Int. J. Appl. Math. Comput. Sci., 2004, Vol. 14, No. 2, 221–232

LINEAR–WAVELET NETWORKS

ROBERTO K.H. GALVÃO*, VICTOR M. BECERRA**
JOÃO M.F. CALADO***, PEDRO M. SILVA***

* Instituto Tecnológico de Aeronáutica
Div. Engenharia Eletrônica
São José dos Campos – SP, 12228–900, Brazil
e-mail: kawakami@ele.ita.br

** University of Reading, Department of Cybernetics
Reading RG6 6AY, United Kingdom
e-mail: v.m.becerra@reading.ac.uk

*** ISEL, Mechanical Engineering Studies Center
1949–014 Lisboa, Portugal
e-mail: {jcalado, psilva}@dem.isel.pt

This paper proposes a nonlinear regression structure comprising a wavelet network and a linear term. The introduction of the linear term is aimed at providing a more parsimonious interpolation in high-dimensional spaces when the modelling samples are sparse. A constructive procedure for building such structures, termed linear-wavelet networks, is described. For illustration, the proposed procedure is employed in the framework of dynamic system identification. In an example involving a simulated fermentation process, it is shown that a linear-wavelet network yields a smaller approximation error when compared with a wavelet network with the same number of regressors. The proposed technique is also applied to the identification of a pressure plant from experimental data. In this case, the results show that the introduction of wavelets considerably improves the prediction ability of a linear model. Standard errors on the estimated model coefficients are also calculated to assess the numerical conditioning of the identification process.

Keywords: wavelet networks, nonlinear models, regression analysis, system identification

1. Introduction

A wavelet network is a nonlinear regression structure that implements input-output mappings as the superposition of dilated and translated versions of a single function, which is localized both in the space and frequency domains (Zhang and Benveniste, 1992). Such a structure can approximate any square-integrable function to an arbitrary precision, given a sufficiently large number of network elements (called "wavelets").

The main advantage of wavelet networks over similar architectures such as multi-layer perceptrons and networks of radial basis functions (RBF) (Haykin, 1998) is the possibility of optimizing the wavelet network structure by means of efficient deterministic construction algorithms (Kan and Wong, 1998; Zhang, 1997). However, owing to the localized nature of the wavelet basis functions, wavelet networks may not be well-suited to dealing with high-dimensional data. In fact, constructing and storing a wavelet basis of large dimension may be computationally prohibitive (Benveniste *et al.*, 1994). To circumvent this problem, Zhang (1997) proposed a construction technique which takes into account only those wavelets whose support contains at least one modelling sample. However, even by doing so, there remains the problem of providing interpolation over those regions of the input space in which modelling data are not available. Such a problem clearly intensifies with the number of inputs to the network.

In the present work, this limitation is alleviated by adding a linear term to the basic wavelet network architecture, resulting in a structure termed "linear-wavelet network" (Galvão and Becerra, 2002). Linear regressors can be seen as appropriate complements to wavelets and vice-versa. In fact, linear functions can more easily provide interpolation when the modelling samples are sparse, whereas wavelets can account for nonlinearities in the system to be identified. Notice that in the approximation of functions that display only small deviations from linearity, linear regressors may replace a much larger number of wavelets, thus allowing a more parsimonious representation to be obtained.

It should be emphasized that the introduction of linear regressors does not impair the approximation capabilities of the wavelet network, since it is equivalent to expanding the library of functions from which the elements of the network are selected. In fact, a similar strategy was adopted in (Galvão *et al.*, 1999; Souza Jr. *et al.*, 2002), where a bias function was added to the wavelets in order to improve the approximation of functions with localized low-frequency features.

A constructive procedure for building a linear-wavelet network on the basis of a given set of input-output pairs is also presented. The procedure is aimed at achieving an accurate representation of the input-output data relationship with the smallest possible number of regressors. For this purpose, a mechanism for redundance (that is, collinearity between the regressors) avoidance is explicitly employed.

For illustration, the proposed modelling technique is employed in a nonlinear system identification framework. The modelling of nonlinear dynamic input-output mappings from experimental measurements is a problem of relevance in many engineering applications (Ljung, 1999). The goal is to obtain a model that can be used to predict the future behaviour of the process from measurements taken up to the present moment. The model can also be used as a mathematical surrogate of the actual system during the design of a controller.

When no particular insight into the system properties is employed, the modelling procedure is termed "black-box" modelling. Among the classical approaches to black-box modelling, one could cite the use of Kernel estimators (Naradaya, 1964; Watson, 1964), Volterra expansions (Rugh, 1981), and B-splines (Schumaker, 1981). More recently, regression structures inspired by artificial intelligence paradigms have been popularized, such as artificial neural networks (Narendra and Parthasarathy, 1990; Poggio and Girosi, 1990), fuzzy (Takagi and Sugeno, 1985) and neuro-fuzzy models (Jang and Sun, 1995).

The representation capabilities of the wavelet network have been exploited for system identification in a number of works (Cannon and Slotine, 1995; Liu *et al.*, 2000; Souza Jr. *et al.*, 2002; Zhang, 1997). The black-box approach employed by those authors will be adopted in two examples presented in this work. In the first of them, the linear-wavelet network is compared to a conventional wavelet network in the identification of a simulated fermentation process (Zhang, 1997). The second example employs experimental data from a real pressure plant. Standard errors on the estimated model coefficients are presented to check the numerical conditioning of the identification process.

This paper is organized as follows: Section 2 presents the wavelet network structure considered in this work and introduces the proposed linear-wavelet network. Section 3 describes the constructive procedure to build a linear-wavelet network from a given set of input-output data. The examples using simulated and experimental data are presented in Sections 4 and 5, respectively. Concluding remarks and suggestions for future work are given in Section 6.

1.1. Notation

Scalars are represented in italic lowercase, vectors in bold type lowercase and matrices in bold type capitals. The symbol $\|\mathbf{x}\|$ means the Euclidean norm of $\mathbf{x}$. The i-th element of $\mathbf{x}$ is denoted by x_i. The hat symbol '$\hat{}$' indicates an estimated value. The Fourier transform of f is denoted by $\mathcal{F}f$. $L^2(\mathbb{R}^d)$ is the space of functions that are square-integrable in $\mathbb{R}^d$, that is, $L^2(\mathbb{R}^d) = \{f : \mathbb{R}^d \to \mathbb{R} \text{ s.t. } \int_{\mathbb{R}^d} |f(\mathbf{x})|^2 \, \mathrm{d}V_x < \infty\}$, where $\mathrm{d}V_x = \mathrm{d}x_1 \, \mathrm{d}x_2 \cdots \mathrm{d}x_d$ is a volume element in $\mathbb{R}^d$.

2. Linear-Wavelet Networks

2.1. Wavelet Networks

A wavelet network with one output y, d inputs $\{x_1, x_2, \ldots, x_d\}$ and L nodes can be parameterized as follows (Zhang, 1997; Zhang and Benveniste, 1992):

$$y = f_{\text{wav}}(\mathbf{x}) = \sum_{j=1}^{L} w_j v_{a_j,\mathbf{b}_j}(\mathbf{x}), \tag{1}$$

where $\mathbf{x} = [x_1 \, x_2 \, \cdots \, x_d]^T$ is the vector of inputs. Functions $v_{a_j,\mathbf{b}_j}$, called wavelets, are dilated and translated versions of a single function $v : \mathbb{R}^d \to \mathbb{R}$:

$$v_{a_j,\mathbf{b}_j}(\mathbf{x}) = a_j^{-d/2} v\left(\frac{\mathbf{x} - \mathbf{b}_j}{a_j}\right). \tag{2}$$

Function v, termed the "mother wavelet", is required to have zero mean, that is,

$$\int_{\mathbb{R}^d} v(\mathbf{x}) \, \mathrm{d}V_x = 0, \tag{3}$$

and also to be localized both in the space and frequency domains (in the sense that $|v(\mathbf{x})|$ and $|\mathcal{F}v(\boldsymbol{\omega})|$ rapidly decay to zero as $\|\mathbf{x}\| \to \infty$ and $\|\boldsymbol{\omega}\| \to \infty$, respectively) (Daubechies, 1992). A number of methods (Cannon and Slotine, 1995; Zhang *et al.*, 1995) are available to construct multidimensional mother wavelets (i.e., with $d > 1$) from one-dimensional functions $\psi : \mathbb{R} \to \mathbb{R}$ with fast decay in space and frequency. For the purposes of this work, the radial approach is adopted, that

is, $v(\mathbf{x}) = \psi(\|\mathbf{x}\|)$. This is the choice made, for instance, in (Zhang, 1997).

In (2), the dilation parameter $a_j \in \mathbb{R}^*$ controls the spread of the wavelet, whereas the translation parameter $\mathbf{b}_j \in \mathbb{R}^d$ determines its central position. It can be shown that, if pairs $(a_j, \mathbf{b}_j)$ are taken from a grid Λ given by

$$\Lambda = \left\{(\alpha^m, \mathbf{n}\beta\alpha^m);\ m \in \mathbb{Z}, \mathbf{n} \in \mathbb{Z}^d\right\} \tag{4}$$

for convenient values of $\alpha > 1$ and $\beta > 0$, then any function $f \in L^2(\mathbb{R}^d)$ can be approximated by (1) to an arbitrary precision, given a sufficiently large number of wavelets (Daubechies, 1992; Zhang, 1997). Typical choices for α, β are $\alpha = 2,\ \beta = 1$, and they are adopted throughout this work.

It is worth noticing that in many practical situations, the function f to be approximated may not belong to $L^2(\mathbb{R}^d)$. Linear functions, for instance, are not square-integrable, and thus cannot be approximated as a linear combination of wavelets over the entire $\mathbb{R}^d$. However, this limitation can be circumvented because in several applications the approximation only needs to be performed over a compact set $\mathcal{X} \subset \mathbb{R}^d$ (Cannon and Slotine, 1995). For instance, suppose that an approximation to a certain function f is to be constructed on the basis of a given modelling set of input-output pairs $\{(\mathbf{x}[k], f(\mathbf{x}[k])),\ k = 1, \ldots, M\}$. For a finite M, the modelling points $\mathbf{x}[k]$ will lie inside a compact set $\mathcal{X} \subset \mathbb{R}^d$. If the approximation of $f(\mathbf{x})$ for $\mathbf{x}$ outside $\mathcal{X}$ is not an issue (that is, one is not attempting to perform extrapolation), then the target function f can be replaced by a modified function $\tilde{f}$ defined as

$$\tilde{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \mathbf{x} \in \mathcal{X}, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

If f is square-integrable in $\mathcal{X}$, then $\tilde{f}$ is square-integrable in $\mathbb{R}^d$ and can thus be approximated to an arbitrary precision by a wavelet network.

It should be noticed that this discussion is mainly of theoretical interest, because an explicit definition of $\mathcal{X}$ is not required in the construction algorithm to be presented later. However, the reasoning discussed above is useful to relax the conditions required for the input-output mapping f to be approximated. Thus, henceforth it will be assumed that f is square-integrable in a region sufficiently large to cover all the $\mathbf{x}$ modelling points.

Remark 1. The requirement of a zero mean for v can be restated for the function ψ used in its generation. In fact, by using the hyperspherical coordinates $(r, \gamma_1, \gamma_2, \ldots, \gamma_{d-1})$ defined in Appendix, the integral in (3) can be rewritten in the following manner:

$$\begin{aligned}\int_{\mathbb{R}^d} v(\mathbf{x})\,\mathrm{d}\mathbf{x} &= \int_{\mathbb{R}^d} \psi\big(\|\mathbf{x}\|\big)\,\mathrm{d}\mathbf{x} \\ &= \int_{\gamma_1=0}^{\pi}\int_{\gamma_2=0}^{\pi}\cdots\int_{\gamma_{d-1}=-\pi}^{\pi}\left(\int_{r=0}^{\infty}\psi(r)r^{d-1}\,\mathrm{d}r\right) \\ &\quad\times\left[\prod_{j=1}^{d-2}(\sin\gamma_j)^{d-j-1}\right]d\gamma_1 d\gamma_2\cdots d\gamma_{d-1},\end{aligned} \tag{6}$$

It follows that a necessary and sufficient condition for the last integral to equal zero is

$$\int_{r=0}^{\infty}\psi(r)r^{d-1}\,\mathrm{d}r = 0 \tag{7}$$

because $(\sin\gamma_j)^{d-j-1} > 0,\ \forall\gamma_j \in (0,\pi),\ j = 1, \ldots, d-2$. Condition (7) is illustrated in the following example.

Example 1. Consider the unidimensional Mexican hat function given by

$$\psi(x) = (c - x^2)e^{-0.5x^2}, \tag{8}$$

where c is a parameter which needs to be adjusted to ensure that $\psi(\|\mathbf{x}\|)$ has zero mean. By introducing the above expression for $\psi(x)$ in (7), it follows that

$$\begin{aligned}&\int_0^{\infty}(c - r^2)e^{-0.5r^2}r^{d-1}\,\mathrm{d}r = 0 \Rightarrow c \\ &\qquad = \frac{\int_0^{\infty}e^{-0.5r^2}r^{d+1}\,\mathrm{d}r}{\int_0^{\infty}e^{-0.5r^2}r^{d-1}\,\mathrm{d}r} = \frac{I_{d+1}}{I_{d-1}}.\end{aligned} \tag{9}$$

By letting $\eta = e^{-0.5r^2}$ and $\mathrm{d}\xi = r^{d-1}\,\mathrm{d}r$, I_{d-1} can be integrated by parts, yielding

$$\begin{aligned}I_{d-1} &= \eta\xi|_{r=0}^{\infty} - \int_{r=0}^{\infty}\xi\,\mathrm{d}\eta = \underbrace{\left.\frac{e^{-0.5r^2}r^d}{d}\right|_{r=0}^{\infty}}_{0} \\ &\quad + \frac{1}{d}\int_{r=0}^{\infty}r^{d+1}e^{-0.5r^2}\,\mathrm{d}r = \frac{I_{d+1}}{d}.\end{aligned} \tag{10}$$

Thus, by using this result in (9), it follows that $c = d$. ♦

2.2. Structure of a Linear-Wavelet Network

The model structure proposed in this work has the form

$$y = f_{\text{lin}}(\mathbf{x}) + f_{\text{wav}}(\mathbf{x}), \tag{11}$$

where

$$f_{\text{lin}}(\mathbf{x}) = \boldsymbol{\theta}_{1\times d}\mathbf{x}_{d\times 1} = \sum_{i=1}^{d}\theta_i x_i, \quad \theta_i \in \mathbb{R} \tag{12}$$

is a linear term and $f_{\text{wav}}(\mathbf{x})$ is implemented by a network of radial wavelets, as in (1). Henceforth, a model with the structure given by (11) will be termed a "linear-wavelet" network (see Fig. 1 for a graphical presentation).

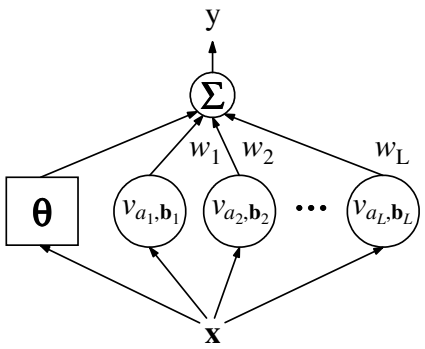

Fig. 1. Structure of a linear-wavelet network. The circles represent wavelet functions with different dilation and translation parameters. The square represents the linear part of the model, which consists of the inner product of the input vector $\mathbf{x}$ with the vector of parameters $\boldsymbol{\theta}$.

It should be noticed that in problems involving the approximation of a function over a compact subset $\mathcal{X} \subset \mathbb{R}^d$, the linear-wavelet network has the same approximation capabilities of a standard wavelet network. In fact, linear functions are square-integrable over any compact subset of $\mathbb{R}^d$ and thus, according to the discussion in the previous subsection, they can be replaced by a linear combination of wavelets. However, many wavelets may be required to approximate a linear function, specifically in high-dimensional domains, because linear functions are not localized in space. Hence, if a function to be approximated is only mildly nonlinear, the use of a linear term may replace a large number of wavelets, thus leading to a more parsimonious representation.

3. Building a Linear-Wavelet Network

A major advantage of wavelet networks over other neural architectures is the availability of efficient construction algorithms for defining the network structure, that is, for choosing convenient values for $(m, \mathbf{n})$ in (4). After the structure has been determined, the weights w_j in (1) can be obtained through linear estimation techniques.

In this work, a constructive method similar to that introduced in (Zhang, 1997) is employed to build a linear-wavelet network. It can be described as follows:

Algorithm 1. (Model construction) Suppose that M modelling samples are available in the form of input-output pairs $(\mathbf{x}[k], y[k]), \quad k = 1, \ldots, M$, where $(\mathbf{x}[k])_{d\times 1}$ is a column vector. Then

1) Normalize the input data to fit within the effective support H of the mother wavelet employed. For radial wavelets, H is a hypersphere in $\mathbb{R}^d$ with radius R. For computational simplicity, H is approximated as a hypercube inscribed in the hypersphere with edges parallel to the coordinate axis.

2) Choose $m_{\min}$ and $m_{\max}$, the minimum and maximum scale levels to be employed.

3) For each sample $\mathbf{x}[k]$ in the modelling set, find I_k, the index set of wavelets whose effective supports contain $\mathbf{x}[k]$:

$$I_k = \big\{(m, \mathbf{n}) \text{ s.t. } \mathbf{x}[k] \in H_{m,\mathbf{n}}\,; \quad m_{\min} \le m \le m_{\max}, \mathbf{n} \in \mathbb{Z}^d\big\}, \tag{13}$$

where $H_{m,\mathbf{n}}$ is a hypercube centred at $\mathbf{n}\beta\alpha^m$ with edges $\alpha^m R\sqrt{2}$.

4) Determine the pairs $(m, \mathbf{n})$ which appear in at least two sets I_{k1} and I_{k2}, $k_1 \neq k_2$. These are the wavelets whose effective supports include at least two samples. This step is different from the algorithm described in (Zhang, 1997), which allows for wavelets with effective supports containing only one sample. Such wavelets are not included here because they would introduce oscillations between neighbour modelling points, which might compromise the generalization ability of the model.

5) Let L be the number of wavelets obtained in the previous step. For notational simplicity replace the double index $(m, \mathbf{n})$ by a single index $j = 1, \ldots, L$.

6) Apply the L wavelets to the M modelling samples and gather the results in a matrix form as follows:

$$\mathbf{V} = \begin{bmatrix} v_1(\mathbf{x}[1]) & v_1(\mathbf{x}[2]) & \cdots & v_1(\mathbf{x}[M]) \\ v_2(\mathbf{x}[1]) & v_2(\mathbf{x}[2]) & \cdots & v_2(\mathbf{x}[M]) \\ \vdots & \vdots & \cdots & \vdots \\ v_L(\mathbf{x}[1]) & v_L(\mathbf{x}[2]) & \cdots & v_L(\mathbf{x}[M]) \end{bmatrix}_{L\times M}. \tag{14}$$

7) Gather the input-output modelling data in a matrix form as

$$\begin{aligned} \mathbf{y} &= \big[y[1]\, y[2] \cdots y[M]\big]_{1\times M}, \\ \mathbf{X} &= \big[\mathbf{x}[1]\, \mathbf{x}[2] \cdots \mathbf{x}[M]\big]_{d\times M}, \end{aligned} \tag{15}$$

and use least-squares regression to estimate the row vectors of linear weights $\boldsymbol{\theta} = [\theta_1\, \theta_2 \cdots \theta_d]$ and

wavelet weights $\mathbf{w} = [w_1\, w_2\, \cdots\, w_L]$ as

$$[\hat{\boldsymbol{\theta}}\, \hat{\mathbf{w}}] = \mathbf{y}\boldsymbol{\Phi}^T(\boldsymbol{\Phi}\boldsymbol{\Phi}^T)^{-1}, \tag{16}$$

where

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{X} \\ \mathbf{V} \end{bmatrix}_{(d+L)\times M}. \tag{17}$$

Remark 2.

1. In the last step of the above algorithm, it is assumed that $\boldsymbol{\Phi}\boldsymbol{\Phi}^T$ is non-singular. If necessary, QR decomposition (Lawson and Hanson, 1974) or Principal Component Analysis (Naes and Mevik, 2001) can be used to deal with the numerical ill-conditioning.

2. The standard error of the estimate $[\hat{\boldsymbol{\theta}}\, \hat{\mathbf{w}}]$ is given by the square root of the diagonal elements of matrix $\mathbf{S}$ given by (Draper and Smith, 1981):

$$\mathbf{S} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T\right)^{-1} \left[\frac{(\mathbf{y}-\hat{\mathbf{y}})(\mathbf{y}-\hat{\mathbf{y}})^T}{M-(d+L)}\right], \tag{18}$$

where $\hat{\mathbf{y}} = [\hat{\boldsymbol{\theta}}\, \hat{\mathbf{w}}]\boldsymbol{\Phi}$. Notice that the term in the square brackets in (18) is the square of the standard error of the estimate for y, adjusted for the number of observations (M) and estimated variables ($d+L$) (Ezekiel and Fox, 1959).

3. It is often the case that Steps 1 to 4 of the construction process result in a large number of wavelets. In order to avoid overfitting problems that result from an overparameterization of the model, it is then important to select a reduced subset of wavelets. An algorithm for the selection of wavelet regressors is described in the next subsection.

3.1. Selection of Wavelet Regressors

In this work, the selection of convenient wavelet regressors is done by choosing rows of $\mathbf{V}$ in a stepwise manner, according to their correlation with $\mathbf{y}$ and also with their degree of independence with respect to the rows already selected. This procedure can be described as follows:

Algorithm 2. (Selection of regressors)

S1) Let $\mathbf{v}_j$ be the j-th row of $\mathbf{V}$, that is,

$$\mathbf{v}_j = \big[v_j(\mathbf{x}[1])\; v_j(\mathbf{x}[2])\; \cdots\; v_j(\mathbf{x}[M])\big],\; j = 1, \ldots, L. \tag{19}$$

S2) (Preliminary pruning) Eliminate all vectors $\mathbf{v}_j$ such that

(a) $\|\mathbf{v}_j\| < \kappa \max_{(l=1,\ldots,L)} \|\mathbf{v}_l\|$ for a fixed $0 < \kappa < 1$

or

(b) $\max_{(k=1,\ldots,M)} |v_j(\mathbf{x}[k])| < 10\% \max_{\mathbf{x}\in\mathbb{R}^d} |v_j(\mathbf{x})|$.

S3) Normalize all the remaining vectors to the unit norm.

S4) (Removing the information conveyed by the linear regressors). Replace $\mathbf{y}$ and all vectors $\mathbf{v}_j$ by their projections onto the subspace orthogonal to the linear regressors, that is,

$$\mathbf{v}_j \leftarrow \mathbf{v}_j(\mathbf{I} - \mathbf{P}_{\mathbf{X}}) \quad \text{and} \quad \mathbf{y} \leftarrow \mathbf{y}(\mathbf{I} - \mathbf{P}_{\mathbf{X}}), \tag{20}$$

where $\mathbf{P}_{\mathbf{X}} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}$ and $\mathbf{I}$ is the identity matrix.

S5) (First selection) For each vector $\mathbf{v}_j$, evaluate the correlation index r_j as

$$r_j = \frac{\left|\mathbf{v}_j\mathbf{y}^T\right|}{\|\mathbf{v}_j\|\|\mathbf{y}\|} \tag{21}$$

and index ρ_j defined as

$$\rho_j = r_j\|\mathbf{v}_j\|. \tag{22}$$

Let $\mathbf{h}_1$ be the vector with the largest value for ρ_j. Let also $i = 1$.

S6) (Projections) Replace $\mathbf{y}$ and all vectors $\mathbf{v}_j$ by their projections onto the subspace orthogonal to $\mathbf{h}_i$, that is,

$$\mathbf{v}_j \leftarrow \mathbf{v}_j(\mathbf{I} - \mathbf{P}_{\mathbf{h}_i}) \quad \text{and} \quad \mathbf{y} \leftarrow \mathbf{y}(\mathbf{I} - \mathbf{P}_{\mathbf{h}_i}), \tag{23}$$

where $\mathbf{P}_{\mathbf{h}_i} = \mathbf{h}_i^T(\mathbf{h}_i\mathbf{h}_i^T)^{-1}\mathbf{h}_i$.

S7) (Selection) For each vector $\mathbf{v}_j$, evaluate index ρ_j as in (22). Let $\mathbf{h}_{i+1}$ be the vector with the largest value for ρ_j.

S8) Let $i = i + 1$ and return to Step S6.

Repeating Steps S6 to S8 n times results in vectors $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n$. The wavelet regressors to be included in the model are those which originated such vectors.

Remark 3.

1. In the preliminary pruning Step S2, the criterion (a) aims at removing regressors with a small signal-to-noise ratio (notice that $\|\mathbf{v}_j\|^2$ is the energy of $\mathbf{v}_j$). The criterion (b) removes regressors which would introduce undesirable oscillations between neighbouring samples (see Fig. 2 for a graphical interpretation in the one-dimensional case). However, in tests carried out by the authors, it was found that this second pruning criterion does not need to be applied when the data are not excessively sparse. Since it is difficult to provide a guideline based on a specific sparsity measure, it is advisable to carry out the pruning using only the criterion (a) and, in case the resulting model does not exhibit a good generalization ability, perform a new selection incorporating the criterion (b).

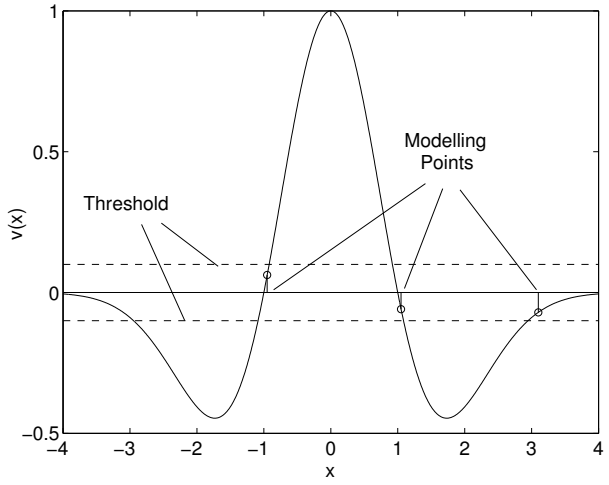

Fig. 2. Logic behind the second pruning criterion (illustrated for a Mexican hat mother wavelet). The dashed lines are at $\pm 10\%$ of the value of the wavelet peak. In this case, for all three modelling samples the wavelet displays values smaller than the $\pm 10\%$ threshold. This wavelet is then discarded.

2. Index ρ_j used in Steps S5 and S7 reflects both the amount of useful information in $\mathbf{v}_j$ (measured by r_j) and its lack of collinearity with the vectors already selected. In fact, if vector $\mathbf{v}_{j1}$ is highly collinear to vector $\mathbf{v}_{j2}$, then the projection of $\mathbf{v}_{j1}$ onto the subspace orthogonal to $\mathbf{v}_{j2}$ will have a small norm. Collinearity avoidance is important to achieve a model with a good generalization ability when least-squares regression is employed (Naes and Mevik, 2001).

3. The entire construction procedure comprising Algorithms 1 (model construction) and 2 (selection of wavelet regressors) is a deterministic process. In fact, it does not require the generation of an initial set of random weights, nor the use of stochastic search algorithms. This is an advantage over the methods employed for training multi-layer perceptron neural networks, such as the classical back-propagation algorithm (Haykin, 1998). Moreover, the construction procedure does not have convergence problems because the number of possible regressors is finite and the regression coefficients are obtained by a matrix inversion operation, rather than an iterative optimization algorithm.

3.2. Choosing the Parameters of the Construction Algorithm

The proposed construction procedure has six parameters: $R, \alpha, \beta, m_{\min}, m_{\max}, \kappa$. Also, a criterion to determine the optimum number n of wavelets should be adopted.

The effective support radius R depends on the mother wavelet employed. If the mother wavelet is generated from the Mexican hat function given by (8), for instance, R can be taken as 4, as can be seen from Fig. 2.

The usual choice for parameters α and β is $\alpha = 2$, $\beta = 1$ as discussed in Section 2.1.

An appropriate choice for parameter $m_{\max}$ (maximum scale level) is $m_{\max} = 0$, because the data are normalized to fit the support of the mother wavelet. It is assumed that interpolation at larger scales is carried out by the linear regressors.

Thus, there remain only $m_{\min}$ and κ to be adjusted according to the application. Parameter $m_{\min}$ depends on the "smoothness" of the function to be approximated: the approximation of sharp peaks and/or edges would require wavelets at small scales and thus a smaller $m_{\min}$ would have to be employed. Parameter κ, which is used in the preliminary pruning procedure, is aimed at reducing the computational workload in the steps that follow. As a rule of thumb, the user should employ the smallest κ that still reduces the computational workload to an acceptable level.

The best number n of wavelets to include in the model can be determined from statistical techniques such as the minimum description length (Rissanen, 1978) and generalized cross-validation (Li, 1986). The goal of such methods is to find a good compromise between the complexity of the model and its ability to fit the training data.

According to (Zhang, 1997), the generalized cross-validation index GCV of a model using s regressors is given by

$$GCV(s) = \frac{1}{M}\sum_{k=1}^{M}\left[y[k] - \hat{f}_s(\mathbf{x}[k])\right]^2 + \frac{2s}{M}\sigma_e^2, \quad (24)$$

where $\hat{f}_s(\mathbf{x}[k])$ is the model prediction for the k-th of M samples used in the identification process and σ_e^2 is the noise variance in the measurement y. The GCV can be used as an estimate of the mean-square error (MSE) that would result from applying the model to samples not used in its development, that is, a measure of the generalization ability.

If the noise variance σ_e^2 is not known, it can be estimated by an iterative procedure, as described in (Zhang, 1997). To do that, start from an initial guess $s^* = s_0$ for the optimum number of regressors and obtain a first estimate of σ_e^2 as the MSE obtained on the modelling set with the resulting linear-wavelet structure. With this estimate of σ_e^2, determine the minimum point of $GCV(s)$ and use it as a new value for s^*. This procedure is repeated until convergence, which usually occurs in a small number of iterations.

3.3. System Identification Applications

In a discrete-time system identification framework, the linear-wavelet network can be used to approximate the functional mapping between the present output of a dynamic system and the information available up to the previous sampling instant. Thus, if $u[k] \in \mathbb{R}$ and $y[k] \in \mathbb{R}$ denote respectively the input and output of the system at the k-th sampling instant, the linear-wavelet network would be used to implement a nonlinear ARX (autoregressive with exogenous input) model of the form

$$y[k] = f\big(\mathbf{x}[k]\big) + e[k], \tag{25}$$

where $e[k]$ is a modelling residual, and

$$\mathbf{x}[k] = \big[\, y[k-1]\, y[k-2] \cdots y[k-n_a] \\ u[k-\delta-1]\, u[k-\delta-2] \cdots u[k-\delta-n_b] \,\big]^T \tag{26}$$

for fixed values of $n_a, n_b > 0$ and $\delta \geq 0$. Notice that the dimension of the input $\mathbf{x}[k]$ to the model is $d = n_a + n_b$.

In this case, it is assumed that f is square-integrable in $\mathbb{R}^d$ or at least in a compact subset $\mathcal{X} \subset \mathbb{R}^d$ where the approximation is to be carried out.

It is also assumed that a set of M modelling samples $\{(\mathbf{x}[k], y[k]), k = 1, \ldots, M\}$ is available for the construction of the linear-wavelet network. The modelling samples must be representative of the functional mapping to be approximated.

The design of an appropriate excitation sequence u to satisfy such a requirement may not be straightforward and is actually a matter of research in the field of system identification (Ljung, 1999). In fact, since the function f is unknown from the start, one does not know *a priori* in which regions of $\mathbb{R}^d$ the approximation of f requires more modelling samples. At this point the designer's experience, or a first-principles engineering analysis, would be of relevance to choose an excitation protocol that would drive the system across the modes of operation that should be captured by the model. In this process, time, cost and safety limitations related to the physical operation of the system should also be taken into account.

Finally, it is worth noticing that the dynamics of the system introduce correlations between the components of the input vector $\mathbf{x}[k]$, thus preventing large variations between those components. This is a fundamental limitation of system identification methods. However, that should not affect the utility of the resulting model, as long as it is used to predict the behaviour of the system along the trajectories similar to those used to generate the modelling samples. That is, the model will not be applied to regions of the $\mathbb{R}^d$ space where the relations between the components of $\mathbf{x}[k]$ are much different from those found in the modelling trajectories of the system.

4. Numerical Example

Consider a fermentation process described by the following Monod model (Aborhey and Williamson, 1978; D'Ans *et al.*, 1972):

$$\frac{\mathrm{d}C}{\mathrm{d}t} = g\frac{CS}{S+p} - Cu, \tag{27}$$

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -qg\frac{CS}{S+p} + (S_{in} - S)u, \tag{28}$$

where C is the microbial concentration, S stands for the substrate concentration (process output), u denotes the dilution rate (process input), g signifies the maximum growth rate, p is the saturation parameter, q means the yield factor, and S_{in} is the inlet substrate concentration.

Suppose that S is observed at discrete time instants:

$$y[k] = S(kT_s) + \varepsilon[k], \quad k \geq 1, \tag{29}$$

where T_s is the sampling period and $\varepsilon[k]$ is the measurement noise.

The values for the model constants were taken from (Zhang, 1997) as

$$g = 0.55, \quad p = 0.15, \quad q = 2, \quad S_{in} = 0.8. \tag{30}$$

The system was simulated in closed loop, as in (Zhang, 1997), with input u being provided by a PI controller with proportional gain $K_p = 0.5$ and integral gain $K_i = 0.05$. The set point for S was changed between three values: 0.2, 0.4 and 0.6. The measurement noise was simulated using a zero-mean white Gaussian noise process with a standard deviation of 0.005. The sampling period adopted was $T_s = 1.0$ time unit. The resulting input ($u[k]$) and output ($y[k]$) signals can be seen in Fig. 3.

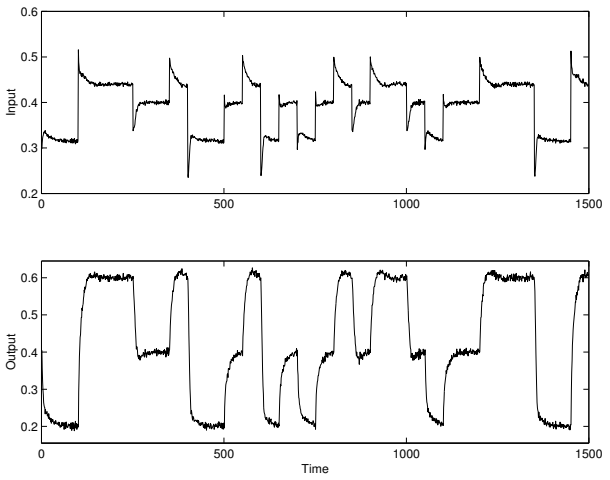

Fig. 3. Input and output data from the simulation.

The first 750 samples were employed for modelling, and the remaining data for validation. The means of the input and output signals were removed during the identification procedures.

For the purpose of illustration, assume that it is desired to obtain a model of the form

$$y[k] = f\big(y[k-1], y[k-2], u[k-1]\big) + e[k] \qquad (31)$$

as proposed in (Zhang, 1997), where f is a nonlinear function to be estimated from the input-output data and $e[k]$ is the modelling residual. Notice that the input to the model is $\mathbf{x}[k] = [y[k-1]\ y[k-2]\ u[k-1]]^T$, so $d = 3$. Each element of $\mathbf{x}[k]$ is associated with a linear regressor, as in (12).

The procedure described in Section 3 was employed to estimate f with a linear-wavelet structure. The mother wavelet employed was generated from the Mexican hat function in (8).

Parameter $m_{\min}$ of Algorithm 1 (construction) was adopted as $m_{\min} = -2$. Using $m_{\min} = -1, -3$ or smaller led to worse approximation results. Steps 1 and 3 resulted in 1907 wavelets, a number which was reduced to 1042 by Step 4. Algorithm 2 (selection) was subsequently applied with $\kappa = 10^{-3}$. After carrying out pruning (Step S2) according to criterion (a), 703 wavelets remained. In this application, it was found that better results were obtained if pruning criterion (b) was not used.

For comparison, a similar construction process was carried out to build a wavelet network. In this case, Step S4 in the selection algorithm was skipped and the vector of wavelet weights $\mathbf{w}$ was obtained by least-squares regression from the vector of output values $\mathbf{y}$ and the matrix of selected wavelet regressors $\mathbf{V}$.

Figure 4 compares the linear-wavelet and the wavelet network models in terms of the mean-square-error of prediction MSE defined for either the modelling or the validation set as

$$MSE(s) = \frac{1}{M}\sum_{k=1}^{M}\Big[y[k] - \hat{f}_s(y[k-1], y[k-2], u[k-1])\Big]^2, \qquad (32)$$

where M is the number of data samples involved and $\hat{f}_s$ is an estimate of f that was generated using s regressors. For the wavelet network, each wavelet corresponds to a regressor (a column of matrix $\mathbf{V}$ in (14)). In the case of the linear-wavelet network, the first three regressors are related to the linear part.

Figure 4 reveals that, for a given number of regressors (which indicate the complexity of the model), the linear-wavelet network yields a smaller mean-square error than the wavelet network both in the modelling and validation sets. Conversely, it can be stated that, for a given degree of the approximation accuracy, the linear-wavelet network is more parsimonious than the wavelet network.

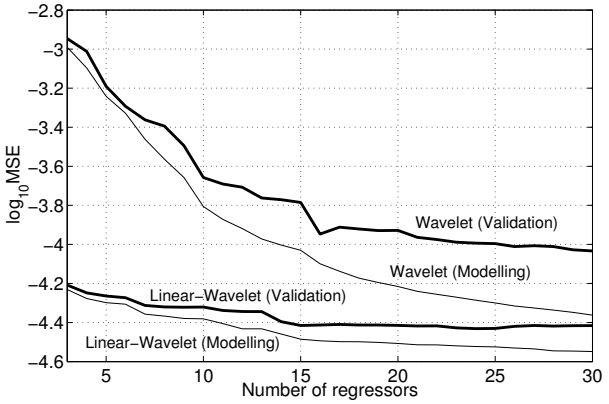

Fig. 4. Mean-square-error (logarithmic scale) for the linear-wavelet and wavelet models. The modelling and validation results are represented by thin and thick lines, respectively.

5. Experimental Example

The plant used in this case study is illustrated in Fig. 5. It consists of a pressure vessel containing air and water. The air pressure is measured at the top of the vessel by means of a pressure transducer. A hydraulic pump is used to create a water flow that enters the vessel through an inlet pipe and so decreases the air volume, thus increasing its pressure. For a given pump rotation speed the system reaches an equilibrium point where no extra water enters the vessel. Furthermore, the direction of flow can be reversed so that the level decreases and so does the air pressure. The input signal, with a range of 0–10 V, is the voltage applied to the power amplifier that drives the DC motor which operates the hydraulic pump. The signals are sent and acquired by a supervisory PC via a Profibus network. The pressure signal ranges between 0 and 100 mBar. The sampling time used was 0.165 s.

Assume that the plant is to be modelled by the structure defined in (25) and (26). Figure 6 presents the input u and output y signals which were used to build and validate the model. As in the previous section, the means of the input and output signals were removed during the identification procedures.

To choose parameters $n_a,\ n_b,\ \delta$ of the model input (cf. Eqn. (26)), a linear identification was initially carried out. Different linear models were identified with $n_a,\ n_b$ varying between 1 and 10, and the delay δ varying between 0 and 9. Figure 7 presents the unexplained variance

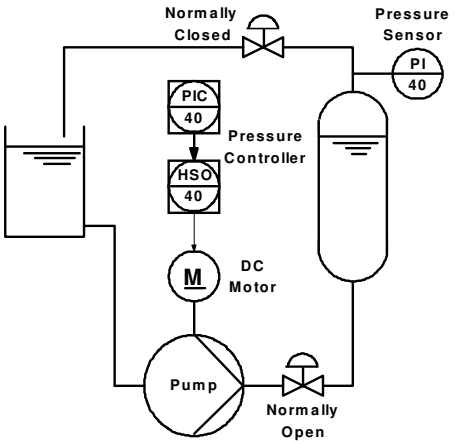

Fig. 5. Schematic diagram of the pilot plant.

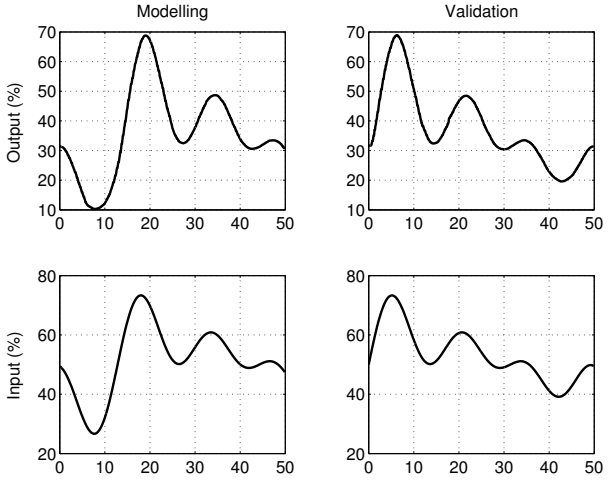

Fig. 6. Input-output data used for modelling and validation. The quantities are shown as percentages of their maximum value (100 mBar for the output and 10V for the input). The horizontal axis is the time in seconds.

in the modelling data as a function of the number of parameters n_a+n_b in the model (in each case, the best result obtained with models of a given complexity are shown). The choice indicated by the arrow, which corresponds to $n_a = 5,\ n_b = 1,\ \delta = 0$, was made to balance the accuracy and complexity of the model. In fact, for a number of parameters greater than 6, the improvements in the model accuracy are minor.

It should be pointed out that choosing the input structure (the elements of $\mathbf{x}[k]$ in (26)) to the linear-wavelet network on the basis of a linear formulation, as described above, is a heuristic procedure. In fact, the best choice of inputs to a nonlinear model does not necessarily correspond to the best inputs to a linear model. However, optimizing the inputs to the full model would involve building and testing different linear-wavelet networks (including the construction and the regressor selection phases) for each combination of $n_a,\ n_b,\ \delta$, which might be impractical.

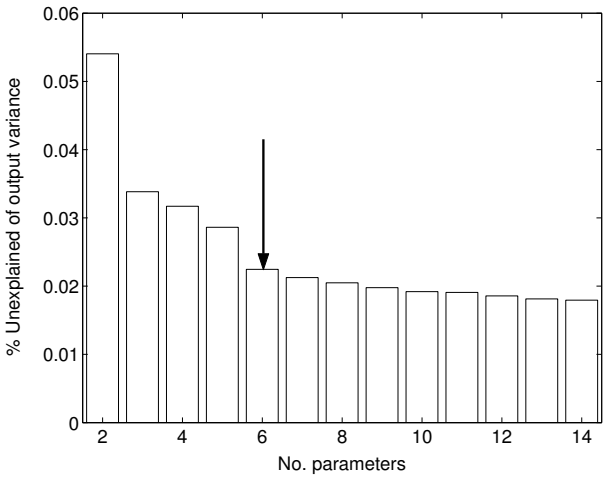

Fig. 7. Selecting the model order.

After the inputs were selected, the procedure described in Section 3 was employed to build the linear-wavelet network. The mother wavelet employed was obtained from the Mexican hat function and the parameters adopted for Algorithms 1 (construction) and 2 (selection) were the same as in the previous example. After the preliminary pruning (Step S2 of the selection algorithm), 241 wavelets remained. In this case, the use of both the pruning critera (a) and (b) was found to be necessary in order to obtain the model with a good generalization ability.

Figure 8 presents the generalized cross-validation index (see Section 3.2) for the linear-wavelet network. The arrows in the graph indicate two inflection points that could be used as a criterion to select the best number of wavelet regressors. For the sake of model parsimony, the point (a), which corresponds to 53 wavelets, was selected.

The validation phase was carried out by using the models to predict the process output in a recursive manner, that is, $\hat{y}[k] = \hat{f}(\hat{y}[k-1], \hat{y}[k-2], \ldots, \hat{y}[k-5], u[k-1])$, starting from the initial conditions $\hat{y}[k] = y[k],\ k = 1, \ldots, 5$.

The results for the linear and linear-wavelet models can be seen in Fig. 9. A comparison between the upper and lower graphs reveals that the use of wavelets considerably improved the prediction ability of the model.

Figure 10(a) presents the coefficients $[\hat{\boldsymbol{\theta}}_{1\times 6}\ \hat{\mathbf{w}}_{1\times 53}]$ (in modulus) and their respective standard errors. It is worth noting that, even though the number of the estimated coefficients is considerably large, the least-squares procedure was not ill-conditioned, since most coefficients

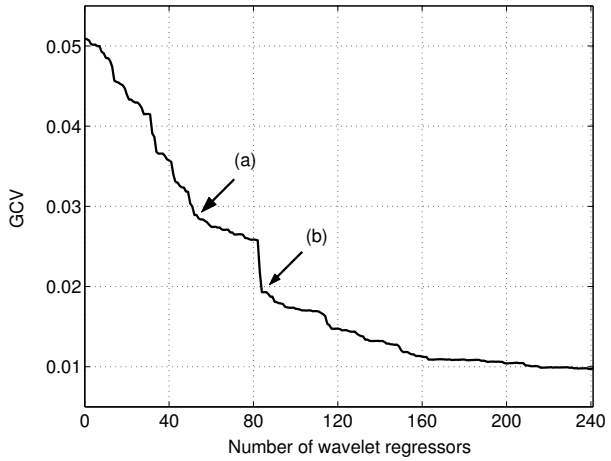

Fig. 8. Generalized cross validation of the linear-wavelet network. The arrows indicate two inflection points.

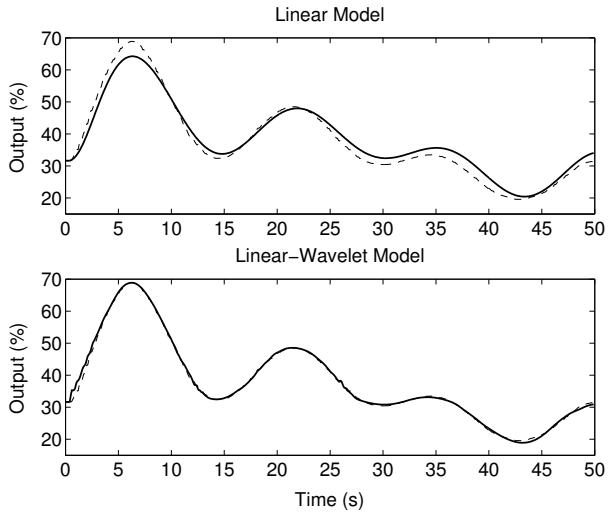

Fig. 9. Validation results. The solid line is the model prediction and the dashed line is the measured output.

are at least twice as large as their standard errors (see Fig. 10(b)). It could be argued that this is a result of the mechanism for collinearity avoidance employed in the selection of wavelet regressors (see Remark 3.2). However, it is possible that a more parsimonious model could be obtained if the wavelets with a small coefficient/standard error ratio in Fig. 10b were eliminated and a new regression performed with the remaining regressors.

6. Conclusion

This paper proposed the combination of linear and wavelet terms in a regression structure for nonlinear function approximation. A deterministic algorithm for constructing a linear-wavelet network from a given set of input-output

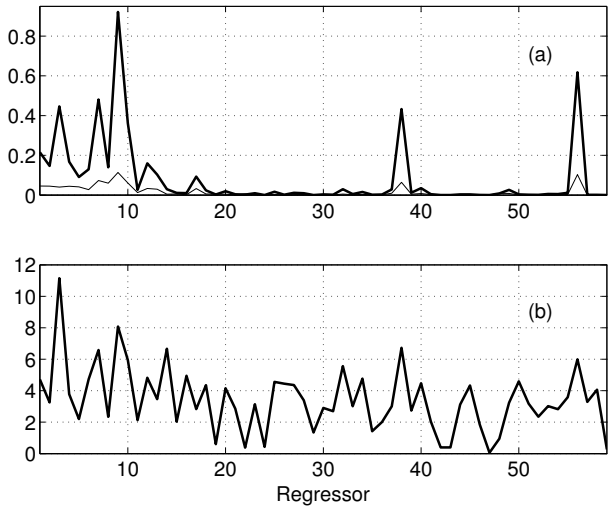

Fig. 10. (a) Coefficients of the linear-wavelet network in absolute values (thick line) and respective standard errors (thin line). The coefficients are plotted in the order the regressors were selected. The first six coefficients correspond to the linear regressors. (b) Absolute value of the coefficients divided by their standard errors.

data was also presented. For illustration, the proposed technique was applied in a dynamic system identification framework.

Results for the identification of a simulated fermentation process revealed that the introduction of the linear term in the wavelet network leads to a more parsimonious model for a given accuracy level. Moreover, results involving experimental data from a real pressure plant revealed that the use of wavelets can indeed improve the prediction ability of a linear model. Those findings corroborate the initial hypothesis that linear regressors might be appropriate complements to wavelets and vice-versa.

A possibility not studied in the present work is the use of the standard error of the model coefficient estimates for the selection of wavelet regressors. This could be done in a backward stepwise procedure, starting with a large wavelet network and pruning those wavelets whose coefficients were not significantly larger than zero, for a given confidence level. This procedure would implicitly take into account collinearity problems, since the standard error tends to increase with collinearity.

Work is being carried out on the use of linear-wavelet networks for predictive control. At each step, the linear term will be employed to generate an initial solution for the sequence of control movements. This solution will then be used as the starting point for an optimization algorithm that takes the whole model into account. Such warm initialization of the optimizer may potentially save computation time, allowing a better solution to be obtained within a fixed time frame. It is worth noting that, in this

application, it would be of interest to assign confidence limits for the model predictions (which could be derived from the standard errors of the estimated coefficients), in order to address robustness issues.

Acknowledgments

The first author acknowledges the support of FAPESP (post-doctorate grant 00/09390–6) and CNPq (PRONEX grant 015/98 and research fellowship). Research collaboration between the University of Reading and ISEL is kindly funded by the British Council under the Treaty of Windsor Programme.

References

Aborhey S. and Williamson D. (1978): *State and parameter estimation of microbial growth processes.* — Automatica, Vol. 14, No. 5, pp. 493–498.

Benveniste A., Juditsky A., Delyon B., Zhang Q. and Glorennec P.Y. (1994): *Wavelets in identification.* — Proc. 10th IFAC Symp. *Syst. Identification*, Copenhagen, pp. 27–48.

Cannon M. and Slotine J.-J.E. (1995): *Space-frequency localized basis function networks for nonlinear system estimation and control*. — Neurocomput., Vol. 9, No. 3, pp. 293–342.

D'Ans G., Gottlieb D. and Kokotovic P. (1972): *Optimal control of bacterial growth.* — Automatica, Vol. 8, No. 6, pp. 729–736.

Daubechies I. (1992): *Ten Lectures on Wavelets*. — Philadelphia: SIAM.

Draper N.R. and Smith H. (1981): *Applied Regression Analysis, 2nd Ed.* — New York: Wiley.

Ezekiel M. and Fox K.A. (1959): *Methods of Correlation and Regression Analysis, 3rd Ed.* — New York: Wiley.

Galvão R.K.H. and Becerra V.M. (2002): *Linear-wavelet models applied to the identification of a two-link manipulator*. — Proc. 21st IASTED Int. Conf. *Modelling, Identification and Control*, Innsbruck, pp. 479–484.

Galvão R.K.H., Yoneyama T. and Rabello T.N. (1999): *Signal representation by adaptive biased wavelet expansions*. — Digital Signal Process., Vol. 9, No. 4, pp. 225–240.

Haykin S.S. (1998): *Neural Networks: A Comprehensive Foundation*. — Upper Saddle River: Prentice-Hall.

Jang J.-S. R. and Sun C.-T. (1995): *Neuro-fuzzy modelling and control*. — Proc. IEEE, Vol. 83, No. 3, pp. 378–406.

Kan K.-C. and Wong K.-W. (1998): *Self-construction algorithm for synthesis of wavelet networks*. — Electronic Lett., Vol. 34, No. 20, pp. 1953–1955.

Lawson C.L. and Hanson R.J. (1974): *Solving Least Squares Problems*. — Englewood Cliffs: Prentice-Hall.

Li K.C. (1986): *Asymptotic optimality of c_l and generalized cross-validation in ridge regression and application to the spline smoothing*. — Ann. Statist., Vol. 14, No. 3, pp. 1101–1112.

Liu G.P., Billings S.A. and Kadirkamanathan V. (2000): *Nonlinear system identification using wavelet networks*. — Int. J. Syst. Sci., Vol. 31, No. 12, pp. 1531–1541.

Ljung L. (1999): *System Identification: Theory for the User*. — Upper Saddle River: Prentice-Hall.

Naes T. and Mevik B.H. (2001): *Understanding the collinearity problem in regression and discriminant analysis*. — J. Chemometr., Vol. 15, No. 4, pp. 413–426.

Naradaya E. (1964): *On estimating regression*. — Theory Prob. Applicns., Vol. 9, pp. 141–142.

Narendra K.S. and Parthasarathy K. (1990): *Identification and control of dynamical systems using neural networks*. — IEEE Trans. Neural Netw., Vol. 1, No. 1, pp. 4–27.

Poggio T. and Girosi F. (1990): *Networks for approximation and learning*. — Proc. IEEE, Vol. 78, No. 9, pp. 1481–1497.

Rissanen J. (1978): *Modeling by shortest data description.* — Automatica, Vol. 14, No. 5, pp. 465–471.

Rugh W.J. (1981): *Nonlinear Systems Theory. The Volterra/Wiener Approach*. — Baltimore: Johns Hopkins University Press.

Schumaker L.L. (1981): *Spline Functions: Basic Theory*. — Chichester: Wiley.

Souza Jr. C., Hemerly E.M. and Galvão R.K.H. (2002): *Adaptive control for mobile robot using wavelet network*. — IEEE Trans. Syst. Man Cybern., Part B, Vol. 32, No. 4, pp. 493–504.

Takagi T. and Sugeno M. (1985): *Fuzzy identification of systems and its applications to modelling and control*. — IEEE Trans. Syst. Man Cybern., Vol. 15, No. 1, pp. 116-132.

Watson G.S. (1964): *Smooth regression analysis*. — Sankhya, Ser. A, Vol. 26, No. 4, pp. 359–372.

Zhang J., Walter G.G., Miao Y. and Lee W.N.W. (1995): *Wavelet neural networks for function learning*. — IEEE Trans. Signal Process., Vol. 43, No. 6, pp. 1485–1496.

Zhang Q. (1997): *Using wavelet network in nonparametric estimation*. — IEEE Trans. Neural Netw., Vol. 8, No. 2, pp. 227–236.

Zhang Q. and Benveniste A. (1992): *Wavelet networks*. — IEEE Trans. Neural Netw., Vol. 3, No. 6, pp. 889–898.

Appendix – Hyperspherical Coordinates

The hyperspherical coordinates $(r, \gamma_1, \gamma_2, \ldots, \gamma_{d-1})$ are defined as

$$x_1 = r\cos\gamma_1, \tag{A1}$$

$$x_k = r\left(\prod_{j=1}^{k-1}\sin\gamma_j\right)\cos\gamma_k, \quad k = 2, \ldots, d-1, \tag{A2}$$

$$x_d = r \prod_{j=1}^{d-1} \sin\gamma_j, \tag{A3}$$

where $r > 0$, $0 \le \gamma_k \le \pi$, $k = 1, 2, \ldots, d-2$ and $-\pi \le \gamma_{d-1} \le \pi$. A volume element is expressed on these variables as $J_d \, \mathrm{d}r \, \mathrm{d}\gamma_1 \, \mathrm{d}\gamma_2 \cdots \mathrm{d}\gamma_{d-1}$, with J_d defined as

$$J_d = \left| \frac{\partial (x_1, x_2, x_3, \ldots, x_d)}{\partial (r, \gamma_1, \gamma_2, \ldots, \gamma_{d-1})} \right|$$

$$= \begin{vmatrix} \frac{\partial x_1}{\partial r} & \frac{\partial x_2}{\partial r} & \cdots & \frac{\partial x_{d-2}}{\partial r} & \frac{\partial x_{d-1}}{\partial r} & \frac{\partial x_d}{\partial r} \\ \frac{\partial x_1}{\partial \gamma_1} & \frac{\partial x_2}{\partial \gamma_1} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_1} & \frac{\partial x_{d-1}}{\partial \gamma_1} & \frac{\partial x_d}{\partial \gamma_1} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ \frac{\partial x_1}{\partial \gamma_{d-2}} & \frac{\partial x_2}{\partial \gamma_{d-2}} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_{d-2}} & \frac{\partial x_{d-1}}{\partial \gamma_{d-2}} & \frac{\partial x_d}{\partial \gamma_{d-2}} \\ \frac{\partial x_1}{\partial \gamma_{d-1}} & \frac{\partial x_2}{\partial \gamma_{d-1}} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_{d-1}} & \frac{\partial x_{d-1}}{\partial \gamma_{d-1}} & \frac{\partial x_d}{\partial \gamma_{d-1}} \end{vmatrix}$$

$$= \begin{vmatrix} \frac{\partial x_1}{\partial r} & \frac{\partial x_2}{\partial r} & \cdots & \frac{\partial x_{d-2}}{\partial r} & \frac{\partial x_{d-1}}{\partial r} & \frac{\partial x_d}{\partial r} \\ \frac{\partial x_1}{\partial \gamma_1} & \frac{\partial x_2}{\partial \gamma_1} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_1} & \frac{\partial x_{d-1}}{\partial \gamma_1} & \frac{\partial x_d}{\partial \gamma_1} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ \frac{\partial x_1}{\partial \gamma_{d-2}} & \frac{\partial x_2}{\partial \gamma_{d-2}} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_{d-2}} & \frac{\partial x_{d-1}}{\partial \gamma_{d-2}} & \frac{\partial x_d}{\partial \gamma_{d-2}} \\ 0 & 0 & \cdots & 0 & -A_d \sin\gamma_{d-1} & A_d \cos\gamma_{d-1} \end{vmatrix}, \tag{A4}$$

where

$$A_d = r \prod_{j=1}^{d-2} \sin\gamma_j. \tag{A5}$$

Define a new variable $\bar{x}_{d-1}$ as $\bar{x}_{d-1} = r \prod_{j=1}^{d-2} \sin\gamma_j$ so that $x_{d-1} = (\cos\gamma_{d-1})\, \bar{x}_{d-1}$ and $x_d = (\sin\gamma_{d-1})\, \bar{x}_{d-1}$. It follows that

$$J_d = \begin{vmatrix} \frac{\partial x_1}{\partial r} & \frac{\partial x_2}{\partial r} & \cdots & \frac{\partial x_{d-2}}{\partial r} & \cos\gamma_{d-1} \frac{\partial \bar{x}_{d-1}}{\partial r} & \sin\gamma_{d-1} \frac{\partial \bar{x}_{d-1}}{\partial r} \\ \frac{\partial x_1}{\partial \gamma_1} & \frac{\partial x_2}{\partial \gamma_1} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_1} & \cos\gamma_{d-1} \frac{\partial \bar{x}_{d-1}}{\partial \gamma_1} & \sin\gamma_{d-1} \frac{\partial \bar{x}_{d-1}}{\partial \gamma_1} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ \frac{\partial x_1}{\partial \gamma_{d-2}} & \frac{\partial x_2}{\partial \gamma_{d-2}} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_{d-2}} & \cos\gamma_{d-1} \frac{\partial \bar{x}_{d-1}}{\partial \gamma_{d-2}} & \sin\gamma_{d-1} \frac{\partial \bar{x}_{d-1}}{\partial \gamma_{d-2}} \\ 0 & 0 & \cdots & 0 & -A_d \sin\gamma_{d-1} & A_d \cos\gamma_{d-1} \end{vmatrix}. \tag{A6}$$

Applying the minors rule to the last line of the above determinant, it can be evaluated as

$$J_d = (-1)^{2d-1}(-A_d \sin\gamma_{d-1}) \times \begin{vmatrix} \frac{\partial x_1}{\partial r} & \frac{\partial x_2}{\partial r} & \cdots & \frac{\partial x_{d-2}}{\partial r} & \sin\gamma_{d-1} \frac{\partial \bar{x}_{d-1}}{\partial r} \\ \frac{\partial x_1}{\partial \gamma_1} & \frac{\partial x_2}{\partial \gamma_1} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_1} & \sin\gamma_{d-1} \frac{\partial \bar{x}_{d-1}}{\partial \gamma_1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{\partial x_1}{\partial \gamma_{d-2}} & \frac{\partial x_2}{\partial \gamma_{d-2}} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_{d-2}} & \sin\gamma_{d-1} \frac{\partial \bar{x}_{d-1}}{\partial \gamma_{d-2}} \end{vmatrix}$$

$$+ (-1)^{2d}(A_d \cos\gamma_{d-1}) \times \begin{vmatrix} \frac{\partial x_1}{\partial r} & \frac{\partial x_2}{\partial r} & \cdots & \frac{\partial x_{d-2}}{\partial r} & \cos\gamma_{d-1} \frac{\partial \bar{x}_{d-1}}{\partial r} \\ \frac{\partial x_1}{\partial \gamma_1} & \frac{\partial x_2}{\partial \gamma_1} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_1} & \cos\gamma_{d-1} \frac{\partial \bar{x}_{d-1}}{\partial \gamma_1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{\partial x_1}{\partial \gamma_{d-2}} & \frac{\partial x_2}{\partial \gamma_{d-2}} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_{d-2}} & \cos\gamma_{d-1} \frac{\partial \bar{x}_{d-1}}{\partial \gamma_{d-2}} \end{vmatrix}$$

$$= A_d \left(\sin^2\gamma_{d-1} + \cos^2\gamma_{d-1} \right) \times \begin{vmatrix} \frac{\partial x_1}{\partial r} & \frac{\partial x_2}{\partial r} & \cdots & \frac{\partial x_{d-2}}{\partial r} & \frac{\partial \bar{x}_{d-1}}{\partial r} \\ \frac{\partial x_1}{\partial \gamma_1} & \frac{\partial x_2}{\partial \gamma_1} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_1} & \frac{\partial \bar{x}_{d-1}}{\partial \gamma_1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{\partial x_1}{\partial \gamma_{d-2}} & \frac{\partial x_2}{\partial \gamma_{d-2}} & \cdots & \frac{\partial x_{d-2}}{\partial \gamma_{d-2}} & \frac{\partial \bar{x}_{d-1}}{\partial \gamma_{d-2}} \end{vmatrix}$$

$$= A_d J_{d-1}, \tag{A7}$$

since $x_1, x_2, \ldots \bar{x}_{d-1}$ are represented in hyperspherical coordinates by $r, \gamma_1, \ldots \gamma_{d-2}$. Then, it can be seen that $J_d = A_d A_{d-1} \cdots A_3 J_2$, where $J_2 = r$ (because an element of area in polar coordinates is given by $r \, \mathrm{d}r \, \mathrm{d}\gamma_1$). Finally, by applying the expression for A_d given by (A5), it follows that

$$J_d = r^{d-1} \prod_{j=1}^{d-2} (\sin\gamma_j)^{d-j-1}.$$

Received: 3 February 2003
Revised: 14 January 2004