Forsch Ingenieurwes
https://doi.org/10.1007/s10010-023-00647-w

ORIGINALARBEITEN/ORIGINALS

Surrogate model based prediction of transmission error characteristics based on generalized topography deviations

Marius Willecke[1] · Jens Brimmers[1] · Christian Brecher[1]

Received: 31 October 2022 / Accepted: 23 January 2023

Abstract

Simulation models of the tooth contact can provide valuable information of the gearbox operational behavior for deviated gears. It is therefore possible to gather geometry and deviation information of the two gears in contact and then estimate how they will perform in the assembled unit before even mounting them. This procedure could save time and costs used to disassemble gear boxes which fail the end of line tests because of acoustic reasons. The major challenge using sophisticated simulation models, which can describe the tooth contact is the tremendous amount of calculation time they need to produce suit-able results. In a productive gearbox manufacturing process this time is just not available. Therefore, a method to predict the operational behavior faster, than with the current used simulation models is needed. Firstly the size of the problem is downsized by introducing the sum deviation surface. It allows to reduce the number of necessary input parameters for a gear topography description to eighteen factors. With the help of that sum deviation surface, 3000 variants within a given variation space are calculated. The resulting training dataset is used to develop a deep neural network meta-model of the gear contact, which can predict the characteristics of the transmission error under load. With the help of that meta-model, the excitation of a gear pair can be predicted faster than real-time.

Modellbasierte Vorhersage von Drehfehlerkennwerten basierend auf verallgemeinerten Topografieabweichungen

Zusammenfassung

Simulationsmodelle des Zahnkontakts können wertvolle Informationen über das Betriebsverhalten eines Getriebes mit abweichungsbehafteten Zahnrädern liefern. Es ist daher möglich mit Informationen über die Geometrie und die Abweichung der beiden sich berührenden Zahnräder abzuschätzen, wie sie sich die montierten Getriebe verhalten werden, bevor sie überhaupt montiert werden. Dieses Verfahren kann Zeit und Kosten für die Demontage von Getrieben einsparen, die aus akustischen Gründen die End-of-Line-Tests nicht bestehen. Die größte Herausforderung bei der Verwendung von hochentwickelten Simulationsmodellen, die den Zahnkontakt für eine solche Vorhersage ausreichend genug beschreiben können, ist die enorme Rechenzeit, die sie benötigen, um nutzbare Ergebnisse zu erzielen. In einem produktiven Getriebeherstellungsprozess ist diese Zeit nicht verfügbar. Daher wird eine Methode benötigt, die das Betriebsverhalten schneller vorhersagt, als mit den derzeit verwendeten Simulationsmodellen. Dazu wird zuerst die Größe des Problems durch die Einführung einer Summenabweichungsfläche verkleinert. Diese erlaubt es, die Anzahl der notwendigen Eingangsparameter für eine Getriebetopographiebeschreibung auf achtzehn Werte zu reduzieren. Mit Hilfe der Summenabweichungsfläche werden 3000 Varianten innerhalb eines vorgegebenen Variationsraums berechnet. Der resultierende Trainingsdatensatz wird verwendet, um ein tiefes neuronales Netz des Zahnkontaktes zu entwickeln, dass die Eigenschaften des Drehfehlers unter Last vorhersagen kann. Mit Hilfe dieses Metamodells kann die Anregung durch ein Abweichungsbehaftetes Zahnradpaar schneller als in Echtzeit vorhergesagt werden.

✉ Marius Willecke
marius.willecke@wzl.rwth-aachen.de

[1] WZL, Campus-Boulevard 30, 52074 Aachen, Germany

Published online: 20 March 2023

1 Introduction and motivation

Tooth contact analysis is an integral part of the gear design process. With the help of simulation tools like ZAKO3D it is possible to calculate the excitation caused by a tooth contact [1]. Usually, the load-free transmission error (TE) or the TE under load is used for this purpose. However, the calculation with the tooth contact analysis ZAKO3D allows only a quasi-static consideration of the excitation. In order to better assess the behavior in the overall system, it is therefore necessary to perform a dynamic simulation. However, the main disadvantage of such dynamic simulations is the much longer computing time compared to quasi-static tooth contact analyses due to the high computational effort.

In the context of measures such as the increase in resource efficiency and due to the rising demands of end customers, the pressure on gearbox manufacturers to produce fewer rejects is growing, while at the same time placing higher demands on the quality of the gearboxes. To ensure that an assembled gearbox meets the acoustic requirements, end-of-line tests are carried out on the finished units [2]. If an anomaly is detected, the unit must be disassembled and overhauled as far as possible or is otherwise scrapped. A significant advantage would be if assemblies relevant for the acoustic behavior, such as the tooth meshes, could already be examined for their excitation behavior before assembly.

Theoretically, this would be possible by simulating the acoustic behavior of the gearbox in a dynamic model, as it is used today in the gear design process, with the real topographies of the gears to be installed. Exemplary work shows that these models have a high agreement with reality [3–5]. The essential problem here is the calculation time of these dynamic models, which prevents a process-time parallel calculation.

To solve this challenge, it is therefore necessary to create a possibility to predict the excitation behavior of the system much faster. One approach, which needs to be investigated for this purpose, is the formation of substitute models that allow considerably faster calculation of the target values. It is relevant that a description of the tooth flanks is used that is as exact as possible in order to be able to determine a realistic assessment of the excitation behavior that arises from the two wheels in the tooth mesh.

2 State of the art

A classical way to optimize the calculation speed of complex simulation models is the usage of meta models. The neural network is the one, which will later be used, therefore the state of the art of this article deals with neural networks and their basic properties. In addition, the adjustment possibilities, with which the properties of the network can be adapted, are discussed.

2.1 Neuronal networks

Neural networks have already been used for some time for the classification of acoustic questions [6]. In the context of gears, neural networks are used, among other things, for classification in the area of condition monitoring. Here they are used to monitor output variables of sensors and to detect deviations that indicate a change in the condition of the gearbox [7–10].

Another use of neural networks for gear modeling is described by CABRERA CANO. In his work, a system of rigid differential equations describing a gearbox in one dimension is set up with the help of a neural network. In this way, the behavior of the original simulation model can be reproduced. The trained neural network is faster than the original simulation model by a factor of ten [11].

BAXMANN undertakes first attempts to calculate parameters describing the application behavior by optimizing the selection of the micro geometry of a bevel gear. He can prove that neural networks are in principle suitable to simulate the behavior of a gearpair and thus accelerate the calculation of the parameters by a factor of 2 [12].

Basic functions of neural networks A neural network is outwardly a function that calculates output variables $\widetilde{y}$ based on input parameters $\widetilde{x}$. It can thus be described as $f(\widetilde{x}) = \widetilde{y}$. At first glance, it does not differ from a conventional function. The main difference between the neural network and a normal function, on the other hand, is that in a normal function the algorithm for transforming the input parameters into the output variables must be known. A neural network, on the other hand, receives a set of different input parameters with the corresponding output parameters and learns the relationships between them itself. Subsequently, previously unknown input parameters can be transformed into the correct output variables. The main advantage of a neural network compared to a normal function is therefore that it becomes possible to model correlations for which no algorithmic correlation can otherwise be found [13].

The schematic structure of a neural network is shown in Fig. 1. The basic building block of a neural network is a neuron, see the left side of Fig. 1. A neuron receives n input parameters, each of which is multiplied by the weights w. These weighted input parameters are then summed together with the constant value b and given as input to an activation function, which calculates the output of this one neuron.

The structure of the network itself again consists of several neurons in one layer. These adjacent neurons together form a layer. Several layers one after the other result in the

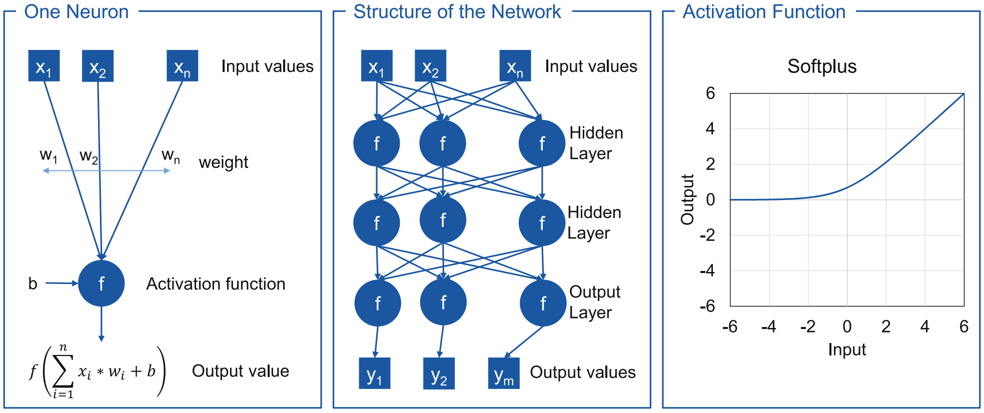


Fig. 1 Design and structure of neural networks [13]

structure of the entire network, see center of Fig. 1. The last layer is called the output layer, the layer of the input parameters is called the input layer, and all other layers in between are called hidden layers. Neural networks with multiple hidden layers are therefore also referred to as deep neural networks, abbreviated DNN in the following [13].

DNN are characterized by various properties. These have a significant influence on the performance and quality of the calculation performed with the DNN. One important property is the number of layers, see Fig. 1. The lower the number of layers, the lower the computational effort of the DNN. However, the accuracy with which the input parameters can be classified decreases. In addition, further settings are possible. The setting of the learning rate describes in principle how fast the DNN converges. A low learning rate leads to low convergence. However, if the learning rate is set too high, the solution may end up fuzzy and the model may no longer converge. The batch size indicates with how many data points of all available variants the algorithm is trained simultaneously. A size of $n=32$ has established itself as the standard value. If the value is chosen too large, the model may not be able to approximate new data. The number of epochs indicates how often the algorithm trains with the entire data set. The larger the number of epochs, the more accurate the DNN can become, but the longer the training takes [14].

The number of neurons per layer becomes clear from Fig. 1. The required number of neurons depends mainly on the number of input parameters and output parameters as well as on the number of layers. A too small number of neurons leads to the fact that the model cannot be approximated correctly, whereas a too large number of neurons leads to a high variance of the results and bad model performance [15].

Another possibility for optimization when using DNN is the normalization of the input data. The normalization of all input data to a value range from zero to one increases the performance, because the activation functions are designed for this value range [14].

Activation function An activation function is the function of a neuron that calculates the output quantity from the weighted input parameters. There are a large number of different functions, which are better or worse suited depending on the application. Since each neuron has its own activation function, different activation functions can be used in a neural network. If the activation functions are nonlinear functions, then already two hidden layers are sufficient to approximate any function, if the number of neurons per layer can be chosen large enough accordingly [13]. Information regarding the various activation functions can be found in [13, 16].

2.2 Conclusion of the state of the art

Up to now, neural networks have been used in the field of gear technology mainly as a decision-making aid for classification problems. Here, the networks are used to infer damage in the gearbox on the basis of recorded data. In addition, there are already first successful attempts to model the behavior of gearbox defects using neural networks. However, these models are still one-dimensional models that do not calculate acoustic parameters.

3 Objective and approach

The objective is to determine the characteristic values of the TE of a deviated gear pair orders of magnitude faster than is possible with a conventional ZAKO3D calculation under load. Therefore, a model is to be developed that is capable of doing this. The method to be developed for calculating the characteristic values of the TE with this model is shown in Fig. 2. It is assumed that each of the two wheels is provided with a deviation. In a first step, these two deviation surfaces (DS) of wheel 1 and wheel 2 are combined to a sum deviation surface (Σ DS). Thus, a substitute tooth contact is created in which only one wheel is provided with deviations and the other wheel has an ideal involute. In

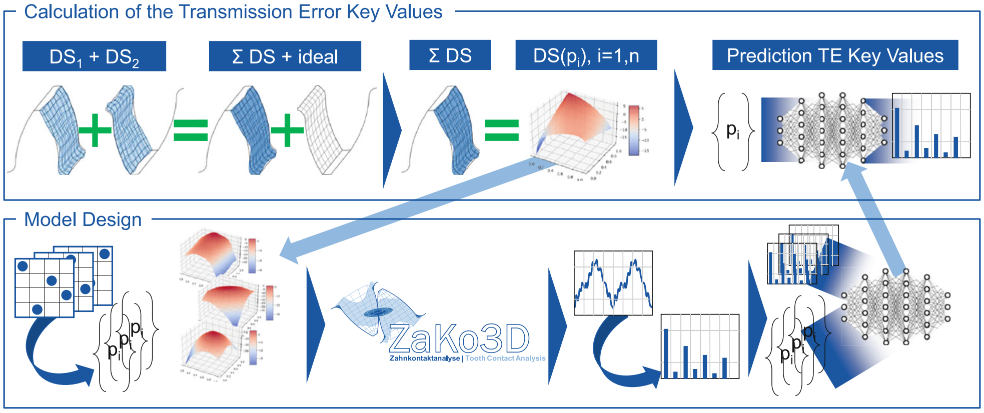


Fig. 2 Calculation method of the TE parameters and procedure for model construction

a second step, the parameters p_i are derived from the Σ DS. These parameters should be able to describe the Σ DS as accurately as possible by using as few of them as possible. In a third step, the parameters p_i are given as inputs to the model, which then returns the TE characteristics.

Corresponding to the individual steps of the method for calculating the TE characteristic values for two gears with deviations, the procedure for achieving this goal can also be divided into several sub-steps. In a first sub-step, the Σ DS must be established. For this purpose, a procedure is worked out with which the various individual topographies can be offset. In a second step a substitute surface is searched, which describes the Σ DS using as few parameters as possible. BRIMMERS points out that radial basis functions are best suited for a description of a topological modification of a gear [17]. For the gear geometry and modification chosen there, a resolution of five points in width direction and three points in height direction was chosen [17]. It is examined whether this resolution is also suitable for the gear and deviations used here.

In a third step, the actual model for calculation is developed. As an approach, a model is to be used that can be trained based on a data set and is then able to predict parameter variants that were not part of the training data set with sufficient accuracy. Therefore, first the training data set has to be created. For this purpose a pool of variants is generated. These are not combined fully factorial in the variation space, but distributed with the help of another experimental design, so that fewer variants are needed to cover the complete variation space. With each variant of this variant pool a ZAKO3D calculation under load is performed. For the thus calculated TE, the TE characteristic values are subsequently calculated. For each calculated variant of the pool, in addition to the parameters p_i as input of the model, the TE characteristic values as output of the model are also known now. The data set created in this way is examined for suitable modeling types using the program NOESIS OPTIMUS. The most promising modeling type is then implemented and tested with variants that are not part of the data pool. Different settings of the metamodeling methodology will be investigated. It is worked out whether the surrogate model is suitable in principle for the description of the problem.

4 Calculation of the characteristic transmission error values

This chapter describes how the parameters for describing the deviation of the gear are derived and how the associated TE characteristic values are calculated. The investigation is based on the gear of the first stage of the *Demonstrator Gearbox* developed within the scope of the Gear Research Circle project *Concept development for a practical Demonstrator Gearbox for high-speed applications of the WZL Gear Research Circle*. The gearbox is exemplary for a transmission of an electrified passenger car with a drive torque of $M_{an} = 120$ Nm at the input shaft. The gear data of the first stage are shown in Table 1.

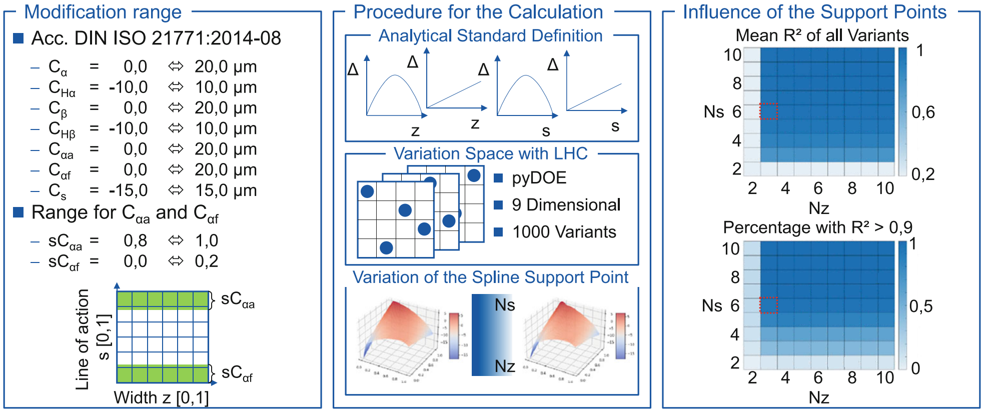
4.1 Introduction and parameterization of the sum deviation surface

If a gear is measured, a DS results for the flank. For each measuring point, the deviation of the real flank from the theoretical involute is specified [18]. The Σ DS is intended to represent the combined deviation of the wheel and pinion by combining the two individual DSs. The advantage of combining the deviations of both contact partners to a common sum deviation surface is that only one of the two gears has to be provided with a deviation in the simulation. The

Table 1 Gear data of stage 1 of the *Demonstrator Gearbox*

z_1/z_2	23/87	β	24.2°
b_1/b_2	27 mm/23 mm	x_{e1}/x_{e2}	0.33166/0.55906
m_n	1.421 mm	d_{a1}/d_{a2}	40.6 mm/137.9 mm
α_n	20°		

Fig. 3 Determination of the number of supporting points of the RBF for the Σ DS


Modification range
Acc. DIN ISO 21771:2014-08
– C_{α} = 0,0 ⇔ 20,0 µm
– $C_{H\alpha}$ = -10,0 ⇔ 10,0 µm
– C_{β} = 0,0 ⇔ 20,0 µm
– $C_{H\beta}$ = -10,0 ⇔ 10,0 µm
– $C_{\alpha a}$ = 0,0 ⇔ 20,0 µm
– $C_{\alpha f}$ = 0,0 ⇔ 20,0 µm
– C_{s} = -15,0 ⇔ 15,0 µm
Range for $C_{\alpha a}$ and $C_{\alpha f}$
– $sC_{\alpha a}$ = 0,8 ⇔ 1,0
– $sC_{\alpha f}$ = 0,0 ⇔ 0,2
Line of action s [0,1]
$sC_{\alpha a}$
$sC_{\alpha f}$
Width z [0,1]
Procedure for the Calculation
Analytical Standard Definition
Δ
z
s
Variation Space with LHC
pyDOE
9 Dimensional
1000 Variants
Variation of the Spline Support Point
Ns
Nz
Influence of the Support Points
Mean R² of all Variants
Ns
Nz
2 4 6 8 10
1
0,6
0,2
Percentage with R² > 0,9
1
0,5
0

other wheel can be modeled with ideal involute. This reduces the number of necessary parameters for the substitute model by half, since only one deviation surface has to be described and not two. The accuracy of this method depends on the amount of the displacement of the contact line under load.

The Σ DS generated in this way is to be approximated by radial basis functions (RBF). BRIMMERS had found them to be suitable for this purpose [17]. In order to determine the optimal number of necessary supporting points of the RBF in tooth width (Nz) and height (Ns) direction, different resolutions are investigated. For this purpose, analytical combinations of the different standardized tooth flank modifications are created in a first step [19]. The variation range of the individual modification amounts is listed on the left side of Fig. 3.

The DS are then approximated by RBF with a different number of supporting points in height and width direction. The number of grid points is varied from two to ten. In the bottom center of Fig. 3, an analytical DS and the corresponding DS approximated by RBF are shown as examples. The difference between the two DSs is examined at 10,000 uniformly distributed points. For the accuracy of the approximation, the R^2 coefficient of determination known from statistics can be calculated and used for the evaluation. In the work of BRIMMERS this value is also used to determine the necessary degree of a surface polynomial for the approximation of the surface [17]. 1000 different variants are compared, which are generated with the Latin hypercube function of the Python package PYDOE in the given limits of the target modifications.

The evaluation of the R^2 value for these 1000 variants is shown on the right side of Fig. 3. The first diagram shows the mean R^2 value for all variants as a function of the supporting points of the RBF in the height direction (Ns) and width direction (Nz). It can be seen that in the width direction, no further improvement of the R^2 value can be achieved from a number of Nz = 3 and in height direction from a number of Ns = 6 grid points.

4.2 Design of experiments

In order to build a learning model or a model converging against real behavior (regression), there must be examples available for the range in which the model should function. The boundaries of this space are shown on the left side of Fig. 3. In the optimal case, all variants are evenly distributed in the space to be developed. A Latin hypercupe experimental design is used for this. The variant space, which provides the basis for the testing of the different models in the further course, is computed from 3000 variants. The library PYDOE is used to create the test plan [20].

4.3 Calculation of the characteristic values

For the variants defined in Chap. 4.2, ZAKO3D (see Chap. 1) calculations are performed and single-number characteristic values are calculated from the values of TE. In this specific ZAKO3D simulation, the surrounding structure is not considered. The used characteristic values are:

- Crest factor
- RMS value
- Fluctuation width
- Skewness
- Kurtosis
- Correlation with negative and positive parabola and straight line
- Gear mesh orders
- Normalized gear mesh orders

The calculation and application of these characteristic values for TE is described by WILLECKE [21]. The TE, which is used as a basis for the calculation of the characteristic values, usually has a constant component. For the calculation of some characteristic values, such as the RMS value or the fluctuation width, this constant component is eliminated before the calculation of the characteristic value. Thus, only the fluctuation component that also leads to an acoustic excitation of the structure is considered. The calculated characteristic values are stored together with the corresponding parameters p_i for all variants in a common HDF5 file. This format allows a fast and efficient data access for programs in different programming languages [22].

5 Development of the model for prediction of the characteristic acoustic values

This chapter describes how a model for the prediction of the characteristic values is selected and further investigated. For this purpose, a preselection within different modeling approaches is carried out with the program NOESIS OPTIMUS. The most promising modeling approach is then built as a model in the programming language PYTHON. Here, properties of the model will be further investigated. The functionality of the model is verified with another test data set, which does not originate from the data set used for the development of the model. The test data set consists of 480 variants, which were created analogously with the procedure described in Chap. 4.2.

5.1 Training of the different models

The data set generated in Chap. 4.3 is used as a basis to test different modeling approaches and to select a suitable approach for the application. To avoid having to implement, train and test each approach individually, the commercial software OPTIMUS from the manufacturer NOESIS is used to generate an initial overview. Here, the data set can be loaded and then a large number of models can be tested automatically. The R^2 value is used to evaluate how well the respective model is suited to represent the data set. For this purpose OPTIMUS performs a cross-validation with the data set. Since the training and evaluation of the large number of models requires some time, only the fluctuation width and the first tooth mesh order were used as result parameters in the first step. The results of the calculation are shown in Table 2.

5.2 Selection of the best model and implementation in PYTHON

The R^2 in Table 2 indicates the lowest error for the deep neural networks (DNN). Therefore, these will be further investigated in the following. The settings used by OPTIMUS are: Normalization of the input data, a group size of 32, an epoch count of 100, a layer count of 3 with 64 neurons per layer, and a learning rate of 0.001 (see Sect. 2.1 for an explanation of the settings).

The model, which is used for the further investigations, is built up in PYTHON based on the library KERAS as part of the total library TENSORFLOW [16, 23]. The library KERAS is chosen for the implementation and the construction of the model, because powerful DNN can be built with it and the complexity and the extent of the modeling can be increased granularly [16].

The construction and training of the model is divided into several sections. In the first step, the data must be prepared for training the model. DNN work most efficiently if the data to be processed are normalized, see Chap. 2.1. The normalization of the data set is done with the library SCIKIT-LEARN [24].

In the second step, the structure of the model is defined. First, the input layer is defined. The model should work with a fixed number of input parameters (the $n_S = 18$ grid points, see Chap. 4.1). Afterwards additional hidden layers are added to the model. These hidden layers are mainly characterized by their neuron number and the activation function of the contained neurons. Different topologies and hyper-parameters were tested here. The results are described in Chap. 5.3. The last layer added is the output layer. This receives as many neurons as output variables to be computed. After the topology of the network is defined, the model is compiled. Here the learning rate and the loss function are defined. A variation of the learning rate shows that a higher value than 0.001 leads to a slightly accelerated learning process, but the achieved loss value is higher at the end. Thus, 0.001 is used as the learning rate. A square error function is used as the loss function, since this is a regression problem and not a classification problem.

The third step is the training of the model. The training data set used for this is described in Chap. 4.2. A group size of 32 is chosen, because the state of the art recommends it, see Chap. 2.1. The training is always performed in blocks of 1000 epochs each, called "training unit" in the following. Afterwards, the result of the loss function is considered and, if necessary, the next 1000 epochs are trained. One "training unit" takes about 40 s. The number of necessary "training units" varies depending on the topology of the DNN.

5.3 Variation of the topology and the hyperparameters of the network

The DNN is investigated by examining the influence of the number of hidden layers, the number of neurons per layer and the activation functions used. As a first step, the settings from NOESIS OPTIMUS are used. Since it could not be determined which activation function is used by OPTIMUS, different activation functions are tested here. In Fig. 4 the ratio between the predicted and the actually calculated value

Table 2 Comparison of the model approaches in NOESIS OPTIMUS

Modelling Strategy	R^2_{cross}	Modelling Strategy	R^2_{cross}
Deep neural networks	0.9348	Random Forrest Regression	0.5994
Radial basis functions	0.8613	Taylor-Polynom	0.572
Kriging	0.835	Optimale Response Surface Modelling	0.4979
Shallow neural networks	0.6899	Akaike information Criterion	0.2664
Relevance Vector Regression	0.6121		

is shown on the left side for the two characteristic values fluctuation width and first tooth meshing order. The RMS value as well as the parabolic correlations and the skewness behave similarly, but are not shown here for reasons of space.

The training data set itself is shown on the left. In a perfect prediction, all points, each representing a variant, would lie on the line of origin in the middle. This is not the case. To better visualize the deviation of the prediction from the ideal result, a regression line is placed through the points, highlighting the deviation from the original line. In the middle of Fig. 4, the same characteristic values and the same model are shown. Here, however, not the training data set but the test data set was predicted.

It can be seen that the deviations between the actual values and the predicted values are increasing. On the right side, a prediction of the training data set is shown again. Here, however, the activation function in the model has been changed from Softmax to Softplus. The prediction accuracy of the characteristic values of the test data set is already better with this activation function than the prediction of the characteristic values of the training data set with the Softmax activation function. In addition to the softmax and softplus activation functions, elu, relu and tanh functions were also tested. However, these showed a lower quality of prediction. After the softplus activation function was determined to be promising for the application, the number of neurons per layer is varied.

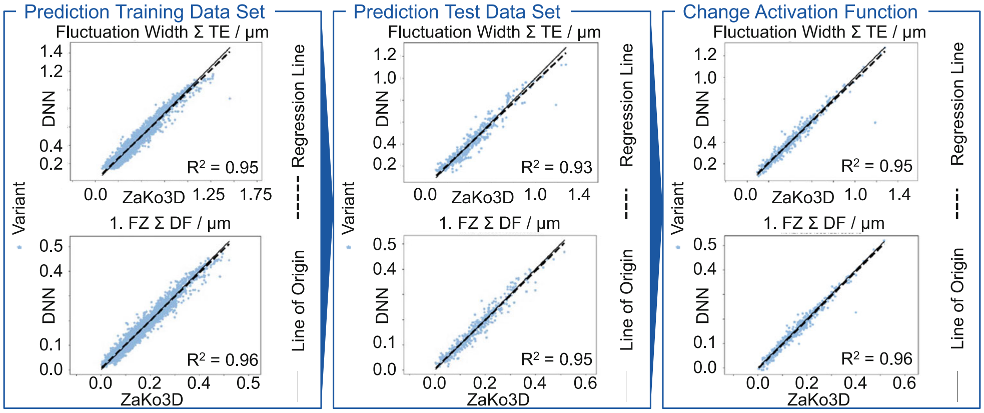
The result of varying the number of neurons is shown in Fig. 5. It can be seen that a higher number of neurons produces a significant improvement in the prediction quality. Likewise, no deterioration due to an overfit can be detected. In the area of the low amplitudes of the first tooth meshing order, outliers can still be seen. The reason for this could be that there are not enough examples of geometry in the training data set that produce a very low first tooth meshing order. This is also supported by the fact that only very few points in the test data set have such a low first tooth meshing order. Both training data set and test data set should in principle have a similar frequency distribution in the boundaries of the variant space.

In the following, it will therefore be test whether a network with more neurons and layers can in principle predict the training data set better than a network with fewer neurons and layers if the training is sufficiently long. In Fig. 6, the upper diagrams in the left and middle columns show the self-prediction of the training data set.

It was trained with a factor ten low learning rate (0.0001 instead of 0.001) until the loss value did not change. No differences can be seen between the two networks in the self-prediction. However, the network with two hidden layers and 64 neurons per layer trained significantly longer than the network with three hidden layers and 256 neurons per layer.

The prediction of the test data set in the lower two diagrams shows an improvement in the mapping when examined closely. Likewise, for the network with two hidden layers and 64 neurons, an overfit already occurs. On the right side of Fig. 6, the model is shown in a state in which it has been trained only up to a loss value L = 0.0085. The "trained

Fig. 4 Influence of the activation function on the prediction



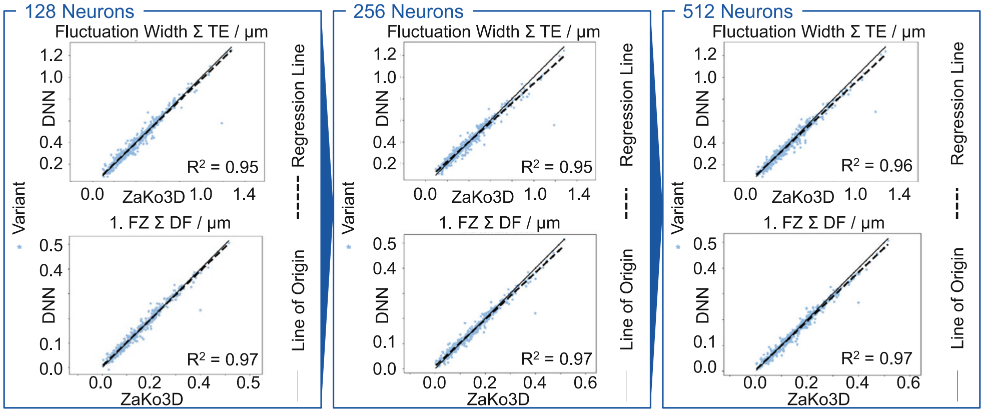


Fig. 5 Influence of the number of neurons on the modeling

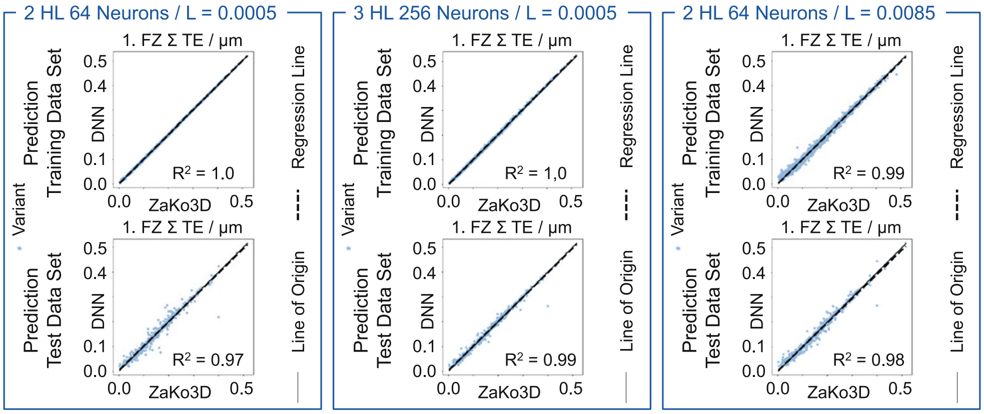


Fig. 6 Maximum prediction accuracy and overfitting of the DNN

out" version on the left shows a loss value of L = 0.0005. It can be seen here that despite the higher loss value, the test data set is better predicted, i.e. there is less overfit.

For the network with three hidden layers and 256 neurons, an overfit could in principle already be present. However, the test data set still shows a better prediction accuracy than the other two models.

The advantage of calculating the excitation with the DNN can be illustrated by an example. For a deviation surface from the micro geometry $C_\alpha = 7\,\mu m$, $C_{H\alpha} = 5\,\mu m$, $C_\beta = 7\,\mu m$, $C_{H\beta} = 5\,\mu m$, $C_{\alpha a} = 5\,\mu m$, $sC_{\alpha a} = 0.9$ the characteristic values of the total angular misalignment are determined with ZAKO3D and with the DNN. ZAKO3D calculates $f_{z1} = 0.098\,\mu m$ for the first tooth meshing order. The DNN calculates $f_{z1} = 0.099\,\mu m$. The value range in the variant space for the first tooth meshing order extends from $f_{z1} = 0.003\,\mu m$ to $f_{z1} = 0.521\,\mu m$. A deviation of $\Delta f_{z1} = 1\,nm$ in the prediction is therefore to be regarded as small. However, relevant differences occur in the calculation duration. While ZAKO3D needs $t_s = 551\,s$ to perform the simulation the DNN has the result after $t_s = 0.026\,s$ and is thus 21,192 times faster. Since the calculation of the DNN can be parallelized better than parallel running ZAKO3D calculations, even higher speed advantages can be achieved on a computer with a 12 core CPU with a large number of calculations. For example, the DNN computes the 480 variants of the test data set with $t_s = 0.038\,s$ ~1,000,000 times faster than ZAKO3D with $t_s = 10\,h$. This speed advantage allows the user new application areas of tooth contact analysis. It would thus be possible to simulative pair all wheels with all pinions for a production lot and thus calculate the pairing setup that ensures the best excitation behavior. In this way, the optimum can be extracted from existing resources. Since the original ZAKO3D simulation model had been validated with real measurements and the DNN predicts the same results as the original model, it can be assumed, that the results of the DNN also match the measurements.

6 Summary and outlook

The aim of the work was to determine the characteristic values of the transmission error (TE) of a gear pair with deviation faster than it is possible with a conventional ZAKO3D calculation under load. To achieve that goal a surrogate modelling approach was chosen. Therefore a training dataset of 3000 topographies was created and calculated in

ZAKO3D. Characteristic values were determined from the calculated TE. Analogously a test dataset of 480 variants was formed. Several surrogate modelling strategies where tested with NOESIS OPTIMUS. The most promising approach, a Deep Neural Network, was built in PYTHON with KERAS as part of TENSORFLOW. For this network, different hyperparameters were tested and the effects on the predictability of the test data set were investigated. It was shown that characteristic values such as the first gear mesh order, the fluctuation width and the RMS value can be predicted well with deep neural networks. It is thus possible to determine characteristic values of the TE under load with sufficient accuracy in a few milliseconds.

During the optimization of the deep neural networks it was found that a higher number of variants and in particular a higher number of variants in certain value ranges could further increase the prediction quality of the meta model. Therefore, it should be investigated whether neural networks with even higher prediction quality can be generated with a higher number of variants.

Furthermore, only quasi-statically calculated TEs were investigated in this work. Therefore, it should be investigated whether the method of the parameterized sum deviation surface as input variable for a meta-model could also be successfully applied in combination with dynamics models.

Due to the speed advantage achieved in quasi-static tooth contact analysis, the developed method could already be used in industrial production. A live quasi-static check of the excitation behavior of the gears to be installed could be carried out. Likewise, the optimum pairings could be found for a larger quantity of gears and pinions.

Acknowledgements The authors gratefully acknowledge financial support by the WZL Gear Research Circle for the achievement of the project results.

Seit 1956
Since 1956
WZL
Getriebekreis
Gear Research Circle

Funding Open Access funding enabled and organized by Projekt DEAL.

Conflict of interest M. Willecke, J. Brimmers and C. Brecher declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Hemmelmann JE (2007) Simulation des lastfreien und belasteten Zahneingriffs zur Analyse der Drehübertragung von Zahnradgetrieben. Diss., RWTH Aachen University
2. Berger V, Wilbertz A, Meyer I (2010) Körperschall Im End-Of-Line-Test Von Doppelkupplungsgetrieben. ATZproduction. https://doi.org/10.1007/BF03224146
3. Matzke D, Schelenz R, Reisch S et al (2018) Validation of the gearbox load calculation of a wind turbine MBS model. J Phys Conf Ser 1037:1–10. https://doi.org/10.1088/1742-6596/1037/6/062025
4. Carl CF (2014) Gehörbezogene Analyse und Synthese der vibroakustischen Geräuschanregung von Verzahnungen. Diss., RWTH Aachen University
5. Sanzenbacher S (2016) Reduzierung von Getriebegeräuschen durch Körperschallminderungsmaßnahmen. Diss., Universität Stuttgart
6. Bothe A, Gärtner U, Krämer S (1995) Neuronale Klassifikatoren in der akustischen Mustererkennung. Tm Tech Mess 62:88–94. https://doi.org/10.1524/teme.1995.62.jg.88
7. Wu J-D, Chan J-J (2009) Faulted gear identification of a rotating machinery based on wavelet transform and artificial neural network. Expert Syst Appl 36:8862–8875. https://doi.org/10.1016/j.eswa.2008.11.020
8. Cheng G, Cheng Y, Shen L et al (2013) Gear fault identification based on Hilbert–Huang transform and SOM neural network. Measurement 46:1137–1146. https://doi.org/10.1016/j.measurement.2012.10.026
9. Cao P, Zhang S, Tang J (2018) Preprocessing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning. IEEE Access 6:26241–26253. https://doi.org/10.1109/ACCESS.2018.2837621
10. Ding J, Xiao D, Li X (2020) Gear fault diagnosis based on genetic mutation particle swarm optimization VMD and probabilistic neural network algorithm. IEEE Access 8:18456–18474. https://doi.org/10.1109/ACCESS.2020.2968382
11. Cabrera Cano M (2016) Neuronale Netze mit externen Laguerre-Filtern zur automatischen numerischen Vereinfachung von Getriebemodellen. Diss., Karlsruher Institut für Technologie
12. Baxmann M (2007) Entwicklung einer Methode zur funktionsorientierten Auslegung und Tolerierung von Kegelradverzahnungen. Diss., RWTH Aachen University
13. Ketkar N (2017) Deep learning with python: a hands-on introduction. Apress, Berkeley
14. Alto V (2019) Neural networks: parameters, hyperparameters and optimization strategies. https://towardsdatascience.com/neural-networks-parameters-hyperparameters-and-optimization-strategies-3f0842fac0a5. Accessed 11 Mar 2022
15. Krishnan S (2021) How to determine the number of layers and neurons in the hidden layer? | by Sandhya Krishnan | Geek Culture | Medium | Geek Culture. https://medium.com/geekculture/introduction-to-neural-network-2f8b8221fbd3. Accessed 11 Mar 2022
16. Keras (2022) Keras: the Python deep learning API. https://keras.io/. Accessed 23 Mar 2022

17. Brimmers J (2020) Funktionsorientierte Auslegung topologischer Zahnflankenmodifikationen für Beveloidverzahnungen. Diss., RWTH Aachen University
18. Richtlinie (2018) Messen und Prüfen von Verzahnungen. Auswertung von Profil- und Flankenlinienmessungen an Zylinderrädern mit Evolventenprofil: Auswertung von Profil- und Flankenlinienmessungen an Zylinderrädern mit Evolventenprofil(2612)
19. Norm (2014) Zahnräder – Zylinderräder und Zylinderradpaare mit Evolventenverzahnung – Begriffe und Geometrie(21771)
20. pyDOE (2022) pyDOE: The experimental design package for python—pyDOE 0.3.6 documentation. https://pythonhosted.org/pyDOE/. Accessed 24 Mar 2022
21. Willecke M, Brimmers J, Brecher C (2021) Virtual single flank testing—applications for industry 4.0. Procedia CIRP 104:476–481. https://doi.org/10.1016/j.procir.2021.11.080
22. The HDF Group (2019) The HDF5® library & file format—the HDF group. https://www.hdfgroup.org/solutions/hdf5/. Accessed 14 Mar 2022
23. TensorFlow (2021) TensorFlow. https://www.tensorflow.org/. Accessed 25 Mar 2022
24. scikit-learn (2022) scikit-learn: machine learning in Python—scikit-learn 1.0.2 documentation. https://scikit-learn.org/stable/. Accessed 25 Mar 2022