11th World Congress on Structural and Multidisciplinary Optimisation
07th -12th, June 2015, Sydney Australia

Combining state of the art meta-models for predicting the behavior of non-linear crashworthiness structures for shape and sizing optimizations

Axel Schumacher, Christopher Ortmann

University of Wuppertal, Faculty D – Mechanical Engineering, Chair for Optimization of Mechanical Structures, Wuppertal, Germany, schumacher@uni-wuppertal.de, ortmann@uni-wuppertal.de

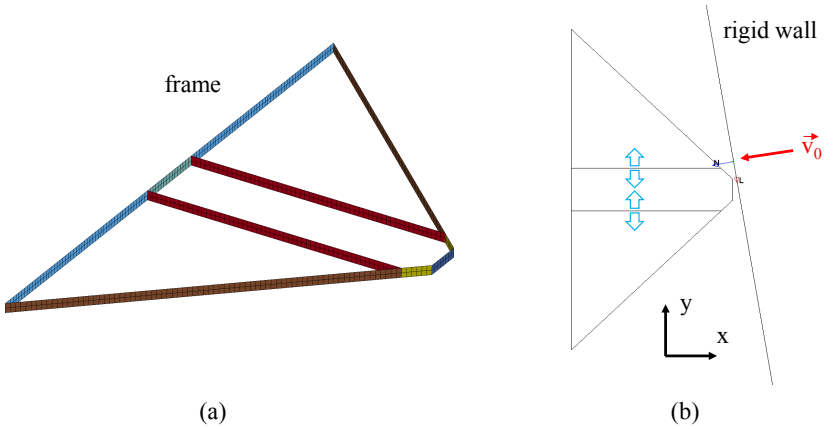
Abstract
In this contribution, we consider a shape optimization problem of a simple crashworthiness structure. The application example is an aluminum frame clamped on one side and impacted on the other side. The structural analysis is carried out with the finite element method using explicit time integration. Contact phenomena, buckling phenomena und non-linear material data are taken into account. With the aid of this example, a large number of different meta-models using radial basis functions, Kriging (Gaussian processes) and neural networks are generated.
The advantages and disadvantages of these different methods and the problems by relying on the common quality criterions "coefficient of determination R^2" and "leave-one-out-cross-validation R^2_{press}" are shown. The combination and superposition of the different meta-model techniques is investigated in order to enhance the meta-model prediction capability.

Keywords: meta-models, crashworthiness, Gaussian Processes, Kriging, neural networks, radial basis functions

1. Introduction
For a limited number of design variables, the use of meta-model techniques is very popular in structural optimization processes. Hereby, we get smooth and continuous descriptions of the structural responses. The quality of the meta-models can be scaled individually depending on the available time. After the creation of the meta-models, the optimization can be performed without further time-consuming finite element simulations using the responses of the meta-models.
When using meta-models for non-linear dynamic mechanical problems (e.g. crashworthiness problems), the nonlinearities and the limited amount of sampling points due to the high computational effort, can cause low prediction capabilities of the meta-models. Further difficulties can arise due to non-smooth behavior of the structural responses, insufficient material data, physical and numerical bifurcation points and finite element mesh dependency.
In commercial and scientific software codes there exist a large number of different meta-model-techniques. The question arises which one is the most suitable for the mechanical problem at hand and which quality criterions can be used for this decision.
In this contribution a large number of different meta-model techniques and some combinations and superpositions of the different meta-models is investigated in order to enhance the meta-model prediction capability.

2. Crashworthiness example
The considered application example shown in figure 1 is an aluminum frame (300 x 150 x 5 mm, wall thickness: 2 mm) clamped on one side and impacted by a rigid wall under an angle of 10 degree. The rigid wall has a mass of 2 kg and an initial velocity of 5 m/s. The design variable describes the y-position of the inner walls (symmetric, see figure 1b). Figure 2 shows the deformation and the reaction force of the frame depending on the value of the design variable.

3. Meta-model techniques
In this chapter the relevant meta-modelling techniques are described and applied to the crashworthiness example. For this purpose 7 sampling points are used, assuming that more sampling points are not available. The quality criterions R^2 and if possible R^2_{press} are calculated. In order to evaluate these quality criterions and the prediction capabilities of the meta-models, 6 additional validation points are used. For this purpose the regression parameter R^2_{val} is calculated for the validation points similarly as R^2 is calculated for the sampling points.



(a) (b)

Figure 1: Aluminum frame [1]

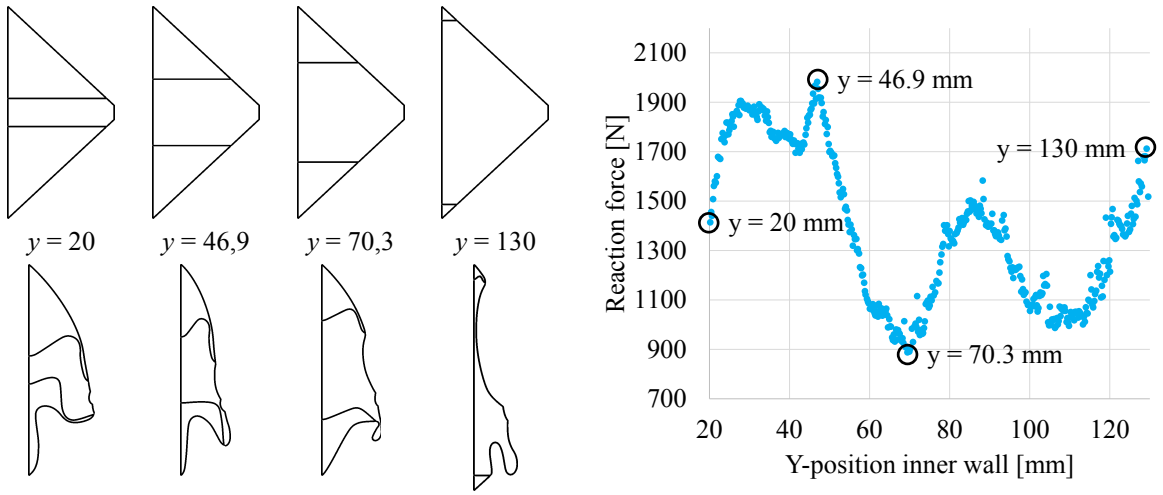


Figure 2: Deformation and reaction force of the frame depending on the value of the design variable [1]

3.1 Radial basis functions

The value of a radial basis function depends on the distance to a center point $C(x_n/y_n)$ called radius $r = \|\vec{x} - \vec{c}\|$. Some common types of radial basis functions are:

- Linear: $\varphi(r) = r$,
- Cubic : $\varphi(r) = r^3$,
- Multiquadric: $\sqrt{a + r^2}$ with the positive shape parameter a.

The interpolation is typically done by a linear combination of *n* (normalized) radial basis functions centered around the *n* sampling points:

$$f(\vec{x}) = \sum_{i=1}^{n} w_i \cdot \varphi(r) \tag{1}$$

The weights w_i can be determined by the following equation system:

$$\begin{bmatrix} \varphi(r_{11}) & \dots & \varphi(r_{1n}) \\ \dots & \dots & \dots \\ \varphi(r_{n1}) & \dots & \varphi(r_{nn}) \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ \dots \\ w_n \end{bmatrix} = \begin{bmatrix} g_1 \\ \dots \\ g_n \end{bmatrix} \tag{2}$$

Figure 3 shows three meta-models with different kinds of radial basis functions for the crashworthiness example. The real (usually unknown) relation between the design variable and the structural response is indicated by blue dots. The training sampling points are pictured by red dots and the validation sampling points by purple dots.

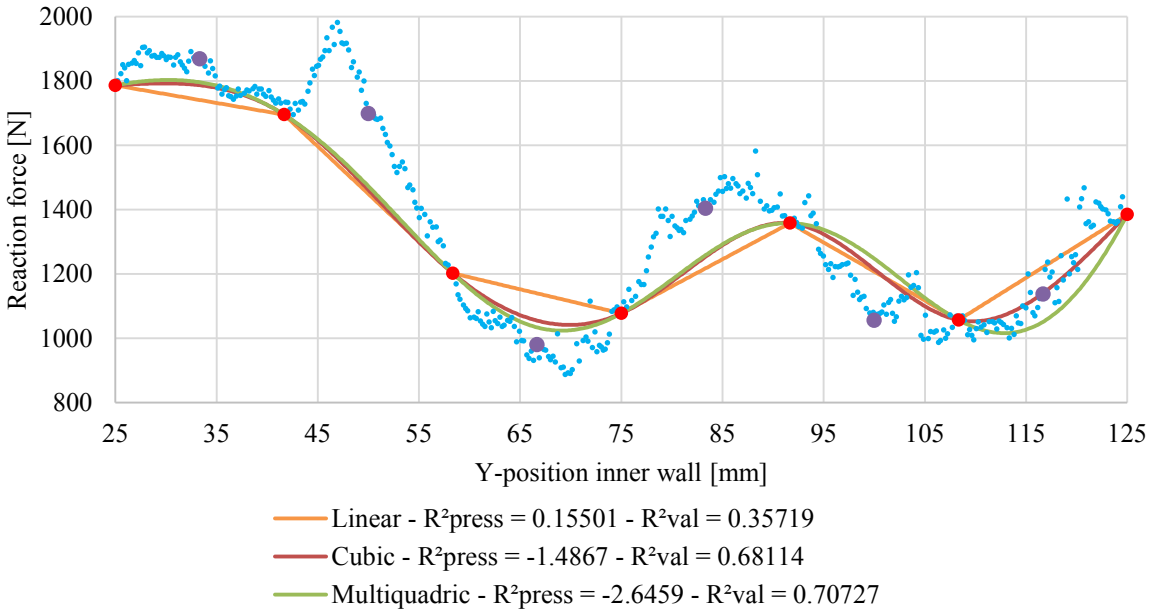


Figure 3: Meta-modeling example with radial basis functions [1]

3.2 Neural network

A neural network consists of three different kinds of neurons: the input neurons (process input data), the output neurons (process output data) and the hidden neurons (internal representation of the environment/problem). A common net type is a feed-forward multi-layer-perceptron net (figure 4).

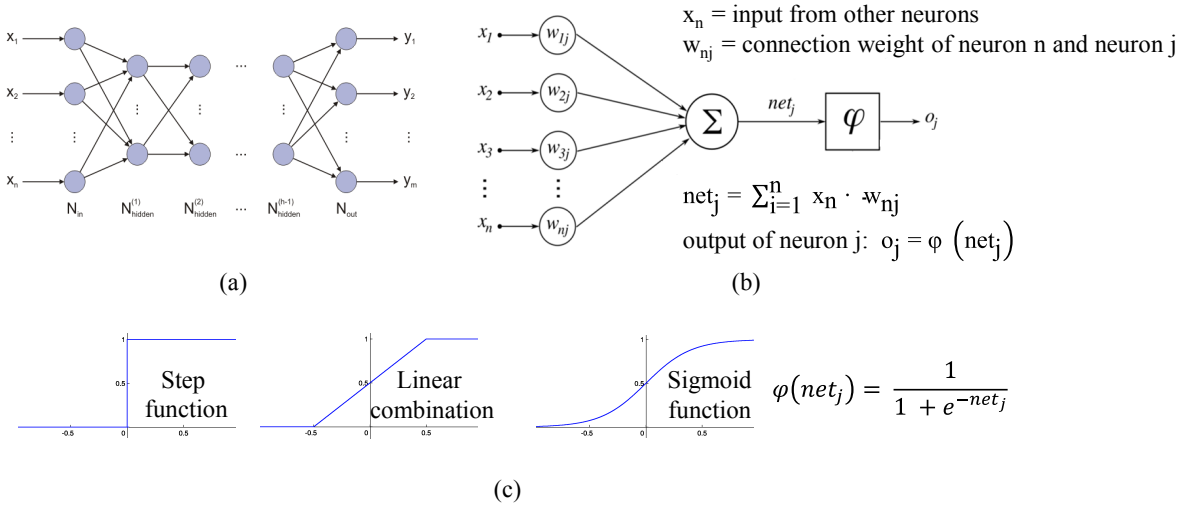


Figure 4: Meta-modeling with neural networks: multi-layer (a), neuron input/output (b), common neuron activation functions (c) [1, 2, 3]

The neural networks learn with training data (here: sampling points). The learning process is an optimization of the weights between the neurons to minimize the error between the neural network outputs and the given sampling points. A common approach for this is the supervised learning with backpropagation, which is a version of the method of steepest descent adapted for neural networks.

Figure 5 shows three different approximations with multi-layer-perceptron networks for the crashworthiness example. Each consists of a single input neuron, a single output neuron and two layers of hidden neurons. The number of hidden neurons in each hidden layer varies.

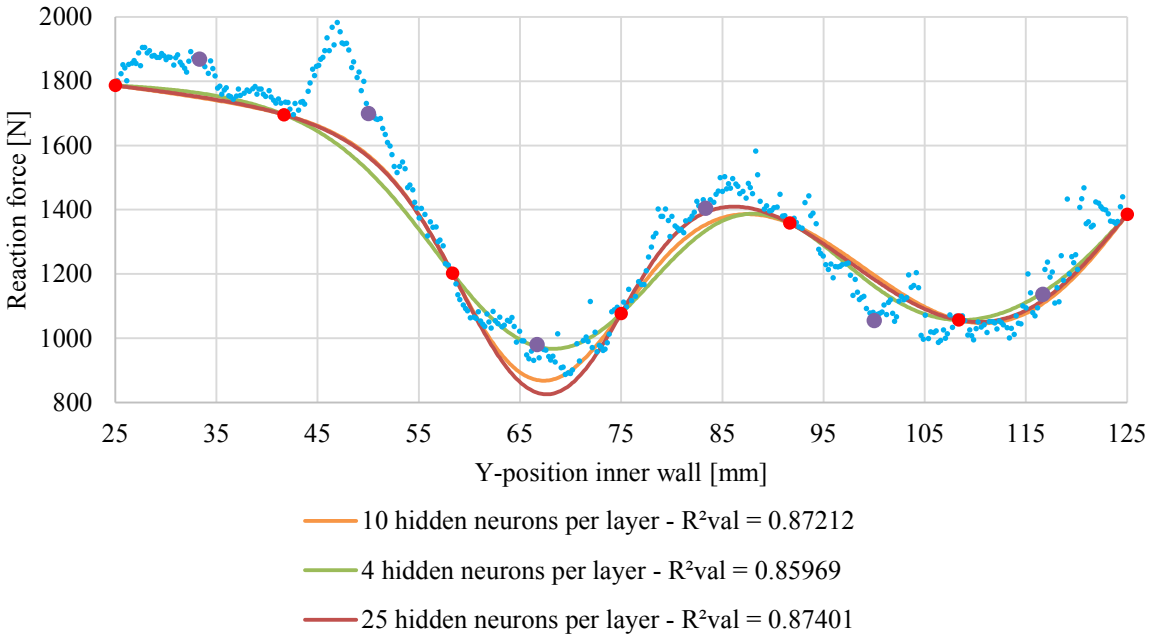


Figure 5: Meta-modeling example with neural networks [1]

3.3 Gaussian Processes (Kriging)

Gaussian processes are also known as Kriging. The similarity influence (figure 6) of a sampling point on an arbitrary point is defined with the covariance function. A common approach is the "squared exponential":

$$Cov(x, x') = \sigma_f^2 \cdot e^{(-(x-x')^2/(2 \cdot l^2))} \tag{3}$$

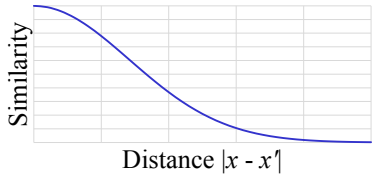


Figure 6: Similarity influence of a sampling point [1]

A covariance matrix between all sampling points with the x-value vector $\vec{p}_x$ can be calculated as:

$$\underline{Cov}(\vec{p}_x) = \begin{bmatrix} Cov(x_1, x_1) & \dots & Cov(x_1, x_n) \\ \dots & \dots & \dots \\ Cov(x_n, x_1) & \dots & Cov(x_n, x_n) \end{bmatrix} \tag{4}$$

The covariance matrix between an arbitrary point and the sampling points can be determined as:

$$\underline{Cov}(x, \vec{p}_x) = [Cov(x, x_1) \quad \dots \quad Cov(x, x_n)] \tag{5}$$

The interpolation $\tilde{g}(x)$ can be defined as:

$$\tilde{g}(x) = \underline{Cov}(x, \vec{p}_x) \cdot \underline{Cov}(\vec{p}_x)^{-1} \cdot \vec{p}_g \tag{6}$$

Three different Kriging interpolations for the crashworthiness example are shown in figure 7. They differ in the choice of the parameter theta, which is similar but not identical to the length scale parameter l in (9). Theta = 8.8315 is optimized for a maximal value of R^2_{press}.

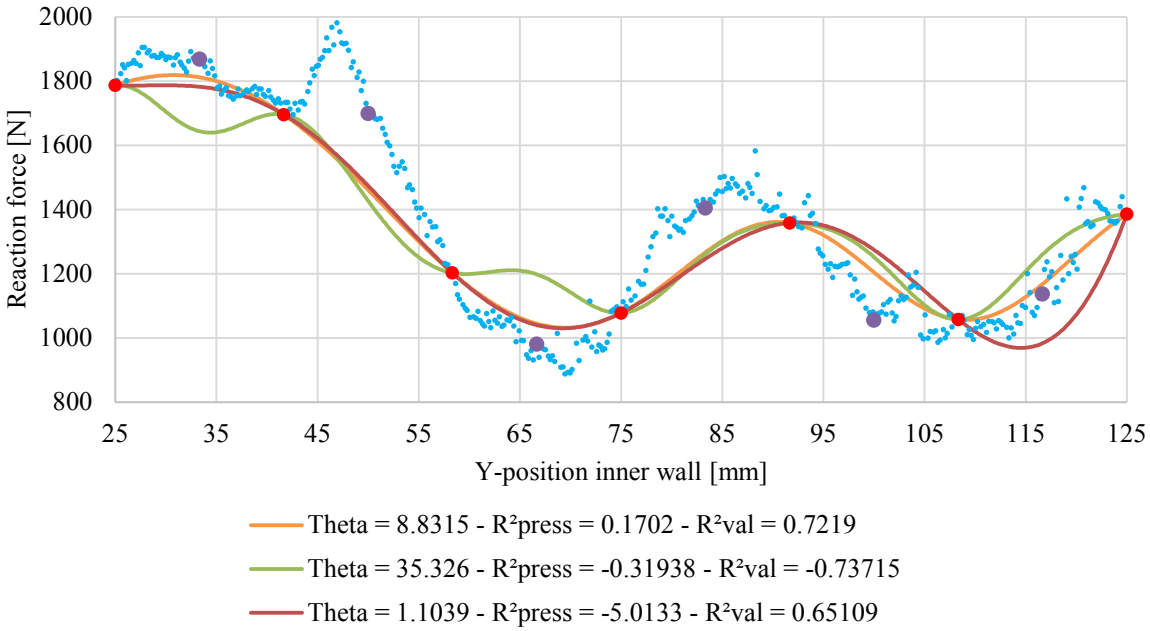


Figure 7: Meta-modeling example with Gaussian processes [1]

3.4 Typical characteristics of the different approaches

Typical characteristics of the three described approaches are summarized in table 1. There exist a lot of other meta-model techniques which are not considered in this contribution, e.g. support vector machine regression, random forest or fuzzy regression model.

Table 1: Typical characteristics of the described approaches for meta-modeling

Radial basis functions	Neural networks	Gaussian Processes (Kriging)
- Interpolation depends on the type of the radial basis function - Smooth transitions between the sampling points - This smoothness limits the versatility	- "Black-box-systems" - Limited versatility due to the activation function - The higher the number of hidden neurons the better the versatility but the higher the risk of overlearning - The training can be computationally expensive - The approximation accuracy of the given sampling points can be poor - High resilience to outlier sampling points	- Usually it is an interpolation - Very versatile and can even adapt to non-smoothness - The length scale parameter l controls the "frequency" - The determination of this parameter can be difficult, a common approach is an optimization of this parameter by the maximum likelihood principle

4. Superposition of the meta-models

The quality criterions R^2 and R^2_{press}, the average absolute error at the validation points $|\varepsilon|_{ave}$ and the regression parameter for the validation points R^2_{val} for each meta-model of the crashworthiness example are summarized in table 2.

Table 2: Validation of the meta-models

Meta-model	R^2	R^2_{press}	$\|\varepsilon\|_{ave}$	R^2_{val}
Linear radial basis function	1	0.15501	159.984148	0.35718969
Cubic radial basis function	1	-1.4867	117.684404	0.68113786
Multiquadric radial basis function	1	-2.6459	129.308549	0.70727437
Neural network 10 hidden neurons	1	-	96.9827279	0.87212213
Neural network 4 hidden neurons	1	-	78.6433099	0.85968893
Neural network 25 hidden neurons	1	-	93.5231771	0.8740076
Kriging Theta = 8.8315	1	0.1702	112.8654049	0.7219
Kriging Theta = 1.1039	1	-5.0133	150.4035117	0.6510925
Kriging Theta = 35.326	1	-0.31938	196.3350898	-0.73714623

Often all available sampling points are used for the training of the meta-model in order to maximize its quality. This is especially true for crashworthiness problems, where due to the high computational effort of each crash simulation only a very limited number of sampling points is available. In these cases there is no validation data and usually R^2_{press} is the main descision criterion for the evaluation of a meta-model.
Astonishingly, for the given crashworthiness example there is no correlation between R^2_{press} and the real prediction capabilities of the meta-models (evaluated by R^2_{val}). For example the radial basis function meta-model with the highest value of R^2_{press} (linear radial basis function) has the lowest value of R^2_{val} of all radial basis function meta-models.
The question arises whether there exists a possibility to reduce the risk of choosing a meta-model with a low prediction capability by relying on common quality criterions like R^2_{val} and R^2_{press}.
In relation to the computational effort for the generation of the sampling points, the computational effort for the creation of meta-models is very low. The idea is to generate a lot of different meta-models and to find a suitable superposition of the models in order to enhance the overall prediction capability or at least to reduce the risk of choosing a meta-model with a low prediction capability.

Figure 9 shows four different meta-models which are based on the superposition of the meta-models shown in the previous chapter. For the superposition the meta-models are weighted equally.
The superposition of all meta-models has a value $|\varepsilon|_{ave}$ of 107.29. The average value of $|\varepsilon|_{ave}$ of all meta-models is 126.192258. This difference is quite surprising, because the superposition is just a simple averaging of the meta-models. Because of the lower average error the R^2_{val} value of the superposition of all meta-models is with 0.73399 also significantly better than the average R^2_{val} value of all meta-models with 0.55414.
This is also true for the $|\varepsilon|_{ave}$ values of all other superpositions compared to the average $|\varepsilon|_{ave}$ value of the meta-models which have been used for the superposition (radial basis functions: 126.1 to 135.66, neural networks: 89.118 to 89.716 and Kriging: 136.15 to 153.2). Whether this is a mere coincidence or a possible strategy for the enhancement of meta-modeling remains the topic of further research.

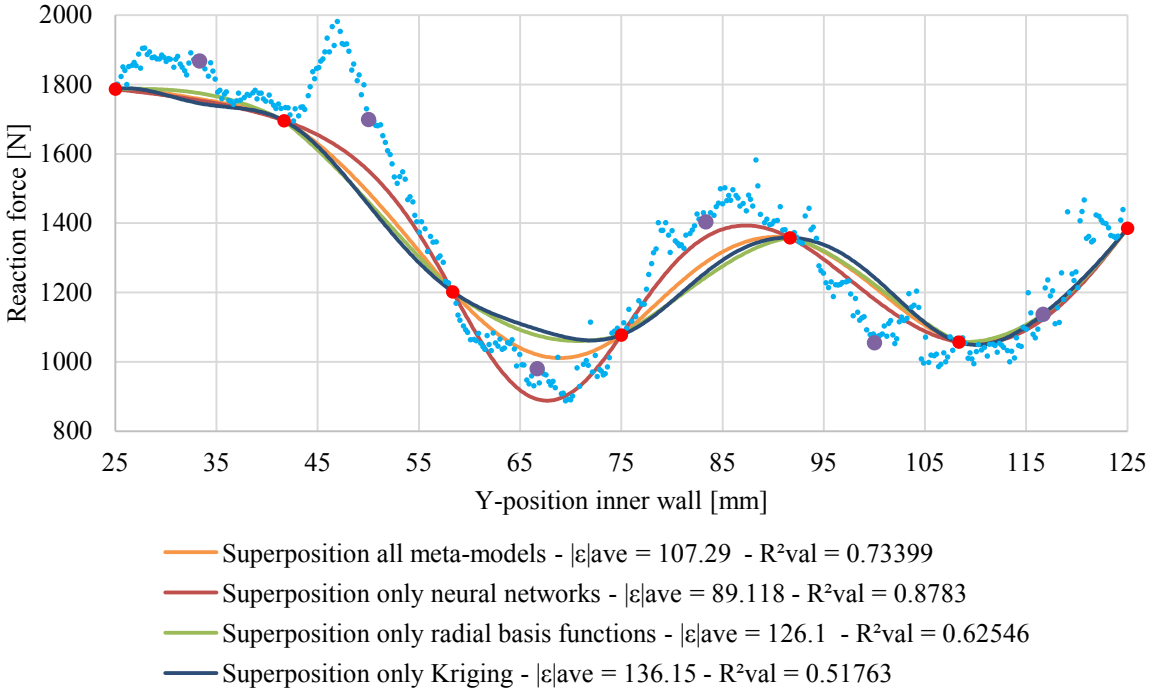


Figure 8: Superposition of the meta-models

7. References

[1] C. Ortmann, A. Schumacher: Meta-models in structural optimization – techniques and strategies", Proceedings of the Automotive CAE Grand Challenge 2014, 15th and 16th of April 2014 in Hanau
[2] http://pi.informatik.uni-siegen.de/Arbeitsgebiete/ci/knn/, access March 26th, 2015
[3] http://de.wikipedia.org/wiki/K%C3%BCnstliches_neuronales_Netz, access March 26th, 2015