

# Identification of pre-sliding and sliding friction dynamics: Grey box and black-box models

K. Worden<sup>a,\*</sup>, C.X. Wong<sup>a</sup>, U. Parlitz<sup>b</sup>, A. Hornstein<sup>b</sup>, D. Engster<sup>b</sup>,  
T. Tjahjowidodo<sup>c</sup>, F. Al-Bender<sup>c</sup>, D.D. Rizo<sup>d</sup>, S.D. Fassois<sup>d</sup>

<sup>a</sup>Department of Mechanical Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK

<sup>b</sup>Drittes Physikalisches Institut, Universität Göttingen, Bürgerstraße 42-44, D-37073 Göttingen, Germany

<sup>c</sup>Department of Mechanical Engineering Division, K.U. Leuven, P.M.A., Celestijnenlaan 300B, 3001 Heverlee (Leuven), Belgium

<sup>d</sup>Department of Mechanical & Aeronautical Engineering, University of Patras, GR 265 00 Patras, Greece

Received 18 April 2005; received in revised form 25 August 2005; accepted 6 September 2005

Available online 8 November 2005

---

## Abstract

The non-linear dependence of pre-sliding and sliding friction forces on displacement and velocity is modelled using different physics-based and black-box approaches including various Maxwell-Slip models, neural networks, non-parametric (local) models and recurrent networks. The efficiency and accuracy of these identification methods is compared for an experimental time series where the observed friction force is predicted from the measured displacement and estimated velocity. All models, although varying in their degree of accuracy, show good prediction capability of friction. Finally, it is shown that better results can be achieved by using an ensemble of the best models for prediction.

© 2005 Elsevier Ltd. All rights reserved.

---

## 1. Introduction

Friction is a complex non-linear phenomenon that exists in mechanical systems. It is the result of interactions between two neighbouring surfaces and is dependent on many parameters, such as: surface topography and materials, presence and type of lubrication and relative motion. The friction phenomenon can usually be divided into two operating regimes, pre-sliding friction and gross sliding friction. Pre-sliding friction is largely dependent on the elastic and plastic deformations of asperities. Gross sliding friction is due to the shearing resistance of the asperities. In reality, the transition between these two regimes in the friction process is a continuous one and a good model should reflect this. A more detailed explanation about friction and its many modelling techniques can be found in various works in the literature [1–3].

The problems caused by friction such as limit cycles, tracking error and stick-slip motion have been studied extensively in engineering. This non-linearity (and other usually less severe—or possibly, easier to model—non-linearities, i.e. backlash, motor saturation, etc.) poses a problem in the control of mechanisms that require

---

\*Corresponding author. Tel./fax: +44 114 222 7758.

E-mail addresses: [k.worden@sheffield.ac.uk](mailto:k.worden@sheffield.ac.uk) (K. Worden), [parlitz@dpi.physik.uni-goettingen.de](mailto:parlitz@dpi.physik.uni-goettingen.de) (U. Parlitz), [farid.al-bender@mech.kuleuven.ac.be](mailto:farid.al-bender@mech.kuleuven.ac.be) (F. Al-Bender), [fassois@mech.upatras.gr](mailto:fassois@mech.upatras.gr) (S.D. Fassois).

a high degree of dynamic accuracy and performance. Many different compensation techniques exist for dealing with friction in mechanisms. Some are non-model based [4,5], while others are model-based methods. Many researchers have recently proposed several accurate friction models. Dahl's model [6], simulates the symmetric hysteresis loops observed in friction small-amplitude sinusoidal forcing. An improvement of this model has been implemented in the LuGre model [7]. The LuGre model includes other effects, such as those associated with the sliding of lubricated contacts (e.g. the Stribeck effect). The Leuven model [3], is based on the experimental finding that the friction force in the pre-sliding regime is a hysteretic function of the position with non-local memory, which was only approximated by the former models. A model based compensation scheme is described in [8].

In the present work, time-domain data is used to construct a friction model for the ultimate purpose of model-based compensation. Because the models are constructed on the basis of measured experimental data, the work falls within the field of *system identification*.

System identification is usually accomplished not only by processing measured data, but also by applying a priori knowledge. According to [10], different types of models are distinguished by the amount of physical prior knowledge they incorporate. These are:

- White-box models;
- Grey-box models;
- Black-box models.

White-box models are distinguished by the fact that they are perfectly specified by prior knowledge and physical insight. Grey-box models are obtained when less physical insight is available and fall into two subclasses: physical models and semiphysical models. The former occurs when prior knowledge is sufficient to supply a model form, but is insufficient to provide values of the model parameters and these must be estimated from data. The latter group of models occurs when prior knowledge is only sufficient to suggest which non-linear combinations of measured data will be relevant for the model, these combinations are then used in a black-box model. A black-box model can be obtained in the absence of a priori physical knowledge or insight and is usually formed from a known basis of model terms which is sufficiently flexible to represent a broad range of behaviours. In the analysis of friction, a good example of a white-box model in the context of automotive engineering can be found in [12]; the same authors have also used grey-box models extensively in the same context and examples can be found in [13,14].

In the case of the friction processes considered here, all three levels of modelling are feasible. However, while some of the authors have proposed a white-box (or largely white) model in [15], this paper is mainly concerned with grey and black-box models. The paper also concentrates on the regime where both pre-sliding and sliding friction are present, a previous paper having covered the case of pre-sliding friction alone [16].

The work described here was carried out by a consortium of participants from: The Katholieke University of Leuven (KUL), Belgium, The University of Patras (UP), Greece, The University of Göttingen (UG), Germany and the University of Sheffield (US), UK. This is reflected in the organisation of the paper. After a section describing the test set-up and data acquisition, the work of the participants is presented sequentially: first the grey-box models are described (KUL and UP), then the black-box models (UG and US). This is followed by a summary section which also describes the work on ensemble modelling. The paper is completed by discussion and conclusions.

## 2. Test set-up and data capture

The experimental data for system identification was obtained from a tribometer at KUL, a schematic is shown in Fig. 1. The rig was constructed on the basis of design rules expressed in [17]. A detailed discussion of the design and commissioning of the tribometer can be found in [18], a description of how the rig was used to obtain pre-sliding data is given in [16]. For the sake of completeness, a brief description of the rig is given here (Fig. 2).

The objective is to measure the friction force in the sliding interface between the friction block and the actuator block (Fig. 1). The instrument can be roughly divided into three parts: an actuator part, a friction

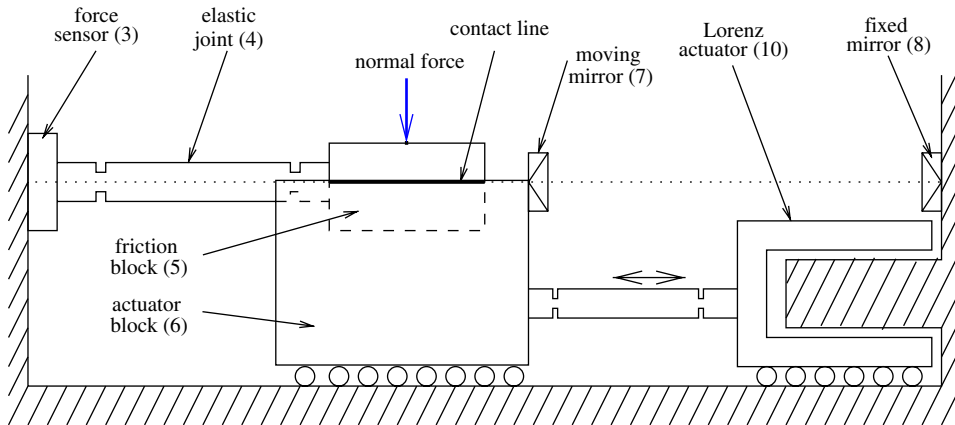


Fig. 1. Schematic of the experiment.

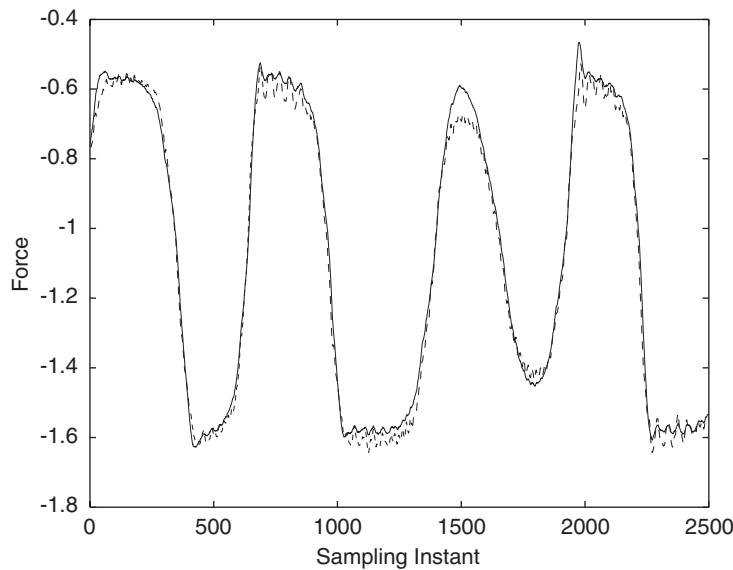


Fig. 2. Actual (solid line) and estimated (dashed) friction force using LuGre model with exponential Stribeck Curve.

part and a loading part. The different parts are decoupled as much as possible, the actuation part and friction part are only coupled by the friction interface under consideration and the loading part and friction part are completely separated by an air-bearing which ensures that all the tangential forces are directed to the force sensor.

The actuator part consists of three main parts: a Lorenz actuator, a moving block and a displacement sensor. The Lorenz actuator linearly forces, by means of a stinger, the moving block which makes contact with the friction block. The displacement of the moving block is measured using a Renishaw laser interferometer which measures the distance between a mirror fixed to the moving block and a mirror fixed to the frame. The Lorenz actuator is current driven and by feeding back the position signal into a controller a desired displacement can be obtained; therefore the set-up is capable of imposing forces or desired displacement trajectories.

The friction part is the critical part of the tribometer and has two important components: the friction block on which the friction force acts, and a force cell, which measures the friction force. There is an elastic joint

between the friction block and the force cell which consists of two pairs of elastic hinges. The purpose of the hinges is to set off small vertical, lateral and rotational alignment errors of the friction block (all perpendicular to the direction of displacement) caused by positioning the friction block on the moving block.

The force and displacement signals were measured on a dSpace system with an accuracy of 24 bits and a sampling frequency of 2500 Hz. 110 000 sample points were recorded. It was established that, as a result of the low level of excitation, there was a substantial quantisation noise component on the position signal. This was therefore low-pass filtered using a Butterworth filter implemented via a MatLab function (`filtfilt.m`). In order to preserve the phase of the position signal, which is critical to the identification, this function passes the data through the filter in both the forward and reverse directions in series. This type of treatment can clearly only be carried out off-line.

The identification exercise here was to produce a predictive model for the force based on the position signal. However, it is well known that although the friction force is dependent on the displacement in the pre-sliding regime, the main dependence in gross sliding is on velocity. The velocity signal was obtained by a process of discrete differentiation.

The identification problem was posed as follows: 110 000 points of force and displacement data were made available. The first 90 000 points were to be used for training and validation in whatever proportion the participants saw fit, the remaining 20 000 points of data were reserved as a test set in order to validate the identified models. As an objective measure of performance, the normalised mean-square-error (*MSE*) was selected, defined by

$$MSE(\hat{F}) = \frac{100}{N\sigma_F^2} \sum_{i=1}^N (\hat{F}_i - F_i)^2, \quad (1)$$

where  $F$  is the friction force,  $\sigma_F^2$  is the force variance and the caret denotes an estimated quantity. If the mean of the output signal  $\bar{F}$  is used as the model (i.e.  $\hat{F}_i = \bar{F}$  for all  $i$ ), the *MSE* is 100. On past experience an *MSE* of less than 5.0 indicates a good fit, while one of less than 1.0 indicates an excellent fit. In addition to the *MSE*, the normalised maximum errors on the test set were determined for the different models, this was defined by

$$\text{MAX}(\hat{F}) = \frac{1}{\sigma_F} \max_{i=1,N} (|\hat{F}_i - F_i|). \quad (2)$$

### 3. Grey-box modelling I: Katholieke Universiteit of Leuven

#### 3.1. Introduction

In this section, two grey-box models of friction will be considered, namely the LuGre and Generalised Maxwell Slip models.

The Generalised Maxwell-Slip (GMS) friction model is an extension of the Leuven model [9]. This model is based on three main components: a Stribeck curve for constant velocities, a hysteresis function with non-local memory in the pre-sliding regime, and frictional memory in the sliding regime. The hysteretic part with non-local memory for the pre-sliding regime is represented by Maxwell-Slip elements. Lampaert et al. [9], have merged the hysteretic property with the non-local memory property of the friction force in the pre-sliding regime, represented by Maxwell-Slip elements, in the original Leuven model. The friction force is given as the summation of the outputs of  $N$  elementary state models.

#### 3.2. The LuGre model

Two sets of 40 000 points each were used for training and validation purposes in the friction identification using the LuGre model. The model is described by the following equations,

$$\frac{dz}{dt} = v - \frac{|v|}{s(v)} z, \quad (3)$$

$$F_f = \sigma_0 z + \sigma_1 \frac{dz}{dt} + \sigma_2, \quad (4)$$

where  $\sigma_0$  is a viscous friction coefficient,  $\sigma_1$  is called the micro-viscous friction coefficient, and  $\sigma_2$  is a damping coefficient. The steady-state characteristic or *Stribeck curve*  $s(v)$  was simply assumed to have an exponential form of

$$s(v) = F_C + (F_S - F_C)e^{-|v|/v_S}. \quad (5)$$

This assumption was made due to the fact that in the steady state, the friction force as a function of velocity is bounded by a static force  $F_S$  and Coulomb force  $F_C$ .

Identification was carried out using a non-linear regression with the Nelder–Mead Downhill Simplex Algorithm, which has a major advantage in its short running time. It converges very rapidly when it finds a minimum. This algorithm also has the advantage that it does not require derivatives; however, it requires more function evaluations than some other algorithms such as the Levenberg–Marquardt algorithm.

The result of the LuGre model identification can be seen in Fig. 1, which shows the actual friction force in comparison to the modelled force. The actual force is represented by a solid line and the modelled force is denoted by a dashed line (this convention is adopted throughout the paper). This identification shows that the LuGre model exhibits an acceptably low error of 1.02% MSE on the testing set. The maximum error obtained was 0.51.

### 3.2.1. The LuGre model in pre-sliding friction

In order to gain a better understanding of the friction behaviour at low velocity, an estimate of the friction force in the pre-sliding regime was investigated.

Experimental friction data in the pre-sliding regime only was acquired by PMA/KUL using the same setup and conditions as the general data (as described in [16]). A validation using the LuGre model results obtained from the sliding case (above) was carried-out to estimate the friction force in this regime. Fig. 3 shows the resulting comparison between the measured data and the friction model. Identification of pre-sliding friction using the LuGre model also exhibits a low error with a MSE of 5.39% and maximum error of 0.1548. This exercise shows how widely applicable the parameters identified for the LuGre model for the sliding case are.

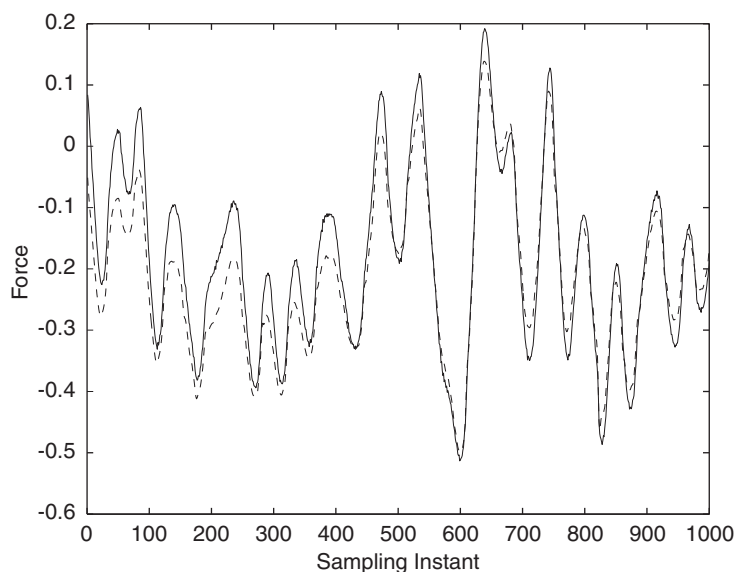


Fig. 3. Actual (solid line) and estimated (dashed) pre-sliding friction force using LuGre model.

### 3.3. Generalised Maxwell-Slip (GMS) model

In the GMS model, as an improvement of the Leuven model, the hysteretic property of non-local memory of the friction force in the pre-sliding regime, represented by Maxwell-Slip elements, was incorporated into the original Leuven model. The friction force is given as the summation of the outputs of  $N$  elementary state models.

The friction behaviour of each elementary model can be written as

$$\frac{dF_i}{dt} = k_i v_i. \quad (6)$$

if the model is sticking, or,

$$\frac{dF_i}{dt} = \text{sgn}(v)C\left(\alpha_i + \frac{F_i}{s(v)}\right) \quad (7)$$

if the model is slipping.

Once the model is slipping, it remains slipping until the direction of movement is reversed back or its velocity approaches a zero value.

By using the same number of points of training and validation sets and also the same optimisation algorithm, the result of this identification method can be seen in Fig. 4. The result shown in Fig. 4 is achieved by utilising 4 Maxwell-Slip elements. This GMS model yields a better MSE value compared to the result of LuGre model, the MSE for the testing set is 0.45%, while the maximum error is 0.26. The major improvement of the GMS model compared to the LuGre model is in the reduction of ‘spikes’ in the friction force which correspond to velocities near zero. A smoother transition between the sliding and pre-sliding regimes is captured with this model.

Fig. 5 shows the result of validating the GMS model obtained in the sliding regime in the pre-sliding regime only. The MSE is 1.65% and the maximum error is 0.1216.

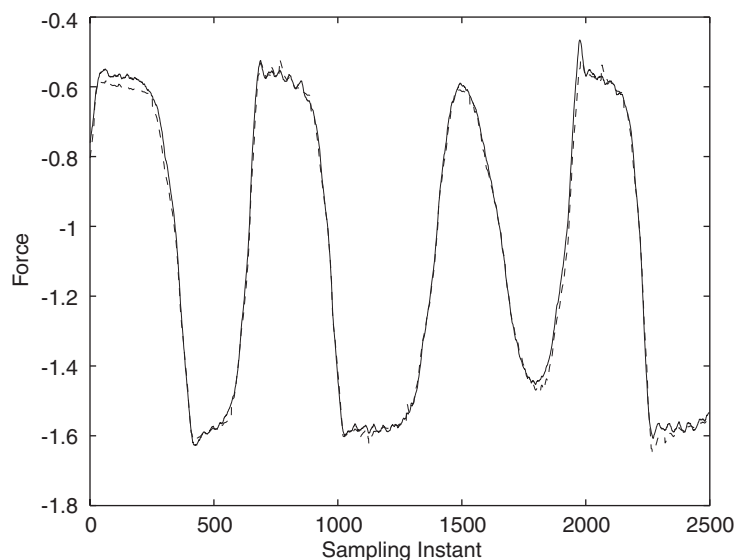


Fig. 4. Actual (solid line) and estimated (dashed) friction force using GMS model with 4 Maxwell-Slip elements and exponential Stribeck Curve.

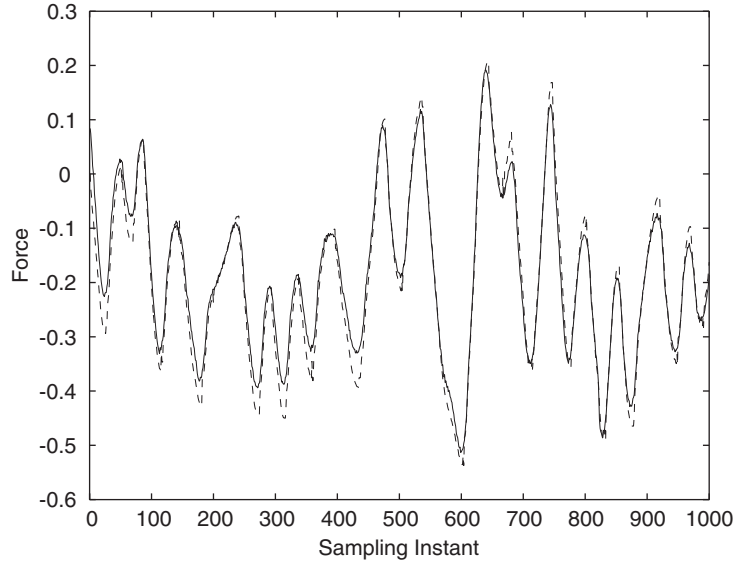


Fig. 5. Actual (solid line) and estimated (dashed) pre-sliding friction force using GMS model.

#### 4. Grey box modelling II: University of Patras

##### 4.1. Identification of physics-based models

The friction identification problem is currently addressed via three ‘physics-based’ methods. The first uses the LuGre model again [7], which accounts for most of the observed frictional characteristics [1]. The other two, designated as *NLR* (NonLinear Regression) and *DNLRX* (Dynamic NonLinear Regression with direct application of eXcitation), are based upon the Maxwell Slip model structure and are thus capable of representing the pre-sliding hysteresis with non-local memory [9,19] (a feature not present in the LuGre model [3]). Among the methods’ main advantages are simplicity and a direct physical interpretation.

All three methods attempt identification via the minimisation of a quadratic cost function of the form

$$\mathcal{J} \triangleq \sum_{t=\lambda}^N e^2(t), \quad (8)$$

where  $N$  is the length of the available signals,  $\lambda \equiv 1$  for the LuGre and *NLR* methods and  $\lambda \equiv \max\{n, n_x\} + 1$  (with  $n$  and  $n_x$  defined in Section 4.3 in the sequel) for the *DNLRX* method.  $e(t)$  is the model error defined as the difference between the measured,  $F(t)$ , and the corresponding model provided,  $\hat{F}(t)$ , friction

$$e(t) \triangleq F(t) - \hat{F}(t). \quad (9)$$

For a more elaborate description and analysis of the methods the reader is referred to Refs. [19,20].

##### 4.2. The LuGre method

This method incorporates the LuGre model structure given by Eqs. (3) and (4), which contain an unmeasurable state variable ( $z$ ) which represents the average deflection of an array of elastic bristles. The time evolution of  $z$  is provided by the following differential equation (3), while the friction is represented by (4). Within this section, the Stribeck curve  $s(v)$  is selected as follows:

$$s(v) = a_1 + \frac{a_2}{1 + \left(\frac{|v|}{v_s}\right)^\mu} \quad \text{with } a_1 = \frac{F_c}{\sigma_0}, \quad a_2 = \frac{F_s - F_c}{\sigma_0}, \quad (10)$$

where  $F_c$  and  $F_s$  are the *Coulomb* and *Static* friction forces, respectively,  $v_s$  the *Stribeck* velocity, and  $\mu$  an extra parameter for providing additional flexibility in the dynamics.

In view of Eq. (4), the LuGre model may be expressed as

$$\text{LG}(\mathbf{g}, \boldsymbol{\theta}_{LG}): F(t) = \boldsymbol{\theta}_{LG}^T \cdot \left[ z(t) \frac{dz(t)}{dt} v(t) 1 \right]^T + e(t) \tag{11}$$

subject to Eqs. (3) and (4). The vectors  $\mathbf{g}$  and  $\boldsymbol{\theta}_{LG}$  incorporate the model parameters to be estimated

$$\mathbf{g} \triangleq [z(1) \ a_1 \ a_2 \ v_s \ \mu]^T, \quad \boldsymbol{\theta}_{LG} \triangleq [\sigma_0 \ \sigma_1 \ \sigma_2 \ b]^T, \tag{12}$$

where  $z(1)$  denotes the initial value of the unmeasurable state variable  $z(t)$  and  $b$  is an extra parameter accounting for experimental friction offset.

### 4.3. NLR/DNLRX methods

Both the *NLR* and *DNLRX* methods are based upon the Maxwell Slip model structure [9]. This model consists of  $M$  massless elasto-slide elements in parallel configuration (all get excited by the same displacement  $x(t)$ —see Fig. 6). Each element  $i$  ( $i = 1, \dots, M$ ) is subject to Coulomb friction and is characterised by stiffness a  $k_i$ . The element’s position is designated as  $x_i(t)$ , the spring deformation as  $\delta_i(t) \triangleq x(t) - x_i(t)$  and the maximum spring deformation (threshold) as  $\Delta_i$  (this is the deformation before the  $i$ th element starts slipping). In mathematical terms, the normalised (with respect to  $\Delta_i$ )  $i$ th spring deformation  $\tilde{\delta}_i(t)$  is given as follows [19]:

$$\tilde{\delta}_i(t + 1) = \text{sgn}[x(t + 1) - x(t) + \delta_i(t)] \cdot \min\{|x(t + 1) - x(t) + \delta_i(t)|, \Delta_i\}. \tag{13}$$

Two friction force representations are employed: the first is provided by the *NLR* method, in which the friction is approximated as the summation of all the elemental spring forces, while the second is by the *DNLRX* method. In the latter case, the friction force is produced by driving the exerted displacement  $x(t)$  through a Finite Impulse Response (FIR) filter of order  $n_x$  and coefficients  $c_r$  ( $r = 0, \dots, n_x$ ), and the normalised spring

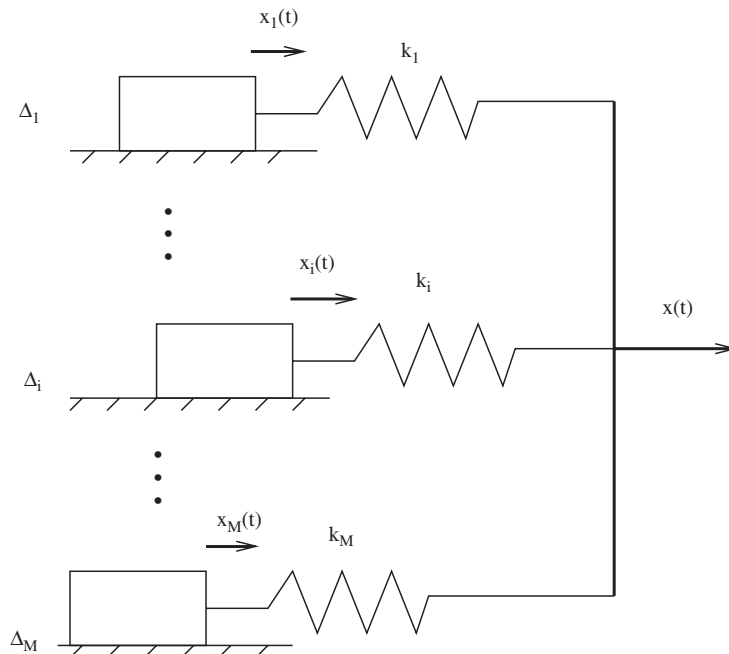


Fig. 6. Schematic representation of the Maxwell Slip model structure.

deformation vector  $\bar{\delta}(t)$ , defined as

$$\bar{\delta}(t) \triangleq [\bar{\delta}_1(t) \ \dots \ \bar{\delta}_M(t)]^T$$

through an  $M$ -dimensional FIR filter of order  $n$  with vector coefficients  $\theta_r$  ( $r = 0, \dots, n$ ).

Thus, the  $NLR$  and  $DNLRX$  model classes are of the following respective forms:

$$NLR(M; \mathbf{d}, \boldsymbol{\theta}_M): F(t) = \sum_{i=1}^M k_i \cdot \Delta_i \cdot \bar{\delta}_i(t) + b + e(t), \quad (14)$$

$$DNLRX(M, n, n_x; \mathbf{d}, \boldsymbol{\theta}_{EM}): F(t) = \sum_{r=0}^{n_x} c_r \cdot x(t-r) + \sum_{r=0}^n \theta_r^T \cdot \bar{\delta}(t-r) + b + e(t), \quad (15)$$

where  $e(t)$  is the corresponding model error, while  $\mathbf{d}$ ,  $\boldsymbol{\theta}_M$  and  $\boldsymbol{\theta}_{EM}$  are the vectors containing the unknown parameters of the corresponding models

$$\mathbf{d} \triangleq [\Delta_1 \ \dots \ \Delta_M]^T, \quad \boldsymbol{\theta}_M \triangleq [k_1 \ \dots \ k_M b]^T \quad \text{and} \quad \boldsymbol{\theta}_{EM} \triangleq [c_0 \ \dots \ c_{n_x} \theta_0^T \ \dots \ \theta_n^T b]^T \quad (16)$$

and  $b$  is again a parameter accounting for the experimental friction offset.

#### 4.4. Parameter estimation

All three types of models [ $LG$ ,  $NLR(M)$  and  $DNLRX(M, n, n_x)$ ] are non-linear in the parameters. As a consequence, minimisation of the cost function  $\mathcal{J}$  (Eq. (8)) leads to a non-linear regression type of estimator for the corresponding unknown parameters. Within this study, the estimation of each model is based upon a two-phase, hybrid, non-linear optimisation scheme [19,20]. The first (*initial optimisation*) phase utilises a Genetic Algorithm (GA) [21] in order to explore large areas of the parameter space and locate regions where global or local minima may exist. The second (*fine-optimisation*) phase utilises the Nelder–Mead Downhill Simplex algorithm [21] for locating the exact global or local minima within the previously obtained regions. The Nelder–Mead algorithm makes use of cost function evaluations but not of derivatives, which, for the  $NLR$  and  $DNLRX$  cases, are not defined everywhere as the cost function is nonsmooth in the parameter space.

This two-phase scheme has been shown to be effective in locating the true global minimum of the cost function and circumvents problems associated with local minima, which are otherwise quite common (especially in connection with the  $NLR$  and  $DNLRX$  methods [19,20]).

#### 4.5. Identification results

The model training set was selected as the data from point 3550 to 18 549 and the validation set was composed of the data from point 18 550 to point 90 000 of the available displacement–friction force signals. However, the  $LG$  method required the velocity signal (Eqs. (3) and (4)) which was unavailable. The velocity signal was thus obtained via first-order differencing of a low-pass filtered version of the displacement signal. Regarding the  $NLR$  and  $DNLRX$  methods, the number  $M$  of the incorporated elements and the orders  $n$  and  $n_x$  of the FIR filters [see Eqs. (14) and (15)] should be selected. A way of doing this was by considering the Mean Square Error (MSE) of candidate  $NLR(M)$  and  $DNLRX(M, n, n_x)$  models within the aforementioned part of the data [19,20]. This procedure led to the selection of an  $NLR(3)$  and a  $DNLRX(5, 4, 2)$  model, respectively.

A comparative performance assessment of the LuGre,  $NLR(3)$  and  $DNLRX(5, 4, 2)$  models is presented in Fig. 7, in which the models MSE and MAX criteria, along with the corresponding model simulation error signals, are presented (these results correspond to the *test set*). As may readily be observed, all models provide an MSE smaller than 1.04%, indicating excellent fits. Comparing the three models, it is however noticed that the  $DNLRX(5, 4, 2)$  model achieves an MSE of 0.29%, which is significantly smaller than that of the LuGre model (1.04%) and the  $NLR(3)$  model (0.76%). The price paid for the improved performance of the  $DNLRX(5, 4, 2)$  model is its increased parametric complexity (that is the number of estimated parameters),

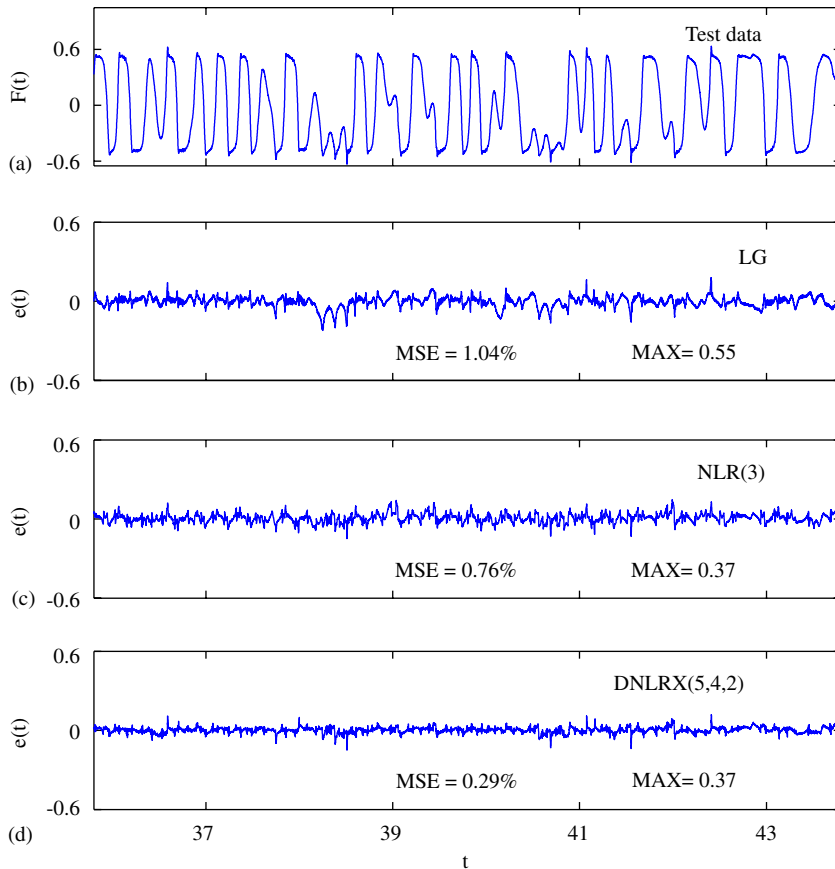


Fig. 7. Assessment of the “physics-based” methods: (a) Measured friction signal (following sample mean removal); (b) LG model-based model error; (c) NLR(3) model-based model error; (d) DNLRX(5,4,2) model-based model error (*Test Set*).

which is almost three and four times higher than that of the LuGre and *NLR*(3) model, respectively (34 parameters compared to 9 and 7, respectively).

## 5. Black-box modelling I: The University of Göttingen

### 5.1. NARX models

The discussion now turns to black-box models starting with NARX models which are a non-linear extension of the well-known ARX<sup>1</sup> models [23–26]. In the regression formalism the model form is described by some arbitrary non-linear function  $\hat{y}_i = f(\mathbf{x}_i|\boldsymbol{\theta})$  where the input vector  $\mathbf{x}_i$  includes past values of the time series  $\{u_i\}$  and values of the exogenous input  $\{w_i\}$ .

How the parameters  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_M)$  enter into the model function is not defined. Practical aspects advise to choose  $f$  as a linear sum of so-called *basis functions*  $\phi_j$

$$f(\mathbf{x}_i|\boldsymbol{\theta}) = \sum_{j=1}^M \theta_j \phi_j(\mathbf{x}_i). \quad (17)$$

<sup>1</sup>AutoRegressive with eXogenous input.

In this way, the parameters  $\theta_j$  contribute only quadratically to the sum of squared errors

$$V(\boldsymbol{\theta}) = \sum_{i=1}^N (y_i - f(\mathbf{x}_i|\boldsymbol{\theta}))^2, \quad (18)$$

reducing the least-squares fit to a simple convex minimisation problem with a well-defined solution.

Theoretically any function type can serve as basis function. Nevertheless it should meet two requirements. First of all the basis functions have to be sufficiently flexible and complex. A combination of them should be able to approximate the potentially complex relationship between the inputs  $\mathbf{x}_i$  and the outputs  $y_i$ . Obviously for non-linear regression this implies that the basis functions have to be also non-linear in some way.

The second requirement is contrary to the first and states that the basis functions should be as simple as possible. This is partly a matter of practicability as simple basis functions reduce computational efforts. However it also refines the control of the complexity of the model. Every basis function  $\phi_j$  with a corresponding non-zero coefficient  $\theta_j$  increases the complexity of the model by a small amount. Choosing an appropriate number  $M$  of basis functions effectively determines the trade-off between accuracy and complexity of the model (see the Forward Orthogonal Regression [23]).

In the models considered here, *monomials* are used for basis functions

$$\phi(\mathbf{x}) = \prod_{i=1}^d x_i^{p_i} \quad \text{for the input } \mathbf{x} \in \mathbf{R}^d \quad (19)$$

with the maximum degree  $p = \sum_i p_i$ . Other popular choices are *radial basis functions*, *rational functions*, and *wavelets*.

## 5.2. Local models

In contrast to the *global* models discussed so far, *local* models do not use any training data until queried with some point  $\mathbf{x}$ . A small neighbourhood of  $\mathbf{x}$  is located in the training set and a simple model using only the training points lying in this neighbourhood is constructed [27].

The most common choice for the neighbourhood is to locate the  $k$  nearest neighbours  $\mathbf{x}_{m_1}, \dots, \mathbf{x}_{m_k}$  of  $\mathbf{x}$  (*fixed mass*), i.e. the  $k$  points in the training set which have the smallest distance to the query point according to some arbitrary metric  $\|\cdot\|$  (usually Euclidean). To find the nearest neighbours a fast algorithm called *ATRIA* is used, which relies on a binary search tree built in a preprocessing stage [28].

The model used in the neighbourhood of the query point is usually fairly simple. A *locally constant* model computes a weighted average of the images of the nearest neighbours

$$\hat{f}(\mathbf{x}) = \frac{\sum_{i=1}^k w_i y_{m_i}}{\sum_{i=1}^k w_i}. \quad (20)$$

Besides their speed of computation, locally constant models are very robust, as their predictions always remain in the data range given by the nearest neighbours. The weights  $w_i$  are usually drawn from a monotonically decreasing weight function, so that the influence of the furthest nearest neighbours is decreased. Otherwise, the model output becomes discontinuous, as shifting the query point  $\mathbf{x}$  results in points suddenly entering or leaving the neighbourhood.

A *locally linear* model fits a linear function

$$\hat{f}(\mathbf{x}) = \mathbf{a}^T \cdot \mathbf{x} + a_0 = \tilde{\mathbf{a}}^T \cdot \tilde{\mathbf{x}} \quad (21)$$

(with  $\tilde{\mathbf{a}} = [\mathbf{a}; a_0]$  and  $\tilde{\mathbf{x}} = [\mathbf{x}; 1]$ ) in the neighbourhood of the query point by minimising the weighted sum of squared errors

$$V(\mathbf{a}, a_0) = \sum_{i=1}^k w_i^2 (y_{m_i} - \tilde{\mathbf{a}}^T \cdot \tilde{\mathbf{x}})^2. \quad (22)$$

The solution for  $\tilde{\mathbf{a}}$  is given by

$$\tilde{\mathbf{a}} = (\mathbf{X}_W^T \mathbf{X}_W)^{-1} \mathbf{X}_W^T \cdot \mathbf{y}_W = \mathbf{X}_W^\dagger \cdot \mathbf{y}_W, \quad (23)$$

where  $\mathbf{X}_W = \mathbf{W} \cdot \mathbf{X}$ ,  $\mathbf{y}_W = \mathbf{W} \cdot \mathbf{y}$ ,  $\mathbf{X} = [\tilde{\mathbf{x}}_{m_1}^T, \dots, \tilde{\mathbf{x}}_{m_k}^T]^T$ ,  $\mathbf{y} = [y_{m_1}, \dots, y_{m_k}]^T$  and  $\mathbf{W} = \text{diag}([w_1, \dots, w_k])$  [29]. The term  $\mathbf{X}_W^\dagger$  denotes the pseudoinverse of  $\mathbf{X}_W$ , which can be calculated using the singular value decomposition  $\mathbf{X}_W = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T$ , where  $\mathbf{S} = \text{diag}([\sigma_1, \dots, \sigma_k])$  with the singular values  $\sigma_i$ . The pseudoinverse is then given by  $\mathbf{X}_W^\dagger = \mathbf{V} \cdot \mathbf{S}^{-1} \cdot \mathbf{U}^T$ .

Locally linear models give usually more accurate estimations than locally constant ones, but they need an additional regularisation method to secure stability. One popular approach for regularisation is the *truncated principal component regression* (TPCR). During the calculation of the pseudoinverse  $\mathbf{X}_W^\dagger$  small singular values in the diagonal matrix  $\mathbf{S}$  are set to zero. This can be further improved by *soft thresholding*, where singular values lying in a specific interval  $[s_1, s_2]$  are smoothly weighted down to zero [29].

For locally linear models, four types of parameters must be chosen: the number of nearest neighbours  $k$  and the metric used to locate these, the weighting function for the weights  $w_i$ , and the regularisation parameters  $s_1, s_2$ . Additionally, one must also find good values for the embedding parameters. Instead of the usually chosen delay embedding with the two parameters dimension and delay, a non-uniform embedding is used, which allows varying delays between the components of the input vector.

For finding good parameter values, one can use a training procedure like *cross validation*. Here, the data set is split into a training and a test set which are used for training and validating the model, respectively. Local models allow an “extreme” form of this procedure, the *leave-one-out cross validation* (LOOCV), where the test set is reduced to one single point. Such an implementation is possible because local models are lazy learners, which delay any model calculations until they are queried. Of course, one has to repeat this validation procedure with enough different test points to get a good estimation of the actual model error.

A genetic algorithm which minimises the LOOCV error is used for optimising the delays of the embedding and the number of nearest neighbours.

For the other parameters a simple type of cyclic optimisation is used, where all parameters are successively optimised with a semi-global line search [29]. Although these optimisation procedures do not necessarily lead to the global minimum in parameter space, they are usually able to improve prediction accuracy compared to manually chosen parameters.

### 5.3. Recurrent neural networks

All methods introduced in the previous subsections require a delay vector as an input. An alternative are Recurrent Neural Networks (RNN) [30,31] that are trained to generate a specific output signal when driven by the corresponding input signal. The non-linear functional relation between input and output is here an emergent feature of generalised synchronisation [16] between the RNNs differ from memoryless models (like NARX, Local Models or feed-forward Neural Networks) that operate on vector-valued input and provide (static) functions between input–output pairs  $(x_i, y_i)$ . In order to use the latter for prediction purposes one has to introduce an external memory by the embedding technique described above. In contrast RNNs do not depend on this technique. They preserve information from the past in the states of their internal elements and can use it for dynamic tasks like prediction or classification.

The local dynamics of the elements in the RNN used here for friction modelling follow

$$x_i(t) = \alpha_i x_i(t-1) + \sigma(I_i(t)) \quad (24)$$

with the local input

$$I_i(t) = w_{i0} + \sum_{j \neq i} w_{ij} x_j(t-1) + v_i u(t), \quad (25)$$

where  $x_i(t)$  is the state of the  $i$ th element at time step  $t$ ,  $w_{ij}$  is the connection strength of the output from the  $j$ th element to the  $i$ th element,  $w_{i0}$  the bias of the  $i$ th element,  $v_i$  the connection strength of the external input signal  $u(t)$  to the  $i$ th element and  $\sigma(\cdot)$  a sigmoid activation function

$$\sigma(z) = \tanh(z). \quad (26)$$

There are many ways to define the topology of a RNN. For current purposes a constructive approach was used that included two phases, an optimisation phase and a growing phase. Starting with a network consisting

of 20 elements the parameters  $\alpha_i$ ,  $v_i$ , and  $w_{ij}$  were chosen randomly. The values for the weights of the internal connections  $w_{ij}$  were set sparse. That means that only about 10% of the connections had a non-vanishing value. This percentage of connectivity was held at the same level throughout the whole construction of the network.

Following the random initialisation, the parameters were adjusted in the optimisation phase. For the optimisation purpose a non-gradient-based algorithm was used, which was a mixture of simulated annealing and the Nelder–Mead simplex search. The optimisation was repeated until it showed no improvements on the validation set. After the optimisation phase the size of the network was increased in the growing phase. Five extra elements were added to the network with random connections to the old structure. Then again the optimisation phase was started followed by a growing phase, and so on. The whole procedure was stopped when no significant improvements on the validation set were achieved between two growing phases.

The output of the network was represented by a linear superposition of the elemental states

$$\hat{y}(t) = \sum_{i=1}^N \theta_i x_i(t) + \theta_0,$$

where the weights  $\theta_i$ ,  $i = 0, \dots, N$  were computed on the training data set with the usual least squares method as already described for NARX Models.

#### 5.4. Identification results

##### 5.4.1. NARX model

The NARX model (Section 5.1) is a polynomial of degree five (including 84 monomials) with a five-dimensional input vector  $\mathbf{x}(t) = (P(t), P(t - 16), P(t - 66), P(t - 67), F(t - 19))$  and output  $y(t) = F(t)$ . The prediction accuracy obtained with this method is shown in Fig. 8 (NMSE = 0.5%).

##### 5.4.2. Local models

Fig. 9 shows the prediction results for a locally linear model (LM) based on 260 nearest neighbours and a linear weight function. For regularisation, a truncated principal component regression with soft thresholding was used.

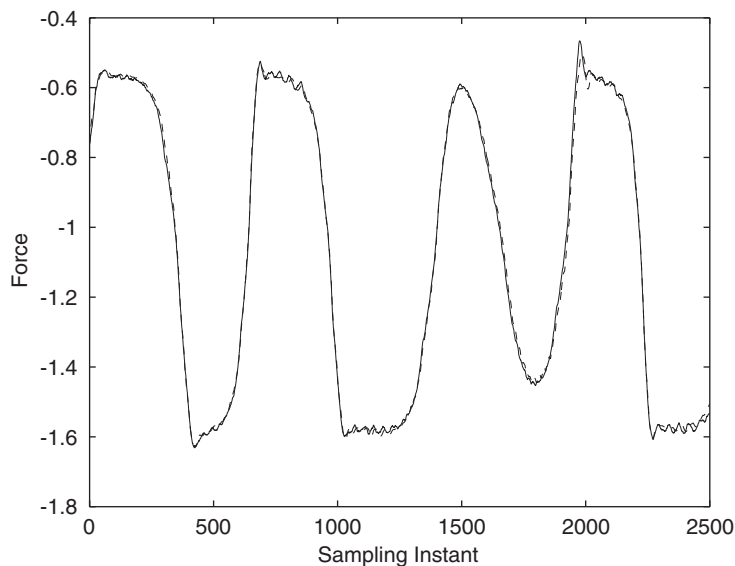


Fig. 8. Actual (solid line) and estimated (dashed) friction force using NARX model.

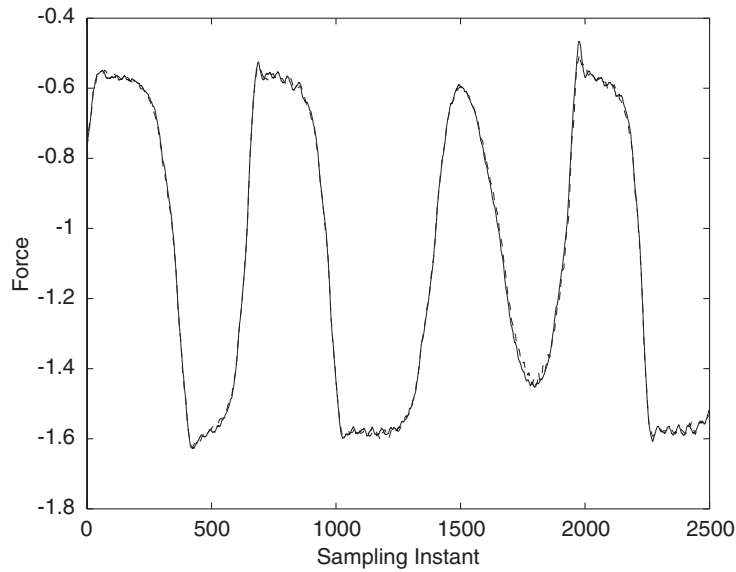


Fig. 9. Actual (solid line) and estimated (dashed) friction force using local model.

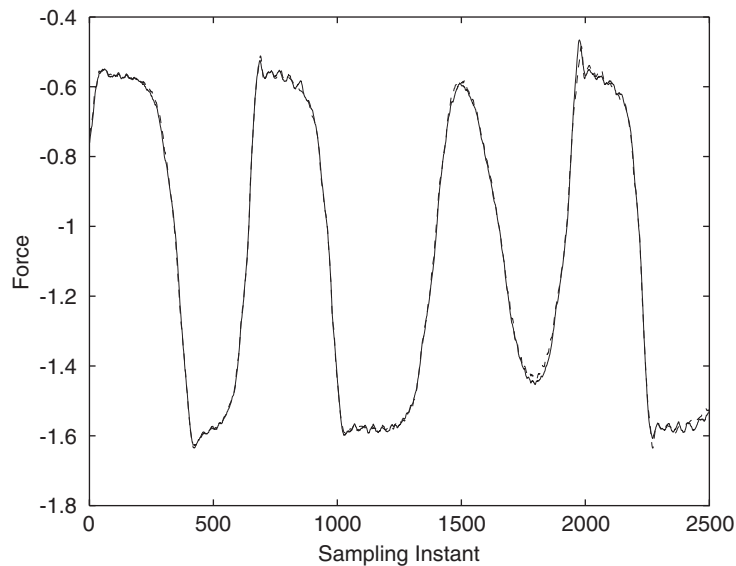


Fig. 10. Actual (solid line) and estimated (dashed) friction force using recurrent neural net model.

The genetic optimisation of the non-uniform embedding leads to a five-dimensional input vector  $\mathbf{x}(t) = (P(t), P(t - 16), P(t - 66), P(t - 67), F(t - 19))$ , consisting of four components from the position and one component from past force values (NMSE = 1.01%).

#### 5.4.3. Recurrent neural networks

Prediction results using a Recurrent Neural Network (RNN) (Section 5.3) are shown in Fig. 10. The final RNN consisted of 45 elements with 225 internal connections, which is a level of connectivity of about 11%. Three inputs were used, the original input signal, a lagged version with a delay of 80 samples and another lagged version with a delay of 160 samples. The prediction accuracy obtained with this method is very high with a MSE of 0.12%.

## 6. Black-box modelling II: The University of Sheffield

### 6.1. Generalised shunting neural network

The *shunting inhibition* mechanism has been used extensively to model visual and cognitive processes. It came about as a plausible explanation in sensory neural information processing [32]. Basically, the shunting inhibition mechanism describes how an excitatory signal into a cellular system is mediated by other signals in the system to deal with the problem of noise and saturation. The cells in this case would be neurons of an artificial neural network.

In the shunting inhibitory network, there exist two ways for information (all the excitatory signals) to arrive to a single neuron; either the feedforward or recurrent path. In the feedforward method, all the input signals are passed into each neuron directly. In the recurrent method, each input signal would only pass through a single neuron, after which the neuron would send signals to the other neurons. The activity of each neuron for the feedforward case can be written in a non-linear differential equation

$$\frac{dx_j}{dt} = I_j - a_j x_j - g \left( \sum_i c_{ji} I_i + c_{j0} \right) x_j + b_j, \quad (27)$$

where  $x_j$  is the output of the  $j$ th neuron;  $I_j$  and  $I_i$  are the inputs to the  $i$ th and  $j$ th neuron, respectively,  $a_j$  is the passive decay rate of the neuron (positive constant),  $c_{ji}$  is the connection weight from the  $i$ th input to the  $j$ th neuron,  $b_j$  represents the bias, and  $g$  is an activation function. For the recurrent case, the inputs  $I_i$  are replaced by  $x_i$ , the activity of the other neurons in the network. Bouzerdoun and colleagues [33] utilised the feedforward method in deriving a static version of the shunting inhibitory network, leading to the so-called SIANN (Shunting Inhibitory Artificial Neural Network). The static version enables the optimisation of the model using backpropagation-type methods without the numerical solution of Eq. (28). This was done by acquiring the steady-state solution of

$$x_j = \frac{I_j + b_j}{a_j + g(\sum_i c_{ji} I_i + c_{j0})}. \quad (28)$$

The state of a static shunting neuron of SIANN would then be explained by Eq. (29). However, it can be seen from Eq. (29) that there would be as many neurons in the network as there are the number of inputs. This might result in a large network that is difficult to train. In order to solve this problem, a modified neuron was proposed, the Generalised Shunting Neuron (GSN) [34].

In the GSN, all the excitatory input is summed and passed through an activation function similarly to a perceptron neuron. The state of a GSN is therefore given as

$$x_j = \frac{b_j + f(\sum_i w_{ji} I_i + w_{j0})}{a_j + g(\sum_i c_{ji} I_i + c_{j0})}, \quad (29)$$

where  $f$  is an activation function,  $b_j$  the bias and  $w_{ji}$  and  $w_{j0}$  the connection weights. The GSN is actually a superset of both the static shunting neuron and the perceptron neuron of a MLP (Multi-Layered-Perceptron) network. The network constructed using the GSN is therefore a superset of both the SIANN and the MLP.

A wide variety of activation functions can be implemented in the GSNN. However for this preliminary study, a restriction is made to that of a hyperbolic tangent numerator function, with a denominator function of either a logarithmic sigmoid, an exponential or the hyperbolic tangent. The structure of the GSNN is also restricted to a 3-layer format with a linear output function, utilising between 1 to 3 neurons. Memory is introduced into the system externally via the NARX (Nonlinear AutoRegressive with eXogeneous inputs) formulation. The output of the GSNN is given as

$$y = \sum_j d_j x_j + d_0, \quad (30)$$

where the inputs to the neurons are the original inputs, the delayed inputs (exogeneous inputs), and the delayed outputs (autoregressive terms) of the model. The original inputs in this case would be

the displacement, velocity and acceleration (the derivatives being calculated using the stable Al-Alaoui IIR filters [35]).

### 6.2. Optimisation routine

Optimisation of ANNs is usually performed using gradient descent-based techniques. One of the most robust techniques is the second-order-based Levenberg–Marquardt (LM) algorithm [36]. A variation of the algorithm, the Optimised Levenberg–Marquardt with Adaptive Momentum (OLMAM) is used [37] utilising simple weight decay [38]. One of the differences between the LM and the OLMAM algorithm is that the change in weights is constrained within a hyperellipse. Conjugacy is also maintained between all the subsequent descent directions, where only descent directions that reduce the objective function sufficiently are accepted. The iterative update rule of OLMAM with weight decay is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\lambda_1}{2\lambda_2} (\mathbf{J}_t^T \mathbf{J}_t + (\mu_t + m))^{-1} (\mathbf{J}_t^T \mathbf{E}_t - m\mathbf{w}_t) + \frac{1}{2\lambda_2} (\mathbf{w}_t - \mathbf{w}_{t-1}), \tag{31}$$

where  $\mathbf{w}$  is the weight vector,  $\mathbf{J}$  is the Jacobian (the partial derivative matrix of the network error with respect to the weights),  $\mu$  is an adaptive parameter that reflects the degree of ‘trust’ the LM algorithm employs in using second-order information.  $\lambda_1$  and  $\lambda_2$  are adaptive hyperparameters,  $m$  is a small parameter that penalises large weights,  $\mathbf{I}$  is a unit identity matrix and  $\mathbf{E}$  is the network error. The subscript  $t$  denotes the current iteration.

The optimisation of the NARX GSNN is slightly more complicated than the NARX MLP model. First of all, the network would be unstable if the denominator function is less than zero. In order to counter this problem, a hard constraint penalty function was used in the training routine. This penalty function enforces an additional criterion when optimising via

$$a_j + g \left( \sum_i c_{ji} I_i + c_{j0} \right) > p, \tag{32}$$

where  $p$  is the lower bound limit for the denominator activation function.

The Jacobian matrix required by the OLMAM algorithm can be easily derived using extensive use of the chain rule of differentiation, just as in the case of standard backpropagation. The Nonlinear Output Error (NOE) method considers the fact that the Jacobians are functions of previous errors as well, caused by the autoregressive terms. The teacher-forced technique (standard backpropagation utilising measured data for the autoregressive terms) ignores this, as all the autoregressive terms would be fixed. The Jacobian computed by the NOE is given as

$$\frac{\partial E(t)}{\partial \mathbf{w}} = \frac{\partial K(t)}{\partial \mathbf{w}} + \sum_{i=1}^{n_y} \left[ \frac{\partial K(t)}{\partial E(t-i)} \right]^T \frac{\partial E(t-i)}{\partial \mathbf{w}}, \tag{33}$$

where  $\partial K(t)/\partial \mathbf{w}$  is the Jacobian calculated using the teacher-forced technique.

In an ideal situation, the gradients should be calculated from the time  $t$  all the way back to the first observation in the training data. The NOE method only approximates the Jacobian by truncating the gradient calculations in a window bounded by  $n_y$ . It would be impractical to calculate the Jacobian for the ideal situation, especially for large datasets. Even if one could calculate the gradients far into the previous time steps, the NARX model would still suffer from the problem of vanishing gradients [39]. Bengio and colleagues [40], have shown that if a model is to robustly latch onto information, (i.e. to model long term dependencies in the presence of noise), the contribution of information far in the past to the gradients would be insignificant; hence the vanishing gradients problem. Therefore, the NOE method still might not be able to capture long-term dependencies well.

The GSNN optimisation is initialised using the teacher-forced technique first and then trained using the NOE technique. This is in done in order to increase the convergence rate and to reduce the problems of instability that the NOE technique might encounter. The instability can sometimes occur due to the feedback nature of the NOE optimisation technique.

### 6.3. Identification results

A wide variety of activation functions can be implemented in the GSNN. However for this preliminary study, the networks have been restricted to that of a hyperbolic tangent numerator function, with a denominator function of either a logarithmic sigmoidal, an exponential or the hyperbolic tangent. The structure of the GSNN is also restricted to that of a 3-layer format with a linear output function, utilising between 1 and 3 neurons. The autoregressive order was constrained between 1 and 6 and the delays of the exogenous inputs between 1 and 3. An exhaustive search conducted within this constrained region was used to produce the final optimised architecture of the GSNN (4 autoregressive order, no delays of the inputs, 3 neurons with hyperbolic tangent numerator and logarithmic sigmoidal denominator). The prediction accuracy produced through the model was 0.35% MSE. The normalised maximum errors are 0.26.

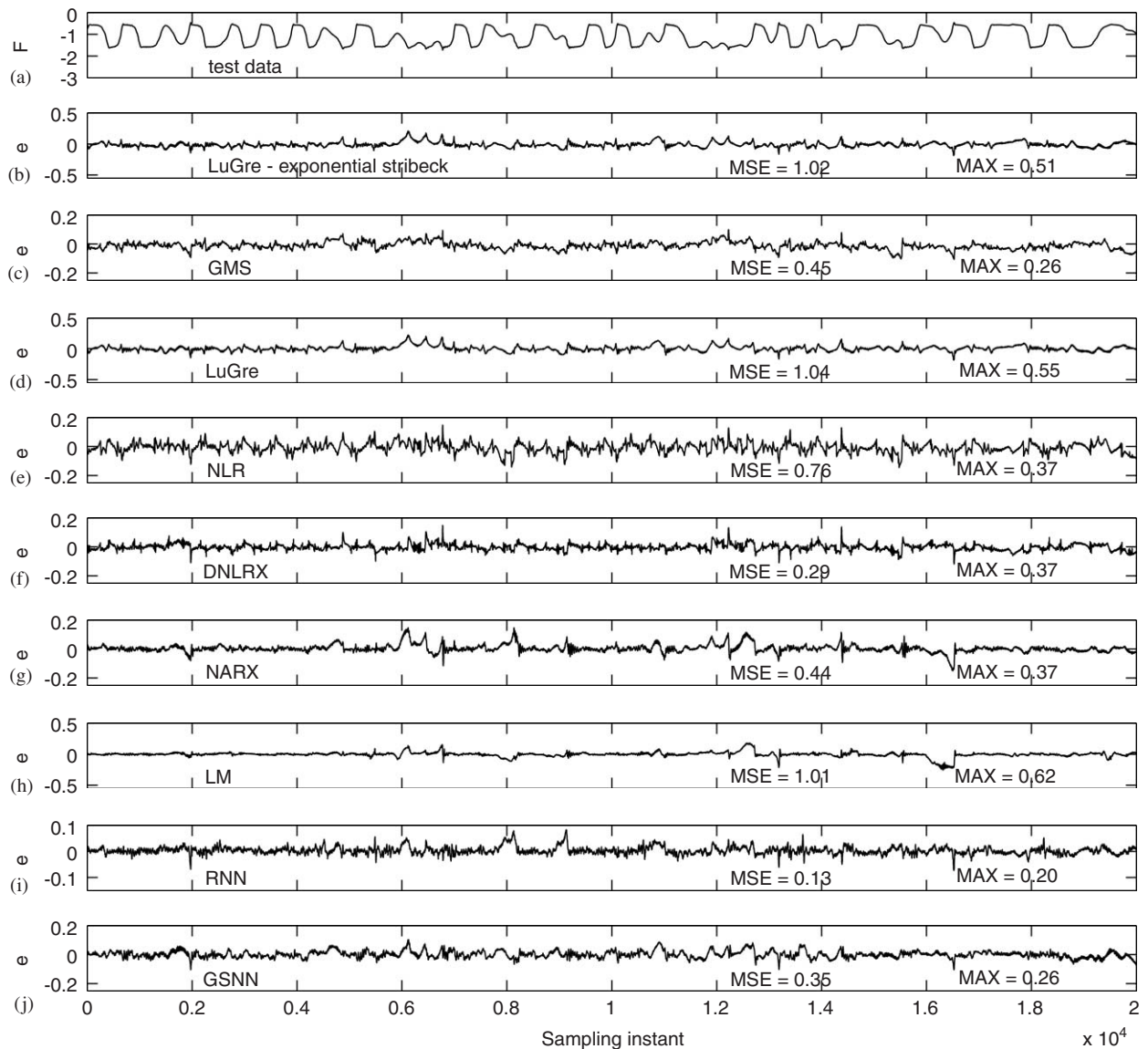


Fig. 11. Summary of model residuals over test set.

### 7. Summary and ensemble modelling

The overall results of the identification exercise are summarised in Fig. 11, which gives the residual errors for each model type, together with its MSE and MAX errors. A comparison between the worst and the best individual models is given in Fig. 12. All the individual models are accurate for a wide range of the data.

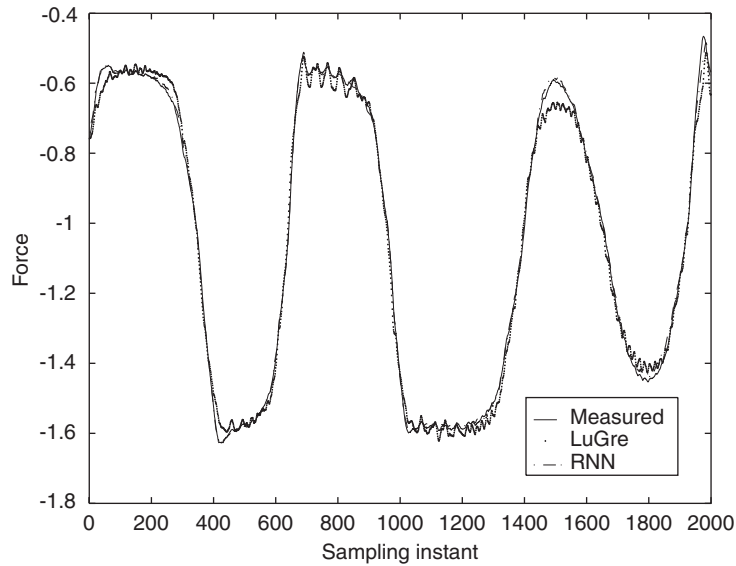


Fig. 12. Comparison showing best and worst individual models.

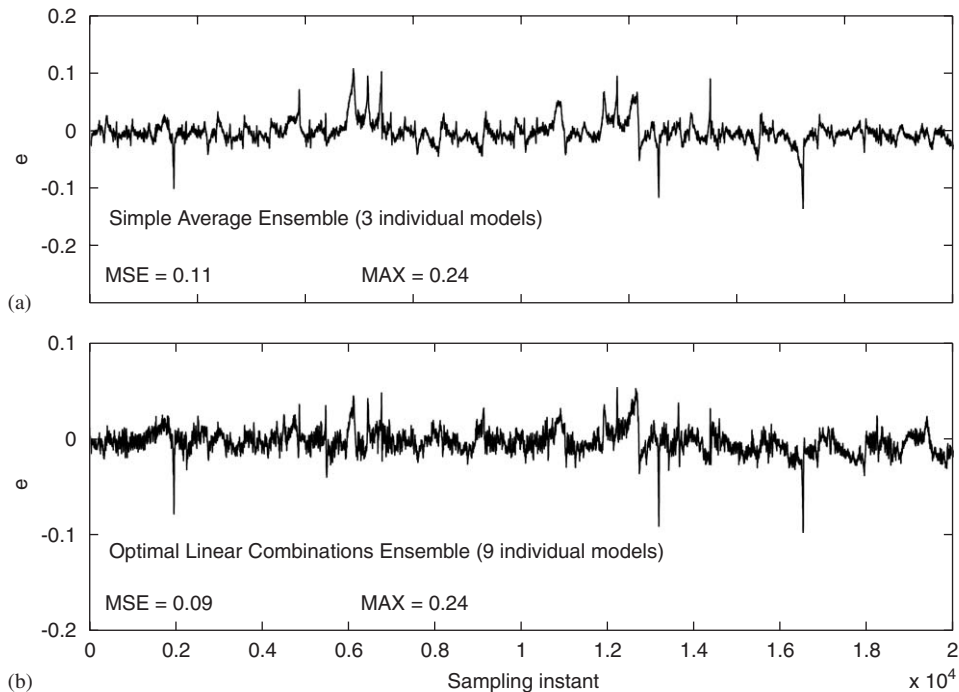


Fig. 13. Residuals for ensemble models computed on first 10000 points of test set.

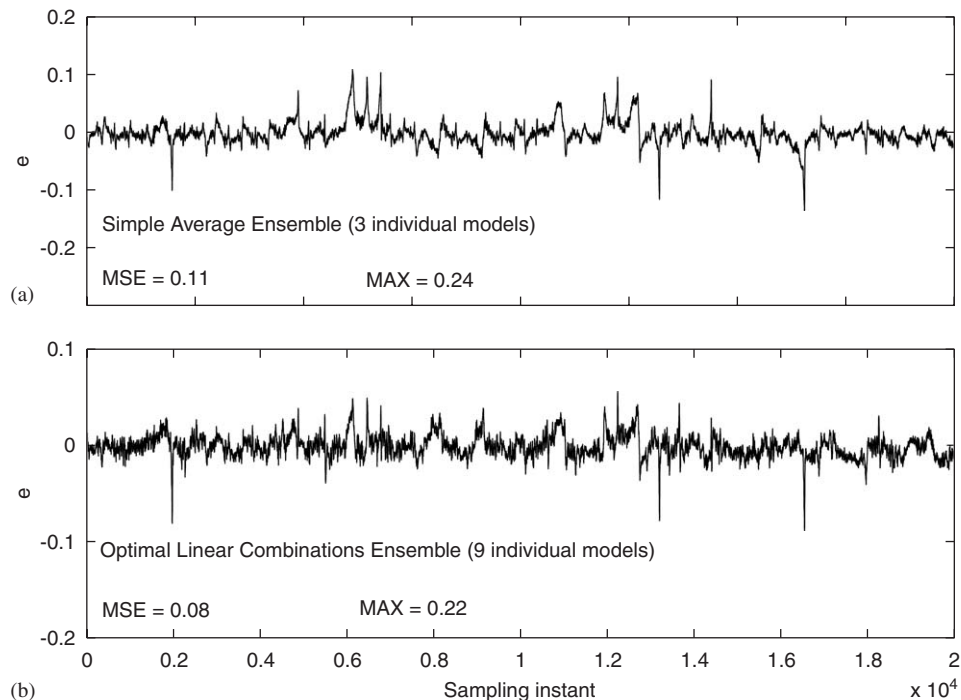


Fig. 14. Residuals for ensemble models computed on all 20 000 points of test set.

Accuracy of the model is mostly determined by how well the models are able to capture dependencies at reversal points for the force output.

Residuals of ensemble models utilising the simple average and the optimal linear combinations technique can be seen in Fig. 13. The simple average ensemble is formed using the DNLRX, the RNN (recurrent networks) and the GSNN (acquired by using an exhaustive search). Note that the maximum errors actually increased compared to the simple RNN. In fact, the RNN by itself, is better than any ensemble model in terms of minimal maximum error. The optimal linear combination was formed using the PRESS statistic based on Orthogonal Forward Regression (PRESS-OFR). The PRESS-OFR selects basis functions (the individual models in this case) in a stepwise manner, leading to linearly independent basis functions that would minimise the leave-one-out cross-validation error of the training set [41]. The 1st 10 000 points were used for training and then the outputs calculated for the whole range of 20 000 points (Fig. 13, this also includes the results for an ensemble of all 9 models). It can be argued that the ensemble model formed using all 20 000 points for training is valid as the PRESS-OFR technique minimises the leave-1-out cross validation error. Such a model is shown in Fig. 14.

## 8. Discussion and conclusions

The identification results presented in this paper are uniformly very good and may well be adequate for control purposes. It is interesting to note that the best model, the recurrent neural network is of the black-box type and does not incorporate any a priori physical knowledge. It is less surprising that the next best model is the physics-based DNLRX. In terms of the parsimony of the models, perhaps the DNLRX is to be preferred. It is not impossible that a black-box model equal or outperform a grey-box model; however two conditions must be fulfilled: (a) the grey-box model cannot incorporate all relevant physics and (b) the black-box model should have an appropriately rich basis to approximate from. The authors believe that both conditions are met here. The LuGre, GMS and DNLRX models clearly do not incorporate *all* known physics, and the

recurrent neural network and GSNN are known to have an appropriately rich basis from their universal approximation theorems.

When the models are combined into an ensemble (with the GSNN model added) it is possible to derive small but important improvements in performance, although this is only observed on the MSE measure; in terms of the MAX error, the RNN model is superior. The choice of an ensemble over an individual model here therefore rests on the question of whether mean performance is more important than worst-case performance.

This paper has presented an overview of a wide range of models used to identify pre-sliding and gross sliding friction and the transition regime. Because friction is always present in machines, it imposes a limit on the precision with which machines can be controlled. The attainable precision can be increased with a greater understanding of the complex non-linear and hysteretic nature of the friction force. This paper has shown that such an understanding is possible through the use of highly accurate testing facilities and state-of-the-art methods of system identification. The next step in the work is to use the models obtained in a control framework.

## Acknowledgements

The authors acknowledge the Volkswagenstiftung for financial support under Grant no. I/76938.

## References

- [1] B. Armstrong-Helouvry, P. Dupont, C. Canudas de Wit, A survey of models, analysis tools and compensation methods for the control of machines with friction, *Automatica* 30 (1994) 1083–1183.
- [2] H. Olsson, Control system with friction, Ph.D. Thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, 1996.
- [3] J. Swevers, F. Al-Bender, C.G. Ganseman, T. Prajogo, An integrated friction model structure with improved presliding behavior for accurate friction compensation, *IEEE Transactions on Automatic Control* 45 (2000) 675.
- [4] M.R. Popovic, D.M. Gorinevsky, A.A. Goldenbery, Accurate positioning of devices with nonlinear friction using fuzzy logic pulse controller, *Proceedings of International Symposium on Experimental Robotics*, 1995, pp. 331–342.
- [5] S. Yang, M. Tomizuka, Adaptive pulse width control for precise positioning under the influence of stiction and Coulomb friction, *ASME Journal of Dynamic Systems, Measurements and Control* 110 (1988) 211–227.
- [6] P.R. Dahl, A Solid Friction Model, vol. TOR-158, The Aerospace Corporation, El Segundo, CA, 1968.
- [7] C. Canudas de Wit, H. Olsson, K.J. Aström, P. Lischinsky, A new model for control of systems with friction, *IEEE Transactions on Automatic Control* 40 (1995) 419–425.
- [8] B. Eriksson, Optimal force control to improve hydraulic drives, Licentiate Thesis, Damek Research Group, Department of Machine Design, Royal Institute of Technology, KTH, Sweden, 1996.
- [9] V. Lampaert, J. Swevers, F. Al-Bender, Modification of the Leuven integrated friction model structure, *IEEE Transactions on Automatic Control* 47 (2002) 683.
- [10] J. Sjöberg, Q. Zhang, L. Ljung, A. Beneviste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, A. Juditsky, Nonlinear black-box modelling in system identification: a unified overview, *Automatica* 31 (1995) 1691–1724.
- [11] Y. Fujii, W.E. Tobler, T.D. Synder, Prediction of wet band dynamic engagement Part 1: Mathematical model development, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 215 (2001) 479–492.
- [12] M. Cao, K.W. Wang, Y. Fujii, W.E. Tobler, A hybrid neural network approach for the development of a friction component dynamic model, *ASME Journal of Dynamic Systems, Measurements and Control* 126 (2004) 144–153.
- [13] M. Cao, K.W. Wang, Y. Fujii, W.E. Tobler, Advanced hybrid neural network (AHNN) automotive friction component model for powertrain system dynamic analysis, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 218 (2004) 845–857.
- [14] F. Al-Bender, V. Lampaert, J. Swevers, A novel generic model at asperity level for dry friction force dynamics, *Tribology Letters* 16 (2004) 81–93.
- [15] U. Parlitz, A. Hornstein, D. Engster, F. Al-Bender, V. Lampaert, T. Tjahjowidodo, S.D. Fassois, D.D. Rzos, C.X. Wong, K. Worden, G. Manson, Identification of pre-sliding friction dynamics, *Chaos* 14 (2004) 420–430.
- [16] H. Nakazawa, *Principles of Precision Engineering*, Oxford University Press, Oxford, 1991.
- [17] V. Lampaert, F. Al-Bender, J. Swevers, Experimental characterisation of dry friction at low velocities on a developed tribometer set-up for macroscopic measurements, *Tribology Letters* 16 (2004) 95–105.
- [18] D.D. Rzos, S.D. Fassois, Presliding friction identification based upon the Maxwell Slip model structure, *Chaos: An Interdisciplinary Journal of Nonlinear Science* 14 (2004) 431–445.
- [19] D.D. Rzos, S.D. Fassois, Friction identification based upon the LuGre and Maxwell-slip models, *IFAC World Congress 2005*, Prague, Czech Republic, 2005.

- [21] O. Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*, Springer, Berlin, 2001.
- [23] S.A. Billings, et al., Identification of MIMO non-linear systems using a forward-regression orthogonal estimator, *International Journal of Control* 6 (1989) 2157–2189.
- [24] L.A. Aguirre, S.A. Billings, Retrieving dynamical invariants from chaotic data using NARMAX models, *International Journal of Bifurcation and Chaos* 2 (1995) 449–474.
- [25] S.A. Billings, D. Coca, Discrete wavelet models for identification and qualitative analysis of chaotic systems, *International Journal of Bifurcation and Chaos* 7 (1999) 1263–1284.
- [26] L.A. Aguirre, et al., Nonlinear multivariable modeling and analysis of sleep apnea time series, *Computers in Biology and Medicine* 29 (1999) 207–228.
- [27] C.G. Atkeson, A. Moore, S. Schaal, Locally weighted learning, *Artificial Intelligence Review* 11 (1997) 11–73.
- [28] C. Merkwirth, et al., Fast nearest-neighbor searching for nonlinear signal processing, *Physical Review E* 62 (2000) 2089–2097.
- [29] J. McNames, *Innovations in local modeling for time series prediction*, Ph.D. Thesis, Stanford University, 1999.
- [30] H. Jaeger, The echo state approach to analysing and training recurrent neural networks, GMD Report German National Research Center for Information Technology, vol. 148, 2001.
- [31] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: a new framework for neural computation based on perturbations, *Neural Computation* 14 (2002) 2531–2560.
- [32] G.G. Furman, Comparison of models for subtractive and shunting lateral-inhibition in receptor-neuron fields, *Kybernetik* 2 (1965) 257–274.
- [33] A. Bouzerdoun, A new class of high-order neural networks with nonlinear decision boundaries, *Proceedings of the Sixth International Conference on Neural Information Processing (ICONIP'99)*, Perth, Australia, 1999, pp. 1004–1009.
- [34] G. Arulampalam, A. Bouzerdoun, A generalized feedforward neural network architecture for classification and regression, *Neural Networks* 16 (2003) 561–568.
- [35] M.A. Al-Alaoui, Novel digital integrator and differentiator, *IEEE Electronics Letters* 29 (1993) 376–378.
- [36] D.M. Bates, D.G. Watts, *Nonlinear Regression and Its Applications*, Wiley, New York, 1988.
- [37] N. Ampazis, S.J. Perantonis, Two highly efficient second-order algorithms for training feedforward networks, *IEEE Transactions on Neural Networks* 13 (2002).
- [38] A. Krogh, J.A. Hertz, A simple weight decay can improve generalisation, *Advances in Neural Information Processing Systems* 4 (1992) 951–957.
- [39] P. Frasconi, M. Gori, G. Soda, Local feedback multilayered networks, *Neural Computation* 4 (1992) 120–130.
- [40] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, *IEEE Transactions on Neural Networks* 5 (1994) 157–166.
- [41] S. Chen, X. Hong, C.J. Harris, P.M. Sharkey, Sparse modelling using orthogonal forward regression with PRESS statistic and regularization, *IEEE Transactions on System, Man, and Cybernetics, Part B* 34 (2004) 898–911.