The architecture of the network repeats the typical structure of all modules (compare Figs. 3a-c). The network ensures the resolution $1/2^{m_0+m_1+...+m_i}$ of x-data and yields after training the estimate $R_N(x;m_0+m_1+...+m_i)$ for $x\in[0,1)$. 'Embedding' of the unit $C_{i-1}(m_0+m_1+...+m_{i-1})$ (ensuring lower resolution $1/2^{m_0+m_1+...+m_{i-1}}$ of the data) in the more 'precise' neural architecture $C_i(m_0+m_1+...+m_i)$ (with higher data resolution capability $1/2^{m_0+m_1+...+m_i}$), $C_{i-1}(m_0+m_1+...+m_{i-1})\subset C_i(m_0+m_1+...+m_i)$, mirrors embedding of the Haar approximation space $V_{m'}$, $m' = m_0+m_1+...+m_{i-1}$, in the approximation space $V_{m''}$, $m'' = m'+m_i$, $V_{m'}\subset V_{m''}$, in the chain in (3.5).

Modular networks of arbitrarily high resolution capability can be readily composed in this manner using the basic unit $C_0(m_0)$ or more complex modules $C_i(m_0+m_1+...+m_i)$, $i = 1,2,...$, as building blocks. As it follows from the above construction, the scale parameters $m_1, m_2, ..., m_i$ are then degrees of freedom i.e. in the class of modular networks the same data resolution ability can be achieved in many ways (by various modular networks). Each of the modular networks can be easily expanded by adding new modules. In result, high resolution of training data, and thus refinement of the network estimate (equations (4.2)-(4.5)), can be achieved without any difficulty, by only employing a suitable number of modules (of freely selected data resolution ability). Within

each of the modular structures, the basic modules $C_0(m_0)$ are trained automatically, according to the routine (4.1a)-(4.1b).

We note that from the viewpoint of the network outcome each modular network $C_i(m_0+m_1+...+m_i)$, $i = 1,2,...$, is equivalent to the network as in Fig. 2 with the artificial scale $m = m_0+m_1+...+m_i$.

6. EFFICIENCY ANALYSIS

Consider the general modular network $C_i(m_0+m_1+...+m_i)$, $i = 1,2, ...$. Further on, the scale m will stand for $m_0+m_1+...+m_i$ and changing m will mean changing of an arbitrary m_i, $i = 1,2, ...$ (a degree of freedom), except m_0 (the scale factor of the basic module $C_0(m_0)$) which will be treated as being fixed. The artificial scale m will be identified with the complexity of the modular network as m is the $\log_2$-cardinality of the set of all perceptron neurons in the modular net. We shall check how should grow the artificial scale m of the modular network (without any distinction among m_i, $i = 1,2, ...$) with the total number of training data N as to ensure recovering of a non-linear characteristic in an optimal manner. To this end, we shall first study the bias error of $g_N(x;m)$ and $f_N(x;m)$ (equations (4.12)-(4.13)) in more detail. In view of Remark 6, the discussion will be confined to the standard task, with the observation model in (2.1).

Let x_0 be an interior point of the interval $[n/2^m, (n+1)/2^m)$. We shall distinguish the following classes of local smoothness of the non-linearity $R(x)$/input probability density $f(x)$ around x_0:

Class 1. $R(x)\in Lip(x_0;r)/f(x)\in Lip(x_0;s)$: The non-linearity $R(x)$/input density $f(x)$ is a locally Lipschitz function around the point x_0, i.e. for $x\in[n/2^m, (n+1)/2^m)$ in a small neighbourhood of x_0 it holds

$$\left|R(x) - R(x_0)\right| \le L_R\left|x - x_0\right|^r \tag{6.1}$$

$$\left|f(x) - f(x_0)\right| \le L_f\left|x - x_0\right|^s \tag{6.2}$$

where $r,s\in(0,1]$ and L_R, L_f are some positive constants.

Class 2. $R(x)\in C^p(x_0)/f(x)\in C^q(x_0)$: The non-linearity $R(x)$/input density $f(x)$ is locally $p \ge 1/ q \ge 1$ times differentiable around x_0, i.e. for $x\in[n/2^m, (n+1)/2^m)$ in a small neighbourhood of x_0 we have (by the Taylor formula):

$$\left|R(x) - R(x_0)\right| \le L_{R1}\left|x - x_0\right| + L_{R2}\left|x - x_0\right|^2 + ... + L_{Rp}\left|x - x_0\right|^p \tag{6.3}$$

$$\left|f(x) - f(x_0)\right| \le L_{f1}\left|x - x_0\right| + L_{f2}\left|x - x_0\right|^2 + ... + L_{fq}\left|x - x_0\right|^q \tag{6.4}$$

where L_{Ri}, L_{fj} are some positive constants.

Class 3. $R(x),f(x)\in Const(x_0)$: The non-linearity $R(x)$/input density $f(x)$ is a locally constant function around the point x_0, i.e. for each $x\in[n/2^m, (n+1)/2^m)$ in a small neighbourhood of x_0 it holds $R(x) = R(x_0)$, respectively $f(x) = f(x_0)$, or equivalently (considering that $\left|x - x_0\right| \le 1/2^m < 1$)

$$\left|R(x) - R(x_0)\right| \le L_{Rc}\left|x - x_0\right|^{r_c} \tag{6.5}$$

$$\left|f(x) - f(x_0)\right| \le L_{fc}\left|x - x_0\right|^{s_c} \tag{6.6}$$

for arbitrarily large exponents $r_c, s_c > 0$, any positive constants L_{Rc}, L_{fc}.

With respect to the function $g(x) = R(x)f(x)$, one can easily observe that local smoothness of this function is determined by local smoothness of more crude function from among $R(x)$ and $f(x)$. For example,

1. If $R(x) \in Lip(x_0;r)$ and $f(x) \in Lip(x_0;s)$ (class 1) then $g(x) \in Lip(x_0;\gamma)$ where $\gamma = \min(r,s)$, and in a small neighbourhood of x_0 we have

$$\left|g(x) - g(x_0)\right| \le L_g\left|x - x_0\right|^{\gamma} \tag{6.7}$$

where the Lipschitz constant is $L_g = M_f L_R + M_R L_f$ (see (2.3)-(2.4) and (6.1)-(6.2)).

2. If $R(x) \in C^p(x_0)$ and $f(x) \in C^q(x_0)$ (class 2) then (by the Leibnitz formula) $g(x) \in C^{\nu}(x_0)$ where $\nu = \min(p,q)$, and for x in a small neighbourhood of x_0 it holds

$$\left|g(x) - g(x_0)\right| \le L_{g1}\left|x - x_0\right| + L_{g2}\left|x - x_0\right|^2 + \ldots + L_{g\nu}\left|x - x_0\right|^{\nu} \tag{6.8}$$

where $L_{gh} = M_f L_{Rh} + M_R L_{fh} + \sum_{l=1}^{h-1} L_{Rl} L_{f,h-l}$ (see (2.3)-(2.4) and (6.3)-(6.4)).

3. If $R(x) \in C^p(x_0)$ (whence $R(x) \in Lip(x_0;1)$) and $f(x) \in Lip(x_0;s)$ then $g(x) \in Lip(x_0;s)$.

4. If $R(x) \in Lip(x_0;r)$ and $f(x) \in C^q(x_0)$ (hence $f(x) \in Lip(x_0;1)$) then $g(x) \in Lip(x_0;r)$.

5. If $R(x)$, $f(x) \in Const(x_0)$ (class 3) then obviously $g(x) \in Const(x_0)$ and in a small neighbourhood of x_0 we have $g(x) = g(x_0)$, or equivalently

$$\left|g(x) - g(x_0)\right| \le L_{gc}\left|x - x_0\right|^{\gamma_c} \tag{6.9}$$

for arbitrarily large exponent $\gamma_c > 0$, any positive constant L_{gc} (for the sake of unification we shall assume that $\gamma_c = \min(r_c, s_c)$ and $L_{gc} = M_f L_{Rc} + M_R L_{fc}$; see expressions in (6.5)-(6.6) and (6.7)).

It is moreover quite obvious that if $R(x) \in Const(x_0)$ but $f(x) \in Lip(x_0;s)$ or $f(x) \in C^q(x_0)$ then, respectively, $g(x) \in Lip(x_0;s)$ and $g(x) \in C^q(x_0)$. If, conversely, $f(x) \in Const(x_0)$ and $R(x) \in Lip(x_0;r)$ or $R(x) \in C^p(x_0)$ then $g(x) \in Lip(x_0;r)$ or $g(x) \in C^p(x_0)$, respectively, i.e. the function $g(x)$ belongs to the same (more rough) class of smoothness as $R(x)$.

Let us now observe (see equation (4.12)) that for the network input $x_0 \in [n/2^m, (n+1)/2^m)$ we have

$$\text{bias}\{g_N(x_0;m)\} = 2^m \int_{n/2^m}^{(n+1)/2^m} (g(x) - g(x_0))\, dx$$

whence

$$\left|\text{bias}\{g_N(x_0;m)\}\right| \le 2^m \int_{n/2^m}^{(n+1)/2^m} \left|g(x) - g(x_0)\right| dx \tag{6.10}$$

and that for m sufficiently large, local properties of $g(x)$ around x_0 hold in the whole interval $[n/2^m , (n+1)/2^m)$. Hence, owing to (6.7), (6.8) and (6.9), we easily conclude that for a large enough m the following bound is valid for the bias error of $g_N(x_0;m)$:

$$| \text{bias}\{ g_N(x_0;m) \} | \le C_b^g \, 2^{-\rho m} \tag{6.11}$$

where $C_b^g = L_g$ and $\rho = \gamma$ for $g(x) \in Lip(x_0;\gamma)$, $0 < \gamma \le 1$; $C_b^g = L_{g1}$ and $\rho = 1$ for $g(x) \in C^{\nu}(x_0)$, $\nu \ge 1$; $C_b^g = L_{gc}$ and $\rho > 0$ and arbitrarily large for $g(x) \in Const(x_0)$.

Similarly, from equation (4.13) and (6.2), (6.4) and (6.6) one can derive that for large m

$$| \text{bias}\{ f_N(x_0;m) \} | \le C_b^f \, 2^{-\xi m} \tag{6.12}$$

where $C_b^f = L_f$ and $\xi = s$ for $f(x) \in Lip(x_0;s)$, $0 < s \le 1$; $C_b^f = L_{f1}$ and $\xi = 1$ for $f(x) \in C^q(x_0)$, $q \ge 1$; $C_b^f = L_{fc}$ and $\xi > 0$ and arbitrarily large for $f(x) \in Const(x_0)$.

Remark 7. In the above bounds we have used that $| x - x_0 | \le 1/2^m$ and that for large values of m the higher-order terms in (6.8) and (6.4) can be neglected. For $g(x) \in Const(x_0)$ and $f(x) \in Const(x_0)$ we factually have $| \text{bias}\{ g_N(x_0;m) \} | = 0$ and $| \text{bias}\{ f_N(x_0;m) \} | = 0$, respectively.

Comparing (6.11) and (6.12) and including the fact that $\xi \ge \rho$ (see the remarks concerning smoothness of the function $g(x)$) we note that generally the bias error of $g_N(x_0;m)$ converges to zero as $m \to \infty$ not faster than the bias of $f_N(x_0;m)$ (cf. Remark 5).

Taking into account (6.11) and (6.12) along with the variance bounds in (4.14) and (4.15), we get asymptotic bounds for the mean square errors in (4.10) and (4.11), at the input point x_0:

$$E\left[g_N(x_0;m) - g(x_0) \right]^2 \le C_V^g \left(\frac{2^m}{N} \right) + C_B^g \, 2^{-2\rho m} \tag{6.13}$$

$$E\left[f_N(x_0;m) - f(x_0) \right]^2 \le C_V^f \left(\frac{2^m}{N} \right) + C_B^f \, 2^{-2\xi m} \tag{6.14}$$

where $C_V^g = (M_R^2 + \sigma_z^2) M_f$, $C_V^f = M_f$, $C_B^g = (C_b^g)^2$ and $C_B^f = (C_b^f)^2$. This further yields

$$E\left[g_N(x_0;m) - g(x_0) \right]^2 \le C_{BV}^g \left(\frac{2^m}{N} + 2^{-2\rho m} \right) \tag{6.13a}$$

$$E\left[f_N(x_0;m) - f(x_0) \right]^2 \le C_{BV}^f \left(\frac{2^m}{N} + 2^{-2\xi m} \right) \tag{6.14a}$$

where $C_{BV}^g = \max\{ C_B^g, C_V^g \}$ and $C_{BV}^f = \max\{ C_B^f, C_V^f \}$. Let m, the artificial scale of modular network, be now a function of N - the number of training data, $m = m(N)$, satisfying the conditions (4.16)-(4.17) and denote

$$2^{m(N)}/N + 2^{-2\rho m(N)} = a_N ; \qquad 2^{m(N)}/N + 2^{-2\xi m(N)} = b_N \tag{6.15}$$

Obviously, a_N, $b_N \to 0$ as $N \to \infty$ and the network estimate $R_N(x_0;m)$ converges to $R(x_0)$ (in probability) for each of the distinguished classes of local smoothness of $R(x)$ and $f(x)$ (see Theorem

1). Employing (6.13a), (6.14a), (6.15) and using Lemma in Appendix C, we infer that for a given scale selection strategy $m = m(N)$ the appropriate rate of convergence is

$$| R_N(x_0;m) - R(x_0) | = O\left(\sqrt{2^{m(N)}/N + 2^{-2\rho m(N)}}\right) \quad \textit{in probability} \tag{6.16}$$

where we have exploited that $\xi \geq \rho$. The rate in (6.16) (efficiency of the modular network) can be optimized by suitable selection of $m(N)$. The following theorem holds.

THEOREM 2. Let $R(x)$ and $f(x)$ belong locally (around x_0) to one of the smoothness classes 1-3. If the artificial scale of modular network is adapted to N, $m = m(N)$, and selected according to the rule

$$m(N) = \left[\frac{1}{2\rho+1}\log_2 N\right] \tag{6.17}$$

then the network estimate $R_N(x_0;m)$ attains asymptotically (for large values of N) optimum convergence rate

$$| R_N(x_0;m) - R(x_0) | = O\left(N^{-\rho/(2\rho+1)}\right) \quad \textit{in probability} \tag{6.18}$$

where $[v]$ stands for the integer part of v and ρ is as in (6.11).

PROOF. The conclusion is straightforward. The optimum scale selection rule in (6.17) follows from standard minimization of the right hand side of (6.16) with respect to m. The rate in (6.18) is obtained after inserting (6.17) into (6.16). Then the two antagonistic components are balanced (see Remark 5). We note that the rule in (6.17) satisfies conditions (4.16) and (4.17).

The optimum scale in (6.17) (optimum network complexity) and the convergence rate in (6.18) depend, through the index ρ, on local smoothness of more crude function from among $R(x)$ and $f(x)$ (cf. definition of ρ in (6.11) and the discussion of smoothness of $g(x)$; items 1-5). We can easily infer specific rates of convergence which can be achieved by the network estimate for specific combinations of smoothness of $R(x)$ and $f(x)$. Examples given below will be used in further considerations.

COROLLARY 1. For $R(x) \in Lip(x_0;r)$ and $f(x) \in Lip(x_0;s)$, $0 < r,s \leq 1$, we get

$$| R_N(x_0;m) - R(x_0) | = O\left(N^{-\gamma/(2\gamma+1)}\right) \quad \textit{in probability}$$

where $\gamma = \min(r,s)$, provided that the artificial scale $m = m(N)$ is selected as

$$m(N) = \left[\frac{1}{2\gamma+1}\log_2 N\right]$$

COROLLARY 2. For $R(x) \in Lip(x_0;r)$, $0 < r \leq 1$ and $f(x) \in C^q(x_0)$, $q \geq 1$, or $f(x) \in Const(x_0)$:

$$| R_N(x_0;m) - R(x_0) | = O\left(N^{-r/(2r+1)}\right) \quad \textit{in probability}$$

provided that the artificial scale $m = m(N)$ is established due to the rule

$$m(N) \;=\; \left[\frac{1}{2r+1}\log_2 N\right]$$

COROLLARY 3. For $R(x)\in C^p(x_0)$, $p \geq 1$, or $R(x)\in Const(x_0)$ and $f(x)\in C^q(x_0)$, $q \geq 1$, or $f(x)\in Const(x_0)$ (except $R(x)$, $f(x)\in Const(x_0)$, simultaneously), we obtain

$$|\, R_N(x_0;m) \;-\; R(x_0)\,| \;=\; O\left(N^{-1/3}\right) \qquad \textit{in probability}$$

if the artificial scale $m = m(N)$ of modular network is chosen according to the rule

$$m(N) \;=\; \left[\frac{1}{3}\log_2 N\right]$$

COROLLARY 4. For $R(x)\in Const(x_0)$ and $f(x)\in Const(x_0)$, we get

$$|\, R_N(x_0;m) \;-\; R(x_0)\,| \;=\; O\left(N^{-1/2}\right) \qquad \textit{in probability}$$

for every artificial scale $m = m(N)$ such that $R(x) = R(x_0)$ and $f(x) = f(x_0)$ for each point x in the interval $[n/2^m, (n+1)/2^m)$ containing x_0.

All rates given above are asymptotic, i.e. take place for large number of training patterns N. They can be achieved by the modular network estimate only if the network complexity (artificial scale m) is appropriately selected and fitted to both the number N of training data and smoothness of $R(x)$ and $f(x)$. The fastest rate of convergence, of order $O(N^{-1/2})$, can be achieved for locally constant functions $R(x)$ and $f(x)$ (Corollary 4). This rate agrees with the best possible parametric rate of convergence in probability (cf. e.g. Bickel and Doksum, 1977; Chapter 4.4). In this particular case the bias error in (6.11) and (6.12) equals zero (cf. Remark 7) and the network solves factually a parametric estimation problem. The constant value $R(x_0)$ (factorized as $R(x_0)f(x_0)/f(x_0)$) plays the role of a constant parameter, θ say, measured in the presence of a zero-mean random noise, $y_k = \theta + z_k$ (cf. equation (2.1)), and the estimate $R_N(x_0;m)$ (a sample mean of y_k for x_k in the neighbourhood of x_0; see equations (4.4)-(4.5)) is standard parametric estimate of θ. Such a rate is no longer attained even if only $f(x)\notin Const(x_0)$ (Corollary 3). For $R(x)$ and $f(x)$ locally differentiable (but not locally constant), the attainable convergence speed is of slower order $O(N^{-1/3})$ (cf. Corollary 3). This is in turn the best possible non-parametric rate of convergence in probability which can be achieved for functions $R(x)$ and $f(x)$ possessing one derivative i.e. for $p = q = 1$ (more generally, for $R(x)$ and $f(x)$ belonging locally to the Lipschitz class $Lip(x_0;1)$; Stone, 1980). As we see, the rate $O(N^{-1/3})$ cannot be improved for smoother functions $R(x)$ and $f(x)$, with $p,q > 1$ derivatives, which is caused by the rough behaviour of the asymptotic bias (the same for each order $p,q \geq 1$; see (6.8), (6.11) and (6.4), (6.12)). This is a disadvantageous effect of the perceptron-based architecture of the networks and consequence of the fact that our nets provide merely piecewise constant approximations of non-linearities. For comparison, the best non-parametric rate of convergence in probability for non-linear characteristics with p derivatives is, as established by Stone (1980), $O(N^{-p/(2p+1)})$. For less smooth functions $R(x)$ or $f(x)$ (Corollaries

1 and 2), the rate of convergence is of order $O(N^{-1/(3+(1/\tau-1))})$, where $0 < \tau \leq 1$ is the smaller Lipschitz exponent of (locally) more rough function from among $R(x)$ and $f(x)$ (i.e. γ or r, respectively), and for $\tau < 1$ is slower than $O(N^{-1/3})$.

7. COMPLEXITY SELECTION

As it was established in Section 6, efficiency of the modular network and optimum network complexity depend on local smoothness of the underlying non-linearity $R(x)$ and the input probability density function $f(x)$ around particular network input. Nevertheless, taking into account Corollaries 1-4, one can recognize the law

$$m(N) = \left[\frac{1}{3}\log_2 N\right] \tag{7.1}$$

as a satisfactorily general rule for selecting the complexity (artificial scale $m = m(N)$) of the modular network, with a relatively wide range of applicability. This choice is asymptotically optimal (in the sense of convergence speed of the network estimate) for a broad class of non-linearities $R(x)$ and input densities $f(x)$, including such remarkable classes of local smoothness of $R(x)$ and $f(x)$ (around x) as $Lip(x;1)$, $C^p(x)$, $p \geq 1$, and $Const(x)$. If $R(x)$ and $f(x)$ belong to one of these smoothness classes for each $x \in [0,1)$, i.e. are at least $Lip(x;1)$ functions everywhere on $[0,1)$, then the guaranteed pointwise rate of convergence (to $R(x)$) of the estimate $R_N(x;m)$ obtained from the network of the size established according to (7.1) is, asymptotically, of order $O(N^{-1/3})$ in probability for each input point x. The rate is better, of order $O(N^{-1/2})$, for locally constant functions $R(x)$ and $f(x)$ (Corollary 4). If $R(x)$ or $f(x)$ is not as smooth and for instance $R(x) \in Lip(x_0;r)$, $r < 1$ or $f(x) \in Lip(x_0;s)$, $s < 1$ around some input $x_0 \in [0,1)$ then for the choice as in (7.1) the guaranteed rate is deteriorated to order $O(N^{-r/3})$, $O(N^{-s/3})$ and $O(N^{-\gamma/3})$, respectively, where $\gamma = \min(r,s)$ (cf. equation (6.16)). Such rates are not only slower than $O(N^{-1/3})$ but also worse than $O(N^{-r/(2r+1)})$, $O(N^{-s/(2s+1)})$ and, respectively, $O(N^{-\gamma/(2\gamma+1)})$ which would be obtained for the adequate optimum selection of the artificial scale $m = m(N)$, according to Corollary 1.

The asymptotic complexity selection law in (7.1) is justified only for large number N of training patterns. For moderate number of data (not exceeding 10^3) the rule in (7.1) is too optimistic and yields too small artificial scale m (requires too small complexity of the modular network) as to ensure satisfactory reduction of the bias (approximation) error in the network estimate. This is apparent from experimental results in Section 8. We propose below a modification of (7.1) providing an approximate formula for selection of the artificial scale $m(N)$ of the modular network for moderate number of data. The derivation of the rule is based on the assumption that convergence of the factor $f_N(x;m)$ (to $f(x)$) is faster than convergence of $g_N(x;m)$ (to $g(x)$) (see (6.13)-(6.14) and the discussion in Section 6 concerning ρ and ξ). The modified rule exploits additional information about $R(x)$ and $f(x)$, namely the bound constants in (2.3)-(2.4) and

(6.1)-(6.2), and ensures faster reduction of the bias error than (7.1).

Assume that N is sufficiently large in order to $f_N(x;m) \approx f(x)$ be valid with a negligible error but still $g_N(x;m) \neq g(x)$. Such asymmetry always takes place when the input density $f(x)$ is smoother than the recovered non-linearity $R(x)$ (cf. Section 6). Then the following approximate formula takes place

$$R_N(x;m) - R(x) \approx \frac{1}{f(x)} \left[g_N(x;m) - g(x) \right]$$

whence

$$E\left[R_N(x;m) - R(x) \right]^2 \approx \frac{1}{f^2(x)} E\left[g_N(x;m) - g(x) \right]^2 \tag{7.2}$$

Consequently, taking into account the expression in (6.13), we get for $\rho = 1$ (we assume that $R(x)$ and $f(x)$ are $Lip(x;1)$ functions, similarly as in the case of the rule in (7.1)):

$$E\left[R_N(x;m) - R(x) \right]^2 \leq \frac{1}{\delta^2} \left[C_V^g \left(\frac{2^m}{N} \right) + C_B^g 2^{-2m} \right] \tag{7.3}$$

where $C_V^g = (M_R^2 + \sigma_z^2) M_f$, $C_B^g = (M_f L_R + M_R L_f)^2$ (see (6.7), (6.11)) and M_R, M_f, δ, L_R, L_f are as in (2.3)-(2.4) and (6.1)-(6.2). Direct minimization of the right hand side of (7.3) with respect to m yields the following modified formula for selection of the artificial scale $m = m(N)$:

$$m(N) = \left[\frac{1}{3} \log_2 N + \frac{1}{3} \log_2 \frac{2 C_B^g}{C_V^g} \right] \tag{7.4}$$

The corresponding network error is then

$$E\left[R_N(x;m) - R(x) \right]^2 \leq B_{opt}(N) \tag{7.5}$$

where

$$B_{opt}(N) = \frac{3 C_B^g}{\delta^2} \left(\frac{C_V^g}{2 C_B^g} \right)^{\frac{2}{3}} \left(N^{-\frac{2}{3}} \right)$$

i.e. for the choice of m as in (7.4) the network estimate $R_N(x;m)$ converges to $R(x)$ (in the pointwise mean square sense) as fast as $O(N^{-2/3})$. Using Lemma in Appendix C we easily see that the appropriate pointwise rate of convergence in probability is then of order $O(N^{-1/3})$, i.e. asymptotically the same as for the rule in (7.1). In the above error bound, the ratio C_V^g / C_B^g represents the variance/bias balance in the network error (7.3) (see (6.13)). If $C_V^g / C_B^g < 1$, i.e. the contribution of the bias error is more essential, the rule in (7.4) yields larger values of m than (7.1), i.e. allows faster reduction of the bias. We see that the modified rule becomes more efficient (i.e. the estimation error in (7.5) decreases more distinctly) when the ratio C_V^g / C_B^g becomes smaller. For N growing large the contribution of the modifying term in (7.4) is diminished and for $N \to \infty$ the rule is reduced to the asymptotically optimal rule in (7.1).

Using (7.4) and (7.5) one can establish an approximate number of training patterns and the required complexity of the modular network which guarantee achievement of a prescribed

estimation accuracy ε:

$$E\left[R_N(x;m) - R(x)\right]^2 \le \varepsilon, \qquad \varepsilon - \text{a small number} \tag{7.6}$$

From (7.4) and the requirement $B_{opt}(N) \le \varepsilon$, after substitution of the explicit expressions for C_V^g and C_B^g, we obtain respectively

$$m(N) = \left[\frac{1}{3}\log_2 N + \frac{1}{3}\left(1 + \log_2 \frac{M_f\left(\frac{L_R}{M_R} + \frac{L_f}{M_f}\right)^2}{1 + \left(\frac{\sigma_z}{M_R}\right)^2}\right)\right] \tag{7.7}$$

$$N_{\min}(\varepsilon) = \left[\frac{1}{2M_f}\left(M_R\frac{M_f}{\delta}\sqrt{3/\varepsilon}\right)^3\left(\frac{L_R}{M_R} + \frac{L_f}{M_f}\right)\left(1 + \left(\frac{\sigma_z}{M_R}\right)^2\right)\right] \tag{7.8}$$

$$m_{\min}(\varepsilon) = \left[\log_2\left(\left(M_R\frac{M_f}{\delta}\sqrt{3/\varepsilon}\right)\left(\frac{L_R}{M_R} + \frac{L_f}{M_f}\right)\right)\right] \tag{7.9}$$

As we see, the larger the Lipschitz constants L_R and L_f (i.e. the 'slopes' of the functions $R(x)$ and $f(x)$) are in the conditions (6.1)-(6.2) and the smaller the ratio σ_z/M_R (i.e. the inverse of the signal-to-noise ratio) is for the problem under investigation, the larger the extra term is in (7.7). The needed number of training patterns in (7.8) and the required network complexity in (7.9) are small for small values of L_R, L_f and the ratio σ_z/M_R (for (7.8)) and they grow for small values of ε in (7.6) (i.e. when higher estimation accuracy is required). These values also grow for small values of δ, i.e. when the identified non-linearity may, potentially, be poorly excited somewhere in the domain of interest (see (2.4)). High efficiency of the networks is achieved for small values of the ratio M_f/δ. Then a given estimation accuracy ε can be obtained from smaller number of training data and by the network of smaller complexity. Since the smallest value $M_f/\delta = 1$ (cf. (2.4)) is achieved for the uniform distribution of the input data (then $M_f = \delta$), the obtained formulas confirm the intuitively clear fact that such kind of non-linearity excitation, not privileging any region in the domain of the identified non-linearity, ensures the best conditions for training the nets and recovering non-linear characteristics from noisy input-output measurements.

Since the rules in (7.1) and (7.7) have been derived for smooth, at least $Lip(x;1)$ non-linearities, they do not guarantee good behaviour of the network estimate in the neighbourhood of possible discontinuity (jump) points of non-linear characteristics. In particular, vanishing of the pointwise bias (approximation) error as $m \to \infty$ does not take place around such points. In turn, assuming that $f_N(x;m) \approx f(x)$, the integrated squared bias (ISB) of the network estimate $R_N(x;m) \approx g_N(x;m)/f(x)$ (i.e. the integrated squared approximation error of $R(x) = g(x)/f(x)$ by the piecewise constant approximator $R(x;m) = g(x;m)/f(x)$ in the Haar space V_m) in the interval $[n/2^m, (n+1)/2^m)$ containing a discontinuity (jump) point of a characteristic $R(x)$ is of order (cf. (7.2), (4.12) and (6.10))

$$\mathrm{ISB}_{\mathrm{discont}}\{R_N(\cdot;m)\} = \int_{n/2^m}^{(n+1)/2^m} [R(x;m) - R(x)]^2\,dx = O\left(\frac{1}{2^m}\right)$$

The same bias error in the interval $[n/2^m, (n+1)/2^m)$ wherein the function $R(x)$ is continuous (Lipschitz) is merely (see the bias term in (7.3))

$$\mathrm{ISB}_{\mathrm{cont}}\{R_N(\cdot;m)\} = \int_{n/2^m}^{(n+1)/2^m} [R(x;m) - R(x)]^2\,dx = O\left(\frac{1}{2^{3m}}\right)$$

Thus, in order to achieve similar rate of decreasing of the (integrated) bias error in the irregularity regions, the complexity of the network (artificial scale m) should be selected there as

$$m_{\mathrm{discont}}(N) = 3\,m(N) \tag{7.10}$$

where $m(N)$ is as in (7.1) or (7.7). This points at the following compromise rule for selection of the complexity of modular network, aimed at recovering both continuous and discontinuous non-linearities

$$m_{\mathrm{comp}}(N) = 2\,m(N) \tag{7.11}$$

where $m(N)$ is computed due to (7.1) or (7.7). Such a rule can be particularly recommended for moderate number of training data (small values of $m(N)$) and under small noise corrupting the non-linearity (small values of the ratio σ_z / M_R), when the bias error is the dominating component in the network error. The choice in (7.11) is a simple balance between $m(N)$ for continuous non-linearities and $3m(N)$ for discontinuous functions, suggested in (7.10). More general rule is

$$m_{\mathrm{general}}(N) = \left[Cm(N) + 0.5\right] \tag{7.12}$$

where $1 \le C \le 3$, with C close to 1 for large values of N and C close to 2 for moderate N. The value of C should also depend on the noise intensity and C should be smaller for large values of the ratio σ_z / M_R . Empirical investigation of the best choice of C is reported in the next section.

8. SIMULATION STUDY

Here, we present results of computer simulation to illustrate performance of the networks for finite number N of training patterns and to provide some empirical indications as to the choice of the constant C in the network complexity selection rule in (7.12). We confine our presentation to the measurement model in (2.1). The situation when the training data come from the Hammerstein system (equation (2.5)), as in principle the same from the viewpoint of the experimental results, is shortly mentioned in the end of the section. The test non-linear functions in the empirical investigation are in part the same as in Zhang *et al.* (1995). Nevertheless, the results presented below cannot be directly compared with those in the cited paper because of distinctly different goals and conditions of the numerical experiment (see Section 1).

In the simulation study, the input excitation $\{x_k; k = ..., -1, 0, 1, 2, ...\}$ of the non-linearity

$R(x)$ is either uniform (i.e. the best one for training the networks; Section 7) or Gaussian. The noise (white) $\{z_k; k = ..., -1, 0, 1, 2, ...\}$ corrupting the output measurements (equation (2.1)) is zero-mean Gaussian $N(0, \sigma_z)$ with the dispersion σ_z set to give (in the main part of the experiment) 5% of the maximum value M_R of $|R(x)|$ (cf. (2.3)), i.e. $\sigma_z / M_R = 5\%$. As an empirical measure of network accuracy, the following mean pointwise relative error (MRE) has been employed

$$MRE(N) = \left\{ \frac{1}{1000} \sum_{i=1}^{1000} \left(\frac{1}{P} \sum_{r=1}^{P} \left[R(x_i) - R_N^r(x_i; m) \right]^2 \right) \Big/ aver(R^2) \right\} \cdot 100\%$$

and the above value was computed at the points $x_i = i/1000$, $i = 0,1,...,999$, from the equidistant x-grid on [0,1) and from $P = 100$ independent random trials, each using the learning sequence of the length N. In the above expression $R_N^r(x;m)$ is the network estimate in the r-th trial and $aver(R^2)$ is the average value of $R^2(x)$ in the interval [0,1). The complexity of the network (artificial scale m) has been selected for each N due to the rule in (7.12) assuming $C = 0.5, 1, 1.5, 2, 2.5$ (Section 7) and $m(N)$ was in this rule computed according to (7.1). The test non-linear functions $R(x)$, $x \in [0,1)$, were selected, in part according to a referee suggestion, as follows:

Function 1: arctan non-linearity

$$R(x) = \arctan(6x)$$

Function 2: pyramid non-linearity (Zhang *et al.*, 1995)

$$R(x) = \sum_j h_j K(x - x_j^0)$$

with

$$\{h_j\} = \{ \quad 1, \quad 1, \quad 1, \quad 0, \quad -1, \quad -1, \quad -1 \}$$

$$\{x_j^0\} = \{ 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875 \}$$

Function 3: hat non-linearity (Zhang *et al.*, 1995)

$$R(x) = \sum_j (g_j x + h_j) K(x - x_j^0)$$

with

$$\{g_j\} = \{ \quad 1, \quad -1, \quad 1, \quad -2, \quad 1, \quad -1, \quad 1 \}$$

$$\{h_j\} = \{ -0.125, 0.25, -0.375, 1, -0.625, 0.75, -0.875 \}$$

$$\{x_j^0\} = \{ 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875 \}$$

Function 4: blocky non-linearity (Donoho and Johnstone, 1994)

$$R(x) = \sum_j h_j K(x - x_j^0)$$

with

$$\{h_j\} = \{4,\ -5,\ 3,\ -4,\ 5,\ -4.2,\ 2.1,\ 4.3,\ -3.1,\ 2.1,\ -4.2\}$$

$$\{x_j^0\} = \{0.1, 0.13, 0.15, 0.23, 0.25, 0.40, 0.44, 0.65, 0.76, 0.78, 0.81\}$$

where in all cases

$$K(x) = \big(1 + \mathrm{sgn}(x)\big)/2$$

and $\{x_j^0\}$ are break-points in the graph of a non-linear characteristic. The above non-linear functions are shown in Figs. 6 and 7 with thin line. Function 1 is an example of a smooth differentiable non-linearity. Function 2, piecewise constant with jumps at dyadic points $x_j^0 = j/2^3$, $j = 1,2,...,7$, represents non-smooth discontinuous non-linearities which however fit well the Haar approximation. Function 3, piecewise linear, is continuous but not differentiable and contains regions of different slope. The break-points are located at dyadic grid as for function 2. Finally, function 4 represents discontinuous wiggly varying characteristics, with jumps lying beyond dyadic grid, which hardly fit the Haar approximations.

The basic simulation tests were carried out for the uniform $U(0,1)$ probability distribution of input data. The run of MRE error for growing number N of training patterns and the network complexity selected according to (7.12) with various values of C is shown in Fig. 4 for arctan, hat and blocky non-linearities (functions 1, 3 and 4). From the plots we see that the best choice of C, ensuring optimum performance of the network, does not factually depend on the shape of the recovered non-linearity (the shape influences only the error value) and for $N \le 10^4$ and 5% noise is $C = 1.5$ (dashed line). For larger values of N better results may be expected for $C = 1$ (solid line). In our examples, for $N > 10^4$ and $C = 2$ (dotted line) the error grows visibly and for $C = 1.5$ it does not decrease further on with growing N. This observation confirms asymptotic optimality of the network complexity selection rule in (7.1).

For moderate number N of training data the best choice of C in (7.12) depends on the noise intensity (the ratio σ_z/M_R). This is illustrated in Fig. 5 in two ways for the hat non-linearity and fixed number of $N = 10^3$ training patterns, as an example. As we see in Fig. 5a, for each $C \in \{1,1.5,2\}$ (the choices $C = 0.5$ and $C = 2.5$ are excluded from the discussion as apparently worse) the error grows linearly with growing σ_z/M_R but at different rate and, for instance, $C = 2$ is a better choice than $C = 1$ for σ_z/M_R < 4%, i.e. for indeed small noise (cf. (7.11) and the associated discussion in Section 7). For σ_z/M_R < 10% the best choice is $C = 1.5$. In the particular case of σ_z/M_R = 5% (as in the main part of the experiment) deterioration of the network accuracy for $C = 1$ and $C = 2$ compared with $C = 1.5$ is nearly the same (Fig. 5b). For σ_z/M_R ≥ 10%, i.e. for a large noise, the best results are obtained for $C = 1$, as is seen from Figs. 5a,b. These empirical observations are consistent with the theoretical conclusions referring to the formulas in (7.7) and (7.11)-(7.12) and stating that artificial scale m of the network should be, for moderate N, smaller for larger values of the ratio σ_z/M_R . It is worth noticing that the choice of

C = 1, yielding the asymptotically optimal selection rule in (7.1), ensures also the smallest fluctuations in the network error for changing values of the ratio σ_z/M_R (Fig. 5b) and guarantees stable, about 5%, error for each noise (Fig. 5a).

According to a referee suggestion, in Fig. 6 the theoretical non-linearities (thin line) together with the noisy measurement points (random training data) versus the network estimates (thick line) are presented jointly to visualize which kind of errors are introduced by the network, and in which regions of the characteristics they appear to be larger. The network estimates were obtained for the uniform $U(0,1)$ input excitation from $N = 10^4$ training patterns and for the ratio σ_z/M_R = 5%, using the artificial scale selection rule in (7.12) with C = 1.5 and $m(N)$ computed according to (7.1). As we see, visually better results are obtained for arctan (smooth) and pyramid (well matching the Haar approximation) non-linearities than for hat or blocky functions, where the bias (approximation) error is visible. The estimation inaccuracy is larger in the regions where the function is (*i*) discontinuous and the jump points lie beyond dyadic grid (compare Figs. 6d and 6b), (*ii*) non-smooth (see the neighbourhood of the break-points in Fig. 6c), or (*iii*) possesses larger slope (compare appropriate regions in Figs. 6a and 6c). The best results are obtained for locally constant functions (see Figs. 6b,c and Fig. 6d far from the jump points) and for smooth (Lipschitz) non-linearities with (locally) small slope (see Fig. 6a). These empirical observations agree with the theoretical results in Section 6 regarding the convergence speed of the network estimate in dependence on local smoothness of the recovered non-linearity, and with the discussion in Section 7 concerning the bias error for continuous and discontinuous non-linearities.

For comparison, the same network estimates - obtained however for the Gaussian $N(0.5,0.15)$ distribution of the input data - are repeated in Fig. 7. In the experiment, the mean value of the input was located at the centre of the interval [0,1) and because of small dispersion $\sigma_x = 0.15$ most of the excitations (including the tail points) were enclosed in [0,1), according to the "3σ" law. Comparing the plots in Figs. 6 and 7 we see that for each non-linearity the network outcome is at present worse than for the uniform distribution of the network input. Some improvement can be only observed in the central part of [0,1) (around x = 0.5) where the Gaussian probability density is large ($\approx 1/\sigma_x\sqrt{2\pi} \cong 2.7 = M_f$; see (2.4)) and hence large number of training patterns is available. In this region the "cloud" of measurement points is dense (compare e.g. Figs. 6c, 7c and Figs. 6d, 7d). Substantial worsening of the network estimate is noticeable in boundary regions of [0,1), where the input density is small ($\approx 0.004M_f \cong 0.01 = \delta$; see (2.4)) and the identified non-linearities are rarely excited giving much smaller number of training data. The "cloud" of measurements is there sparse. This observation agrees with equations (7.8) and (7.9) and the associated discussion in Section 7 referring to the influence of the ratio M_f/δ on efficiency of the networks. In our case the ratio $M_f/\delta \cong 270$ is large and according to (7.8)-(7.9) larger number N of training data and larger artificial scale m are required in order to achieve the same estimation accuracy as for the uniform distribution of the input data, with $M_f/\delta = 1$.

In summary, the empirical results verify good performance of the networks (producing local mean value estimators of non-linearity; Sections 4 and 5) for locally smooth or locally constant functions and worse in different cases. This is presented in Figs. 6 and 7. They also confirm that accuracy of the networks depends on the relationship between the ratio σ_z / M_R and the value of the constant C in the network complexity selection rule in (7.12). For $N \leq 10^4$ the recommended choice is $C = 1.5$ for $\sigma_z / M_R < 10\%$ and $C = 1$ for $\sigma_z / M_R \geq 10\%$. For $N > 10^4$ the artificial scale m should be selected according to the asymptotic rule in (7.1).

Remark 8. Similar results and conclusions are obtained when training data are generated by the Hammerstein system, according to equation (2.5). In Fig. 8 we present, for illustration, the estimate of arctan non-linearity (function 1) obtained from the network under the same conditions as the plot in Fig. 6a, however in the situation when the non-linearity was followed by the linear dynamical filter (Fig. 1b) with the impulse response $\{\lambda_p = 1/2^p,\ 0 \leq p \leq 3;\ \lambda_p = 0,\ p \geq 4\}$. Then (see (2.6b)) $c = 1$, $d = (7/8)\int_0^1 \arctan(6x)\,dx \cong 1$ and the network has provided the estimate of the shifted non-linearity $\overline{R}(x) \cong \arctan(6x) + 1$ (cf. Remark 6 and equation (2.6a) in Remark 1).

9. CONCLUSIONS

We have proposed and examined a class of modular networks for recovering non-linear characteristics from random noisy measurements. The networks have been based on the Haar approximation of functions. Each network in the class is a parallel connection of a number of modules (sub-networks) which work autonomously. The networks are self-similar: they possess the same general architecture and more complex nets repeat the structure of modules of which they are composed (Fig. 3a-c). It has been shown that the proposed networks provide pointwise consistent estimates for a broad class of non-linear characteristics and input probability densities and it has been checked in which way the efficiency of the networks depends on local smoothness of both the target non-linearity and the probability density function of the input excitation. The optimum network size selection problem has been solved. Specifically, we have answered the following questions: 1) What is the sufficient condition on the resolution ability of the modular network to yield consistent estimates of non-linearities? 2) How can the needed resolution be obtained in the modular network? 3) What is the efficiency of the network in dependence on their complexity? 4) How can the needed complexity be approximated for a given number of training data?

The main advantages of the proposed modular networks can be summarized as follows:

1. The structure of the networks is simple, flexible and easy to expand.
2. High resolution of data (high sensitivity to the details of the recovered non-linearity) may be achieved without any difficulty; it suffices to apply a suitable number of neural modules.

3. There is a freedom in selecting component modules; the same resolution capability may be achieved in many ways, by using neural modules of various resolution ability (Section 5).
4. The networks are easy to train; training of the networks is based on simple recursive routines (Section 4.1).
5. Neural modules are standardized and easy for hardware realization as chips; similar architecture is preserved in each module starting from the basic perceptron-type component (Section 4 and 5).

As weaker points one can indicate the following:

1. For finite number N of training patterns and finite artificial scale m we obtain a crude step-wise approximation of non-linearity, with quite a large approximation error in some cases. This is an effect of a rough nature of the Haar approximation and the simplicity of perceptron-based neural modules (Sections 4-8).
2. After expanding the network (by adding new modules) the net must be re-trained. This is not a serious drawback as training procedure is easy to repeat (Section 4.1).

Applicability of the networks has been investigated for the measurement models corresponding to the memoryless and Hammerstein systems. In fact, the range of application is much wider. The networks can be used for recovering non-linear characteristics of all systems where the non-linearity to be recovered (or its version) can be expressed as a regression function (Section 2). A class of such systems is characterized by the property that the non-linearity under examination can be isolated from the rest of the system and represented by a separate static block, and the remaining part of the system acts, in a sense, as a source of nuisance noise (as was the case in (2.5)-(2.7) for the Hammerstein system). We refer to Pawlak and Hasiewicz (1998) for general description of such a class and particular examples of systems.

In the paper we have focused on the Haar wavelet approximation and modular networks composed of identical modules. Future research can be directed towards more general modular networks and application of more general wavelet approximation bases. First, modular networks composed of various kinds of modules, with discriminated resolution capability, can be considered. Such modular networks of the mixed type ensure various resolution of data in different regions of non-linearity and may be useful when individual estimation precision in individual sectors of non-linear characteristics is required. Next, a class of self-organizing wavelet modular networks with adaptive (x-driven) activation of modules may be investigated. Such networks, arising from the multiresolution and wavelet approximation of functions, may use adaptive block thresholding of wavelet approximation coefficients (referring to whole resolution levels; see, e.g., Mallat, 1998 or Härdle *et al.*, 1998 for various thresholding techniques). For such generalized modular networks the fundamental problems are the selection of appropriate wavelet basis, efficiency

analysis and hardware realization of neural modules (in contrast to the Haar basis functions, most wavelet bases are not given in the explicit form; Daubechies, 1992). Of the particular theoretical interest will be then (as pointed out by a reviewer) the analysis of the relationship between the smoothness (regularity) of the non-linear characteristic under examination and the input probability density function, on the one hand, the regularity of the mother wavelet of the implemented wavelet basis, on the other, and their combined impact on the modular networks precision. Our considerations set the stage for studies into this direction.