PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 47, NO. 3–4, PP. 253–267 (2003)

NON-LOCAL HYSTERESIS FUNCTION IDENTIFICATION AND COMPENSATION WITH NEURAL NETWORKS

Péter BERÉNYI[*] Gábor HORVÁTH[*], Vincent LAMPAERT[**] and Jan SWEVERS[**]

[*]Department of Measurement and Information Systems
Budapest University of Technology and Economics
H–1521 Budapest, Hungary
[**]Department of Mechanical Engineering, Division PMA
Katholieke Universiteit Leuven
Leuven, Belgium

Received: Oct. 20, 2003

Abstract

This paper discusses the on-line identification of non-local static hysteresis functions, which are encountered in mechanical friction, magnetic materials, and piezoelectric actuators and cause problems by the design of controllers. In this article we want to introduce a compensation method for friction in presliding regime, based on the simplified Leuven Friction Model and on technology borrowed from neural networks. We present a solution how to identify the hysteresis caused by the friction, and how to use this identified model for the compensation of the friction effects. Results from both simulations and experiments will be shown.

Keywords: hysteresis, neural networks, identification, friction compensation.

1. Introduction

Hysteresis phenomena appear in many systems of engineering interest, such as piezoelectric actuators [1, 2] structures being stressed beyond their elastic limit, magnetic materials in the presence of alternating electromagnetic fields, and friction force as a result of micro displacements [3, 4, 5]. Accurate control of systems with hysteresis requires a model of this non-linear phenomenon. According to MAYERGOYZ [6], hysteresis non-linearities can be classified into two categories:

- non-linearities with local memory where the future output depends only upon the future input and the present output,
- non-linearities with non-local memory where the future output does not only depend upon the current output and the future input but also on the past extreme values of the input. This type of effect can be observed in magnetic and piezoelectric materials and also in friction.

The modelling of the second type of hystereses presents more difficulties because it requires a solution that is able to store all extreme values of the input even through a long time period.

This paper describes a model for static hysteresis functions with non-local memory, where the term static indicates that the speed of the input variations has no influence on the branch of the hysteresis curve. The Preisach model [7] is a well known approach to model the considered hysteresis functions. The Preisach model has two important properties, which must be valid for the hystereses being modelled: the congruency and the wiping-out properties. The first is about the similarity of the loops of the hysteresis, and the second ensures that the loops are closed and the input history of the hysteresis is overwritable. The Preisach model is a good generic theoretical hysteresis model, however, it requires an accumulation of the past extreme values of the input, which needs in a practical implementation a dynamic allocation facility and the model parameters are also difficult to compute.

Another modelling approach is developed by KRASNOSELSKI and POKROVSKI [8]. The model is based on a weighted superposition of many elementary hysteresis operators, which differ in one or more parameters depending on the operator type. A small amount of elementary operators suffice for a quite accurate modelling of arbitrary hysteresis functions due to the continuity of the operators. KUHNEN et al. [5] use the so called linear stop operator (LSO). LAMPAERT et al. [9] use an extended version of the linear stop operator, called the extended linear stop operator. The model described in this paper uses a simplified version of the original Preisach model, as explained in Section 2. The models in this second group are equivalent regarding their modeling capabilities which correspond to a subset of the hystereses that are modellable with the Preisach model.

One way to identify the parameters of a hysteresis model is curve fitting [10], which has a drawback that it is an off-line procedure. KUHNEN et al. use a gradient estimator technique which is an on-line identification method. LAMPAERT et al. use the recursive least squares (RLS) method for estimation. This paper uses neural approximation as explained in Section 2. This is not the first example of using neural networks for hysteresis approximation, WEI and SUN in [11] describe a similar approach, however, they approximate the continuous weight function of the Preisach model with the neural network, while KUCZMANN in [12] and [13] describes an approximation method for magnetic hystereses using multi-layer perceptrons.

Simulation and experimental results for static systems are presented in Section 3 and 4. The on-line identification methods (like the gradient estimator) are preferable against the off-line procedures because they allow the adaptation of the generated models during the operation of the system in which the model is used. That allows a quicker reaction to the changing operational environment.

2. Neural Hysteresis Model

The neural hysteresis modelling system is based on a simplified version of the original PREISACH model [7]. The main difference is that this model uses a one-dimensional weight function instead of the two-dimensional one of the Preisach model. A further difference is that the weight function is not continuous, it is approximated by a step-like function. (See *Fig.* 1 for an example of weight functions. The weights are defined as a function of the differences of the input values u – the input distances – and will be determined during the identification process.)

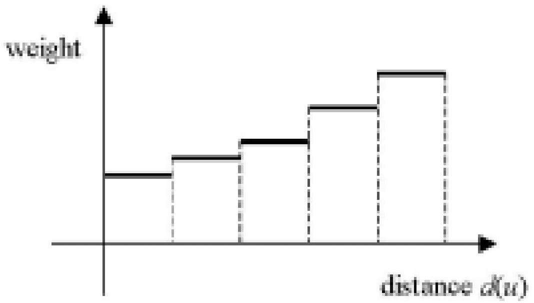

Fig. 1. An example weight function

Using the weights defined by this weight function as the weights of a simple neural network, an easy-to-calculate, elementary neural hysteresis model can be constructed, which can model only symmetric hystereses (not due to the step function but due to the 1D function). For the current modelling task this should prove satisfactory, and using a combination of these simple hystereses asymmetric models can also be constructed.

The hysteresis model is actually a one-layer neural network with N inputs (see *Fig.* 2) combined with a special non-linear transformation that maps the hysteresis input values to the input values of the neural network, and allows the storage of previous extreme values infinitely long.

As it can be seen from the figures above, in the following we will have to distinguish the input of the whole model, i.e. the position signal, and the input of the neural network that is actually the model input after the non-linear transformation.

The input range of the hysteresis to be modelled is divided into N different intervals equal to the number of weights (see *Fig.* 2). (The input range of the hysteresis is the interval between the first full saturation points in positive and negative directions.) The weights of the neural network are assigned to selected intervals of the input range. In the example above, the intervals are equally distributed over the range which is a simple but not necessary solution. Other interval selections allow, for example, the changing of approximation precision, or better approximation with fewer weights using some a priori knowledge about the properties of the hysteresis being modelled. Each interval i got by this division has one state x_i (this will be called activation). The input of the neural network consists of a vector built from these activation values. Let us now introduce the first part of the neural hysteresis model, the non-linear mapping.

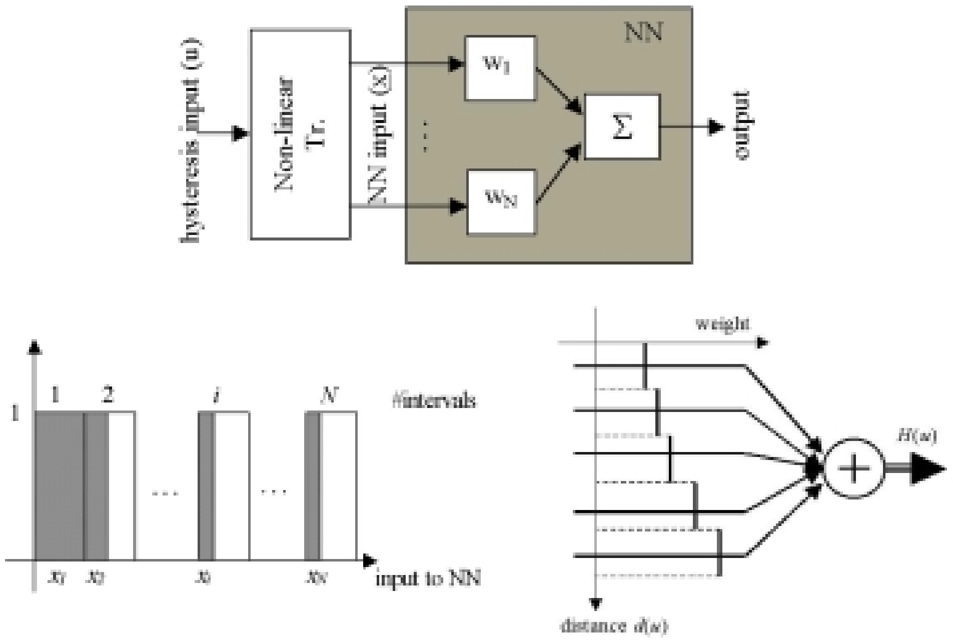

Fig. 2. The neural model structure

2.1. Calculation of the Activations

The activations are calculated with a non-linear mapping from the input of the hysteresis, i.e. from the position signal. This section describes how this non-linear transformation works.

The non-linear transformation needs the storage of a few state variables, such as the previous input $[u(k-1)]$ and the previous activation values. The non-linear transformation will map the values of the real input into a distance based system.

To calculate the next values of the activation values – the real input $[u(k)]$ of the hysteresis is first converted into a distance $[d(k)]$ and a direction $[\text{dir}\,(k)]$ using the stored value of the previous input $[u(k-1)]$. The distance is

$$d(k) = |u(k) - u(k-1)| \tag{1}$$

and the direction is

$$\text{dir}\,(k) = \text{sgn}\,[u(k) - u(k-1)]. \tag{2}$$

For further explanation see *Fig.* 3.

These variables are used to calculate the activation values according to the following method.

Let us assume that the system was started with a positive input. If the actual input moves the system in the same positive direction, as before, the calculated

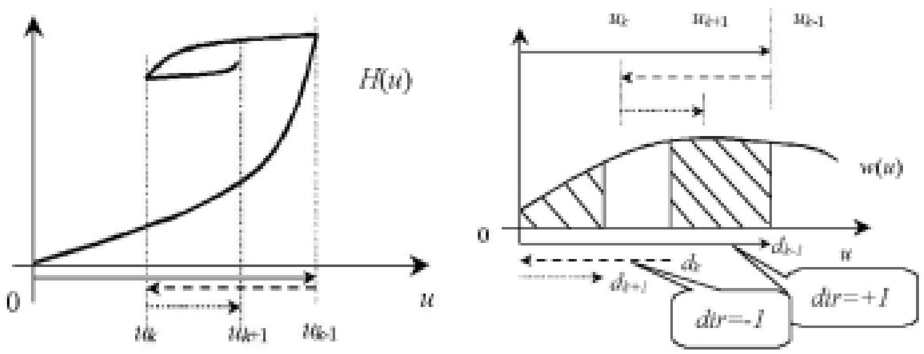

Fig. 3. a.) The input-output relaion of a hysteretic system, b.) The weight function

distance will be added to the interval used last time. Here we compare the current distance with the part of the interval not activated until now, starting with the first interval, activate as many as possible and subtract the size of the part activated in this step from the distance. If the distance is still bigger than zero, the same procedure will be carried out on the next interval until all the distance is used to activate the model or the model is fully saturated. (*Fig.* 4).

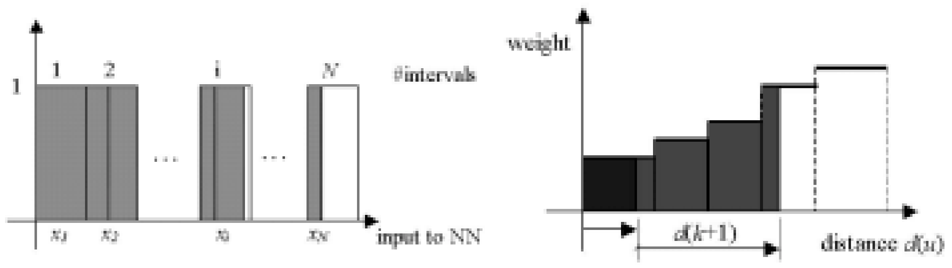

Fig. 4. The activation of the intervals in positive direction

If the current direction of the system is negative, the process differs from the above-mentioned algorithm only in the fact that the distance will be used to deactivate the intervals.

If the input changes the direction, the calculated distance will be used from the first interval in the same way as described before. So a positive distance will be used to activate, and negative distance to deactivate the intervals.

In a more mathematical way, the calculation of the weight activation for the different cases can be given as follows:

$$x_i(k) = \min\left\{ x_i(k-1) + \max\left[d(k) - \sum_{j=0}^{i-1}(a_j - x_j(k-1)), 0 \right], a_i \right\} \tag{3}$$

if the actual direction is positive, where a_j is the size of the working interval of the

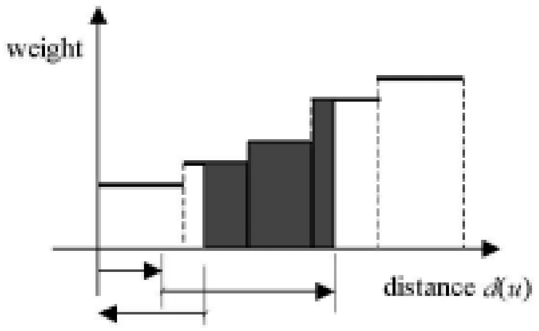

Fig. 5. Deactivating the intervals after a direction change

jth weight. If the actual direction is negative the formula has the following form:

$$x_i(k) = \max\left\{x_i(k-1) - \max\left[d(k) - \sum_{j=0}^{i-1} x_j(k-1), 0\right], 0\right\}. \tag{4}$$

The output of the system will be the weighted sum of the activation values x_i. This model doesn't require the storage of previous turning points of the input because over the working interval of one weight the weight value is constant, so it is not relevant where exactly the activated part lies within that interval. As we will see next, this property will make it also possible to model easily the initial curve (anhysteretic state of the systems).

Now that we have formulated a method to calculate the activation values of the intervals from the previous one using the actual input of the system, we only have to define a starting point for our algorithm, an initial setup for the interval activation values.

As it is known about systems with hysteresis, they all have an initial state that is called anhysteretic state. It would be just natural if the initial state of the model had represented the anhysteretic state and the behavior of systems with hysteresis.

This special initial activation setup is very simple; each interval needs to be activated halfway, as it can be seen in the *Fig.* 6.

The anhysteretic state of a hysteretic system (for example a magnetic material) can be reached by applying a dampening periodic input signal to the system. If we translate this to the calculation method of the modelling system, we get that thin lines of activated and not activated intervals follow each other along the input range. Each activated interval has inactive intervals as neighbours and vice versa. However, in each weight interval the activated and inactive parts can be summarized, thus leading to the result that the starting point or the anhysteretic state of the model is by half activating each weight interval in the model:

$$x_i(0) = \frac{a_i}{2}, \tag{5}$$

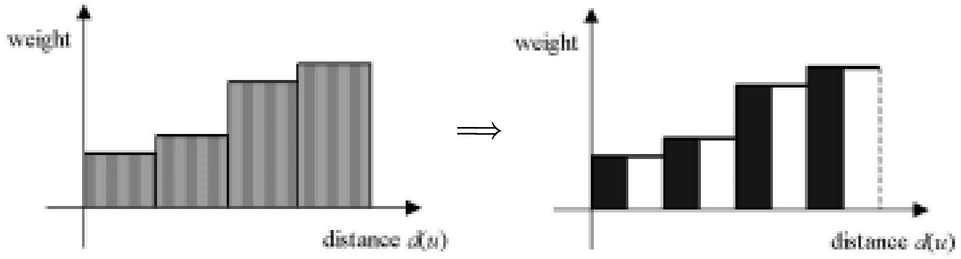

Fig. 6. The initial setup to simulate the anhysteretic behavior

where x_i is still the input of the ith weight – the ith activation value – and q the size of the working interval of the same.

With this type of networks, by setting up different weight combinations, we are able to model different hystereses. However, our goal is to set up the weights according to the measured hysteresis data, so we need some training procedure.

2.2. *Calculation of the Model Output*

The output of the model was calculated in most of the investigated cases with the simplest possible neural network, a linear perceptron. The output of the model is defined by the following formula:

$$h(u(k)) = \sum_{i=0}^{N} w_i \cdot x_i(k) + w_{\text{bias}}, \tag{6}$$

where N is the number of weights, w_i is the value of the ith weight, $x_i(k)$ is the actual activation of the ith weight and w_{bias} is the bias weight.

However, more complex models can also be constructed using such simple models over different input ranges and combining the results with other simple neural networks.

2.3. *Training of the Model*

As it can be seen from the above description, the modelling system is similar to a neural network, but the normal neural training procedure must be reconsidered, because the hysteresis is a multivalued function if defined as $h(u)$. By the training, the memory effect of the hysteresis must also be taken into account therefore only a serial training procedure can be used with a weight modification rule, similar to normal neural networks.

The training uses the well-known gradient method to adjust the weights of the neural network. The error on the output is:

$$\text{error }(k) = H(u(k)) - h(u(k)), \tag{7}$$

where $H(u(k))$ is the desired hysteresis value.

A weight modification rule can be formulated through the derivation of the quadratic output error function as follows (similarly as in [12]):

$$dw_i(k) = \mu \text{ error }(k) x_i(k), \tag{8}$$

where μ is the learning factor.

3. Friction Identification with the Neural Hysteresis Model

The above described model was tested on a setup using a sliding weight, a DC motor to move the weight, a dSpace 1103 board to run the algorithm and a laser interferometer to measure the position of the weight (see *Fig.7*).

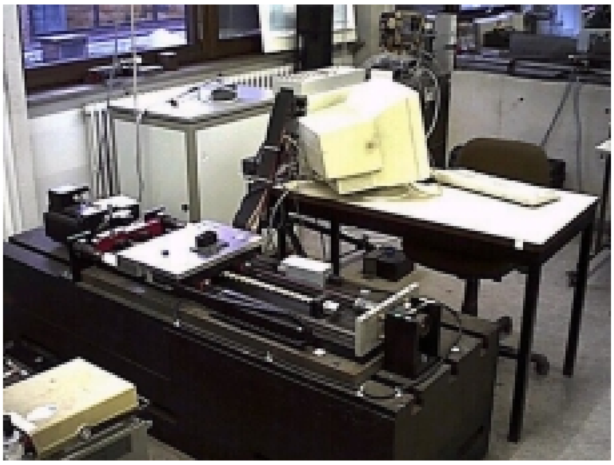

Fig. 7. The measurement setup in the laboratory KU Leuven

Simulations of an ideal environment were also made, using the Leuven Friction Model [14] as a reference for the simulation and the neural model to identify and compensate the friction hysteresis. A scheme of the setup can be seen in *Fig7*. The first one shows the identification setup and the second one the compensation setup.

These schemes are also valid for the real setup, as far as the simplified Leuven Friction Model is correct. As we can see, the original system with friction is modelled by an accelerating mass and a position feedback through a built-in hysteresis. In the simulation the stop model of the hysteresis was used to simulate the

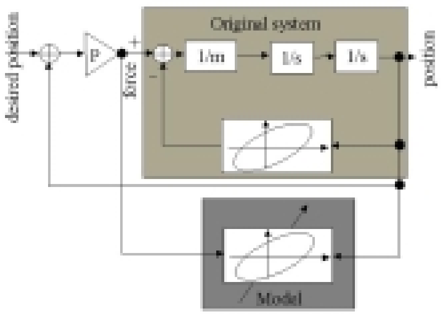

Fig. 8.a. Scheme of the system for the identification

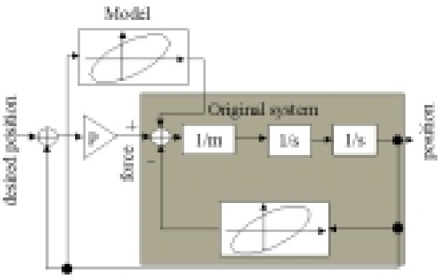

Fig. 8.b. Scheme of the system for compensation

required hysteresis for the Leuven Friction Model. The acceleration of the mass was controlled by a position-driven proportional controller. The hysteresis force was measured on the output of the controller – F_{applied}, – while the output of the system was the position of the mass – *position*. So the friction hysteresis is defined as

$$F_{\text{applied}} = h(\text{position}). \tag{9}$$

In the identification phase these two values were used to train a neural hysteresis model that was used later to predict the hysteresis force and by connecting it to the model – as it is done in *Fig.* 8.b – to compensate the hysteresis effect. In the experiments the applied force was controlled by the input voltage of the DC motor and the position was measured in micrometers. Thus, the resultant plots show the relation between the position and the input voltage.

The stop model of the hysteresis is a good reference for the simulated evaluation of the neural hysteresis model because it can be easily set up to create different symmetric hystereses modifying the weights of the model and the stop model and the neural model, can model the same type of hystereses. But there is also a significant difference in the training properties of the two models that justifies the research of the neural model. Due to the activation mode of the stop hysterons in the stop model, each non-zero input produces a non-zero output value on all of the hysterons. So the weight modification changes in each step all the weights, while in case of the neural model it is well probable that even after many non-zero hysteresis inputs there will be intervals in the non-linear transformation that have zero activation value. This has the effect that the weight modification does not affect any weights, so the model fits better the local properties of the hysteresis. Naturally, by certain training sets this leads to worse generalization, however, local errors in the training data appear localized in the trained model as well.

The same experiments were done using the simulated and the real setup to gain comparable results and to be able to validate the model.

The following sections show the results for the identification and compensation experiments on both the real system and the simulated setup.

3.1. Identification Results on the Simulated Setup

Using the simulated setup, the biggest problem was to find correct controller parameters for the proportional controller to get a good approximation of the friction hysteresis without unnecessary oscillations, thus to create the simulated setup and to simulate the measurement process. The best possible way to do this was the trial-and-error method. The problem was mainly the high sensitivity of the simulation against the parameters that resulted in instability in many cases.

However, after finding a correct gain, the hysteresis model had no trouble approximating the 'measured' friction hysteresis.

Even the presence of some artificially generated measurement noise caused no noticeable differences.

The convergence speed was reasonably fast as well ($\sim$ 60 training cycles with 1000 samples pro cycle for an error limit of 5% of the output) to try the same identification method in the real setup.

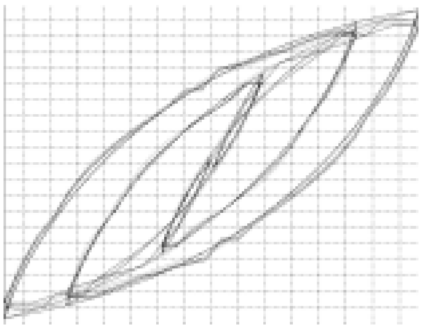

Fig. 9.a. Simulation – identifying the hysteresis in ideal case

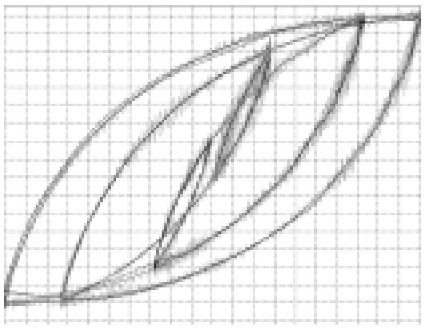

Fig. 9.b. Simulation – identifying the hysteresis in the presence of measurement noise

Fig. 9.*a* and 9.*b* show the results of two identification tests on the simulated system using the same number of intervals (20) in the model. The only difference is that in the second test a small noise – with an amplitude of about 5% of the maximum difference – was added to the simulated measured position values.

3.2. Identification Results in the Real Setup

The experiments in the real setup, however, have shown a quite different image.

First, the convergence of the approximation was much slower than in the simulation. In the beginning, the measurement noise was thought to be responsible for the difference, but this hypothesis was dropped after the experiments with artificial noise on the simulation because there even the visible noise caused no problems, and in the real setup the measurement noise was nearly below the visible level.

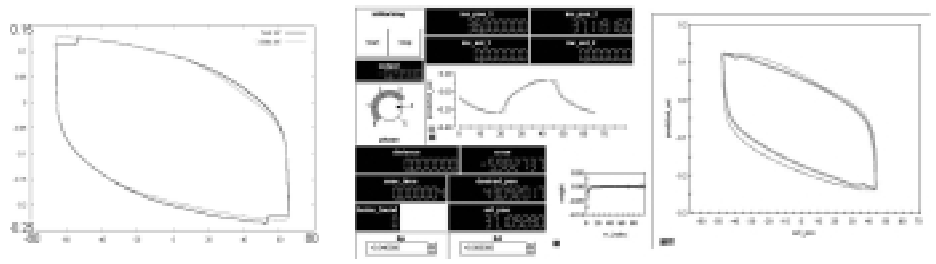

Fig. 10. Real setup – identifying the hysteresis using the main loop

After examining the measured friction hysteresis in detail, a possible source of the problems could be identified: the measured hysteresis was not exactly symmetric and also the congruency property [7] was not exactly valid (as it can be observed in the next figures). (However, according to friction theory, both properties should be valid.) In *Fig. 11.a* we have plotted the main loop of the measured hysteresis with a continuous line and its mirrored counterpart with a dashed line. If the loop would be symmetric, the two lines should overlap each other so we would see only the solid line. In *Fig. 11.b* we have plotted the raising and falling sections of the major and minor loops setting the turning points to the origo. If the measured hysteresis were congruent, there would be a maximum of two lines visible as the minor loops would follow the same trajectory as the major loop. (The two trajectories would mean that the hysteresis is not symmetric but congruent.)

So while the simulation worked with a symmetric and congruent hysteresis, in reality neither properties were exactly right. As it seemed unlikely that such a minor flaw in these properties could cause a major slowdown in the training, to test the effect about the same flaws were introduced into the simulated setup. The result was that in the simulation there also appeared a drop in the convergence speed. So it seems that these properties – symmetry and congruency – are overall important for good approximation results.

The approximation performance was also tested when the training was done using only samples from the major loop, and when samples also from minor loops were used. In the first case, as the next figures show (*Fig.12.a* and *12.b*), the estimated curve was smoother, but the approximation error on the minor loops was large.

In the second case the performance on the minor loops got better but decreased

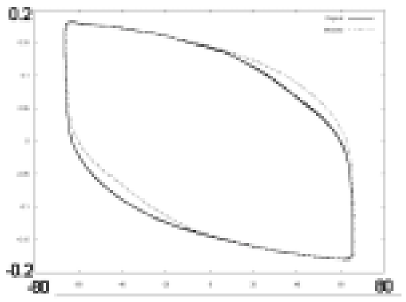

Fig. 11.a. Simmetry test

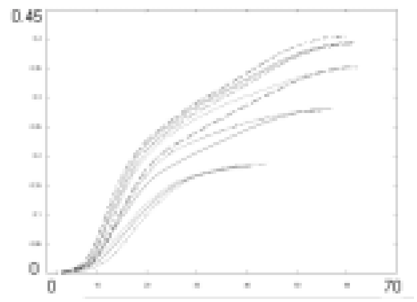

Fig. 11.b. Congruency test (plotted with a small linear distortion to emphasize the differences between the curves)

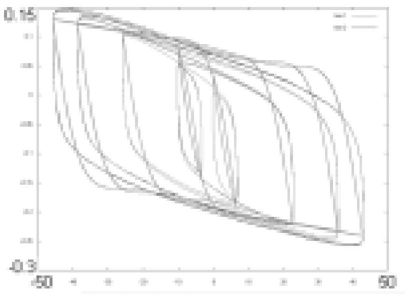

Fig. 12.a. Performance in minor loops

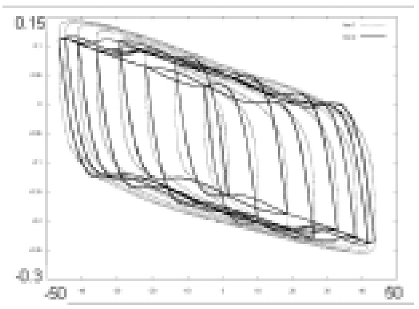

Fig. 12.b. Identifying the hysteresis using also minor loops

on the major loop, however, the average performance was better than in the first case. This behavior was also verified using the simulated setup with an asymmetric hysteresis model.

As an interesting byproduct of these experiments, a possible way of speeding up the convergence on the asymmetric training data could be determined. Additional output noise made the training faster because it blurs the asymmetry in the data. The speed gain using measured data from the real setup was larger than in the simulation, but it was also less asymmetric than the simulated data.

4. Friction Compensation with the Neural Hysteresis Model

As mentioned before, the identified model was tested by using it to compensate the friction hysteresis present in the real setup and in the simulation. The following figures show the desired position – actual position graphs for the different setups. Now all units are in μm.

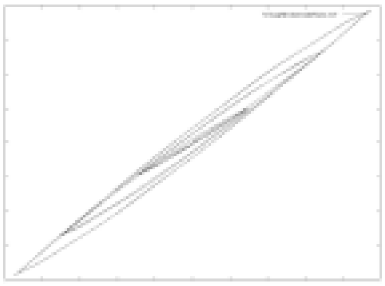

Fig. 13.a. Simulation – desired position, actual position without friction compensation

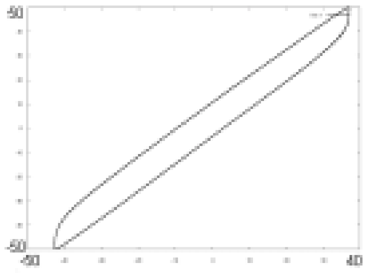

Fig. 13.b. Real data – desired position, actual position without friction compensation

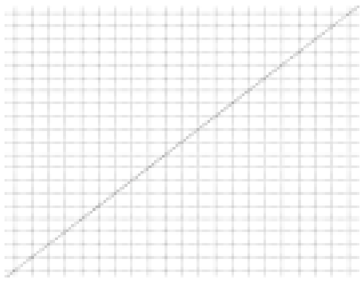

Fig. 14.a. Simulation – compensation with the identified model in the ideal case

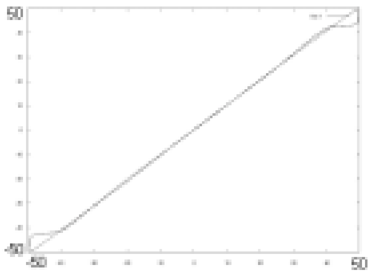

Fig. 14.b. Real data – compensation with the identified model – major loop

As it could be expected from the training results, the compensation capability of the model in the simulation was extraordinarily good, but in the real setup the approximation errors cause visible ‘bumps’ of about 5 μm. These appear due to the fact that the model does not fit correctly the steepest parts of the hysteresis branches, and, of course, the simple linear controller that was used together with the hysteresis model cannot compensate this strongly non-linear difference. Currently further input interval setups are under investigation that are thought to provide better approximation on the fast changing parts of the hysteresis. However, the best way to deal with the problem would be an algorithmic solution to automatically optimise the interval setup during the training process.

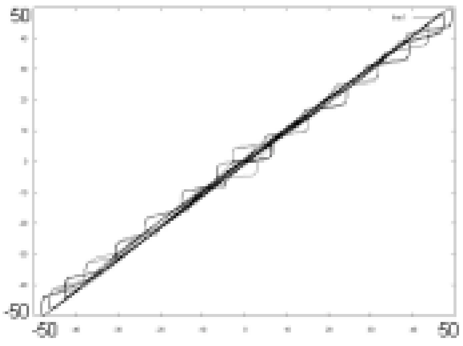

Fig. 15. Compensation in minor loops

5. Conclusions

This article introduced and investigated a modelling approach based on neural networks for hysteresis effects. The introduced neural model is a much simplified version of the traditional Preisach hysteresis model using only a one dimensional and discrete weight function to calculate the output. Thus its modelling ability is also simpler, limited to the domain of symmetric hystereses. However, to function properly the model still requires that the two necessary properties – congruency and wiping-out property – for the Preisach model are fulfilled. The model was found to be very effective in the simulated environment, but it has a considerable error in a real setup. As these result were examined, the asymmetry of the measured hysteresis and the lack of congruency in the minor loops could be determined as a possible cause for this effect. Naturally, the real physical effect fulfils these properties, however, even a slight distortion in the measurements can cause quite big problems in the modelling by disturbing these properties. So the neural modelling approach must be reconsidered and refined using these experiences.

References

[1] GE, P. – JOUANEH, M., Modeling Hysteresis in Piezoactuators, *Precision Engineering,* **17** No. 3 (1995), pp. 211–221.

[2] GOLDFARB, M. – CELANOVIC, N., A Lumped Parameter Electromechanical Model for Describing the Non-linear Behavior of Piezoelectric Actuators, *Trans. of the ASME, Journal of Dynamic Systems, Measurement, and Control,* **119** (1997), pp. 478–485.

[3] FUTAMI, S. – FURUTANI, A. – YOSHIDA, S., Nanometer Positioning and its Micro-Dynamics, *Nanotechnology,* **1** No. 1 (1990) pp. 31–37.

[4] SWEVERS, J. – AL-BENDER, F. – GANSEMAN, C. – PRAJOGO, T., An Integrated Friction Model Structure with Improved Presliding Behaviour for Accurate Friction Compensation, *ITAC,* **45** No. 4 (2000), pp. 675–686.

[5] KUHNEN, K. – JANOCHA, H., Adaptive Inverse Control of Piezoelectric Actuators with Hysteresis Operators, *Proceedings of European Control Conference,* 1999.

[6] MAYERGOYZ, I. D., *Mathematical Models of Hysteresis*, Springer-Verlag, New York, 1991.

[7] PREISACH, F., On Magnetic Aftereffect, *Zeitschrift für Physiks*, **97** (1935), pp. 277–302.

[8] KRASNOSELSKI, M. A. – POKROVSKII, A. V., *Systems with Hysteresis*, Springer-Verlag, New York, 1989.

[9] LAMPAERT, V. – SWEVERS, J., On-Line Identification of Hysteresis Functions With Non-local Memory, *Proceedings of the International Conference on Advanced Intelligent Mechatronics*, **1** (2001), pp. 833–837.

[10] AWABDY, B. A. – SHIH, W.-C. – AUSLANDER, D. M., Nanometer Positioning of a Linear Motion Stage Under Static Loads, *IEEE/ASME Trans. on Mechatronics*, **3** No. 2 (1998), pp. 113–119.

[11] WEI, J.-D. – SUN, C.-T., Constructing Hysteretic Memory in Neural Networks, *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics*, **30** No. 4 (2000), pp. 601–609.

[12] KUCZMANN, M., A New Neural-Network-Based Scalar Hysteresis Model, *IEEE Trans. On Magn.*, **38** (2002) pp. 857–860.

[13] KUCZMANN, M., Neural Network Model of Magnetic Hysteresis, *COMPEL, The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, **21** (2002), pp. 364–376.

[14] LAMPAERT, V. – SWEVERS, J. – AL-BENDER, F., Modification of the Leuven Integrated Friction Model Structure, *Transactions on Automatic Control*, **47** No. 4 (2002), pp. 683–687.