

Article

Approximation of Permanent Magnet Motor Flux Distribution by Partially Informed Neural Networks

Marcin Jastrzębski *  and Jacek Kabziński 

Institute of Automatic Control, Lodz University of Technology, 90-537 Lodz, Poland; jacek.kabzinski@p.lodz.pl

* Correspondence: marcin.jastrzebski@p.lodz.pl

Abstract: New results in the area of neural network modeling applied in electric drive automation are presented. Reliable models of permanent magnet motor flux as a function of current and rotor position are particularly useful in control synthesis—allowing one to minimize the losses, analyze motor performance (torque ripples etc.) and to identify motor parameters—and may be used in the control loop to compensate flux and torque variations. The effectiveness of extreme learning machine (ELM) neural networks used for approximation of permanent magnet motor flux distribution is evaluated. Two original network modifications, using preliminary information about the modeled relationship, are introduced. It is demonstrated that the proposed networks preserve all appealing features of a standard ELM (such as the universal approximation property and extremely short learning time), but also decrease the number of parameters and deal with numerical problems typical for ELMs. It is demonstrated that the proposed modified ELMs are suitable for modeling motor flux versus position and current, especially for interior permanent magnet motors. The modeling methodology is presented. It is shown that the proposed approach produces more accurate models and provides greater robustness against learning data noise. The execution times obtained experimentally from well-known DSP boards are short enough to enable application of derived models in modern algorithms of electric drive control.

Keywords: permanent magnet synchronous motor; magnetic flux model; neural networks; extreme learning machine



Citation: Jastrzębski, M.; Kabziński, J. Approximation of Permanent Magnet Motor Flux Distribution by Partially Informed Neural Networks. *Energies* **2021**, *14*, 5619. <https://doi.org/10.3390/en14185619>

Academic Editor: Ryszard Palka

Received: 19 July 2021

Accepted: 2 September 2021

Published: 7 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

According to [1], about 45% of all electricity is consumed by electric motors. It is commonly understood that the greatest potential for improving energy efficiency can be found in the intelligent use of electrical energy. For this reason, it is important to constantly improve control algorithms that allow for minimizing losses in electric motors. The increasing number of electric vehicles seems to confirm this thesis. Permanent magnet synchronous motors (PMSMs), and especially interior permanent magnet synchronous motors (IPMSMs), provide high torque-to-current ratio, high power-to-weight ratio and high efficiency together with compact and robust construction, especially when compared to asynchronous motors. As the magnetization of the rotor of a permanently excited synchronous motor is not generated via in-feed reactive current, but by permanent magnets, the motor currents are lower. This results in better efficiency than can be obtained with a corresponding asynchronous motor [2].

Further improvement can be achieved by smart control of speed-variable drives with IPMSMs. The maximum torque of the traction motor and minimum energy losses can be guaranteed by using the maximum torque per ampere (MTPA) control strategy [3,4]. Such control requires reliable information about the motor flux distribution. Modeling the motor flux is a complex problem and finding an accurate but practically applicable analytical model remains an unfulfilled dream of control engineers.

If a complete motor construction is known, it is possible to derive the d/q-axis flux linkage distribution of the PMSM model using finite element methods (FEMs). It is

necessary to know not only all motor dimensions, for instance, precise magnet location in the rotor, but also the physical parameters of all materials used. This information is usually restricted and manufacturers never publish confidential data of the motor design. Therefore the FEM analysis method, which is able to provide tabulated data on motor magnetic flux as a function of current and rotor angle, is mainly used in the motor design and optimization process. Numerous examples of this approach may be found in the huge bibliography of the subject—works [5–8] are typical of thousands of papers concerning interior permanent magnet machine designs published in the 21st century. Analytical calculation of d- and q-axis flux distributions is also possible [9,10], but it requires almost the same knowledge as numerical FEM analysis and the acceptance of rigorous assumptions. Therefore, several methods were developed to model the flux linkages from numerical data obtained from experiments conducted with a real machine. For instance, analysis of phase back electromotive force (EMF) provides information about flux as a function of position angle and flux harmonic representation [11], although information on the current influence (saturation) is lost. Flux identification may be treated as an optimization problem—d/q axis voltages and torques calculated from assumed flux are compared to the data obtained from the real machine [12].

The process of designing an effective controller for a drive with a real permanent magnet motor consists of three main steps:

1. collecting the numerical data which allow for identifying the model of the drive, including the model of the motor,
2. constructing and identifying the model using the data,
3. designing the controller according to the control aim.

The data collected at the first stage may be inaccurate (such as nominal resistances and inductances) and disturbed by measurement noise and outliers (like almost all data collected by measurement), but some of them may be precise and reliable—such as a number of pole pairs or a pitch size.

This paper concentrates on the second stage. It is assumed and explained that the description of d/q flux as a function of current and rotor position (let us call them flux surfaces) is an important component of the complete model of the drive. Obtaining flux surfaces from numerical, discrete data which may be corrupted by noise and outliers is the main problem addressed here. A new method of obtaining such a practical description is proposed and investigated. The main challenge for the presented idea is:

- to develop an artificial neural model of flux distribution,
- to equip the neural network modeling the flux with any available reliable information about the motor,
- to obtain a fast and accurate model allowing practical applications.

To face this challenge, we propose an artificial neural model of motor flux surfaces based on an extreme learning machine (ELM) approach. We demonstrate the effectiveness of ELM neural networks for the approximation of permanent magnet motor flux distribution. We introduce two original modifications of the network, using a priori information about the modeled relationship. Therefore, the novelty of the presented contribution is twofold:

- a new method of neural network approximation of discrete data is proposed, which improves the accuracy of approximation by including any preliminary, reliable information into the network structure,
- a new, convenient method of d/q flux distribution modeling is proposed, its reliability is tested and demonstrated and practical applicability is demonstrated.

Use of the flux distribution information by the controller is a separate problem. It is evident that exact information about the flux will enable more effective control aiming at torque ripple elimination—one of the most important problems for interior permanent magnet machines (see [13] or [14] as exemplary recent research in this field). Controller synthesis based on the flux models developed here is a separate task and remains outside the scope of this paper. However, let us mention the main benefits of using the proposed

neural networks in the control algorithm. First, as the training process of the developed network is very fast, the model constructed offline can be improved online if more data are collected. Next, the proposed model is a linear-in-parameters one (if the weights of the last layer are considered as parameters) and this type of model is especially attractive for adaptive controller design. This feature of neural models was intensively used in our previous works [15–17].

In the next section, the problem of efficient flux distribution modeling is formulated and discussed. Section 3 contains the description of the standard ELM network. In Section 4, two original modifications of the standard network are introduced and explained. Numerical experiments allowing us to compare the effectiveness of the proposed modeling techniques and their DSP applicability are presented in Section 5. Finally, conclusions are presented.

2. Motor Flux Distribution Modeling

If an ideally sinusoidal flux distribution is assumed (meaning: sinusoidal radial flux density in the air gap), a motor is modeled by well-known simplified equations in the rotor oriented reference frame (notation is explained in Table 1):

$$\begin{aligned} L_d \frac{d}{dt} i_d &= \omega_e L_q i_q + \omega_e \psi_q - R_s i_d + V_d, \\ L_q \frac{d}{dt} i_q &= \omega_e L_d i_d - \omega_e \psi_d - R_s i_q + V_q. \end{aligned} \quad (1)$$

In this case, flux-related parameters are constant:

$$\psi_q = 0, \psi_d = \text{const}. \quad (2)$$

Designing a controller based on the MTPA strategy for such a simplified model is a fairly well-known task and numerous versions of this approach are described [18–20].

Table 1. Signals and parameters in permanent magnet motor model.

Notation	Signal/Parameter	Unit
L_d	d-axis inductance	[H]
L_q	q-axis inductance	[H]
R_s	phase resistance	[Ω]
ψ_d, ψ_q	d and q permanent magnet flux components	[Vs/rad]
$\Psi_d = \Psi_{di} + \psi_d$	d current dependant and magnet flux components	[Vs/rad]
$\Psi_q = \Psi_{qi} + \psi_q$	q current dependant and magnet flux components	[Vs/rad]
$\tilde{L}_d = \frac{\delta}{\delta i_d} \Psi_{di}$	d flux derivative with respect to currents	[H]
$\tilde{L}_q = \frac{\delta}{\delta i_q} \Psi_{qi}$	q flux derivative with respect to currents	[H]
θ_e	electric rotor position	[rad]
$\omega_e = \frac{d}{dt} \theta_e$	electric rotor velocity	[rad/s]
V_d, V_q	d- and q-axis voltages	[V]

Unfortunately, in a real permanent magnet motor, the flux distribution is never perfectly sinusoidal, even if the manufacturer claims it is. Such flux distortions cause higher harmonics in no-load EMF and torque deformations. The influence of motor construction on the flux distribution, EMF, and torque are well described [21–23]. Especially, for interior permanent magnet motors, where the magnets are embedded inside the rotor, deformation of flux distribution is non-negligible and the simplified model (1) is unacceptable. In this case, each flux component is a non-linear function of current and rotor position [24], so a more reliable model is [25]:

$$\begin{aligned}\tilde{L}_d \frac{d}{dt} i_d &= \omega_e L_q i_q + \omega_e \Psi_q(i_d, \theta_e) - R_s i_d + V_d, \\ \tilde{L}_q \frac{d}{dt} i_q &= \omega_e L_d i_d - \omega_e \Psi_d(i_d, \theta_e) - R_s i_q + V_q.\end{aligned}\quad (3)$$

The flux components $\Psi_q(i_q, \theta_e)$, $\Psi_d(i_d, \theta_e)$ used in (3) may be complicated, non-linear functions, describing surfaces with multiple extremes.

Still, even in this situation, it is possible to minimize the losses by an MTPA approach (see, for example [26]), although it is very beneficial to have a reliable flux distribution model and to be able to compensate deformations. Having trustworthy models of flux surfaces is particularly useful in numerous applications. In addition to the energy aspect, it allows for analyzing the motor performance (torque ripples, for example) and to identify motor parameters [27,28]. They may also be used in the control loop to compensate flux and torque variations [28–31]. As analytical models obtained from field equations are too complex for practical or online applications, we concentrate on artificial neural network models.

Several methods may be used to create numerical data representing the surfaces $\Psi_d(i_d, \theta_e)$, $\Psi_q(i_q, \theta_e)$: starting from detailed 3D modeling of the motor magnetic field, ending with observers of a different type, for example, as described in [27,28,32,33]. All these methods produce data degraded (to a certain degree) by noise or outliers. An exemplary set of data is presented in Figure 1. It was obtained from a permanent magnet synchronous motor with B-202-C-21 rotor embedded permanent magnets manufactured by Kollmorgen. The diagram and the photo of the rotor are presented in Figure 2. The motor parameters are given in Table 2.

Although both flux surfaces are complex and non-linear, some regularities and repetitions are easily visible. The periodicity follows from the motor construction constant and known distance between the poles.

Table 2. Motor parameters.

Notation	Signal/Parameter
Rated power	1.5 kW
Rated velocity	6200 r/min
Rated torque	2.59 Nm
Rated current	5 A
Inertia	10^{-5} kgm ²
EMF constant	0.1147 Vs
Torque constant	0.49 Nm/A _{RMS}
Phase resistance	2.34 Ω
Phase inductance	25 mH

The shortest possible time of training is a crucial feature of a network, as the model is supposed to operate online or as a part of an embedded controller. Therefore, we decided to apply the so-called extreme learning machine (ELM) [34,35]—a neural network with activation function parameters selected at random. The most important advantage of the ELM is the extremely short training time, as training means solving a linear mean square problem and it is carried out in just one algebraic operation.

We start with a presentation of a classical ELM and discuss the effectiveness of its application to the motor flux modeling problem. Motivated by the features of this problem, having in mind the well-known drawbacks of ELMs, we present two new network structures that allow us to use the available information about the data effectively, preserving the simplicity and short training time.

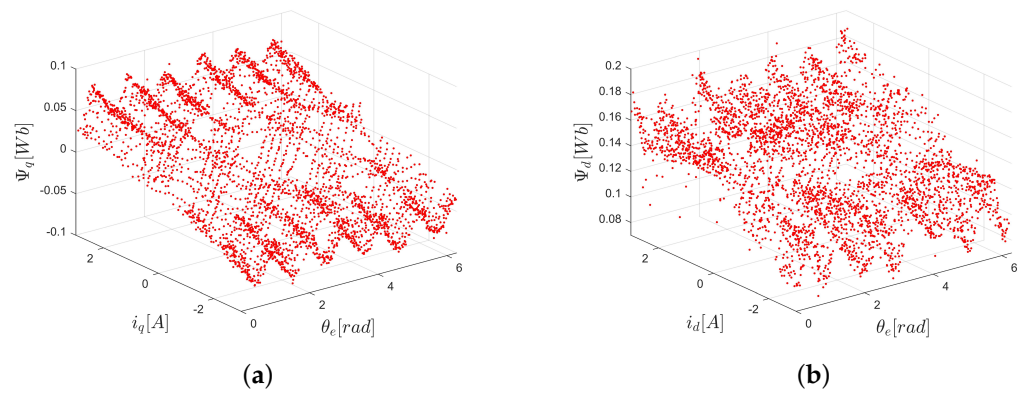


Figure 1. Example of flux surfaces: (a) $\Psi_d(i_d, \theta_e)$, (b) $\Psi_q(i_q, \theta_e)$. Data collected from an identification experiment.

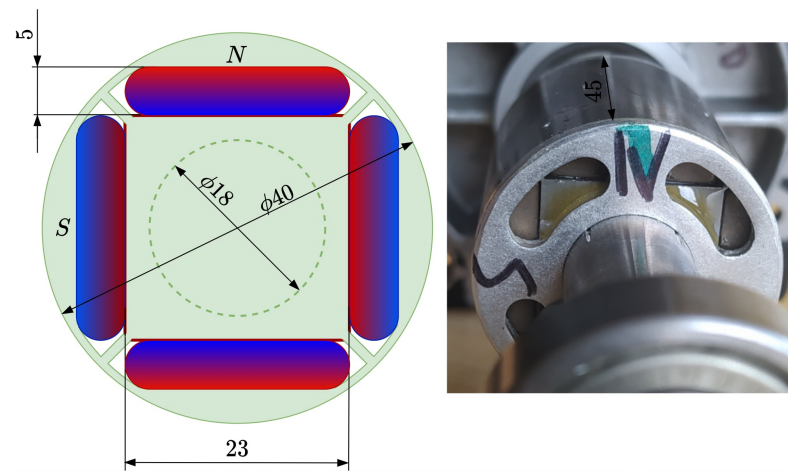


Figure 2. The rotor of the motor used in the experiment. Dimensions in millimeters.

3. Standard ELM

Architecture of a standard ELM is presented in Figure 3.

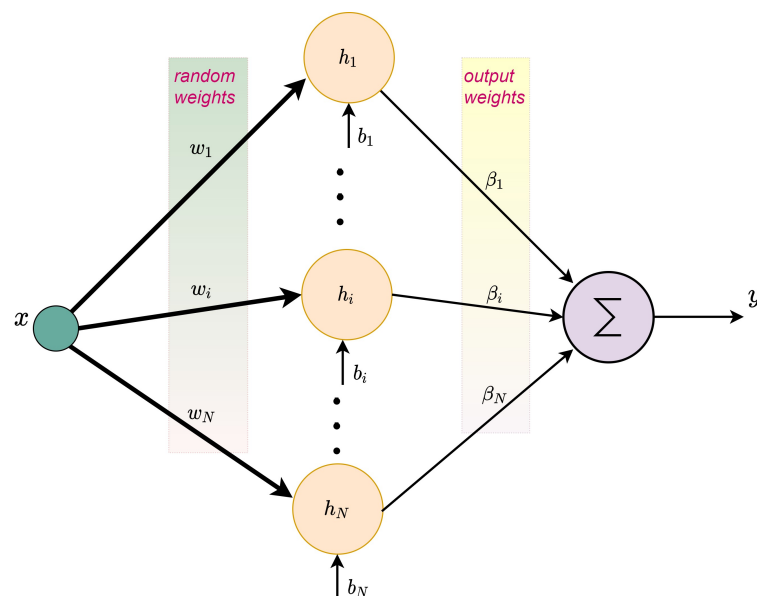


Figure 3. Architecture of a standard ELM.

As was proven [34], the selection of a particular activation function (AF) is not critical for the network performance. Assuming sigmoid AFs for all hidden neurons is commonly accepted:

$$h_i(\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x}^T \mathbf{w}_i + b_i)}. \quad (4)$$

It is typical for a standard ELM approach that the weights $\mathbf{w}_i = [w_{i,1}, \dots, w_{i,n}]^T$ and the biases b_i are selected at random, according to the uniform distribution in $[-1, 1]$ [34,35]. The training data for an n -input ELM form a batch of M samples $\mathbf{x}_k \in R^n, k = 1, \dots, M$ and corresponding desired outputs $\mathbf{t} = [t_1, \dots, t_M]^T$. It is assumed that each input is normalized to the interval $[0, 1]$. The batch of M samples is transformed by an ELM with $N \leq M$ hidden neurons into M values of output:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} h_{1,1} & \dots & h_{1,N} \\ \vdots & \ddots & \vdots \\ h_{M,1} & \dots & h_{M,N} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_N \end{bmatrix} = \mathbf{H}\boldsymbol{\beta}, \quad (5)$$

where $h_{k,i}$ is the value of the i -th AF calculated for the k -th sample and β_i is the output weight of the i -th neuron. Optimal output weights $\boldsymbol{\beta}_{opt}$ minimize the network performance index

$$E = \|\boldsymbol{\beta}\| + C\|\mathbf{H}\boldsymbol{\beta} - \mathbf{t}\|^2, \quad (6)$$

hence

$$\boldsymbol{\beta}_{opt} = \left(\frac{1}{C} \mathbf{I} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{t}. \quad (7)$$

The design parameter $C > 0$ is introduced to avoid high condition coefficients of the matrix $\mathbf{P} := \frac{1}{C} \mathbf{I} + \mathbf{H}^T \mathbf{H}$. This approach is called Tikhonov regularization [36,37]. A smaller value of C makes the structure of \mathbf{P} closer to the identity matrix, but degrades the approximation accuracy, as $\|\boldsymbol{\beta}\|$ has a stronger impact on the performance index. A high value of C makes $\boldsymbol{\beta}_{opt}$ closer to $\boldsymbol{\beta}_{min} = \mathbf{H}^+ \mathbf{t}$ where \mathbf{H}^+ is the Moore–Penrose generalized inverse of matrix \mathbf{H} . The vector $\boldsymbol{\beta}_{min}$ minimizes $E_0 = \|\mathbf{H}\boldsymbol{\beta} - \mathbf{t}\|^2$.

A standard ELM possesses the universal approximation property [34,35,37]. This means that by increasing the number of hidden neurons, we may decrease the approximation error arbitrarily. Unfortunately, the large number of neurons increases the probability that some columns of \mathbf{H} become almost co-linear, and generates numerical difficulties and high output weights. Tikhonov regularization is supposed to help, but it is not easy, or may even be impossible, to find a compromise value of parameter C . Several approaches to solve this dilemma were proposed (see [38–40] and the references therein), but none of them is perfect.

It is well recognized that insufficient variation in activation functions is responsible for numerical problems of ELMs [39,40]. Sigmoid functions with the weights and biases distributed uniformly in $[-1, 1]$ behave like linear functions in the unit hypercube and may demonstrate insignificant variation. A simple improvement was proposed in [40] and is implemented here. The first step to enlarge variation in the sigmoid activation functions is to increase the range of weights $w_{k,i} \in [-w_{max}, w_{max}]$. The weights must be large enough to expose the non-linearity of the sigmoid AF, and small enough to prevent saturation. Higher weights allow us to generate steeper surfaces and should correspond with slopes of the data.

Next, the biases are selected to guarantee that the range of each sigmoid function is sufficiently large. The minimal value of the sigmoid function $h_k(\mathbf{x})$ in the unit hypercube $0 \leq x_i \leq 1, i = 1, \dots, n$ is achieved at the vertex selected according to the following rules: $w_{k,i} > 0 \Rightarrow x_i = 0, w_{k,i} < 0 \Rightarrow x_i = 1, i = 1, \dots, n$, and equals $h_{k,min} = \frac{1}{1 + \exp(-(\sum_{i:w_{k,i} < 0} w_{k,i} + b_k))}$, while the maximum value is achieved at the vertex: $w_{k,i} > 0 \Rightarrow$

$x_i = 1, w_{k,i} < 0 \Rightarrow x_i = 0, i = 1, \dots, n$, and equals $h_{k,max} = \frac{1}{1 + \exp(-(\sum_{i:w_{k,i}>0} w_{k,i} + b_k))}$.

Therefore, if the biases are selected at random from intervals,

$$[\bar{b}, \tilde{b}] := \left[\sum_{i:w_{k,i}>0} w_{k,i} - \ln\left(\frac{1}{r_2} - 1\right), -\ln\left(\frac{1}{r_1} - 1\right) - \sum_{i:w_{k,i}<0} w_{k,i} \right], \quad (8)$$

the sigmoid function $h_k(\mathbf{x})$ has a chance to cover the interval $[r_1, r_2]$.

This approach—enhancing variation in activation functions—is applied to model flux surfaces in all experiments presented in this paper. Still, some drawbacks of the standard ELM modeling are noticeable, therefore, we propose two new network architectures to improve the modeling quality.

4. ELM with Input-Dependent Output Weights

4.1. New Network Structure

Very short learning times and the simplicity of the algorithm are the most attractive features of ELMs. The greatest disadvantage of a standard ELM is that a non-linear transformation with randomly selected parameters may be unable to represent all important features of the input space. Hence, a large number of neurons is necessary, which generates the numerical problems described above and in [39–44].

If we have some, even approximate, information about the structure of the modeled non-linearity, we may pass this knowledge through the network. As it is trusted information, we do not allow the network to modify it deeply. However, because it is still only partial, approximate, and incomplete knowledge, we accept a slight modification.

Motivated by this philosophy, we propose the following network structure. If we assume that L known non-linear functions of the input $f_l(\mathbf{x})$, $l = 1, \dots, L$ should be represented in the model output, we plug in those functions into the output weights β , making each weight a function of input. The network output is now given by

$$y = \sum_{i=1}^N h_i(\mathbf{x}) \beta_i(\mathbf{x}), \quad (9)$$

$$\beta_i(\mathbf{x}) = \beta_{i,0} + \sum_{l=1}^L \beta_{i,l} f_l(\mathbf{x}).$$

The information represented in $f_l(\mathbf{x})$, $l = 1, \dots, L$ has a strong impact on the output, but it can still be modified by coefficients $\beta_{i,l}$. The standard ELM structure is represented by the weights $\beta_{i,0}$ and the standard ELM is a special case of the structure presented in (9) obtained for $\beta_{i,l} = 0$. Therefore, the proposed network preserves the universal approximation property of the standard ELM.

Expanding Formula (9) provides

$$y = \sum_{i=1}^N h_i(\mathbf{x}) \beta_{i,0} + \sum_{i=1}^N h_i(\mathbf{x}) f_1(\mathbf{x}) \beta_{i,1} + \dots + \sum_{i=1}^N h_i(\mathbf{x}) f_L(\mathbf{x}) \beta_{i,L}. \quad (10)$$

Architecture of the proposed ELM is presented in Figure 4.

Hence, Formula (9) is equivalent to a standard ELM structure with $N(l+1)$ hidden neurons equipped with activation functions $h_i(\mathbf{x})$, $i = 1, \dots, N$, $h_i(\mathbf{x}) f_1(\mathbf{x})$, $i = 1, \dots, N$, \dots , $h_i(\mathbf{x}) f_L(\mathbf{x})$, $i = 1, \dots, N$. Sigmoid functions $h_i(\mathbf{x})$ with randomly selected parameters represent a random, non-linear transformation of inputs into the feature space, while $f_l(\mathbf{x})$, $l = 1, \dots, L$ code an assumed knowledge about the data structure. We expect that using this knowledge directly inside the network will reduce the total number of neurons required to obtain the desired modeling accuracy, compared with a standard ELM.

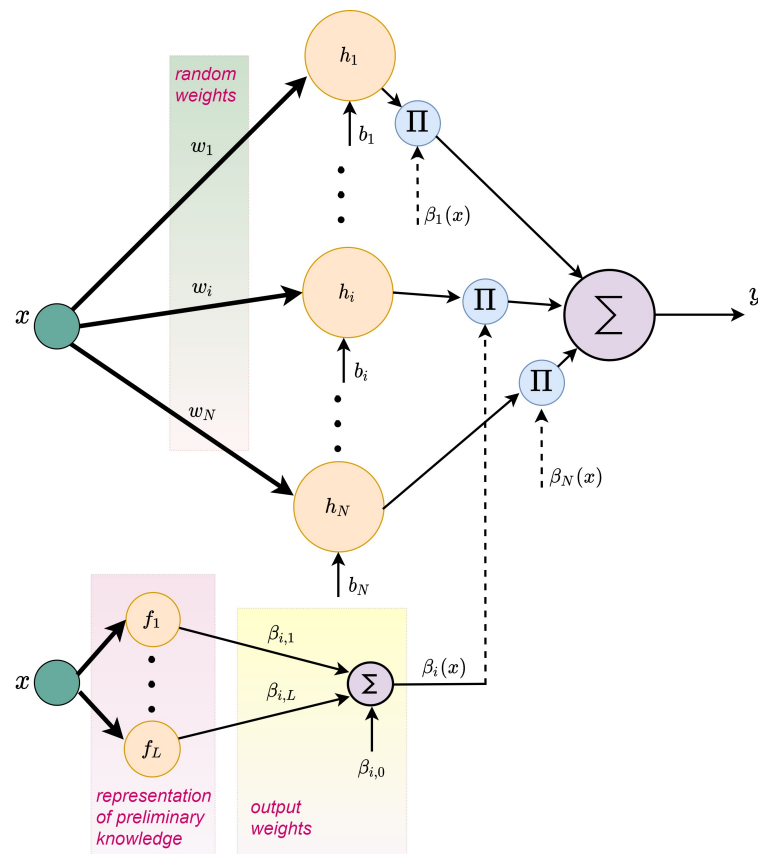


Figure 4. Architecture of an ELM with input-dependent output weights.

A batch of M samples generates M output values

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} g_{1,1} & \cdots & g_{1,N(L+1)} \\ \vdots & \ddots & \vdots \\ g_{M,1} & \cdots & g_{M,N(L+1)} \end{bmatrix} \hat{\boldsymbol{\beta}} = \mathbf{G}\hat{\boldsymbol{\beta}}, \quad (11)$$

where $\hat{\boldsymbol{\beta}} = [\beta_{1,0}, \dots, \beta_{N,0}, \beta_{1,1}, \dots, \beta_{N,1}, \dots, \beta_{1,L}, \dots, \beta_{N,L}]^T$ and $g_{i,k}$ are the value of the k -th function from the sequence

$$[g_1(\mathbf{x}), \dots, g_{N(L+1)}(\mathbf{x})] := [h_1(\mathbf{x}), \dots, h_N(\mathbf{x}), h_1(\mathbf{x})f_1(\mathbf{x}), \dots, h_1(\mathbf{x})f_L(\mathbf{x}), \dots, h_N(\mathbf{x})f_1(\mathbf{x}), \dots, h_N(\mathbf{x})f_L(\mathbf{x})] \quad (12)$$

calculated for the i -th sample. Optimal weights, minimizing the performance index $\hat{E} = \|\hat{\boldsymbol{\beta}}\| + C\|\mathbf{G}\hat{\boldsymbol{\beta}} - \mathbf{t}\|^2$, are given by

$$\hat{\boldsymbol{\beta}}_{opt} = \left(\frac{1}{C} \mathbf{I} + \mathbf{G}^T \mathbf{G} \right)^{-1} \mathbf{G}^T \mathbf{t}. \quad (13)$$

4.2. Reduction of Output Weight Number

The coefficient $h_i(\mathbf{x})$ which multiplies each function $f_l(\mathbf{x})$ in (10) depends on the actual value of input \mathbf{x} and randomly selected parameters of the activation function. So, to a certain degree, it is a random number from the interval $[0,1]$. If our aim is to simplify the network (10) and to reduce the number of output weights, while still preserving the information represented in functions $f_l(\mathbf{x})$, we may take a random gain for each $f_l(\mathbf{x})$ and

use one output weight for a group of functions $f_l(\mathbf{x})$. For example, modification of the standard ELM according to

$$y = \sum_{i=1}^N h_i(\mathbf{x}) \beta_i(\mathbf{x}), \quad (14)$$

$$\beta_i(\mathbf{x}) = \beta_{i,0} + \beta_{i,1} \sum_{l=1}^L a_{i,l} f_l(\mathbf{x}),$$

where parameters $a_{i,l}$, $l = 1, \dots, L$ are randomly selected results in

$$y = \sum_{i=1}^N h_i(\mathbf{x}) \beta_{i,0} + \sum_{i=1}^N h_i(\mathbf{x}) \left(\sum_{l=1}^L a_{i,l} f_l(\mathbf{x}) \right) \beta_{i,1}. \quad (15)$$

Architecture of the reduced ELM is presented in Figure 5.

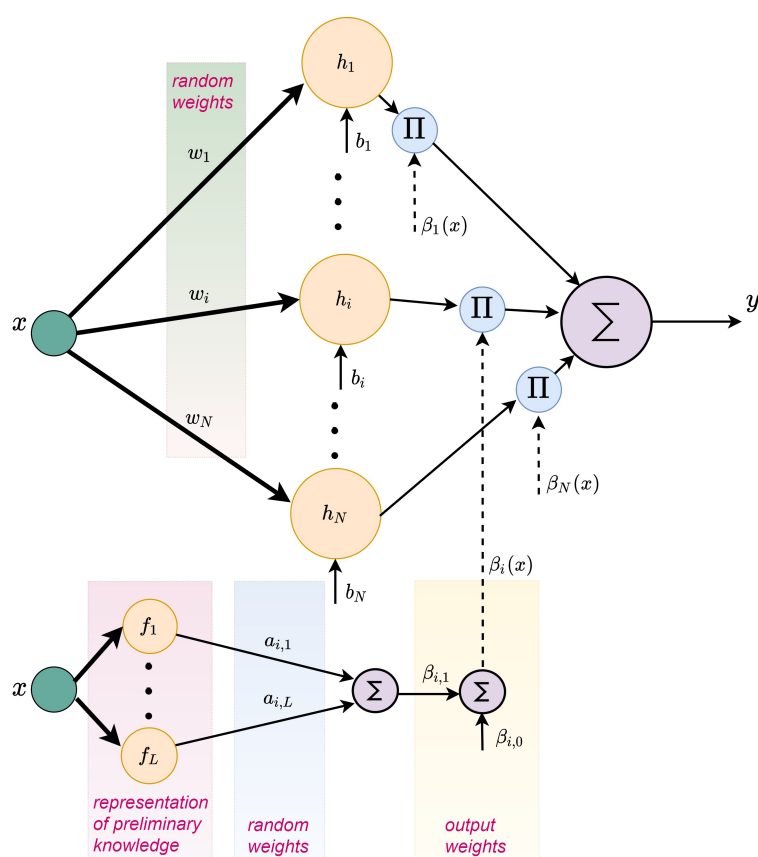


Figure 5. Architecture of a reduced ELM with input-dependent output weights.

Hence, (15) is equivalent to a standard ELM with $2N$ hidden neurons and activation function $h_i(\mathbf{x})$, $i = 1, \dots, N$ and $h_i(\mathbf{x}) \left(\sum_{l=1}^L a_{i,l} f_l(\mathbf{x}) \right)$, $i = 1, \dots, N$. Again, the standard ELM is a special case of (15), so the network (15) possesses the universal approximation property.

Responding to the batch of M samples (15) generates M output values

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} r_{1,1} & \dots & r_{1,2N} \\ \vdots & \ddots & \vdots \\ r_{M,1} & \dots & r_{M,2N} \end{bmatrix} \tilde{\boldsymbol{\beta}} = \mathbf{R} \tilde{\boldsymbol{\beta}}, \quad (16)$$

where $\tilde{\boldsymbol{\beta}} = [\beta_{1,0}, \dots, \beta_{N,0}, \beta_{1,1}, \dots, \beta_{N,1}]^T$ and $r_{i,k}$ are the value of the k -th function from the sequence

$$[r_1(\mathbf{x}), \dots, r_{2N}(\mathbf{x})] := [h_1(\mathbf{x}), \dots, h_N(\mathbf{x}), h_1(\mathbf{x}) \sum_{l=1}^L a_{1,l} f_l(\mathbf{x}), \dots, h_N(\mathbf{x}) \sum_{l=1}^L a_{N,l} f_l(\mathbf{x})] \quad (17)$$

calculated for the i -th sample. Optimal weights, minimizing the performance index $\tilde{E} = \|\tilde{\beta}\| + C\|\mathbf{R}\tilde{\beta} - \mathbf{t}\|^2$, are given by

$$\tilde{\beta}_{opt} = \left(\frac{1}{C} \mathbf{I} + \mathbf{R}^T \mathbf{R} \right)^{-1} \mathbf{R}^T \mathbf{t}. \quad (18)$$

Although the network (15) is simpler than (9), it is not necessarily less accurate for the same number of output weights.

5. Comparison of Networks

5.1. Introductory Example

To clearly demonstrate the idea of the proposed modifications, we start with a simple one-dimensional example. The curve to be approximated is given by

$$F(x) = 10f_0(x) + 5f_1(x) + 2f_2(x) = 10 \sin(2\pi x) + 5 \sin(6\pi x) + 2 \sin(14\pi x) \quad (19)$$

$$0 \leq x \leq 1.$$

The training set consists of $N_{tr} = 300$ pairs $(x_i, F(x_i))$ where x_i are equidistantly distributed in $[0,1]$, while the test data are formed by $N_{test} = 1000$ such points. The final result is judged by the mean value of

$$E_{test} = \sqrt{\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (F(x_i) - y(x_i))^2}, \quad (20)$$

where y is the model output obtained from 1000 experiments.

Three networks are compared:

- ELM1: The standard ELM given by (5), (7) with the input weights and biases selected randomly, according to the enhanced variation mechanism (8) with $r_1 = 0.1$, $r_2 = 0.9$.
- ELM2: The network with input-dependent output weights, according to (9):

$$\beta_i(x) = \beta_{i,0} + \beta_{i,1} f_1(x) + \beta_{i,2} f_2(x), \quad (21)$$

where the partial knowledge about the output is used.

- ELM3: The network modified according to (14):

$$\beta_i(x) = \beta_{i,0} + \beta_{i,1}(a_{i,1} f_1(x) + a_{i,2} f_2(x)), \quad (22)$$

where $a_{i,1}, a_{i,2}$ are selected at random from the interval $[-1,1]$.

We compare the networks with the same number of the output weights, so, if we use N hidden neurons in ELM1, the corresponding ELM2 has $N/3$, and ELM3— $N/2$ neurons.

In this simple example, all three networks work properly, in the sense that we can obtain a correct approximation from any of them. An exemplary plot is presented in Figure 6. For other networks, we obtain similar results, but important differences are illustrated in Figures 7–9.

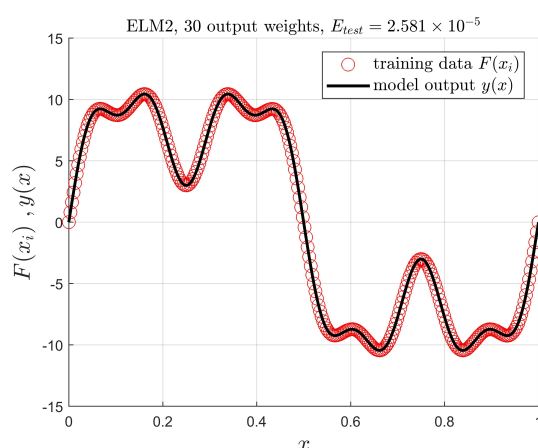


Figure 6. The modeled curve and the network response.

Enhancement of the variance of activation functions was applied for each network. Figure 7 demonstrates that a sufficient range of input weights is important, but it also illustrates that:

- The standard network (ELM1) is far more sensitive to a small range of input weights than modified networks (ELM2 and ELM3).
- The standard network (ELM1) generates a higher test error than modified networks (ELM2 and ELM3), in spite of the range of the input weights.
- The standard network (ELM1) generates much higher output weights than modified networks (ELM2 and ELM3), so the standard model demonstrates much worse numerical properties.

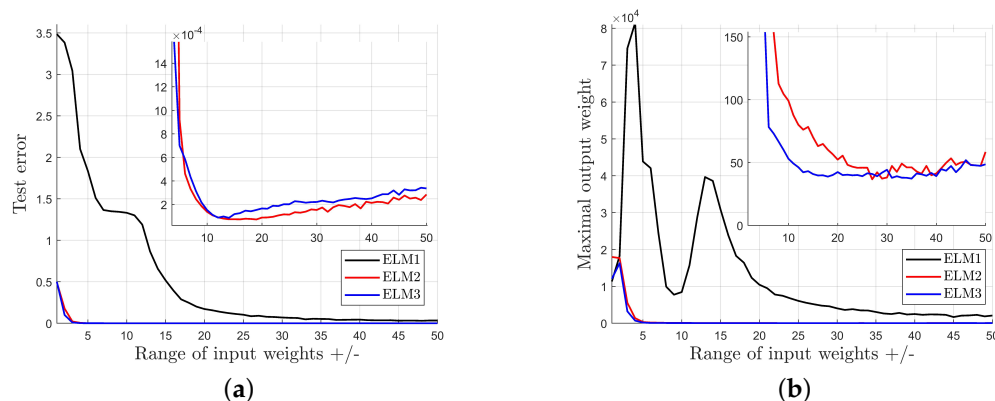


Figure 7. Test error (a) and the biggest output weight (b) as a function of the range of input weights. $N = 48$, $C = 10^8$.

The influence of the number of output weights (number of hidden neurons) is presented in Figure 8. It is shown that:

- The standard network (ELM1) generates higher test error than modified networks (ELM2 and ELM3) for the same number of output weights and requires a much larger number of hidden neurons to obtain a similar test error as ELM2 or ELM3.
- The standard network (ELM1) generates much higher output weights for any number of hidden neurons than modified networks (ELM2 and ELM3), so the standard model demonstrates much worse numerical properties.

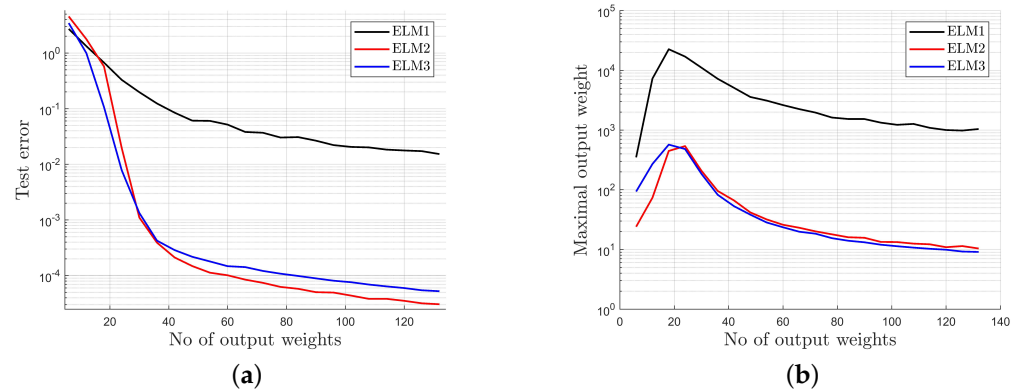


Figure 8. Test error (a) and the biggest output weight (b) as a function of the number of output weights. $w_{max} = 30$, $C = 10^8$.

The impact of the regularization parameter C is presented in Figure 9. It is evident that:

- The standard network (ELM1) generates a much higher test error for any C than modified networks (ELM2 and ELM3).
- The standard network (ELM1) requires strong regularization (small C to decrease output weights), resulting in poor modeling accuracy. The modified networks (ELM2, ELM3) preserve moderate output weights for any C —regularization is not necessary.

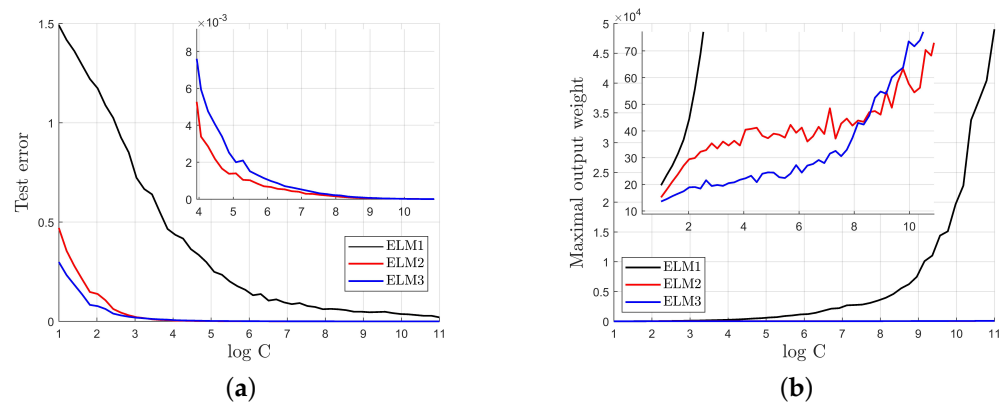


Figure 9. Test error (a) and the biggest output weight (b) as a function of the regularization coefficient C . $w_{max} = 30$, $N = 48$.

5.2. Motor Flux Modeling

To compare the performance of the discussed networks precisely, a numerically generated surface is used. The surface is presented in Figure 10, and it is similar to $\Psi_q(i_q, \theta_e)$ presented in Figure 1b. The formula generating the surface is:

$$t(x_1, x_2) = 0.1p_1(2x_2 - 1) + 0.02 \sin(12\pi x_1)(2x_2 - 1) + 0.3p_2(10x_2 - 5),$$

$$p_1(\delta) = \frac{2}{1 + \exp(-2\delta)} - 1, \quad p_2(\delta) = \exp(-\delta). \quad (23)$$

Using artificial data allows us to calculate the training and test errors accurately. For each experiment, three sets of data are generated:

- training data $t_i(x_{1,i}, x_{2,i})$, $i = 1, \dots, N_{tr}$, where $(x_{1,i}, x_{2,i})$ are randomly selected from the input area $[0,1] \times [0,1]$,
- training data corrupted by noise $t_i^*(x_{1,i}, x_{2,i}) = t_i(x_{1,i}, x_{2,i})(1 + \gamma_i)$, $i = 1, \dots, N_{tr}$, where γ_i is a random variable possessing normal distribution $N(0, 0.1)$,
- testing data $t_i^0(x_{1,i}^0, x_{2,i}^0)$, $i = 1, \dots, N_{test}$ randomly selected from the input area $[0,1] \times [0,1]$, different from the training data. $N_{test} = 3000$ is used for all experiments.

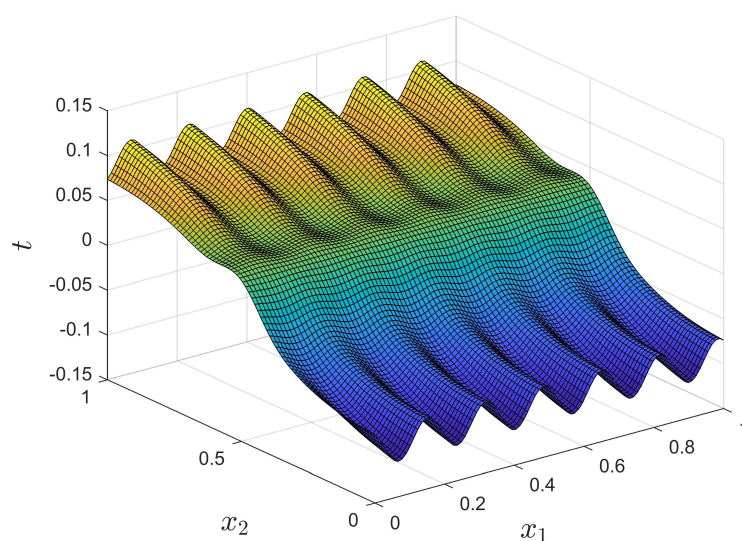


Figure 10. The flux-like surface.

The main index used to compare the networks is the test error:

$$E_{test} = \sqrt{\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left(t_i^0(x_{1,i}^0, x_{2,i}^0) - y(x_{1,i}^0, x_{2,i}^0) \right)^2}, \quad (24)$$

where y denotes the actual network output. The test error E_{test} compares the network output with accurate data, even if noisy data were used for training.

As each network depends on randomly selected parameters, 100 experiments are performed and mean values of obtained test errors are used for comparison. The average value from the modeled surface is about 0.05, so an error smaller than 0.005 provides the relative error $\sim 10\%$.

Three networks are compared:

- ELM1: The standard ELM given by (5), (7) with the input weights selected randomly, according to a uniform distribution, from the interval $[-w_{max}, w_{max}] = [-30, 30]$ and the biases selected according to (8) for $r_1 = 0.1$, $r_2 = 0.9$. This approach provides activation functions with a sufficient variance, as presented in Figure 11.
- ELM2: The network with input-dependent output weights, according to (9):

$$\begin{aligned} \beta_i(x) &= \beta_{i,0} + \beta_{i,1}f_1(x) + \beta_{i,2}f_2(x), \\ f_1(x) &= \sin(12\pi x_1) \quad f_2(x) = \cos(12\pi x_2), \end{aligned} \quad (25)$$

where the knowledge about the motor construction (number of pole pairs) is used to propose $f_1(x)$, $f_2(x)$.

- ELM3: The network with input-dependent output weights, according to (14):

$$\begin{aligned} \beta_i(x) &= \beta_{i,0} + \beta_{i,1}(a_{i,1}f_1(x) + a_{i,2}f_2(x)), \\ f_1(x) &= \sin(12\pi x_1) \quad f_2(x) = \cos(12\pi x_2), \end{aligned} \quad (26)$$

where $a_{i,1}$, $a_{i,2}$ are selected at random from interval $[-1, 1]$.

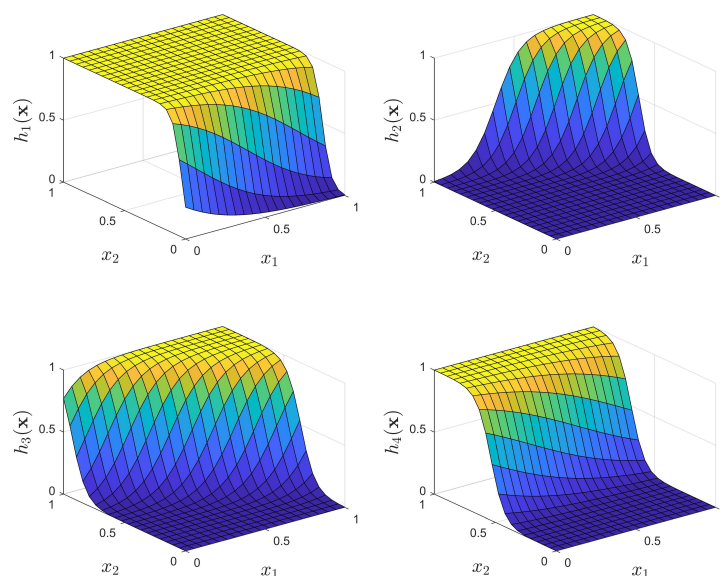


Figure 11. Exemplary activation functions of ELM1.

As carried out previously, we compare the networks with the same number of output weights, so if we use N hidden neurons in ELM1, the corresponding ELM2 has $N/3$, and ELM3— $N/2$ neurons.

The test errors of compared networks as functions of output weight number are presented in Figure 12 for noiseless data $N_{tr}=3000$ and $N_{test} = 3000$, $C = 10^{10}$, $w_{max} = 30$.

The modified networks (ELM2 and ELM3) provide significantly lower modeling errors than the standard one (ELM1). The surface generated by one of the modified networks is presented in Figure 13, it is indistinguishable from the original one plotted in Figure 10.

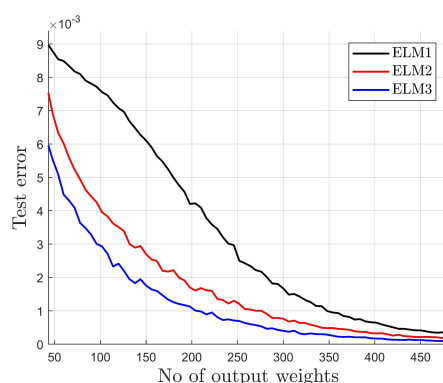


Figure 12. Test error as a function of the number of output weights.

The advantage of modified networks (ELM2 and ELM3) becomes more visible when the information about the surface is poorer. Figure 14 presents the test error as the function of the number of training points N_{tr} for $N_{test} = 3000$, $C = 10^{10}$, $w_{max} = 30$, $N = 240$.

The modified networks also offer smaller test errors if the training data are corrupted by noise. This situation is presented in Figure 15 for $N_{tr} = 3000$ and $N_{test} = 3000$, $C = 10^{10}$, $w_{max} = 30$. The advantage of modified networks is even more significant, as in the presence of the noisy data it is impossible to increase the number of neurons arbitrarily. A large number of hidden neurons causes an increase in the test error as the network loses generalization properties due to overfitting. The analysis of the error surface structure demonstrates that the smaller number of neurons gives a smoother modeling error surface with a smaller number of local extremes.

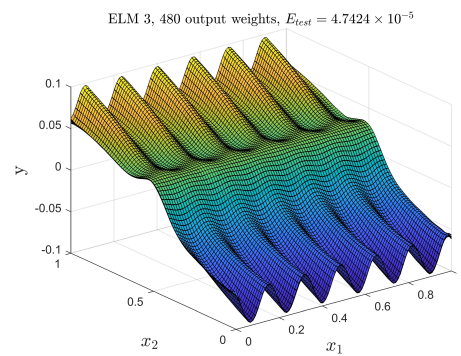


Figure 13. The surface generated by one of the modified networks.

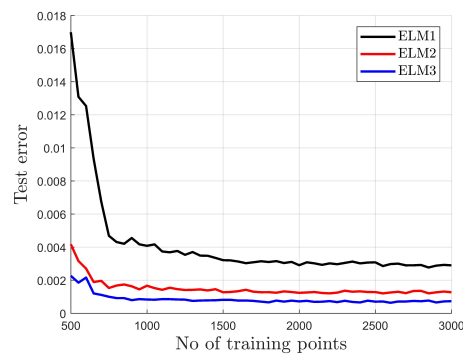


Figure 14. Test error as a function of the number of training points.

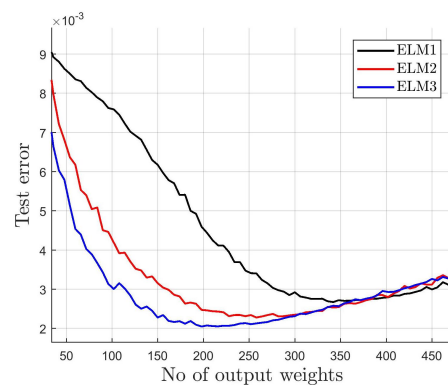


Figure 15. Test error as a function of the number of output weights. The training data corrupted by noise.

5.3. Modeling of Experimental Data

A similar comparison of the networks is repeated with the experimental data presented in Figure 1. The data set was divided into the training data $N_{tr} = 20,000$ and the test data $N_{test} = 20,000$. Of course, as the accurate value of the flux is unknown, the test error is defined with respect to the experimental data from the test data set. The training and the test error behave similarly, so only the training error is plotted in Figure 16. The result of the comparison is similar: for both flux surfaces, the modified networks (ELM2 and ELM3) provide significantly lower modeling errors than the standard one (ELM1).

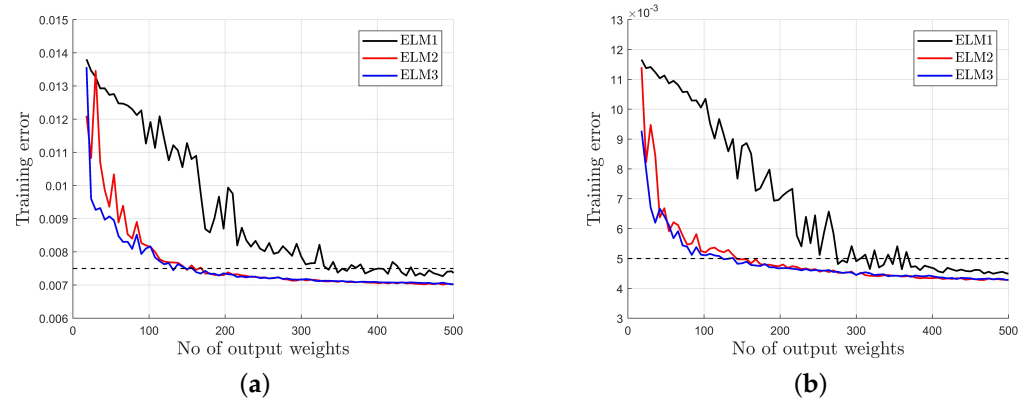


Figure 16. Modeling of: (a) $\Psi_d(i_d, \theta_e)$ and (b) $\Psi_q(i_q, \theta_e)$. Training error as a function of the number of output weights.

The number of hidden neurons and output weights necessary to obtain a training error smaller than 0.0075 for $\Psi_d(i_d, \theta_e)$ or smaller than 0.0050 for $\Psi_q(i_q, \theta_e)$ is presented in Table 3.

Table 3. Number of hidden neurons and output weights necessary to obtain the desired training error value.

Axis		ELM1	ELM2	ELM3
q	No of hidden neurons	336	50	64
	No of output weights	336	150	128
d	No of hidden neurons	276	48	63
	No of output weights	276	144	126

The surfaces generated by the networks described in Table 3 are presented in Figures 17 and 18. Of course, currents and position are rescaled to the original range in amperes and radians. Application of networks ELM2 and ELM3, where the knowledge about the motor construction is used, allows us to smooth the modeled surface, to reduce the number of extremes, and to obtain a continuous transition at $\theta_e = 0/\theta_e = 2\pi$. The models obtained from the modified networks (ELM2 and ELM3) are more regular, with a smaller number of extremes.

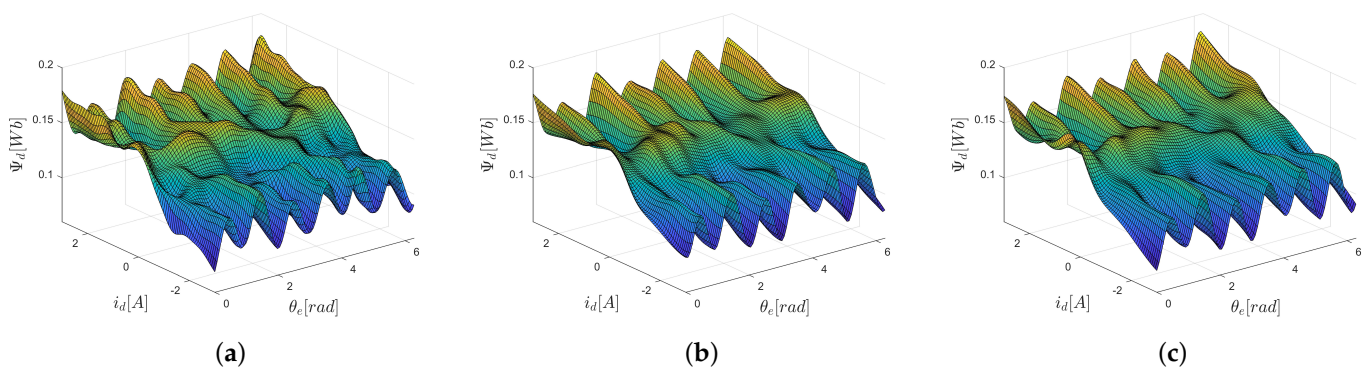


Figure 17. Models of $\Psi_d(i_d, \theta_e)$ generated by: (a) ELM1, (b) ELM2, (c) ELM3.

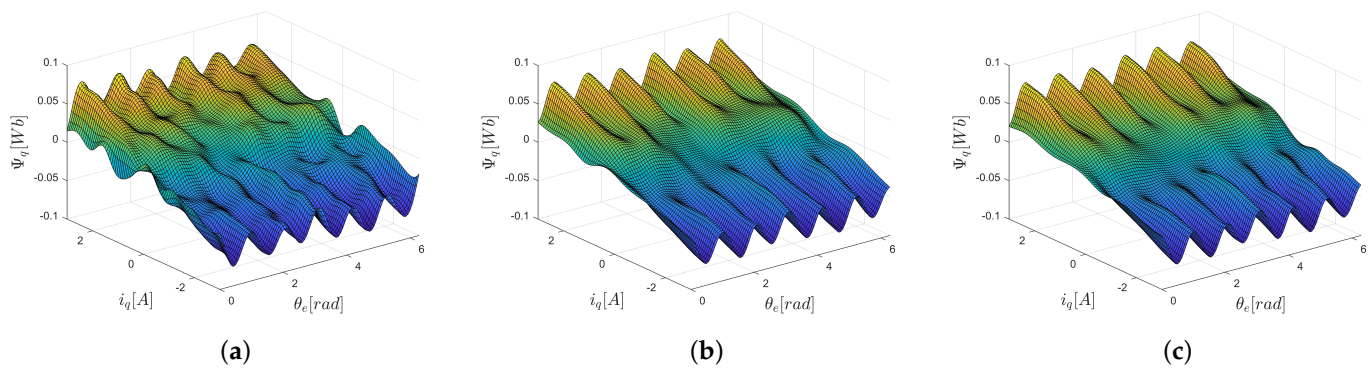


Figure 18. Models of $\Psi_q(i_q, \theta_e)$ generated by: (a) ELM1, (b) ELM2, (c) ELM3.

The time history of the flux generated for a given current–position trajectory is presented in Figure 19. The flux obtained from the modified networks (ELM2 and ELM3) is continuous for $\theta_e = n2\pi$, while the one generated from the standard network (ELM1) is not. Therefore, the model generated from ELM1 violates physical principles of motor operation, while modified networks benefit from the knowledge coded in functions $f_1(x)$ and $f_2(x)$ included in the network.

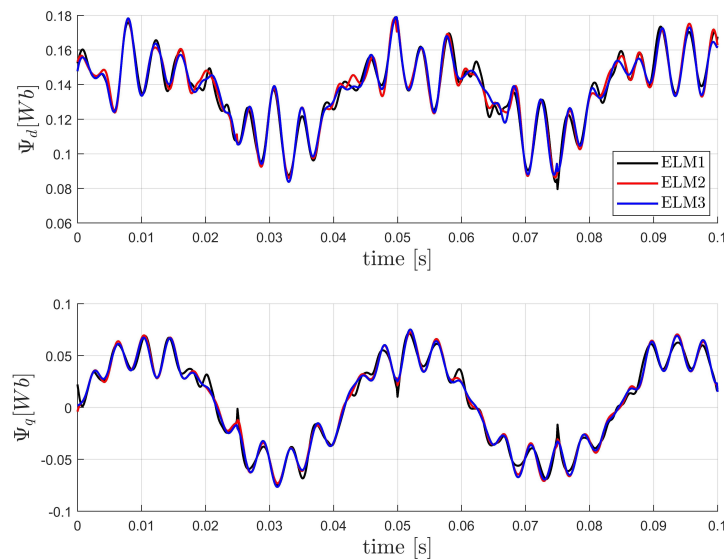


Figure 19. The time history of the flux generated for a given current–position trajectory.

The signal presented in Figure 19 is not periodical. It is a time history of the flux generated under variable speed and current. Therefore, Fourier analysis of this signal does not provide any useful information. Instead of this, fast Fourier transform (FFT) analysis of the outputs of the considered models obtained for a constant velocity and for several fixed values of the q-axis current is presented. In this case, the flux is a periodical function of a rotor position. The results of such an analysis, shown in Figure 20, confirm the existence of the sixth harmonic with high amplitude, which is in line with the expectations. The model created with the use of the standard network (ELM1) generates non-negligible content of higher harmonics, especially for small current values. The presence of all subsequent higher harmonics in the FFT of the output of ELM1 demonstrates an undesirable ability to generate noise by this model. The modified networks ELM2 and ELM3 generate waveforms with a lower content of higher harmonics, which is undoubtedly an advantage.

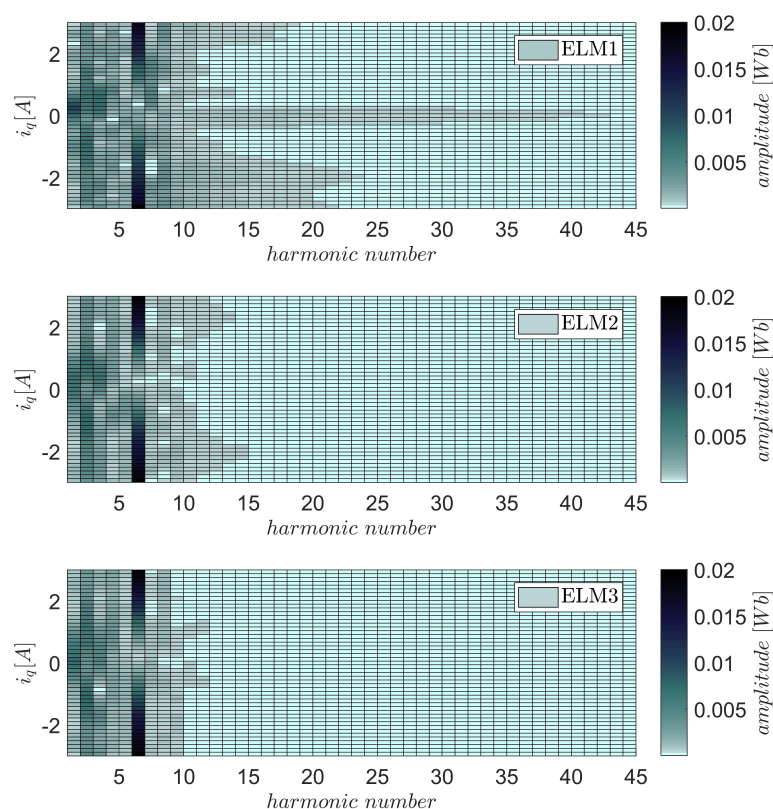


Figure 20. Amplitude of harmonics in q-axis flux generated from models ELM1–3 for a constant speed and several constant values of q-axis current.

Finally, a practical, hardware implementation of the proposed networks was considered. The time necessary for model execution on some popular DSP boards is presented in Table 4. The obtained results encourage the implementation of ELMs in control algorithms.

Table 4. Time necessary for model execution.

DSP Board	ELM1	ELM2	ELM3
DS1104	470 μ s	70 μ s	70 μ s
DS1006	30 μ s	12 μ s	12 μ s

6. Conclusions

Both proposed modified structures of the ELM allow for incorporating preliminary, partial, imprecise information about modeled data into the network structure. It was demonstrated that the modified network may be interpreted as a network with input-dependent output weights, or as a network with modified activation functions. The proposed approach preserves all the attractive features of the standard ELM:

- the universal approximation property,
- fast, random selection of parameters of activation functions,
- extremely short learning time, as learning is not an iterative process, but is reduced to a single algebraic operation.

It was demonstrated by numerical examples that both modified networks outperform the standard ELM:

- offering better modeling accuracy for the same number of output weights and a smaller number of parameters, while assuring the same accuracy, therefore reducing the problem of dimensionality,

- generating lower output weights and better numerical conditioning of output weight calculation,
- being more flexible for Tikhonov regularization,
- being more robust against data noise,
- being more robust against small training data sets.

It is difficult to decide which of the proposed modifications is “better”. For the presented examples, they are comparable and the selection of one of them depends on the specific features of a particular problem.

It was shown that the proposed modified ELMs are suitable for modeling motor flux versus position and current, especially for interior permanent magnet motors. The modeling methodology was presented. The extra information about flux surfaces is available from the motor construction (pole pitch) and may be easily included. The obtained models preserve flux continuity around the rotor, and provide good agreement with measured signals (like torque and EMF), so they may be considered trustworthy.

The modified networks provide significantly lower modeling errors than the standard one and this feature becomes more visible when the information about the surface is poorer (fewer samples are available), or the training data are corrupted by noise. FFT analysis of the networks’ periodical outputs demonstrates that the modified networks generate more reliable spectra, corresponding to theoretical expectations, while the standard one generates a visible amount of high-harmonic noise.

The obtained neural models may be used for control or identification, working online. The execution times obtained from well-known DSP boards are short enough for modern algorithms of electric drive control. Therefore, we claim that the recent control methods of PMSM drives might be improved by taking flux deformations into account.

Author Contributions: Conceptualization, M.J.; methodology, M.J. and J.K.; formal analysis, M.J. and J.K.; writing—original draft preparation, M.J. and J.K.; writing—review and editing, M.J. and J.K. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Lodz University of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors are grateful to Tomasz Sobieraj for his support in obtaining experimental data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Waide, P.; Brunner, C.U. *Energy-Efficiency Policy Opportunities for Electric Motor-Driven Systems*; International Energy Agency: Paris, France, 2011; 132p. [\[CrossRef\]](#)
2. Melfi, M.J.; Rogers, S.D.; Evon, S.; Martin, B. Permanent Magnet Motors for Energy Savings in Industrial Applications. In Proceedings of the 2006 Record of Conference Papers—IEEE Industry Applications Society 53rd Annual Petroleum and Chemical Industry Conference, Philadelphia, PA, USA, 11–15 September 2006; pp. 1–8. [\[CrossRef\]](#)
3. Baek, S.K.; Oh, H.K.; Park, J.H.; Shin, Y.J.; Kim, S.W. Evaluation of Efficient Operation for Electromechanical Brake Using Maximum Torque per Ampere Control. *Energies* **2019**, *12*, 1869. [\[CrossRef\]](#)
4. Tochenko, O. Energy Efficient Speed Control of Interior Permanent Magnet Synchronous Motor. In *Applied Modern Control*; IntechOpen: London, UK, 2019. [\[CrossRef\]](#)
5. Mellor, P.; Wrobel, R.; Holliday, D. A Computationally Efficient Iron Loss Model for Brushless AC Machines That Caters for Rated Flux and Field Weakened Operation. In Proceedings of the 2009 IEEE International Electric Machines and Drives Conference, Miami, FL, USA, 3–6 May 2009; pp. 490–494. [\[CrossRef\]](#)
6. Tseng, K.; Wee, S. Analysis of Flux Distribution and Core Losses in Interior Permanent Magnet Motor. *IEEE Trans. Energy Convers.* **1999**, *14*, 969–975. [\[CrossRef\]](#)
7. Chakkarapani, K.; Thangavelu, T.; Dharmalingam, K.; Thandavarayan, P. Multiobjective Design Optimization and Analysis of Magnetic Flux Distribution for Slotless Permanent Magnet Brushless DC motor using evolutionary algorithms. *J. Magn. Magn. Mater.* **2019**, *476*, 524–537. doi:10.1016/j.jmmm.2019.01.029. [\[CrossRef\]](#)

8. Lee, J.G.; Lim, D.K. A Stepwise Optimal Design Applied to an Interior Permanent Magnet Synchronous Motor for Electric Vehicle Traction Applications. *IEEE Access* **2021**, *9*, 115090–115099. [\[CrossRef\]](#)
9. Liang, P.; Pei, Y.; Chai, F.; Zhao, K. Analytical Calculation of D- and Q-axis Inductance for Interior Permanent Magnet Motors Based on Winding Function Theory. *Energies* **2016**, *9*, 580. [\[CrossRef\]](#)
10. Michalski, T.; Acosta-Cambranis, F.; Romeral, L.; Zaragoza, J. Multiphase PMSM and PMaSynRM Flux Map Model with Space Harmonics and Multiple Plane Cross Harmonic Saturation. In Proceedings of the IECON 2019—45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 14–17 October 2019; Volume 1, pp. 1210–1215. [\[CrossRef\]](#)
11. Hua, W.; Cheng, M.; Zhu, Z.Q.; Howe, D. Analysis and Optimization of Back EMF Waveform of a Flux-Switching Permanent Magnet Motor. *IEEE Trans. Energy Convers.* **2008**, *23*, 727–733. [\[CrossRef\]](#)
12. Khan, A.A.; Kress, M.J. Identification of Permanent Magnet Synchronous Motor Parameters. In Proceedings of the WCX™ 17: SAE World Congress Experience, Detroit, MI, USA, 4–6 April 2017; SAE International: Warrendale, PA, USA, 2017. doi:10.4271/2017-01-1237. [\[CrossRef\]](#)
13. Kumar, P.; Bhaskar, D.V.; Muduli, U.R.; Beig, A.R.; Behera, R.K. Disturbance Observer based Sensorless Predictive Control for High Performance PMBLDCM Drive Considering Iron Loss. *IEEE Trans. Ind. Electron.* **2021**. [\[CrossRef\]](#)
14. Rehman, A.U.; Choi, H.H.; Jung, J.W. An Optimal Direct Torque Control Strategy for Surface-Mounted Permanent Magnet Synchronous Motor Drives. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7390–7400. [\[CrossRef\]](#)
15. Jastrzębski, M.; Kabziński, J.; Mosiolek, P. Adaptive and Robust Motion Control in Presence of LuGre-type Friction. In Proceedings of the 2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 28–31 August 2017; pp. 113–118. [\[CrossRef\]](#)
16. Kabziński, J.; Orłowska-Kowalska, T.; Sikorski, A.; Bartoszewicz, A. Adaptive, Predictive and Neural Approaches in Drive Automation and Control of Power Converters. *Bull. Pol. Acad. Sci. Tech. Sci.* **2020**, *68*, 959–962. [\[CrossRef\]](#)
17. Kabziński, J. *Advanced Control of Electrical Drives and Power Electronic Converters*; Springer: Cham, Switzerland, 2017. [\[CrossRef\]](#)
18. Sun, J.; Lin, C.; Xing, J.; Jiang, X. Online MTPA Trajectory Tracking of IPMSM Based on a Novel Torque Control Strategy. *Energies* **2019**, *12*, 3261. [\[CrossRef\]](#)
19. Dianov, A.; Anuchin, A. Adaptive Maximum Torque per Ampere Control of Sensorless Permanent Magnet Motor Drives. *Energies* **2020**, *13*, 5071. [\[CrossRef\]](#)
20. Zhou, Z.; Gu, X.; Wang, Z.; Zhang, G.; Geng, Q. An Improved Torque Control Strategy of PMSM Drive Considering On-Line MTPA Operation. *Energies* **2019**, *12*, 2951. [\[CrossRef\]](#)
21. Yoon, K.Y.; Baek, S.W. Performance Improvement of Concentrated-Flux Type IPM PMSM Motor with Flared-Shape Magnet Arrangement. *Appl. Sci.* **2020**, *10*, 6061. [\[CrossRef\]](#)
22. Huynh, T.A.; Hsieh, M.F. Performance Analysis of Permanent Magnet Motors for Electric Vehicles (EV) Traction Considering Driving Cycles. *Energies* **2018**, *11*, 1385. [\[CrossRef\]](#)
23. Jang, H.; Kim, H.; Liu, H.C.; Lee, H.J.; Lee, J. Investigation on the Torque Ripple Reduction Method of a Hybrid Electric Vehicle Motor. *Energies* **2021**, *14*, 1413. [\[CrossRef\]](#)
24. Krishnan, R. *Permanent Magnet Synchronous and Brushless DC Motor Drives*; CRC Press: Boca Raton, FL, USA, 2010.
25. Kabziński, J. Fuzzy Modeling of Disturbance Torques/Forces in Rotational/Linear Interior Permanent Magnet Synchronous Motors. In Proceedings of the 2005 European Conference on Power Electronics and Applications, Dresden, Germany, 11–14 September 2005; p. 10. [\[CrossRef\]](#)
26. De Castro, A.G.; Guazzelli, P.R.U.; Lumertz, M.M.; de Oliveira, C.M.R.; de Paula, G.T.; Monteiro, J.R.B.A. Novel MTPA Approach for IPMSM with Non-Sinusoidal Back-EMF. In Proceedings of the 2019 IEEE 15th Brazilian Power Electronics Conference and 5th IEEE Southern Power Electronics Conference (COBEP/SPEC), Santos, Brazil, 1–4 December 2019; pp. 1–6. [\[CrossRef\]](#)
27. Lin, H.; Hwang, K.Y.; Kwon, B.I. An Improved Flux Observer for Sensorless Permanent Magnet Synchronous Motor Drives with Parameter Identification. *J. Electr. Eng. Technol.* **2013**, *8*, 516–523. [\[CrossRef\]](#)
28. Sobieraj, T. Selection of the Parameters for the Optimal Control Strategy for Permanent-Magnet Synchronous Motor. *Prz. Elektrotech.* **2018**, *94*, 91–94. [\[CrossRef\]](#)
29. Wójcik, A.; Pajchrowski, T. Application of Iterative Learning Control for Ripple Torque Compensation in PMSM Drive. *Arch. Electr. Eng.* **2019**, *68*, 309–324. [\[CrossRef\]](#)
30. Mosiolek, P. Sterowanie Adaptacyjne Silnikiem PMSM z Dowolnym Rozkładem Strumienia. *Prz. Elektrotech.* **2014**, *90*, 103–108. [\[CrossRef\]](#)
31. Kabziński, J. Oscillations and Friction Compensation in Two-Mass Drive System with Flexible Shaft by Command Filtered Adaptive Backstepping. *IFAC-Pap.* **2015**, *48*, 307–312. doi:10.1016/j.ifacol.2015.09.203. [\[CrossRef\]](#)
32. Veaser, F.; Braun, T.; Kiltz, L.; Reuter, J. Nonlinear Modelling, Flatness-Based Current Control, and Torque Ripple Compensation for Interior Permanent Magnet Synchronous Machines. *Energies* **2021**, *14*, 1590. [\[CrossRef\]](#)
33. Kabziński, J.; Sobieraj, T. Auto-Tuning of Permanent Magnet Motor Drives with Observer Based Parameter Identifiers. In Proceedings of the 10th International Conference on Power Electronics and Motion Control, Dubrovnik & Cavtat, Croatia, 9–11 September 2002.
34. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme Learning Machine: Theory and Applications. *Neurocomputing* **2006**, *70*, 489–501. doi:10.1016/j.neucom.2005.12.126. [\[CrossRef\]](#)

35. Huang, G.B.; Zhou, H.; Ding, X.; Zhang, R. Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2012**, *42*, 513–529. [[CrossRef](#)] [[PubMed](#)]
36. Tikhonov, A.N.; Goncharsky, A.V.; Stepanov, V.V.; Yagola, A.G. *Numerical Methods for the Solution of Ill-Posed Problems*; Springer: Dordrecht, The Netherlands, 1995. [[CrossRef](#)]
37. Huang, G.; Huang, G.B.; Song, S.; You, K. Trends in Extreme Learning Machines: A Review. *Neural Netw.* **2015**, *61*, 32–48. doi:10.1016/j.neunet.2014.10.001. [[CrossRef](#)] [[PubMed](#)]
38. Kabziński, J. Rank-Revealing Orthogonal Decomposition in Extreme Learning Machine Design. In *Artificial Neural Networks and Machine Learning—ICANN 2018*; Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 3–13.
39. Akusok, A.; Björk, K.M.; Miche, Y.; Lendasse, A. High-Performance Extreme Learning Machines: A Complete Toolbox for Big Data Applications. *IEEE Access* **2015**, *3*, 1011–1025. [[CrossRef](#)]
40. Kabziński, J. Extreme learning machine with diversified neurons. In Proceedings of the 2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 17–19 November 2016; pp. 000181–000186. [[CrossRef](#)]
41. Miche, Y.; Sorjamaa, A.; Bas, P.; Simula, O.; Jutten, C.; Lendasse, A. OP-ELM: Optimally Pruned Extreme Learning Machine. *IEEE Trans. Neural Netw.* **2010**, *21*, 158–162. [[CrossRef](#)] [[PubMed](#)]
42. Feng, G.; Lan, Y.; Zhang, X.; Qian, Z. Dynamic Adjustment of Hidden Node Parameters for Extreme Learning Machine. *IEEE Trans. Cybern.* **2015**, *45*, 279–288. [[CrossRef](#)]
43. Miche, Y.; van Heeswijk, M.; Bas, P.; Simula, O.; Lendasse, A. TROP-ELM: A Double-Regularized ELM Using LARS and Tikhonov Regularization. *Neurocomputing* **2011**, *74*, 2413–2421. [[CrossRef](#)]
44. Parviainen, E.; Riihimäki, J. A Connection Between Extreme Learning Machine and Neural Network Kernel. *Commun. Comput. Inf. Sci.* **2013**, *272*, 122–135. [[CrossRef](#)]