

# Is Extreme Learning Machine Effective for Multisource Friction Modeling?

Jacek Kabziński

► **To cite this version:**

Jacek Kabziński. Is Extreme Learning Machine Effective for Multisource Friction Modeling?. Richard Chbeir; Yannis Manolopoulos; Ilias Maglogiannis; Reda Alhajj. 11th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI 2015), Sep 2015, Bayonne, France. IFIP Advances in Information and Communication Technology, AICT-458, pp.318-333, 2015, Artificial Intelligence Applications and Innovations. <10.1007/978-3-319-23868-5\_23>. <hal-01385367>

**HAL Id: hal-01385367**

**<https://hal.inria.fr/hal-01385367>**

Submitted on 21 Oct 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Is Extreme Learning Machine Effective for Multisource Friction Modeling?

Jacek Kabziński

Institute of Automatic Control, Lodz University of Technology  
jacek.kabzinski @p.lodz.pl

**Abstract.** The aim of this contribution is to discuss suitability of extreme learning machine (ELM) approach for modeling multisource friction for motion control purposes. The specific features of multisource friction in mechatronic systems are defined, the main aspects of friction modeling by a standard ELM are investigated and some modifications are proposed to make it more suitable for specific demands of the discussed task. This allows to formulate some general remarks concerning properties of ELM for function approximation.

**Keywords:** extreme learning machine, friction modeling, adaptive control

## 1 Introduction

Extreme Learning Machine (ELM) is relatively new neural modeling technique. It is intensively promoted by numerous researchers and attracts increasing attention for its fast and simple training algorithm. Several review papers [1,2,3] describe the basic form of ELM and discuss characteristic features of the method. The range of possible applications of ELM covers as well regression and classification problems and learning schemes include batch processing and sequential on-line training. In this contribution we concentrate on function approximation problems with batch data processing approach. The authors of ELM stress that the network structure is simple, hidden neurons parameters are selected on random “without seeing the data” and remain constant, output weights are calculated once for a batch of data, with the use of Moore-Penrose generalized inverse. Hence, without iterative tuning of parameters by back-propagation, ELM achieves very short learning times and, with sufficient number of neurons, sufficient accuracy in terms of modeling RMSE. On the other hand, several publications report that ELM may: create ill-condition numerical problems [4], introduce overfitting, require too big number of neurons [2], that the influence of number of neurons and parameters of random weights selection on the resulting performance is unclear and requires analysis [5].

Having in mind the “no free lunch” rule, i.e. remembering that it is impossible to find “the best” solution for all possible problems, we investigate the effectiveness of ELM modeling for a special problem - friction (i.e. friction force or torque) modeling for motion control in mechatronic systems. This problem is important for applications, and it constitutes an interesting benchmark for modeling techniques. Although several

memoryless models of friction as a function of relative speed of moving surfaces are available [6], parametrical identification of these models is difficult because of high nonlinearity. The problem becomes more complex if several sources of friction are present in the plant, acting together against the motion. In this case the use of AI models is acceptable, and the control algorithms benefit from linear parameterization of such models [7]. In this contribution the main features of friction modeling by a standard ELM are investigated and some modifications are proposed to make it more suitable for specific demands of the discussed task.

## 2 Multisource Friction Model

The friction modeling and parametric identification is extremely complex if the drive system has to match friction forces from several sources. A simple example is a mover acting between two different surfaces. As a matter of fact the presence of several sources of friction is a typical situation. Usually friction is caused by a load machine, bearings, a motor, etc. Of course it is possible to use several different models of friction – one for each recognized source, but this will increase the number of parameters drastically and will make the whole model too complicated. As all friction forces accumulate, working against the motion, it will be impossible to distinguish among the models and to identify all parameters.

The multisource friction model may be defined as

$$F_F(v) = \sum_{k=1}^S [f_{ck} + (f_{sk} - f_{ck})g(v, v_{sk})] \operatorname{sgn}(v) + \sum_{k=1}^S B_k v \quad (1)$$

where  $S$  is the number of sources,  $v$  is the velocity between two surfaces in contact. For each source of friction  $f_{ck}$  is the Coulomb friction, the Stribeck effect is described by  $f_{sk}$  and the Stribeck velocity is  $v_s$ , while  $B_k$  denotes a coefficient of viscous friction [6]. The function  $g(v, v_s)$  influences the shape of the characteristics. Several possibilities were reported in the literature [6], for example

$$g(v, v_s) = e^{-\left(\frac{v}{v_s}\right)^2}, g(v, v_s) = e^{-\frac{|v|}{v_s}}, g(v, v_s) = \frac{1}{1+\frac{|v|}{v_s}} \quad (2)$$

The exact form of  $g(v, v_s)$  is not justified theoretically and must be adjusted to the considered application. Generally, it represents the transient between the boundary lubrication and the hydrodynamic lubrication zones, it starts with 1 for  $v = 0$  and disappears for big  $v$ .

The data collected to identify the friction model are obtained during some experiments, from available measurements, or from a proper observer output [8]. Let us denote the measured or estimated values of friction by  $\hat{F}_F(v)$ . The common features of the obtained data are:

- Changing parameters  $f_{ck}, f_{sk}, v_{sk}$  results in various shapes of the characteristics  $F_F(v)$ , but several values of parameters may lead to the same

curve  $F_F(v)$ , so the identification of all parameters in (1) by any curve fitting method is impossible.

- The data are subject to measurement or observation inaccuracies, that may be represented as a uniformly distributed noise.
- It is possible to identify the resulting viscous friction coefficient  $B = \sum_{k=1}^S B_k$  from the steady-state data, collected for sufficiently big velocities, as  $\Delta F_F \approx B \Delta v$ , although it is impossible to distinguish among  $B_k$ .
- The number of the data is limited.
- It is possible to find reasonable bound for the biggest Stribeck velocity  $\max\{v_{sk}\} < v_{smax}$ .

The benchmark problem considered here is obtained from the model (1) with parameters given in table 1.

Table 1. Parameters of 3-source friction model (SI units).

	$f_{ck}$	$f_{sk}$	$v_{sk}$	$B_k$
$k = 1$	4	8	0.01	2.5
$k = 2$	6	16	0.005	50
$k = 3$	5	20	0.02	5

Measurement/estimation generates an uniform random noise in  $[-0.5, 0.5]$  N. The data are normalized to the range  $[0, 1]$  by scaling factors  $K_{out}, K_{in}$  for the output and input respectively. As  $B = \sum_{k=1}^S B_k$  is identified using linear regression for the data obtained from big speed motion and  $g(v, v_{sk}) \approx 0$  for  $v > 3v_{smax}$ , the final model, obtained from the data  $\{(x_i, t_i) = (K_{in}v_i, K_{out}(\hat{F}_F(v_i) - Bv_i)), i = 1, \dots, N\}$  will be

$$F_{Fm}(v) = \begin{cases} K_{out} AIM(K_{in}v)sign(v) + Bv & \text{for } |v| < 3v_{smax} \\ K_{out} AIM(3K_{in}v_{smax})sign(v) + Bv & \text{for } |v| > 3v_{smax} \end{cases} \quad (3)$$

where  $AIM(\cdot)$  stands for the Artificial Intelligence Model, that will be provided by EML or other methods. It is assumed that the Stribeck curve is symmetrical and modeling for positive velocity allows to obtain a complete model, although generalization to nonsymmetrical characteristics is straightforward.

$N = 200$  points are selected on random from  $[0, 3v_{smax}]$  and are used for training. The point  $(0, F_F(0))$ , where  $F_F(0)$  is calculated as the mean of measurements for four smallest values of velocity, and  $(3v_{smax}, F_F(3v_{smax}))$  are added to the learning dataset to obtain the model covering the velocity interval  $[0, 3v_{smax}]$ . The friction curves are plotted in fig. 1 and the normalized data  $(x_i, t_i)$  are presented in fig. 2. The testing dataset consists of another 200 points selected on random from  $[0, 3v_{smax}]$ . RMSE is used to measure the training error  $E_{train}$ , or the test error  $E_{test}$

$$E_{train} = \sqrt{\frac{1}{N} \sum_{i=1}^N (AIM(x_i) - t_i)^2} \quad (4)$$

Also so called “theoretical testing error”, comparing the model with the exact function used to generate the data, is considered in simulations. It is calculated using  $Nt=500$  values  $v_k$  distributed equidistantly in  $[0, 3v_{smax}]$

$$E_{Ttest} = \sqrt{\frac{1}{Nt} \sum_{i=1}^{Nt} (AIM(x_i) - K_{out} (F_F(v_i) - Bv_i))^2} \quad (6)$$

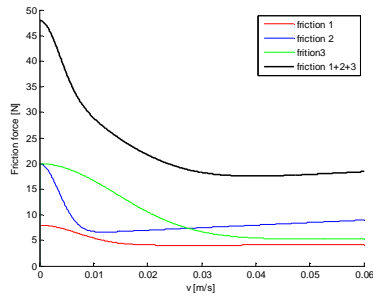


Fig.1 3-source friction.

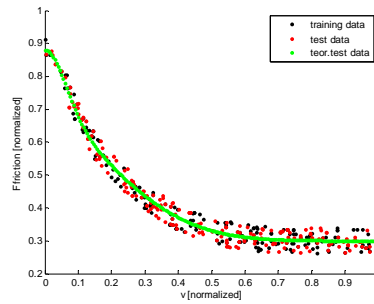


Fig. 2. The data collected for training and testing.

### 3 Fuzzy Friction Modeling

The fuzzy friction modeling technique was discussed in details in [8]. It is known that the careful initial model selection [7] enables to obtain accurate and relatively simple Takagi-Sugeno-Kang models, and the use of nonlinear consequences helps to improve the model performance [9]. Here fuzzy modeling is used only for comparison with EML models. The standard initialization technique implemented as `genfis1` in Matlab fuzzy toolbox is used together with ANFIS approach to model training [10]. The results obtained for different number of bell-shaped membership functions (MFs) are presented in fig. 2, 3.

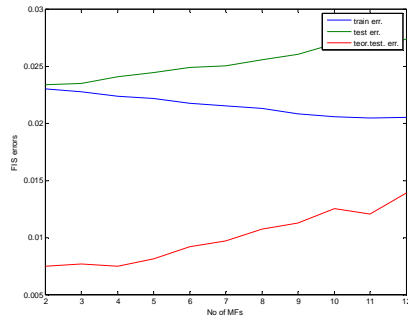


Fig. 3 Training and test errors versus no of bell-shaped membership functions

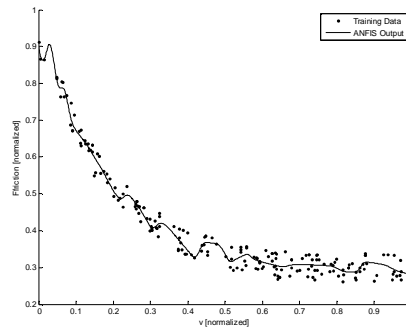
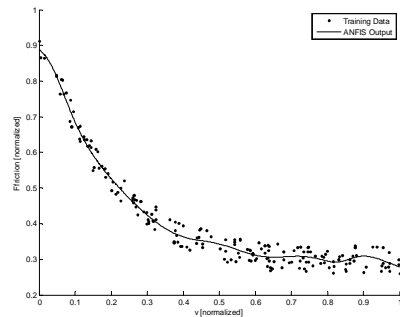
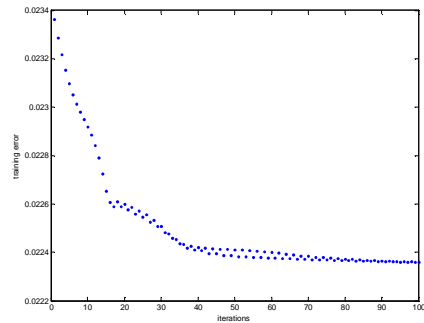


Fig. 4 Overfitting, 12 bell-shaped membership functions

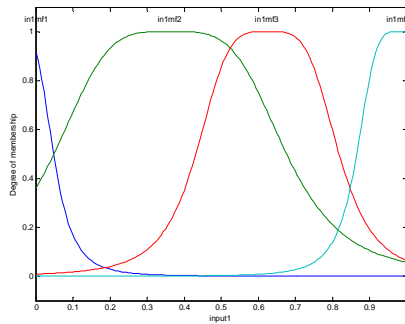
Although the training error is decreasing with number of membership functions, the test error is increasing, starting from very moderate number of MFs. The overfitting is visible in fig. 2. Hence the lack of generalization is the most important drawback of fuzzy modeling. The choice of 4 membership functions seems reasonable, it makes the model to be linear subject to 8 parameters and nonlinear subject to 12 more, so 20 parameters are optimized. The performance of 4 membership functions model is presented in fig. 5,6,7. The obtained values of train, test and theoretical test errors are  $[E_{Ttest}, E_{test}, E_{Ttest}] = [0.0224, 0.0241, 0.0075]$ . Similar results were obtained with other types of membership functions. For example the errors achieved with Gauss-type MFs are plotted in fig. 8.



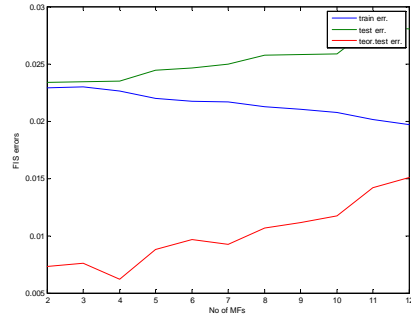
**Fig. 5** Fuzzy model with 4 MFs and the trading data.



**Fig.6** Training and test errors versus no of Gaussian-type membership functions.



**Fig. 7** Bell-shaped MFs of the trained fuzzy model



**Fig.8** Training and test errors versus no of Gaussian-type membership functions

## 4 Standard EML Model

ELM was originally proposed by Huang et al. [1] as a learning algorithm for the single-hidden-layer neural network. The main concepts behind the ELM are that: (i) the weights and biases of the hidden nodes are generated randomly and are not adjusted and (ii) the output weights are determined analytically.

The output function of ELM is of the form (for a one output case,  $x$  is the input variable, i.e. the normalized velocity in friction modeling problem):

$$F_{ELM}(x) = \sum_{k=1}^M \beta_k h_k(x) \quad (7)$$

where  $M$  is a number of hidden neurons,  $h(x) = [h_1(x), \dots, h_m(x)]^T$  is the output vector of the hidden layer with respect to the input  $v$ , and  $\beta = [\beta_1, \dots, \beta_M]^T$  is the vector of the output weights (between the hidden layer and the output node).

Different activation functions (AFs) may be used in the hidden neurons. The most popular are: sigmoid, Gaussian, multiquadric, step, triangular and sine AFs. The sigmoid AF is:

$$h_k(x) = \frac{1}{1 + \exp(-(w_k x + b_k))} \quad (8)$$

where  $w_k$ ,  $k = 1, \dots, M$  is the input weight of the  $k$ -th hidden neuron,  $b_k$  is its bias.

The parameters of hidden nodes  $w_k$  and  $b_k$ , are randomly generated according to any continuous probability distribution instead of being explicitly trained. That process is independent of the training data. If

$$\{(x_i, t_i) = (K_{in} v_i, K_{out} (\hat{F}_F(v_i) - B v_i)), \quad i = 1, \dots, N\} \quad (9)$$

compose a set of learning data, the  $N$ -dimensional vector of model outputs  $F_m$  is compared with the target vector  $F$ , where

$$F_m = [F_{ELM}(x_1), \dots, F_{ELM}(x_N)]^T, \quad F = [t_1, \dots, t_N]^T \quad (10)$$

As

$$F_m = \begin{bmatrix} h_1(v_1) & \cdots & h_M(v_1) \\ \vdots & \ddots & \vdots \\ h_1(v_N) & \cdots & h_M(v_N) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_M \end{bmatrix} = H\beta \quad (11)$$

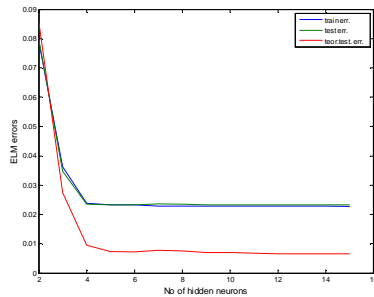
the output weights  $\beta_k$  are found by minimizing the approximation error:

$$E = \|H\beta - F\|^2 \quad (12)$$

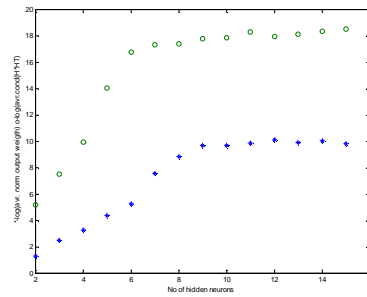
The optimal solution is:

$$\beta_{opt} = H^+ F \quad (13)$$

where  $H^+$  is the Moore–Penrose generalized inverse of matrix  $H$ . The standard configuration of ELM assumes sigmoid AFs with random parameters uniformly distributed in  $[-1,1]$ . This was used for the benchmark modeling.

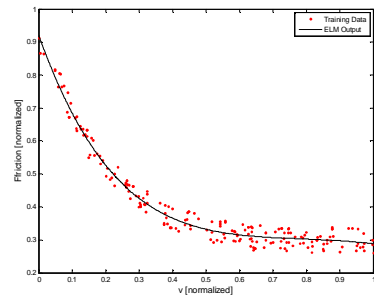


**Fig. 9** Average training and test errors versus no of hidden neurons.

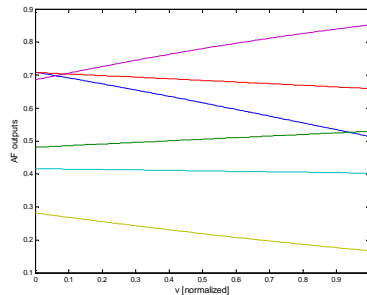


**Fig.10** Average norm of output weights and average  $cond(H^T H)$

The model performance is demonstrated in fig. 9. Mean values from 20 runs are presented. As it is plotted in fig. 10, the output weights and the condition coefficient of matrix  $HH^T$  became very big numbers. Therefore using more than few hidden neurons does not make any sense. The results obtained for 6 hidden neurons are presented in fig. 11. The achieved modeling errors are  $[E_{train}, E_{test}, E_{Ttest}] = [0.0232, 0.0233, 0.0072]$  and are comparable with the scores obtained by the fuzzy model. The generalization properties of standard ELM are much better – no overfitting with increasing number of neurons is observed. The noise corrupting the data set does not allow to decrease the modelling errors below the limits obtained with several hidden neurons, and from numerical point of view the model becomes ill-conditioned for bigger number of neurons.



**Fig. 11** Modeling with 6-neurons ELM



**Fig.12** Activation functions of 6-neurons ELM

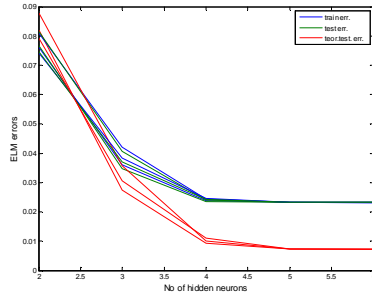


## 5 Improving ELM

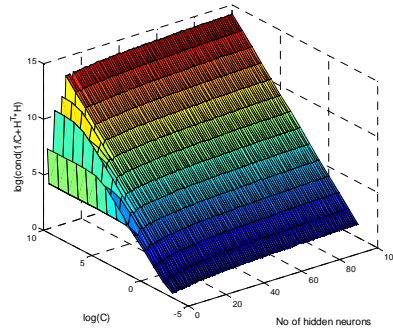
The ELM drawbacks described in the previous section motivate the search for modifications that will improve the modeling performance, i.e. the model accuracy, ability of generalization and the numerical stability.

### Activation functions

The change of the activation function type from sigmoid to any other infinitely differentiable function, with the same weights and biases selection procedure, will not improve the model. The errors behave very similarly for different smooth AFs (fig. 13) and also  $cond(H^T H)$  and  $\|\beta_{opt}\|$  remain at comparable levels. Of course the application of AFs with a compact support, like membership functions used in fuzzy modeling will change the situation, making ELM to behave like a fuzzy model.



**Fig. 13** Average training and test errors versus no of hidden neurons for sigmoid, sinus and radial basis AFs.



**Fig.14**  $cond\left(\frac{1}{C}I + H^T H\right)$  as a function of number of neurons and regularization coefficient  $C$

### Regularization

The well-known remedy for numerical problems in ELM, caused by ill-conditioned set of linear equation, is regularization of the least-square problem [1]. Instead of minimizing (12) the weighted problem is considered, with the performance index:

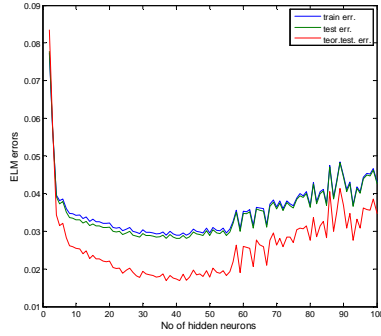
$$E_C = \|\beta\|^2 + C\|H\beta - F\|^2 \quad (14)$$

where  $C$  is a design parameter. The optimal solution is now [1]:

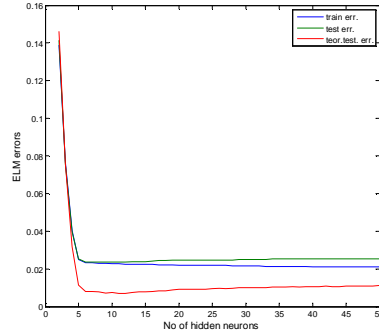
$$\beta_{Copt} = \left(\frac{1}{C}I + H^T H\right)^{-1} H^T F \quad (15)$$

and decreasing  $C$  allows to improve the condition number of  $\frac{1}{C}I + H^T H$ . Inevitably this will degrade the modelling accuracy, so a compromise is required. The proper choice of  $C$  depend on the problem structure and it is difficult to formulate general rules. In

the considered case (200 samples, sigmoid AFs) an exemplary plot of  $cond\left(\frac{1}{C}I + H^T H\right)$  is presented in fig. 14. Unfortunately, if  $C$  provides acceptable condition coefficient, the accuracy of the model is too poor (fig. 15). Moreover the model instability (growing errors) is observed for  $N > 50$ . Smaller values of  $C$  will stabilize the model but the modeling errors will be bigger.



**Fig. 15** Average training and test errors versus no of hidden neurons for  $C=10^5$  and sigmoid AFs.



**Fig.16** Average training and test errors versus no of hidden neurons for enlarged variability of sigmoid AFs.

#### Activation function variability

The ELM is not able to learn the features from the data, as a fully trained neural network does. It projects the input data into the feature space the randomly chosen weights happen to specify, and learns a linear mapping in that space. Therefore this nonlinear mapping of the data into a feature space should be able to extract the features sufficient for predicting the target variable of a regression task. The importance of sufficient AFs variability was noticed previously [5]. The point is, to find the way to correct the random mechanism of weights and biases creation to increase the variability without losing too much from the simplicity of random selection.

The first step to enlarge the variability of sigmoid AFs is to increase the range of uniform random selection of input weights. It must be suitably fixed to guarantee that the sigmoid operation neither remains linear nor too strongly saturates in the input interval. In the presented application the range  $w_k \in [-10,10]$  seems suitable. Next the biases may be selected to ensure that the range of sigmoid function is sufficiently large. For a positive input weight  $w_k > 0$  sigmoid function is strictly increasing in  $[0,1]$ . Assuming that the minimal value of the sigmoid AF shall be smaller than the given  $r_1$ , and the maximal value should be bigger than  $r_2$ ,  $0 < r_1 < r_2 < 1$  yields

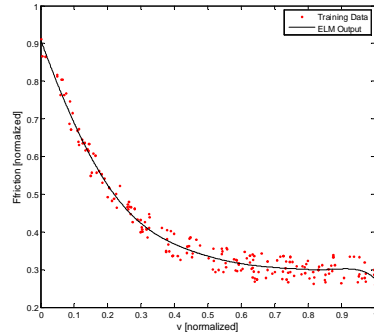
$$h_k(0) = \frac{1}{1 + \exp(-b_k)} < r_1, \quad h_k(1) = \frac{1}{1 + \exp(-w_k - b_k)} > r_2 \quad (16)$$

so the bias  $b_k$  shall be selected on random, uniformly from  $b_k \in [\bar{a}, \tilde{a}]$ , where

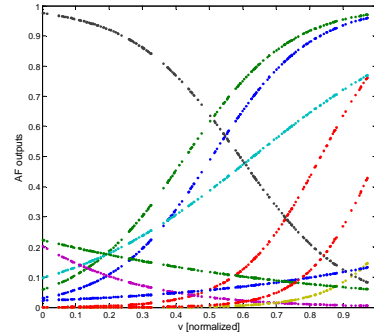
$$\bar{a} = -w_k + \ln\left(\frac{1}{r_2} - 1\right), \quad \tilde{a} = -\ln\left(\frac{1}{r_1} - 1\right) \quad (17)$$

If the selected weight is negative  $w_k < 0$ , the bias  $b_k$  shall be selected on random, uniformly from  $b_k \in [\tilde{a}, \bar{a}]$ , where  $\tilde{a}, \bar{a}$  are given in (17).

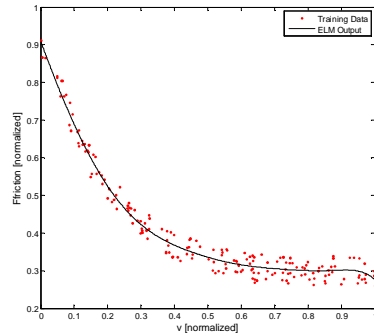
The application of the proposed method to the considered friction modeling problem results in the performance presented in fig. 16. The achieved modeling errors are  $[E_{train}, E_{test}, E_{Ttest}] = [0.0210, 0.0234, 0.0070]$ . The results for 10 neurons are presented in fig. 17,18. Comparing figures 18 and 12 we notice that the variability of AFs was noticeably increased. The output weights are smaller and ill-conditioning is eliminated.



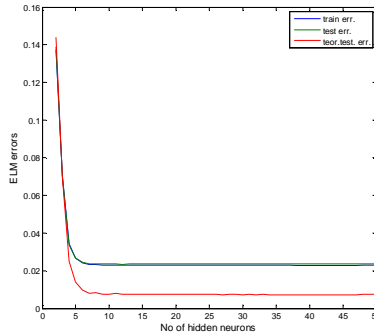
**Fig. 17** Modeling with 10-neurons ELM and enlarged variability of AFs.



**Fig.18** Activation functions of 10-neurons ELM and enlarged variability of AFs.



**Fig. 19** Modeling with 10-neurons ELM, enlarged variability of AFs and regularization with  $C=10^5$ .

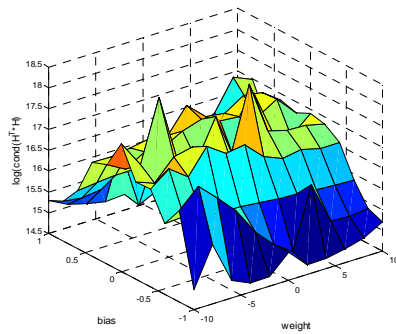


**Fig. 20** Average training and test errors versus no of hidden neurons for enlarged variability of sigmoid AFs and regularization with  $C=10^5$ .

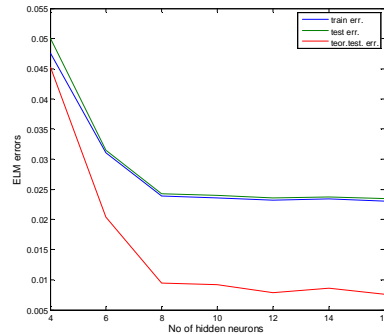
Even better results are obtained applying the variability enlargement and the regularization together. The results for 10 neurons and  $C=10^5$  are plotted in fig. 19. The achieved modeling errors are  $[E_{train}, E_{test}, E_{Ttest}] = [0.0231, 0.0234, 0.0074]$ . As it is visible from fig. 20 the generalization ability is preserved for increasing number of neurons – the training and test errors remain at the same level. No overfitting was observed for number of neurons  $< 50$ . It is remarkable that the norm of output weights remains small:  $\|\beta\| < 6$ .

#### Direct minimization of the condition coefficient

Good conditioning of the linear equation (11) is vital to derive accurate ELM with moderate output weights. Therefore some hidden neurons parameters may be not selected randomly, but may be used to minimize  $cond(H^T H)$  directly. So, the hidden neurons will be divided among two subsets, say RAND and OPT. First, the AF parameters of neurons in RAND will be selected on random, next the parameters of neurons in OPT will be calculated to minimize  $cond(H^T H)$ .



**Fig. 21** An exemplary plot of  $\log(cond(H^T H))$  as a function of the 11<sup>th</sup> neuron parameters.



**Fig. 22** Average training and test errors versus no of hidden neurons for optimization of conditioning by the half of neurons.

Of course in this case we cannot claim that the weights and biases selection process is so fast and simple as in the basic ELM algorithm. The choice of the optimization algorithm is also an open problem. It is difficult to assume anything about the shape of the minimized function, as it depends on the training data and randomly selected parameters of neurons in RAND. The exemplary plot of the condition number as a function of the weight and the bias, when first ten sigmoid neurons belong to RAND and the neuron number 11 is the only one in OPT, is presented in fig. 21. As it is visible, the minimized function is not smooth and multimodal. Therefore gradient type algorithms are not applicable, and good heuristic algorithm, that is able to avoid the local minima is recommended. The optimization result will be influenced by the initial condition, as well as by the range of the decision variables, i.e. the weights and biases of the neurons in OPT. All these makes the whole process quite complicated and time consuming. The result highly depends on the effectiveness of the applied optimization algorithm.

Selecting a half of neurons as RAND with random selection with the enlarging the AFs variability approach, and the other half in OPT with minimization performed by Particle Swarm Optimization algorithm, provides results presented in fig. 22. The achieved modeling errors are  $[E_{train}, E_{test}, E_{Ttest}] = [0.0231, 0.0235, 0.0076]$ . The mean value of the output weights norm is smaller than 17 for any number of neurons. So the results are comparable with these provided by application of AF variability enlargement and regularization together.

## 6 ELM Models in Adaptive Control

The final test for the ELM model of friction is its behaviour in a control system. Here we investigate the applicability of the proposed models in a simple adaptive backstepping, tracking position control.

The motion dynamics of the considered system is given by

$$\frac{dx}{dt} = v \quad m \frac{dv}{dt} = F_e - F_{friction}. \quad (18)$$

where  $m$  is a forcer mass,  $F_e$  is a thrust force, and  $F_{friction}$  represents all types of friction forces. The mover speed is  $v$  and the position is  $x$ . Although equation (18) is written according to linear motion convention, it may be also used for rotational movement description, if we read  $m$  as a moment of inertia and consider torques instead of forces. It is assumed that the thrust force is proportional to the control variable (the motor current)

$$F_e(t) = \varphi \cdot i(t), \quad (2)$$

and coefficient  $\varphi$  is not known exactly. The desired smooth position trajectory will be denoted by  $x_d$ , actual position by  $x$  and the tracking error by  $e_1 = x_d - x$ . The velocity will be ‘virtual control’ for position tracking. If we choose the desired velocity  $v_d$  according to

$$v_d = \dot{x}_d + k_1 \cdot e_1, \quad (3)$$

where  $k_1 > 0_1$  is a design parameter, we will be able to describe the tracking error dynamics as

$$\dot{e}_1 = \dot{x}_d - \dot{x}_d - k_1 \cdot e_1 + e_2 = -k_1 \cdot e_1 + e_2, \quad (4)$$

$$e_2 = v_d - v = \dot{x}_d + k_1 \cdot e_1 - v \quad (5)$$

$$m_o \cdot \dot{e}_2 = m_o \cdot \dot{v}_d - m_o \cdot \dot{v} = m_o \cdot \dot{v}_d - i - \frac{1}{\varphi} F_{friction}, \quad m_o = \frac{m}{\varphi}, \quad (6)$$

$$\dot{v}_d = \ddot{x}_d + k_1 \cdot (-k_1 \cdot e_1 + e_2). \quad (7)$$

The control variable  $i$  has to compensate the function

$$D = m_o \cdot \dot{v}_d - \frac{1}{\varphi} F_{friction}, \quad (8)$$

and to assure fast tracking. We will use a model  $\hat{D}$  for  $D$ , incorporating the fuzzy friction model (17,18). The general structure of  $\hat{D}$  will be given by

$$\hat{D} = \hat{\mathbf{A}}^T \boldsymbol{\xi}, \quad (9)$$

where  $\hat{\mathbf{A}}^T$  is a vector of adaptive parameters and  $\boldsymbol{\xi}$  is known. The structure with two adaptive parameters:

$$\hat{\mathbf{A}}^T = [\hat{m}_o, \hat{k}] \quad \boldsymbol{\xi} = \begin{bmatrix} \dot{v}_d \\ \frac{1}{\varphi_N} F_{Fm}(v) \end{bmatrix}, \quad (10)$$

was selected, where  $F_{Fm}(v)$  is the friction model (3) (with the derived ELM model inside),  $\hat{m}_o$  is supposed to adapt the changing inertia and  $\hat{k}$  corrects  $\frac{1}{\varphi_N} F_{Fm}$  to the actual value of  $\frac{1}{\varphi} F_{friction}$ .

It is assumed that “the best” adaptive parameters  $\mathbf{A}^{*T}$  exist, such that the model  $D^* = \mathbf{A}^{*T} \boldsymbol{\xi}$  provides bounded estimation error  $\varepsilon = D - D^*$ ,  $|\varepsilon| < \varepsilon_{max} < \infty$ . The difference between actual and “the best” parameters is denoted as  $\tilde{\mathbf{A}} = \mathbf{A}^* - \hat{\mathbf{A}}$ .

The control law

$$i = \hat{D} + k_2 \cdot e_2 + e_1 \quad (28)$$

provides the tracking error dynamics

$$m_o \cdot \dot{e}_2 = \varepsilon + \tilde{\mathbf{A}}^T \boldsymbol{\xi} - k_2 \cdot e_2 - e_1. \quad (29)$$

The Lyapunov function

$$V = \frac{1}{2}(e_1^2 + m_o \cdot e_2^2 + \tilde{\mathbf{A}}^T \mathbf{\Gamma}^{-1} \tilde{\mathbf{A}}). \quad (11)$$

with positive definite symmetric matrix  $\mathbf{\Gamma}$  is selected to investigate the system stability. Taking any of the adaptation laws

$$\dot{\hat{\mathbf{A}}} = e_2 \mathbf{\Gamma} \xi, \text{ or } \dot{\hat{\mathbf{A}}} = e_2 \mathbf{\Gamma} \xi - \delta \mathbf{\Gamma} \mathbf{A}, \text{ or } \dot{\hat{\mathbf{A}}} = e_2 \mathbf{\Gamma} \xi - \delta \sqrt{e_1^2 + e_2^2} \mathbf{\Gamma} \mathbf{A}. \quad (12\text{a,b,c})$$

with small positive  $\delta$ , allows to prove that the Lyapunov function derivative (32) is negative outside a certain, bounded set, and so  $e_1, e_2$  are uniformly ultimately bounded [11]. For example with adaptation performed according to (31a) we get

$$\dot{V} = -k_1 \cdot e_1^2 - k_2 \cdot e_2^2 + e_2 \cdot \varepsilon \leq -k_1 e_1^2 - \left(k_2 - \frac{1}{2}\right) e_2^2 + \frac{1}{2} \varepsilon^2 \quad (13)$$

and is negative outside

$$e_1^2 + e_2^2 > \frac{1}{k} \varepsilon_{\max}^2, \quad k = \min\left(k_1, k_2 - \frac{1}{2}\right) \quad (14)$$

Finally the control algorithm with the adaptation law (31c) was selected. Cooperation with several friction models was tested (fuzzy model with 4 membership functions and 3 different ELM models). Although all models may be used for adaptive control, it must be stressed that the more accurate fuzzy model results in smaller control signals. Taking this aspect into account, the ELM model with 10-neurons, enlarged variability of AFs and regularization (presented in fig. 19) was the best one. The application of models with big values of output weights is sensitive to numerical aspects of control implementation – the discretization time must be shortened.

The proposed modeling technique was also tested on the real life data collected from a linear permanent magnet motor connected with a pneumatic linear motor. So at least three sources of friction are present: 1) the friction in liner bearings of the electric motor, 2) the friction in linear bearings of the pneumatic motor trolley, 3) the friction between the pneumatic motor cylinder and piston. The results confirm conclusions from simulations presented here.

## 7 Conclusions

The conclusions from the presented model comparison are collected in table 2. The presented grade concerns the friction modeling problem, but some conclusions seem to be general for curve approximation by ELM.

- The increase of number of hidden neurons is often limited by numerical instability or brings insufficient benefits because of noisy data. Engineering applications with moderate number of neurons will provide appropriate accuracy of models and facilitate practical implementations.
- The regularization itself is often not sufficient to improve the conditioning and strongly spoils the accuracy.
- The sufficient variability of the applied activation functions is very important. It does pay to modify the random choice of parameters to improve the variability and to make the feature mapping more effective.

Table 2. Suitability of various models for multisource friction modeling

	Accuracy	Generalization	Training time	Numerical conditioning and stability
Fuzzy model	sufficient	poor	moderate	good
Basic ELM	sufficient	moderate	short	bad
ELM with regularization	poor	poor	short	good - if we compromise with the accuracy
ELM with enlarged variability of AFs	good	good	short	good
ELM with enlarged variability of AFs + regularization	good	the best	short	the best
ELM enlarged variability of AFs and conditioning optimization	good	good	long	good

The presented research demonstrates that extreme learning machine provides an interesting alternative to fuzzy modeling, although the application is not so universal and straightforward as it is recommended by many publications.

## 8 References

1. Gao Huang, Guang-Bin, Huang, Shiji Song, Keyou You: Trends in extreme learning machines: A review, *Neural Networks*, vol. 61, pp. 32-48, (2015)



2. Xia Liu; Shaobo Lin; Jian Fang; Zongben Xu: Is Extreme Learning Machine Feasible? A Theoretical Assessment (Part I), *Neural Networks and Learning Systems*, IEEE Transactions on , vol.26, pp.7-20, (2015)
3. Shaobo Lin; Xia Liu; Jian Fang; Zongben Xu: Is Extreme Learning Machine Feasible? A Theoretical Assessment (Part II), *Neural Networks and Learning Systems*, IEEE Transactions on , vol.26, , pp.21-34, (2015)
4. Zhixiang X. Chen, Houying Y. Zhu, Yuguang G. Wang: A modified extreme learning machine with sigmoidal activation functions, *Neural Comput & Applic*, vol.22, pp. 541–550, (2013).
5. Parviainen, E., Riihimäki, J.: Connection between Extreme Learning Machine and Neural Network Kernel, in *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, Springer Berlin Heidelberg, pp. 122-135, (2013)
6. Wojewoda, J., Stefanski, A., Wiercigroch, M., Kapitaniak, T.: Hysteretic effects of dry friction: modelling and experimental studies. *Phil. Trans. R. Soc. A* 366, pp. 747–765, (2008)
7. Kabziński, J.: One-Dimensional Linear Local Prototypes for Effective Selection of Neuro-Fuzzy Sugeno Model Initial Structure, *IFIP WG 12.5 International Conference "Artificial Intelligence Applications and Innovations*. Larnaca, Cyprus Springer, IFIP Series Berlin, pp. 62 – 69, (2010)
8. Kabzinski,J.: Fuzzy Friction Modeling for Adaptive Control of Mechatronic Systems. *Artificial Intelligence Applications and Innovations*. IFIP Advances in Information and Communication Technology, vol. 381, pp.185-195, Springer, Berlin Heidelberg (2012)
9. Kabziński J., Kacerka J.: TSK Fuzzy Modeling with Nonlinear Consequences. in: *Artificial Intelligence Applications and Innovations IFIP Advances in Information and Communication Technology*. p.498-507 , Springer, Berlin Heidelberg (2014)
10. Jang J.R.: ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* 23, pp. 665 – 684 (1993)
11. Khalil H.: *Nonlinear Systems*, Macmilan Publishing Co., New York (1992)