Journal of Sound and Vibration (2001) **248**(2), 247–265
doi:10.1006/jsvi.2001.3737, available online at http://www.idealibrary.com on IDEAL®

NEURAL IDENTIFICATION OF NON-LINEAR DYNAMIC STRUCTURES

R. Le Riche

Lab. de Mécanique de Rouen, INSA/CNRS UMR 6138, *Av. de l'Université*, 76800 *Saint Etienne du Rouvray, France. E-mail: Rodolphe.Leriche@insa-rouen.fr*

D. Gualandris, J. J. Thomas

PSA, Dir. de la Recherche et de l'Innovation Automobile, ch. de la Malmaison, 91570 *Bièvres, France*

AND

F. Hemez

Eng. Sciences and Applications, Los Alamos National Lab., Los Alamos, NM 87545, *USA*

(*Received* 27 *July* 2000, *and in final form* 19 *February* 2001)

Neural networks are applied to the identification of non-linear structural dynamic systems. Two complementary problems inspired from customer surveys are successively considered. Each of them calls for a different neural approach. First, the mass of the system is identified based on acceleration recordings. Statistical experiments are carried out to simultaneously characterize optimal pre-processing of the accelerations and optimal neural network models. It is found that key features for mass identification are the fourth statistical moment and the normalized power spectral density of the acceleration. Second, two architectures of recurrent neural networks, an autoregressive and a state-space model, are derived and tested for dynamic simulations, showing higher robustness of the autoregressive form. Discussion is first based on a non-linear two-degree-of-freedom problem. Neural identification is then used to calculate the load from seven acceleration measurements on a car. Eighty three per cent of network estimations show below 5% error.

1. INTRODUCTION

Economical constraints call for a faster cycle of development and use of structural models. They also call for safeguards against errors occurring in material characterization, boundary conditions description and manufacturing. System identification and neural network techniques are more directly geared towards respecting these constraints than traditional modelling for two reasons: they involve simpler representation than the finite element method and directly account for errors when the model is created.

Linear system identification has been thoroughly described by Ljung [1]. Numerous applications of system identification to control, signal processing and pattern recognition exist (see references [1, 2]). In structural dynamics, linear system identification is a valuable approach for random vibration problems [3].

Neural networks have received massive attention in the last two decades. Complementary review papers have been written (references [4, 5]). Reference [6] gives an

0022-460X/01/470247 + 19 $35.00/0

excellent introduction to neural networks in a probabilistic context. Early use of feed-forward and Hopfield neural networks for structural analysis and design are found in reference [7]. Many applications of feed-forward neural networks to material engineering are presented in reference [8]. In reference [9], the identification of flutter and boundary conditions of a cantilever beam is performed using feed-forward networks.

Since the work of Chen *et al.* [10], neural networks have provided a framework for non-linear system identification. Their contribution to identification and control of non-linear dynamic systems has been analyzed in reference [11]. Neuro control of structures has rapidly emerged as a primary application of the technique [12–14].

Typically, during customer surveys in the automobile industry, data on accelerations are gathered. Other information is then inferred from the acceleration measurement. This paper deals with the identification of the vertical dynamics of a vehicle from acceleration measurements using neural networks. Two interwoven problems are addressed. First, the mass of a vehicle driving on an unknown road is identified from the acceleration measurement. Feed-forward neural networks whose variance is controlled by early stopping and cross-validation provide a methodology for extracting features from accelerations in order to best characterize the mass of the system. Second, knowing that the mass of the system is estimated, the dynamics of the vehicle is learned by recurrent neural networks. Performances of state-space and autoregressive neural architectures are compared.

The discussion relies mainly on a simple two-degree-of-freedom one-wheel system because it permits the explicit derivation of network structures and the comparison between feature extraction and sensitivity analysis. The last part of the paper provides an application to a real car where the load transmitted from the wheel to the suspension is derived from seven acceleration measurements at other car locations.

2. NEURAL SYSTEM IDENTIFICATION VERSUS STRUCTURAL MODELLING

Modelling in structural vibrations usually results from writing fundamentally principles (Newton's law, equilibrium principle and material constitutive law) as partial differential equations and integrating them over time and space in an adequate manner. Popular modelling and analysis techniques in structural mechanics include the finite element method and the boundary element method. The resulting models potentially represent the entire mechanical structure. Time integration, hence, provides the response at any location of the discretization in terms of acceleration history, stress, pressure, etc. For linear systems, modal analysis consists in characterizing the behavior of the system in terms of resonance. This analysis technique is very popular because modal superposition principles can be applied to approximate the response at any level of accuracy.

Likewise, most system identification techniques developed in the field of structural dynamics are based on the notion of resonant modes and frequencies. Given the system's vibration time history measured at a few discrete locations, these algorithms attempt to reconstruct the system's frequency response functions. These may further be characterized in terms of resonant frequencies, modal masses, modal damping ratios and deformation shapes. Although successful applications are documented in the literature to either identify the dynamics of complex systems, detect the occurrence of structural damage or implement health monitoring platforms [15], the mathematical assumptions behind linearity and time-to-frequency transformation restricts these algorithms to linear regimes, low-frequency dynamics and deterministic systems. A recent review shows that very little literature is currently available in the field of structural dynamics for identifying non-linear, stochastic structural dynamics and correlating test data to numerical predictions [16].

The approach described in this paper, termed system identification, also, sidesteps the integration of fundamental principles to focus on the input/output relationship. Emphasis is shifted from physics to statistics. Therefore, a "statistically correct" model is sought which may not necessarily be "mathematically correct" in the sense that fundamental principles of structural mechanics may not be satisfied. The reason is that inverse models capable of representing arbitrarily complex non-linear systems are generally not available, especially when investigating systems that may feature material non-linearity, localized contact and impact dynamics and transient responses [17]. Another important motivation is the identification of stochastic systems, a field of open research to a great extent in non-linear structural dynamics. Here, the "black-box" linking inputs to outputs is a neural network, that can be seen as a family of parameterized non-linear functions. The parameters are tuned by minimizing a distance between network and real system outputs. Neural network properties regarding universal approximation [18, 19], complexity control, parsimony, and the amount of learning strategies (optimization algorithms, regularization, see reference [6]) developed in the last 15 years make them a particularly attractive representation tool.

Although neural network-based system identification does not include as much physics as traditional modelling does, it offers the following important advantages:

(1) Only experimental measurements of inputs and outputs are necessary to represent the system. Modelling is possible even if no physical model exists.

(2) Because of their universal approximation properties, neural networks can identify non-linear and stochastic systems. Accounting for uncertainty and environmental variability is critical in applications where, for example, assembling tolerances may not have been fully complied with; when material characteristics may vary with temperature and humidity conditions; or when an input to the system (such as the road profile in the following numerical application) is an unknown, stochastic variable.

(3) System identification [1] and neural networks [6] offer a probabilistic framework to the representation. For example, when the input/output relationship is learned by minimizing a least-square distance between the neural network response and the experiments, the neural network learns the average response of the system conditioned on the input (assuming a large number of data and neurons). System identification can therefore be used in cases where the system is not well-defined. If one of its characteristics, such as a boundary condition or a material property, is unknown, the neural network still predicts the response. Such predictions would be difficult to achieve, if not impossible, with models based on the conventional principles of structural mechanics. The type of prediction produced (maximum likelihood, conditional average, etc.) depends mainly on the norm minimized for identifying the neural network. Noise on the data can be accounted for by controlling the network effective complexity.

(4) Once they have proceeded through a "tuning" step, neural networks are numerically very efficient. They can replace complex engineering or physics simulations that would otherwise require several hours of computing time on the most advanced platforms. It may therefore be advantageous for re-analysis, sensitivity analysis, parametric studies or optimization to replace the full-order calculation by a "fast running model" or meta-model formed of neural networks.

Finally, it is emphasized that system identification strategy proceeds differently from its conventional counterparts. Instead of inverting a mathematical model that would represent the forward problem, which is what the over whelming majority of techniques attempt to achieve in structural dynamics, a neural network will infer the inverse model directly from test data without having, first, to model the forward problem. This tends to reduce the ill-conditioning difficulties encountered when operators that represent the discretized, forward problem are numerically inverted. Of course, there is a cost to pay for bypassing the

need of a physical description of the system. It is the training step during which test data must be used for tuning the parameters of the neural network. Depending on the complexity and performance expected, this first step may yield significant computational requirements. Several strategies are discussed in the remainder for training simple networks in the most efficient possible way.

3. MASS IDENTIFICATION

3.1. PRINCIPLE

The final goal of this work is the prediction of loads in a car component knowing accelerations at other locations. The relation between acceleration measurements and loads is conditioned by the car mass. This mass varies with passengers, luggage, gas tank level, etc. A preliminary problem is, therefore, the determination of the car mass from accelerations.

The driver being free to take the car where he pleases, the road profile is not known in a deterministic way. It is assumed that the road can be described by an ergodic stationary random process. It implies that at a given mass, speed, and road curvatures, acceleration measurements are also an ergodic and stationary random process. In that case, a unique mapping exists between acceleration statistics and the mass. The determination of the statistics and associated mapping solves the mass identification problem. A difficulty arises, however, from the stationarity assumption: the car can be assumed to move in a straight line at constant speed only over a short time period. Acceleration statistics are noisy over such finite sampling time. The mass identification problem can finally be stated as follows: knowing accelerations during ΔT s, find the statistical features and associated mapping that predict the car mass in a robust way.

To construct the mapping, it is assumed that N_{exp} experiments have been carried out, yielding a database $\mathscr{D}$ of N_{exp} pairs [accelerations, mass], $[\ddot{\mathbf{x}}_1^{(j)}, dm^{(j)}]$, $j = 1, N_{exp}$. $\ddot{\mathbf{x}}_1$ denotes the vector of accelerations $[\ddot{\mathbf{x}}_1(t_1), \ldots, \ddot{\mathbf{x}}_1(t_M)]^{\mathrm{T}}$, $(t_M - t_1) = \Delta T$. To simplify notations, $\ddot{\mathbf{x}}_1$ will sometimes be written as y. There are too many acceleration measurements for them to be directly used in the mapping. Accelerations are thus pre-processed to produce N_F statistical features, $u_i(\ddot{x}_1) = u_i$, $i = 1, N_F$. The working database $\mathscr{D}$ is finally composed of paris [acceleration statistics, mass], $[\mathbf{u}^{(j)}, dm^{(j)}]$. Finding the inverse mapping from $\ddot{\mathbf{x}}_1$ to dm involves two coupled subproblems:

(1) Determine a subset $u_{\mathscr{I}}$ of features used by the mapping.
(2) Find an associated mapping,

$$\widehat{dm} = f(u_{\mathscr{I}}(\ddot{x}_1), w), \tag{1}$$

where $\hat{}$ means estimate. The function f is modelled using a feed-forward neural network with one hidden layer of sigmoidal units and an output layer of linear units (see references [4–6] for introductory comments on neural networks). Training of the network is done by calculating w that minimizes a least-squares error function between dm and $\widehat{dm}$ over a part $\mathscr{D}_D$ of $\mathscr{D}$, $\mathscr{E}(\mathscr{D}_D, w)$. Minimization is performed by a Levenberg–Marquardt algorithm [20, 21]. Assuming a large database $\mathscr{D}$ and a large number of hidden neurons, f therefore tends to the regressor of dm knowing $u_{\mathscr{I}}$,

$$f(u_{\mathscr{I}}, w) \to E(dm|u_{\mathscr{I}}), \tag{2}$$

where E denotes expectation. A backpropagation algorithm [22] calculates gradients of the least-squares error function.

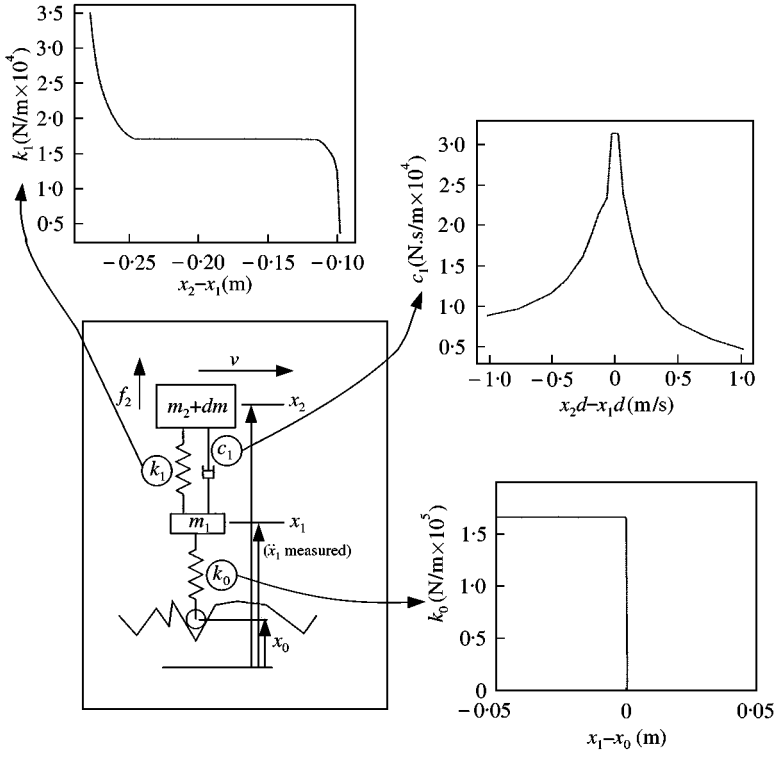

Figure 1. One-wheel system. Behavior non-linearities, $k_0(x_1 - x_0)$, $k_1(x_2 - x_1)$ and $c_1(\dot{x}_2 - \dot{x}_1)$ are plotted. $m_1 = 27\ kg$, $m_2 = 331$ kg. Linear case: $k_0 = 166000$ N/m, $k_1 = 17000$ N/m, $c_1 = 14000$ N/m.

A two-degree-of-freedom linear or non-linear one-wheel system is considered (see Figure 1). Acceleration at the wheel hub, $\ddot{x}_1(t_i)$ is known at discrete times t_i with a time step $\Delta t = 0{\cdot}001$ s and a total recording time $\Delta T = 14{\cdot}336$ s. Other data on the one-wheel system are given in Figure 1. Seven road samples are considered for 11 masses ranging from $dm = -5\%$ of m_2 to $dm = +20\%$ of m_2 (every 2·5% of m_2). There is a total of $N_{exp} = 7 \times 11 = 77$ experiments. Despite its small dimensionality, this problem has difficulties associated with the non-linearities. In addition, the input (the road profile) is a random variable. Therefore, conventional system identification procedures cannot be applied here.

3.2. CANDIDATE FEATURES

Nineteen candidate features are being considered as potential inputs to the network. Candidate features are:

(1) The first four statistical moments,

$$\hat{E}(\ddot{x}_1) = \hat{E}(y) = \frac{1}{M}\sum_{i=1}^{M} y_i, \tag{3}$$

$$\hat{E}((y - E(y))^i) = \frac{1}{M}\sum_{i=1}^{M} (y_i - \hat{E}(y))^i, \quad i = 2, 4. \tag{4}$$

(2) The power spectral density, $\widehat{S_{yy}}(\omega_k)$, calculated at the discrete frequencies ω_k. $\widehat{S_{yy}}(\omega_k)$ is obtained using Welch's algorithm [23] with a window of size 256.

(3) The energy of y in the low- and medium-frequency range,

$$\widehat{S0_y} = \sum_{\omega_i \in [\omega_{min}, \omega_{max}]} \widehat{S_{yy}}(\omega_i); \qquad \omega_{min} = 0 \text{ Hz}, \omega_{max} = 40 \text{ Hz}. \tag{5}$$

(4) The normalized power spectral density,

$$\widehat{N_{yy}}(\omega_k) = \widehat{S_{yy}}(\omega_k)/\widehat{S0_y}. \tag{6}$$

(5) The four spectral moments,

$$\widehat{\mathscr{M}_y^i} = \sum_{\omega_j \in [\omega_{min}, \omega_{max}]} |\omega_j|^i \widehat{S_{yy}}(\omega_j), \quad i = 1, 4. \tag{7}$$

(6) The normalized spectral moments,

$$\widehat{\mathscr{N}_y^i} = \sum_{\omega_j \in [\omega_{min}, \omega_{max}]} |\omega_j|^i \widehat{N_{yy}}(\omega_j), \quad i = 1, 4. \tag{8}$$

(7) The average minima and maxima of acceleration amplitude, $\widehat{\max(y)}$ and $\widehat{\min(y)}$. $\ddot{x}_1(t_i)$ is defined as a maximum (a minimum respectively) if $\ddot{x}_1(t_i) > \ddot{x}_1(t), \forall t \in [t_i - 0{\cdot}1 \text{ s}, t_i + 0{\cdot}1 \text{ s}$, ($<$ respectively).

(8) The coefficients a_i of an nth order autoregressive linear process and the associated prediction error, $\hat{e}$. The a_i's are calculated by least-squares minimization of the distance between $\ddot{x}_1(t)$ and a model,

$$\hat{\ddot{x}}_1(t_k) = a_1 \ddot{x}_1(t_{k-1}) + a_2 \ddot{x}_1(t_{k-2}) + \cdots + a_n \ddot{x}_1(t_{k-n}). \tag{9}$$

The order of the model is taken equal to 8. The prediction error is

$$\hat{e} = \| \hat{\ddot{x}}_1 - \ddot{x}_1 \|. \tag{10}$$

Most of the candidate features, namely $\widehat{S_{yy}}$, $\widehat{S0_y}$, $\widehat{N_{yy}}$, $\widehat{\mathscr{M}_y^i}$ and $\widehat{\mathscr{N}_y^i}$, are variations on the power spectral density. They all attempt to characterize how mass changes affect energy frequency distribution. Again, the difficulty arises from the limited size of road samples which perturbs the features. Ideally, one looks for features that are sensitive to mass variations while being as little sensitive to road sampling as possible. The trade-off is already present in power spectral densities of the linear one-wheel system,

$$S_{yy}(\omega) = |H(\omega)|^2 S_{x_0 x_0}(\omega), \tag{11}$$

where $H(\omega)$ is the frequency response function between x_0 and the acceleration y given in equation (A.2). The finite size of the sample induces variations on $S_{x_0 x_0}(\omega)$, while the mass of the system influences $H(\omega)$. Both effects are mixed in accelerations, i.e., in $S_{yy}(\omega)$. Sensitivity analysis of the linear one-wheel system is carried out in Appendix A. It is seen that, at 4 Hz, the sensitivity of the power spectral density to the mass, $(\partial S_{yy}(\omega)/\partial m_2)$, is maximal.

Visual examination of some of the candidate features illustrates the trade-off between mass and road sensitivities. Power spectral densities $S_{x_0 x_0}$ of seven road samples are shown in Figure 2. Some of the features are sensitive to the mass: the first normalized spectral

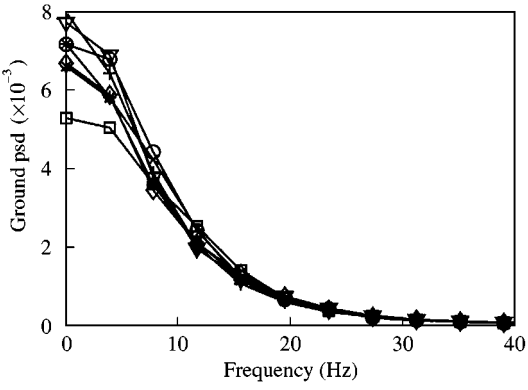

Figure 2. Power spectral density of the road, x_0, for the seven road samples.

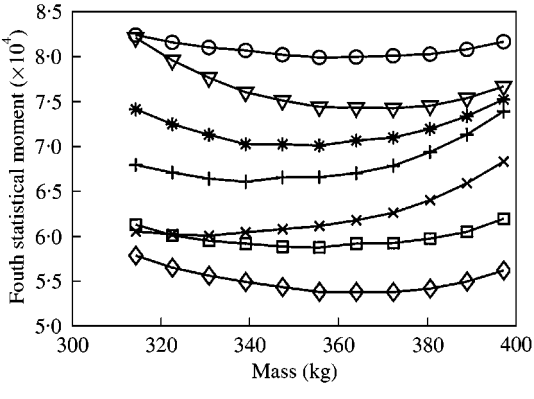

Figure 3. First normalized spectral moment, $\widehat{\mathscr{N}_y^1}$, for varying masses and road samples. Each symbol is associated with a road sample.

moment, $\widehat{\mathscr{N}_y^1}$ (cf. Figure 3), the third statistical moment, $\hat{E}((y-\hat{E}(y))^3)$, the power spectral density at 4 Hz, $\widehat{S_{yy}}$ (4), the normalized power spectral density at 4 Hz, $\widehat{N_{yy}}$ (4), the second, third and fourth spectral moments, $\widehat{\mathscr{M}_y^2}$, $\widehat{\mathscr{M}_y^3}$, $\widehat{\mathscr{M}_y^4}$ and the average amplitude of minima, $\widehat{\min(y)}$. Other candidate features are mainly influenced by road samples: the fourth statistical moment, $\hat{E}((y-\hat{E}(y))^4)$ (cf. Figure 4), the average, $\hat{E}(y)$, the variance, $\hat{E}((y-\hat{E}(y))^2)$ and the cumulated power spectral density $\widehat{S0_y}$. As will be seen later, an important feature for mass identification is $S_{x_0x_0}$ (4), which is unknown since it relates to the road. Spearman's coefficients [24] provide a measure of correlation between ranks where $+1$ indicates perfect positive correlation, -1 perfect negative correlation and 0 no correlation. Spearman's correlation coefficients between mass insensitive features and $S_{x_0x_0}$ (4) are, by decreasing values: 0·77 ($\hat{E}((y-\hat{E}(y))^4)$), 0·63 ($\hat{E}(y)$), 0·56 ($\widehat{S0_y}$) and 0·55 ($\hat{E}((y-\hat{E}(y))^2)$).

Note that for the linear one-wheel system, the mass can directly be read from $\widehat{N_{yy}}$ (4) with 2·5% accuracy. Masses cannot be discerned visually from $\widehat{N_{yy}}$ (4) in the non-linear case, as is shown in Figure 5 for two masses different by 10%. More elaborated feature processing is required in the non-linear case.

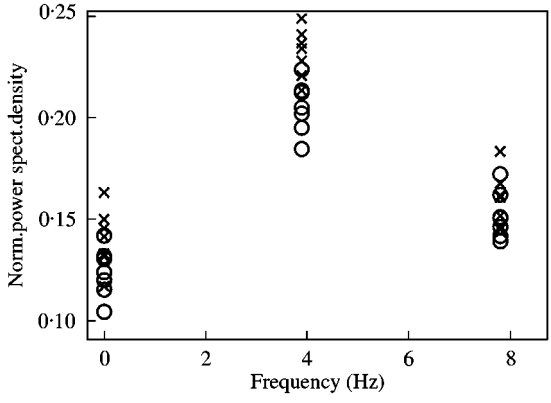

Figure 4. 4th statistical moment of $\ddot{x}_1$ for different mass and road samples.

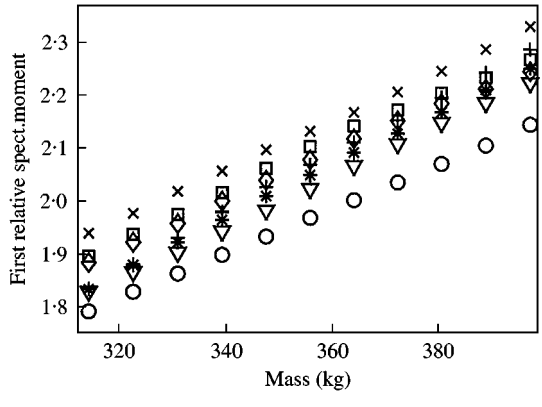

Figure 5. Normalized power spectral density of $\ddot{x}_1$ for seven road samples and two masses: x means $dm = 0$ and o means $dm = +10\%$ of m_2.

3.3. IDENTIFICATION PROTOCOL

A trade-off between mass and road sample sensitivities of the candidate features u_i has been presented. Another trade-off between bias and variance of the mapping f (cf. equation 1) is now introduced and drives the identification protocol (cf. reference [6] for a detailed discussion on bias and variance). Bias is the average error between f and $E(dm|u_{\mathscr{I}})$, the ideal network's response (cf. equation (2)). Variance measures sensitivity of identified mappings f to the choice of databases used in learning, $\mathscr{D}_L$. The average error committed by a mapping f over several choices of learning database, $\mathscr{D}_L$, is the sum of bias and variance. Minimization of bias and variance are two antagonistic goals. For example, bias is reduced by increasing the number of hidden neurons, i.e., increasing the flexibility of the mapping. But this increases variance. For a given network architecture, variance can be decreased by including more experiments in the database. However, considering the cost of testing on real vehicles, it is realistic to assume that a limited number of experiments is available. Special care is then needed in addressing the bias–variance trade-off. Four precautions are taken here.

First of all, neworks with architectures of varying flexibilities are considered: 1 linear hidden neuron, 2 and 4 sigmoidal hidden neurons.

Second, a strategy called early stopping is used to prevent excessive variance. The idea of early stopping is to divide the learning database $\mathscr{D}_L$ into two sets, $\mathscr{D}_D$ and $\mathscr{D}_S$, $\mathscr{D}_L = \mathscr{D}_D \cup \mathscr{D}_S$, $\mathscr{D}_D \cap \mathscr{D}_S = \varnothing$. $\mathscr{D}_D$ defines a direction of change of the network parameters, typically a variation on the gradient of the least-squares error, $\partial \mathscr{E}/\partial w$. Learning stops when the least-squares error on $\mathscr{D}_S$ increases for N_{inc} successive iterations. By adjusting network parameters based on a set of experiments, $\mathscr{D}_D$, and deciding when to stop based on another set of experiments, $\mathscr{D}_S$, it is expected that the noise present in $\mathscr{D}_D$ is not learned, thereby reducing variance.

Third, high variance means that performance might be so much related to the choice of $\mathscr{D}_L$ that no conclusions can be drawn. Let $\mathscr{D}_T$ denote the testing set, $\mathscr{D}_T \cup \mathscr{D}_L = \mathscr{D}$, $\mathscr{D}_T \cap \mathscr{D}_L = \varnothing$. Cross-validation aims at improving error estimates by considering average errors over several choices of training and learning sets, $\mathscr{D}_T$ and $\mathscr{D}_L$ in $\mathscr{D}$. Test error for a particular choice of $\mathscr{D}_T$ is written as $\mathscr{E}(\mathscr{D}_T)$ and the average test error, $\mathscr{E}_T$.

Finally, neural network structures are redundant, so that there will typically be several solutions to learning the least-squares problem. If there are non-linear hidden neurons, local minima may appear. Learning is susceptible of yielding results that are a function of the initial network weight setting. Local minima are accounted for in testing by averaging test results over N_r runs.

The identification protocol is summarized in Figure 6.

3.4. RESULTS

It is obviously not possible to test all 2^{19} combinations of the $N_F = 19$ candidate features. Instead, 33 sets of input features $u_{\mathscr{I}}$ are arbitrarily chosen. For each choice of inputs, three network architectures are tested (one linear, two and four sigmoidal hidden neurons respectively). Results are obtained using a database of 77 experiments, corresponding to seven road samples of 11 masses. Cross-validation averages results over three choices of learning and testing sets. The seven road samples and all associated masses are assigned either to $\mathscr{D}_L$ or to $\mathscr{D}_T$. $\mathscr{D}_L$ contains five road samples, four in $\mathscr{D}_D$ and one in $\mathscr{D}_S$. $\mathscr{D}_T$ has the remaining two samples. Results are summarized in Table 1 for $N_{inc} = 3$ and $N_T = 100$. Each test error in the Table stems from $100 \times 3 \times 3 = 900$ runs. 80% confidence

- For each choice of candidate features $u_{\mathcal{I}}$,
  - For each network architecture,
    - For each choice of $\mathcal{D}_T$, $\mathcal{D}_D$, $\mathcal{D}_S$,

      Learn Nr networks with early stopping,

      Average test errors on $\mathcal{D}_T \longrightarrow \mathcal{E}(\mathcal{D}_T)$
    - End.

    Average test errors $\mathcal{E}(\mathcal{D}_T) \longrightarrow \mathcal{E}_T$.
  - End.
- End.

Figure 6. Mass identification protocol flow chart.

TABLE 1

Best network architectures and associated test errors. **L**: *linear*, **H**: *sigmoidal*

Inputs $\{u_s\}$	Test error $\mathscr{E}_T$ [80% confidence int.]	arch.
$\widehat{N_{yy}}(\omega_{1,\ldots,4})$, $\hat{E}((y-E(y))^4)$	5·67e − 03 [5·19e − 03; 6·15e − 03]	L
$\widehat{\mathscr{N}_y^1}$, $\hat{E}((y-E(y))^4)$	7·24e − 03 [6·89e − 03; 7·60e − 03]	2H
$\hat{E}((y-E(y))^3)$, $\hat{E}((y-E(y))^4)$	7·35e − 03 [7·10e − 03; 7·59e − 03]	4H
$\widehat{\mathscr{N}_y^1}$, $\widehat{S0_y}$	7·80e − 03 [7·32e − 03; 8·29e − 03]	L
$\widehat{N_{yy}}(\omega_{1,\ldots,4})$	8·34e − 03 [7·71e − 03; 8·97e − 03]	2H
$\widehat{N_{yy}}(\omega_{1,\ldots,4})$, $\widehat{\min(y)}$	9·57e − 03 [8·84e − 03; 1·02e − 02]	L
$\widehat{\mathscr{N}_y^1}$, $\hat{E}((y-E(y))^4)$, $\hat{E}((y-E(y))^3)$, $\hat{E}((y-E(y))^2)$	9·78e − 03 [9·29e − 03; 1·02e − 02]	4H
$\widehat{\mathscr{N}_y^1}$	9·93e − 03 [9·31e − 03; 1·05e − 02]	2H
$\widehat{\min(y)}$, $\hat{E}((y-E(y))^3)$, $\widehat{\mathscr{N}_y^1}$, $\hat{E}((y-E(y))^4)$	1·03e − 02 [9·79e − 03; 1·08e − 02]	4H
$\widehat{\mathscr{N}_y^1}$, $\widehat{\min(y)}$, $\hat{E}((y-E(y))^4)$	1·05e − 02 [9·98e − 03; 1·11e − 02]	2H
$\widehat{N_{yy}}(\omega_{1,\ldots,4})$, $\widehat{\mathscr{N}_y^1}$	1·12e − 02 [1·06e − 02; 1·19e − 02]	2H
$\widehat{\mathscr{N}_y^1}$, $\widehat{\mathscr{N}_y^2}$	1·28e − 02 [1·19e − 02; 1·37e − 02]	2H
$\widehat{\mathscr{N}_y^3}$, $\hat{E}((y-E(y))^4)$, $\hat{E}((y-E(y))^3)$	1·47e − 02 [1·38e − 02; 1·56e − 02]	4H
$\widehat{N_{yy}}(\omega_{1,\ldots,4})$, $\widehat{S0_y}$	1·66e − 02 [1·51e − 02; 1·80e − 02]	L
$\widehat{\mathscr{M}_y^3}$, $\widehat{S0_y}$	1·69e − 02 [1·57e − 02; 1·80e − 02]	4H
$\widehat{N_{yy}}(\omega_{1,\ldots,4})$, $\hat{E}((y-E(y))^3)$	1·88e − 02 [1·82e − 02; 1·95e − 02]	4H
$\widehat{\mathscr{M}_y^1}$, $\widehat{\mathscr{M}_y^2}$, $\widehat{\mathscr{M}_y^3}$, $\widehat{\mathscr{M}_y^4}$	1·94e − 02 [1·86e − 02; 2·02e − 02]	2H
$\widehat{S_{yy}}$, $\hat{E}((y-E(y))^4)$	2·03e − 02 [1·93e − 02; 2·12e − 02]	4H
$\widehat{\mathscr{M}_y^3}$, $\hat{E}((y-E(y))^4)$	2·21e − 02 [2·10e − 02; 2·32e − 02]	4H
$\widehat{\mathscr{M}_y^1}$, $\widehat{\mathscr{M}_y^2}$	2·33e − 02 [2·23e − 02; 2·44e − 02]	4H
$\widehat{\max(y)}$, $\widehat{\min(y)}$	2·35e − 02 [2·24e − 02; 2·47e − 02]	L
$a_1, \ldots, a_8$,	4·39e − 02 [4·19e − 02; 4·59e − 02]	2H
$a_1, \ldots, a_8, \hat{e}$	2·49e − 02 [2·37e − 02; 2·61e − 02]	L
$\hat{E}(y)$, $\hat{E}((y-E(y))^2)$, $\hat{E}((y-E(y))^3)$, $\hat{E}((y-E(y))^4)$	3·07e − 02 [2·97e − 02; 3·17e − 02]	4H
$\hat{E}((y-E(y))^4)$, $\widehat{S0_y}$	4·34e − 02 [4·28e − 02; 4·39e − 02]	4H
$\widehat{S_{yy}}$	5·22e − 02 [5·03e − 02; 5·42e − 02]	4H
(hereafter, inputs use road information)		
$\widehat{N_{yy}}(\omega_{1,\ldots,4})$, $\widehat{S_{xx}}(4)$	2·45e − 03 [2·30e − 03; 2·60e − 03]	L
$\widehat{N_{yy}}(\omega_{1,\ldots,4})$, $\widehat{S0_x}$	3·40e − 03 [3·19e − 03; 3·61e − 03]	L
$\widehat{\mathscr{N}_y^1}$, $\widehat{S_{xx}}(4)$	9·88e − 03 [9·46e − 03; 1·02e − 02]	L
$\widehat{\mathscr{N}_y^1}$, $\hat{E}((y-E(y))^3)$, $\widehat{S_{xx}}(4)$	1·38e − 02 [1·32e − 02; 1·44e − 02]	L
$\widehat{S_{yy}}(\omega_{1,\ldots,4})$, $\widehat{S_{xx}}(4)$, $\widehat{S0_y}$	1·61e − 02 [1·51e − 02; 1·71e − 02]	L
$\hat{E}((y-E(y))^3)$, $\widehat{S_{xx}}(4)$	4·04e − 02 [3·88e − 02; 4·20e − 02]	2H
$\hat{E}((y-E(y))^3)$, $\widehat{S0_x}$	4·36e − 02 [4·27e − 02; 4·45e − 02]	L

intervals are given. They show meaningful differences in mass identification performance of varying networks.

The best prediction model found is linear and takes $\widehat{N_{yy}}(\omega_{1,\ldots,4})$ and $\widehat{S_{xx}}(4)$ as inputs ($\mathscr{E}_T = 2{\cdot}45e-03$). It is, however, of no direct use because it relies on information about the road through $\widehat{S_{xx}}(4)$, which is not available. Yet, it shows that $\widehat{S_{xx}}(4)$ usefully completes $\widehat{N_{yy}}(\omega_{1,\ldots,4})$ to identify the mass. It confirms, for the non-linear problem, what linear sensitivity analysis hinted at (relation (11) and Appendix A). The pendant of this best overall network that does not taken information on the road is linear and has $u_{\mathscr{I}} = \{\widehat{N_{yy}}(\omega_{1,\ldots,4})\ \hat{E}((y-E(y))^4)\}$ as inputs ($\mathscr{E}_T = 5{\cdot}67e-03$). This is logical since, among all candidate features, $\hat{E}((y-E(y))^4)$ has the highest correlation with the missing information, $\widehat{S_{xx}}(4)$. Predictions of this network are illustrated in Figure 7. The average error on the mass is 1·5% and the error is always below 5%. The second-best network (not using road information) processes $\widehat{\mathscr{N}_y^1}$ and $\hat{E}((y-E(y))^4)$. It shows that the most important aspect of normalized power spectral density for mass identification is how it is distributed in the frequency range. This is the aspect of N_{yy} that is carried onto $\mathscr{N}_y^1$. More generally, Table 1

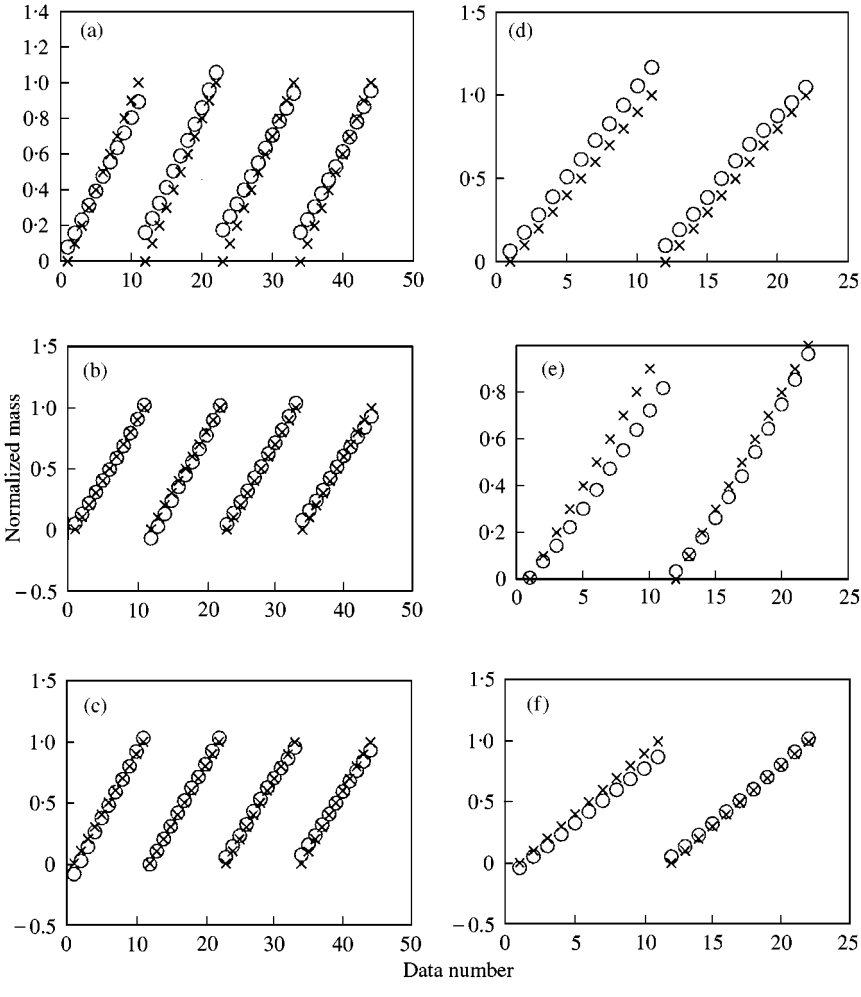

Figure 7. Typical mass predictions of linear network on three learning and three testing sets: (a) Learning $\mathscr{D}_{D1}$; (b) Learning $\mathscr{D}_{D2}$; (c) Learning $\mathscr{D}_{D3}$; (d) Testing $\mathscr{D}_{T1}$; (e) Testing $\mathscr{D}_{T2}$; (f) Testing $\mathscr{D}_{T3}$ sets: Inputs are $u_{\mathscr{I}} = \{\hat{E}(\ddot{x}_1 - \hat{E}(\ddot{x}_1))^4, \widehat{N_{yy}}(\omega_{1,\ldots,4})\}$. Test error = $5{\cdot}33e-03$. Key: ×, measurement data; ○, network output.

underlines how mass identification is best performed from a co-operation between features sensitive to mass and features insensitive to mass and sensitive to road samples. The four best networks are such examples. Another conclusion stemming from Table 1 is that $\widehat{N_{yy}}$ is a much more reliable feature for mass identification than $\widehat{S_{yy}}$. Sensitivity analysis in the linear case (Appendix A) explains this advantage by the robustness of $\widehat{N_{yy}}$ to road sampling. Finally, Table 1 illustrates the bias–variance trade-off: for the same information content and a given database, the smaller the number of inputs (the smaller the number of weights), the better. For example, remembering that $N_{yy} = S_{yy}/S0_y$, $\{\widehat{N_{yy}}(\omega_{1,\ldots,4}), \widehat{S_{xx}}(4)\}$ and $\{\widehat{S_{yy}}(\omega_{1,\ldots,4}), \widehat{S_{xx}}(4), \widehat{S0_y}\}$ have the same input content but the first network should be preferred. Similarly, $\{\widehat{N_{yy}}(\omega_{1,\ldots,4})\}$ outperforms $\{\widehat{N_{yy}}(\omega_{1,\ldots,4}), \widehat{\mathcal{N}_y^1}\}$.

4. DYNAMIC SYSTEM IDENTIFICATION

The preceding discussion on mass identification aims at illustrating how a vehicle can be characterized from acceleration measurements. Next, its dynamics can be learned by recurrent neural networks. The specific problem tackled here is load calculations at vehicle locations knowing accelerations at other locations. Such a problem serves for customer surveys in the automobile industry.

The difficulties of dynamic identification are of a different nature from those of mass identification. Mass identification suffered from the lack of data from which to learn. On the contrary, dynamics identification typically benefits from a very large database: at a sampling frequency of 1000 Hz. Thousand data points are provided every second. Two practical difficulties of dynamic identification are the determination of network architectures and the numerical cost of learning.

4.1. RECURRENT NEURAL NETWORK STRUCTURES

Two principal recurrent network architectures exist and will be considered here: autoregressive networks and state-space networks. Autoregressive networks predict the next observed outputs based on a finite collection of previous inputs and observed outputs. State-space networks synthesize the next observed outputs based on observed and non-observed state variables at the previous time step. State variables play a role comparable to memory. No noise model, that accounts for the difference between the prediction and real response, is included in the present network structures since, during simulations, the real response of the system is not known.

To keep formal agreement with the one-wheel test case of Figure 1, observed outputs of the system are denoted as $f_2(k)$ (force on mass m_2) and inputs are denoted as $\ddot{x}_1(k)$ (accelerations at the wheel hub), where k is the kth time step. Autoregressive output error model are described as

$$\begin{aligned} &\text{regressor: } \phi(k) = [\widehat{f_2}(k-1), \ldots, \widehat{f_2}(k-n_0), \ddot{x}_1(k), \ldots, \ddot{x}_1(k-n_u)], \\ &\text{predictor: } \widehat{f_2}(k) = f(\phi(k), w). \end{aligned} \tag{12}$$

Denoting by $\chi(k)$ the state variable vector at time k, state-space models are written as

$$\begin{aligned} &\text{regressor: } \phi(k) = [\widehat{\chi_1}(k), \ldots, \widehat{\chi_n}(k), \ddot{x}_1(t), \dot{x}_1(t)], \\ &\text{predictor: } \hat{\chi}(k+1) = f(\hat{\chi}(k), w), \qquad \hat{\chi}(k) = [\widehat{\chi_1}(k), \ldots, \widehat{\chi_n}(k)]^{\mathrm{T}}, \\ &\qquad\qquad \widehat{f_2}(k) = \mathbf{C}(w)\hat{\chi}(k). \end{aligned} \tag{13}$$

$\mathbf{C}(w)$ is a visibility matrix, translating state variables into observed outputs. Note that state-space networks considered here have a special connectivity, known as multivariate canonical form, which guarantees identifiability for linear hidden neurons (see reference [1], Appendix 4A). This structure is generalized here to non-linear networks, in the hope that good identifiability properties are kept.

For both autoregressive and state-space recurrent neural network structures, learning is implemented by minimizing least-squares distances to experimental data. The Levenberg–Marquardt algorithm [20, 21] solves the optimization problem. Derivatives of least-squares errors are calculated by backpropagation through time (cf. reference [25]).

4.2. ONE–WHEEL TEST CASE

Structures of autoregressive and state-space recurrent networks can be derived analytically for a linear one-wheel problem by discretizing the differential equations (A.1). These structures are then generalized to the non-linear problem. Discretization of equation (A.1) is performed here by approximating derivatives by finite differences. Another approach would be to integrate the equations of motion over one-time step, assuming the input $\ddot{x}_1(t)$ remains constant [26].

To simplify notations and since time is discretized, time indices instead of real times are used in the following functions of time, so that $x(t-1)$ really means $x((t-1)\Delta t)$. The second equation of equation (A.1) can be written at two successive time steps, and resulting relations subtracted. This yields

$$f_2(t) - f_2(t-1) = -c_1(\Delta\dot{x}_2(t) - \Delta\dot{x}_1(t)) - k_1(\Delta x_2(t) - \Delta x_1(t)), \tag{14}$$

where

$$\Delta x(t) = x(t) - x(t-1), \quad \Delta\dot{x}(t) = \dot{x}(t) - \dot{x}(t-1). \tag{15}$$

For small time steps Δt, a first order Taylor approximation can be used on the right side of equation (14) to increase the degree of derivation,

$$f_2(t) - f_2(t-1) = -c_1\Delta t(\ddot{x}_2(t-1) - \ddot{x}_1(t-1)) - k_1\Delta t(\dot{x}_2(t) - \dot{x}_1(t)) + \mathcal{O}(\Delta t^2). \tag{16}$$

Writing equation (16) at two consecutive times, subtracting and using a first order approximation one more time gives

$$m_2 f_2(t) = (2m_2 - c_1\Delta t)f_2(t-1) + (-m_2 - k_1\Delta t + c_1\Delta t)f_2(t-2) + m_2c_1\Delta t\ddot{x}_1(t-1) + (m_2k_1\Delta t^2 - m_2c_1\Delta t)\ddot{x}_1(t-2) + \mathcal{O}(\Delta t^2). \tag{17}$$

Equation (17) corresponds to the autoregressive structure (12). For small time steps, it suggests that an autoregressive network with $\ddot{x}_1$ as inputs, f_2 (equivalently $\ddot{x}_2 = f_2/m_2$) as output and $n_o = n_u = 2$, would correctly describe the linear one-wheel system.

In a similar fashion, a state-space recurrent network structure can be obtained. A system composed of, first, equation (16) with $\dot{x}_1$ kept as in equation (14), and second, Taylor

expansion of $\dot{x}_2(t)$, is a state-space expression:

$$\begin{bmatrix} 1 & c_1 \\ 0 & m_2 \end{bmatrix} \begin{pmatrix} f_2(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{bmatrix} 1 & c_1 - k_1 \Delta t \\ \Delta t & m_2 \end{bmatrix} \begin{pmatrix} f_2(t-1) \\ \dot{x}_2(t-1) \end{pmatrix} + \begin{pmatrix} c_1 \dot{x}_1(t) + (k_1 \Delta t - c_1)\dot{x}_1(t-1) + \mathcal{O}(\Delta t^2) \\ \mathcal{O}(\Delta t^2) \end{pmatrix}. \quad (18)$$

This equation corresponds to the state-space structure (13) with $\chi = [f_2 \dot{x}_2]^{\mathrm{T}}$. Note that the above derivations of autoregressive and state-space expressions illustrate how the number of states and sizes of regression horizons are related. Finite differences translate regression steps into state variables. Both are equal to 2 in the linear one-wheel problem. Equations (17) and (18) could easily be represented as neural networks with linear hidden units. They are used in the following tests to choose the network inputs and outputs. However, contrary to the above analysis, the tests are non-linear, making more flexibility in the networks necessary. For this reason, sigmoidal hidden units, regressions and non-observed state variables will be added.

Both families of networks are now tested on the non-linear one-wheel problem. Learning and testing databases contain 2000 points each. The autoregressive network is composed of three sigmoidal and one linear hidden neurons. To be conservative, horizons of size 3 are chosen (one more than linear analysis suggests), $n_o = n_u = 3$. Early trials showed that a state-space network is competitive with an autoregressive network provided it has more non-observed state variables than the horizon size of the autoregressive network. Consequently, it also needs more hidden neurons. The final state-space network has f_2 and $\dot{x}_2$ as observed state variables, six non-observed state variables, $\mathbf{C} = [1\ 0\ 0\ 0\ 1\ 0\ 0\ 0]$ and 16 hidden neurons. Initial tests showed high-frequency oscillations and accidental offsets in autoregressive and state-space network simulations. These flaws disappeared on regularizing the least squares through weight decay with $\nu = 0{\cdot}01$ [6]. Training stops after 15 iterations for the autoregressive network and after 30 iterations for the state-space network. Test error of the autoregressive network yields $\mathscr{E}_T = 0{\cdot}24\mathrm{e} - 03$. The associated simulation is illustrated in Figure 8. Test error of the state-space network is $\mathscr{E}_T = 3{\cdot}25\mathrm{e} - 03$. At such accuracy levels, network and data responses seem superimposed.

Even though both networks produce satisfactory dynamic simulations, the autoregressive network should be preferred. It has a smaller test error, but mainly it is of

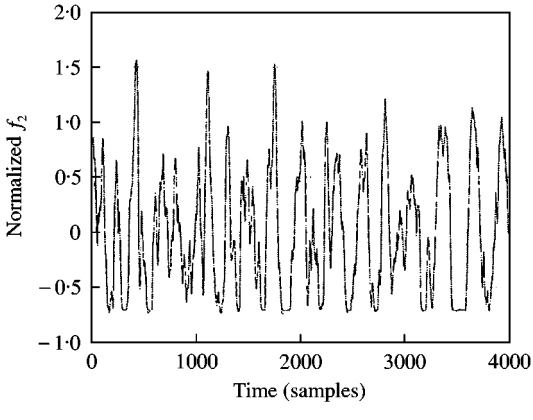

Figure 8. Comparison of autoregressive network and test data on $f_2(t)$, non-linear one-wheel test, $\mathscr{E}_T = 0{\cdot}24e - 03$. ——, data; - - -, network.

a smaller size, making it both less susceptible to high variance and numerically cheaper to identify. An autoregressive output error model is thus chosen to simulate real car dynamics.

4.3. IDENTIFICATION OF A CAR VERTICAL DYNAMICS

The aforementioned autoregressive neural network simulation is now applied to vertical load calculation at a car suspension chamber based on seven acceleration recordings. The accelerations provided are vertical accelerations at the front wheel steering knuckles (2), vertical accelerations at front suspension spring supports (2), longitudinal, vertical and transversal accelerations at the center of gravity of the car. Horizontal accelerations are included because, contrary to vertical accelerations, they are dominated by low frequencies and it is essential to account for the coexistence of low (below 5 Hz and medium to high (between 5 and 50 Hz) frequency phenomena in car dynamics. The load was recorded by a dynamometric wheel. The database is made up of 292·96 s recording at a sampling rate of 256 Hz, for a total of about 75 000 points. Assumptions underlying the one-wheel test case (constant speed, curvature and ground probabilistic model) are not satisfied.

Because experimental conditions are not well controlled, a neural structure minimizing variance is chosen to identify the load history. It is a linear autoregressive model with horizons of $n_u = 4$ and $n_o = 6$ (cf., equation (12)). Regularization by weight decay is applied with $v = 0{\cdot}1$. The other characteristics of network learning are similar to those exposed earlier in section 4.1.

The learning database is made up of three sample intervals [70–101], [109–125] and [203–223] s. The rest of the recording constitutes the testing database. Complete recording and network simulation are shown in Figure 9. One notices how different parts of the road have different load responses. Overall, 83% of the network simulation shows error below 5%, 97% below 10% and 99·34% below 20%. A detailed view of a time interval not included in learning is given in Figure 10.

5. CONCLUSIONS

Two complementary neural network-based techniques were applied to a dynamic system identification problem. First, the system's mass was identified from accelerations using

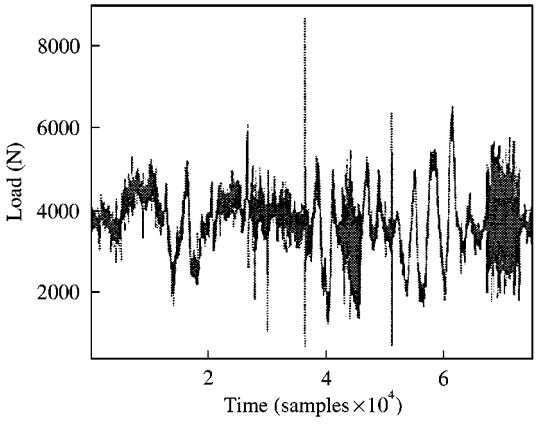

Figure 9. Comparison of load recording and autoregressive simulation, complete data (training and testing), car test. ——, data; - - -, network.

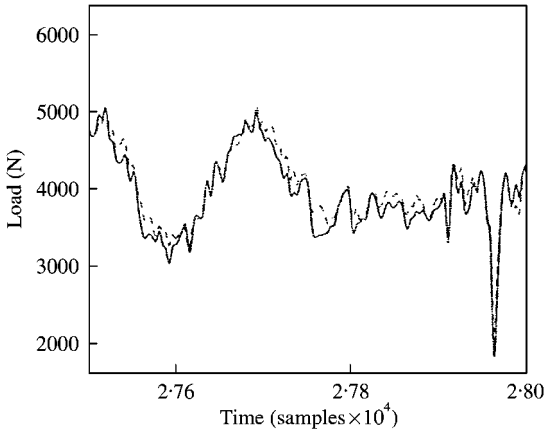

Figure 10. Comparison of load recording and autoregressive simulation, zoom on some of the test data, car test. ——, data; - - -, network.

feed-forward neural networks. Careful statistical tests were necessary to account for the limited number of data. In particular, two techniques for controlling variance, early stopping and cross-validation, have been used. Beyond the derivation of an identification program, the method showed synergy of the fourth statistical moment and the normalized power spectral density to identify the system's mass.

Second, recurrent neural networks have been tested for simulating loads with known accelerations at specific system's components. For a one-wheel system, autoregressive and state-space structures were derived by discretizing the linear differential equation. It was shown that horizon-based autoregressive networks should be preferred to state-space networks: they performed slightly better with a lower model complexity, i.e., at a lower numerical cost. It is likely that this result can be generalized to most dynamic systems which do not have long memory effects, as opposed to non-linear material behaviors, for example. An encouraging application of an autoregressive network to simulate a real vehicle dynamics has been described. The robustness of the method to the deterioration of mathematical assumptions (stationarity and ergodicity) in real testing conditions brings to the fore the potential of neural system identification for rapid structural dynamic modelling.

ACKNOWLEDGMENTS

The authors would like to thank S. Canu, P. Leray and A. Rakotomamonjy for enlightening discussions. The help of P. Leblanc for parts of the programming is acknowledged.

REFERENCES

1. L. Ljung 1987 *System Identification: Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall.
2. E. Walter and L. Pronzato 1994 *Identification de Modèles Paramétriques à Partir de Données Expérimentales* (in French). Paris: Masson.
3. K. J. Åström and C. G. Källström 1976 *Automatica* **12**, 9–22. Identification of ship steering dynamics.
4. R. P. Lippmann April 1987 *IEEE ASSP Magazine* **4**, 4–22. An introduction to computing with neural nets.

5. D. R. HUSH and B. G. HORNE January 1993 *IEEE Signal Processing Magazine*, 8–39. Progress in supervised neural networks: what's new since Lippmann?
6. C. M. BISHOP 1995 *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.
7. P. HAJELA and L. BERKE 1991 *Computers and Structures* **41**, 657–667. Neurobiological computational models in analysis and design.
8. Z. WASZCZYSZYN 1998 *Advances in Engineering Computational Technology*, 173–187. Edinburgh: Civil-Comp Press. Some new results in applications of backpropagation neural networks in structural and civil engineering.
9. I. TAKAHASHI 1999 *Journal of Sound and Vibrations* **228**, 857–870. Identification for critical flutter load and boundary conditions of a beam using neural networks.
10. S. CHEN, S. A. BILLINGS and P. M. GRANT 1990 *International Journal Control* **51**, 1191–1214. Non-linear system identification using neural networks.
11. A. U. LEVIN 1992 *Yale University Ph.D Thesis*. Neural networks in dynamical systems.
12. G. G. YEN 1994 *IEEE Transactions on Control Systems Technology* **2**, 362–370. Reconfigurable learning control in large space structures.
13. Y.-K. WEN, J. GHABOUSSI, P. VENINI, and K. NIKZAD 1995 *Smart Material Structure* **4**, 149–157. Control of structures using neural networks.
14. H. M. CHEN, K. H. TSAI, G. Z. QI, J. C. S. YANG and F. AMINI 1995 *Journal of Computing in Civil Engineering* **9**, 168–176. Neural networks for structure control.
15. S. W. DOEBLING, C. R. FARRAR, M. B. PRIME and D. W. SHEVITZ May 1996 Los Alamos National Laboratory, New Mexico, Report LA-13070-MS. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review.
16. F. M. HEMEZ and S. W. DOEBLING September 2000 *Proceedings of the 25th International Conference on Noise and Vibration Engineering* (*ISMA*-25), *Leuven*, *Belgium*. Inversion of structural dynamics simulations: State-of-the-art and orientations of the research, to appear.
17. F. M. HEMEZ and S. W. DOEBLING February 2000 *Proceedings of the 18th International Modal Analysis Conference* (*IMAC-XVIII*), *San Antonio*, *Texas*, 124–132. Validation of nonlinear modeling from impact test data using probability integration.
18. A. R. BARRON 1993 *IEEE Transactions on Information Theory* **39**, 930–945. Universal approximation bounds for superposition of a sigmoidal function.
19. L. K. JONES 1990 *Proceedings of the IEEE* **78**, 1586–1589. Constructive approximations for neural networks by sigmoidal functions.
20. K. LEVENBERG 1944 *Quarterly of Applied Mathematics* **2**, 164–168. A method for the solution of certain nonlinear problems in least squares.
21. D. W. MARQUARDT 1963 *Journal of SIAM* **11**, 431–441. An algorithm for least squares estimation of nonlinear parameters.
22. D. E. RUMELHART, G. E. HINTON and R. J. WILLIAMS 1986 *Nature* **323**, 533–536. Learning representations by back-propagating errors.
23. P. D. WELCH 1967 *IEEE Transactions on Audio Electroacoustics* **AU-15**, 70–73. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms.
24. J. T. MCCLAVE and F. H. DIETRICH 1982 *Statistics*. Dellen Publishing Company, second edition.
25. B. A. PEARLMUTTER 1995 *IEEE Transactions on Neural Networks* **6**, 1212–1228. Gradient calculations for dynamic recurrent neural networks: a survey.
26. K. J. ÅSTRÖM and B. WITTENMARK 1997 *Computer-controlled Systems*. Englewood Cliffs, NJ: Prentice-Hall, third edition.

APPENDIX A: MASS SENSITIVITY ANALYSIS

In this section, mass sensitivity analysis is carried out for the linear one-wheel problem sketched in Figure 1. Equations of motion are

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{pmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{pmatrix} + \begin{bmatrix} c_1 & -c_1 \\ -c_1 & c_1 \end{bmatrix} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} + \begin{bmatrix} k_0 + k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -m_1 \mathbf{g} \\ -m_2 \mathbf{g} \end{pmatrix} + \begin{pmatrix} k_0 x_0 \\ 0 \end{pmatrix}. \tag{A.1}$$

The system frequency response function between the road deflection and the hub acceleration $\ddot{x}_1$ is

$$H(\omega) = -\omega^2 X_1, \tag{A.2}$$

where X_1 is the solution of

$$\begin{bmatrix} k_0 + k_1 + \mathrm{i}c_1\omega - m_1\omega^2 & -k_1 - \mathrm{i}c_1\omega \\ -k_1 - \mathrm{i}c_1\omega & k_1 + \mathrm{i}c_1\omega - m_2\omega^2 \end{bmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} k_0 \\ 0 \end{pmatrix}. \tag{A.3}$$

The best frequency, ω, for mass identification using either the power spectral density $S_{yy}(\omega)$ or the normalized power spectral density $N_{yy}(\omega)$ is sought. Such optimal frequency maximizes the sensitivity to the mass and minimizes the sensitivity to $S_{x_0x_0}(\omega)$ (see also equation (11)):

$$\max_{\omega} |\partial S_{yy}(\omega)/\partial m_2| = \max_{\omega} |(\partial |H(\omega)|^2/\partial m_2)|\, S_{x_0x_0}(\omega)|, \tag{A.4}$$

or,

$$\max_{\omega} |\partial N_{yy}(\omega)/\partial m_2| = \max_{\omega} |(\partial (|H(\omega)|^2/S0_y)/\partial m_2) S_{x_0x_0}(\omega)| \tag{A.5}$$

and, simultaneously,

$$\min_{\omega} |\partial S_{yy}(\omega)/\partial S_{x_0x_0}(\omega)| = \min_{\omega} ||H(\omega)|^2|, \tag{A.6}$$

or,

$$\min_{\omega} |\partial N_{yy}(\omega)/\partial S_{x_0x_0}(\omega)| = \min_{\omega} |(|H(\omega)|^2 S0_y - |H(\omega)|^4 S_{x_0x_0}(\omega))/S0_y^2|. \tag{A.7}$$

Log-derivatives of $S_{yy}(\omega)$ and $N_{yy}(\omega)$ with respect to mass and road are shown in Figures A.1 and A.2. First of all, the largest mass sensitivities are obtained both for $S_{yy}(\omega)$ and $N_{yy}(\omega)$ around 4 Hz, which is a local maximum of $|H(\omega)|$. Next, although $S_{yy}(4)$ is more sensitive to the mass than $N_{yy}(4)$, it is also more sensitive to road variations. Normalizing

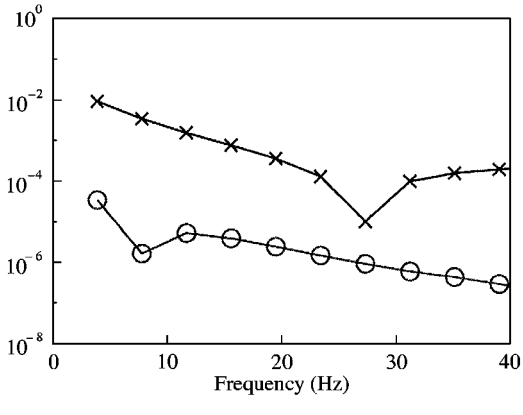

Figure A.1. Comparison of sensitivities of $S_{yy}(\omega)$ and $N_{yy}(\omega)$ to mass for various frequencies ω. $m_2 = 331$ kg, $\Delta T = 100{\cdot}352$ s. —×—, $|[\partial S_{yy}(\omega)/\partial m_2]/[S_{yy}(\omega)]|$; —○—, $|[\partial N_{yy}(\omega)/\partial m_2]/[N_{yy}(\omega)]|$.

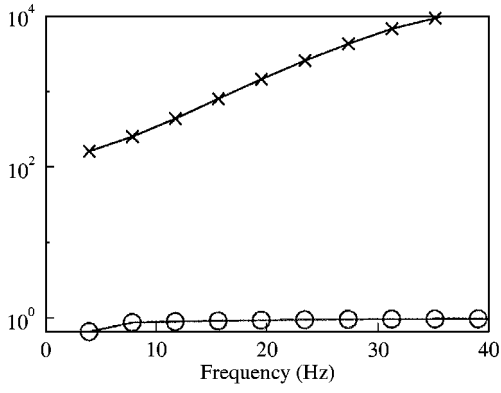

Figure A.2. Comparison of sensitivities of $S_{yy}(\omega)$ and $N_{yy}(\omega)$ to $S_{x_0x_0}$ for various frequencies ω. $m_2 = 331$ kg, $\Delta T = 100{\cdot}352$ s. —×—, $|[\partial S_{yy}(\omega)/\partial m_2]/[S_{yy}(\omega)]|$; —○—, $|[\partial N_{yy}(\omega)/\partial m_2]/[N_{yy}(\omega)]|$.

the power spectral density by $S0_y = (\sum_{\omega_i} \widehat{S_{yy}}(\omega_i))$ yields a more robust estimate with respect to road variations. This explains why, in section 3.4, $N_{yy}(\omega)$ is a more adequate input to the neural network than $S_{yy}(\omega)$.