

Auto-Extraction of Modelica Code from Finite Element Analysis or Measurement Data

The-Quan Pham¹, Alfred Kamusella², Holger Neubert²

¹ OptiY e.K., Aschaffenburg Germany, e-mail: pham@optiy.eu

² Technische Universität Dresden, Institute of Electromechanical and Electronic Design

Abstract

This paper presents a new approach to extract Modelica codes from finite element analysis or measurement data automatically. The finite element model or the real system on the test bench is adaptively sampled while applying the Gaussian process with a few number of model calculations or measurement points. Based on these support points, a meta- or surrogate model of the system is built. Thus, Modelica codes can be generated automatically. These algorithms are implemented in the multidisciplinary design software OptiY[®]. Its application is demonstrated on the example of an electromagnetic actuator.

Keywords: Gaussian Process; Kriging; Surrogate Modeling; Meta Modeling

1 Introduction

The manual modeling of technical systems with network elements is a challenging and time-consuming process. Considerable experience and knowledge on the working principles is necessary. Commercial software systems such as Dymola, Matlab/Simulink, SimulationX, etc. provide ready-to-use model libraries for this approach. However, they do not support the elaboration of an adequate network structure. In order to achieve a sufficient accuracy of the models, an expensive parameter identification has to be performed frequently. This can be achieved by comparing the simulation results with those from experimental investigations and then adjusting the network parameters.

An increasing demand for automatic model generation emerges from this (Fig. 1). Two ways are possible. The first way uses measurement data from a real product or process as a basis of the model. It is not necessary that the working principles or mathemati-

cal relations which describe the system are known. The system is assumed to be a black box.

The second way models the system rigorously, starting from real system geometry, discretizing the system in time and space and using partial differential equations. Mostly, the finite element method is used. Both approaches allow automatic model code generation for the usage in system simulation.

When the equation system is known and the finite element method is applied than a model order reduction (MOR) is possible [7]. However, MOR is not applicable on experimental data or black box models. Alternatively, in such cases, the adaptive Gaussian process [1-5] can be used to generate model codes automatically as we demonstrate in this paper. Meta or surrogate models are derived from these black box systems. Therefore, this approach is more general than MOR.

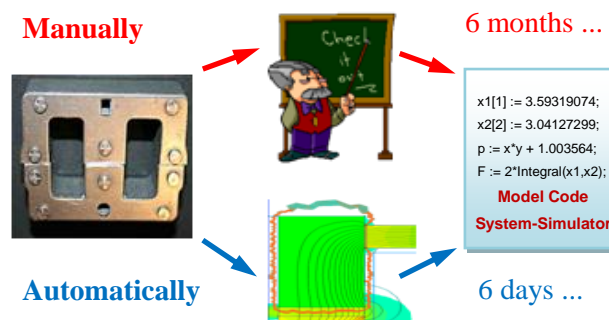


Fig 1: Different modeling approaches

2 Adaptive Gaussian Process

2.1 Gaussian Process

The Gaussian process, also known as Kriging, extends a multivariate normal distribution to infinite dimensionality [1, 2, 3]. The Gaussian process model $Y(\mathbf{x})$ is composed of the global function $f(\mathbf{x})$ and the stochastic process $Z(\mathbf{x})$ representing the deviation from the global function:

$$Y(\mathbf{x}) = \sum_{i=1}^p \beta_i \cdot f_i(\mathbf{x}) + Z(\mathbf{x}).$$

$f(\mathbf{x})$ is the polynomial regression function of any order, $\boldsymbol{\beta}$ are the unknown regression coefficients, and $Z(\mathbf{x})$ is a stationary stochastic process having mean zero, variance σ^2 and correlation function $R(\bullet)$. \mathbf{x} is a m -dimensional vector of input parameters.

It is assumed, that the training data or support points consist of the simulation or measurement data at the input set $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ and that $y(\mathbf{x}_0)$ is the predicted vector. The Gaussian process model implies the points $Y_0 = Y(\mathbf{x}_0)$ and $\mathbf{Y}_n = [Y(\mathbf{x}_1), \dots, Y(\mathbf{x}_n)]^T$ having the multivariate normal distribution:

$$\begin{pmatrix} Y_0 \\ \mathbf{Y}_n \end{pmatrix} \approx N_{n+1} \left[\begin{pmatrix} \mathbf{f}_0^T \\ \mathbf{F} \end{pmatrix} \boldsymbol{\beta}, \quad \sigma_z^2 \begin{pmatrix} 1 & \mathbf{r}_0^T \\ \mathbf{r}_0 & \mathbf{R} \end{pmatrix} \right].$$

$\mathbf{f}_0 = f(\mathbf{x}_0)$ is the $(p, 1)$ -vector of regression function for $Y(\mathbf{x}_0)$. $\mathbf{F} = [f_i(\mathbf{x}_i)]$ is the (n, p) -matrix of regression functions for the training data or support points. $\mathbf{r}_0 = [R(\mathbf{x}_0, \mathbf{x}_1), \dots, R(\mathbf{x}_0, \mathbf{x}_n)]^T$ is the $(n, 1)$ -vector of correlation functions of \mathbf{Y}_n with $Y(\mathbf{x}_0)$, and $\mathbf{R} = R(\mathbf{x}_i, \mathbf{x}_j)$ is the (n, n) -matrix of correlation functions among \mathbf{Y}_n .

Therefore, the best linear unbiased predictor for $Y(\mathbf{x}_0)$ is the mean value of the multivariate normal distribution. It represents the response surface, also known as meta- or surrogate model, of the real system:

$$\hat{Y}(\mathbf{x}_0) = \mathbf{f}_0^T \boldsymbol{\beta} + \mathbf{r}_0^T \mathbf{R}^{-1} (\mathbf{Y}_n - \mathbf{F} \boldsymbol{\beta}).$$

The uncertainty of the predicted value is characterized by the variance of the multivariate normal distribution:

$$\sigma^2 = \sigma_z^2 (1 - \mathbf{r}_0^T \mathbf{R}^{-1} \mathbf{r}_0 + (\mathbf{f}_0 - \mathbf{F}^T \mathbf{R}^{-1} \mathbf{r}_0)^T (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F})^{-1} (\mathbf{f}_0 - \mathbf{F}^T \mathbf{R}^{-1} \mathbf{r}_0)).$$

2.2 Correlation Function

The correlation function is the crucial ingredient in the Gaussian process predictor because it contains assumptions about the function to be predicted. It interpolates between support points in which its value smoothly changes between 0 and 1.

Several stationary correlation functions have been investigated to approximate a lot of real functions or systems [1]:

- Gamma exponential

$$R(\mathbf{x}_1, \mathbf{x}_2) = \exp \left\{ - \sum_{i=1}^m w_i^\gamma \cdot |\mathbf{x}_1 - \mathbf{x}_2|^\gamma \right\}$$

- Matérn class

$$R(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{\Gamma(\nu) 2^{\nu-1}} \left(\frac{2\sqrt{\nu} |\mathbf{x}_1 - \mathbf{x}_2|}{\theta} \right)^\nu K_\nu \left(\frac{2\sqrt{\nu} |\mathbf{x}_1 - \mathbf{x}_2|}{\theta} \right)$$

- Rational quadratic

$$R(\mathbf{x}_1, \mathbf{x}_2) = \left(1 + \frac{w^2 \cdot |\mathbf{x}_1 - \mathbf{x}_2|^2}{\alpha} \right)^{-\alpha}.$$

w, λ and α are hyper-parameters, which have to be identified using optimization methods in order to maximize the likelihood function of the multivariate normal distribution. In some cases, the Gamma-exponential correlation function is either exponential ($\gamma=1$) or square exponential ($\gamma=2$). For the Matérn class, the correlation functions with $\nu=3/2$ and $\nu=5/2$ are frequently used.

2.3 Adaptive Gaussian Process

With the variance σ^2 of the multivariate normal distribution, the confidence interval (3σ) of the response surface is available at any point. Thus, it is possible to measure the accuracy of the meta model. Besides the variance, the expected improvement (EI) has been introduced as a second factor for meta model evaluation purposes [4]. EI is defined as a potential improvement which is achieved by investigating the input parameter x :

$$EI = \sigma \left\{ \frac{Y_{\min} - \hat{Y}}{\sigma} \Phi \left(\frac{Y_{\min} - \hat{Y}}{\sigma} \right) + \Psi \left(\frac{Y_{\min} - \hat{Y}}{\sigma} \right) \right\}.$$

$\Phi(\bullet)$ and $\Psi(\bullet)$ are commulative distribution functions and probability density function of the normalized normal distribution. A third evaluation factor is the statistical low bounding (SLB) [5]:

$$SLB = \hat{Y} - k \cdot \sigma \quad \text{with } k = 1, 3, 5, \dots$$

Based on these three factors, the accuracy of the meta model can be improved by using additional support points, which are suggested by the optimization procedures to:

- maximize EI,
- maximize 3σ and
- minimize SLB.

Meta modeling is an adaptive process, which involves several loops of the Gaussian process. Starting from the initial sampling points, the response surface of the modeled system is built. Based on this response surface, additional support points are suggested and make possible that the new response surface is rebuilt more accurately. The process comes to an end either if a predetermined number of support points is computed or if a specified value of maxi-

imum EI is achieved. The necessary number of support points for a specific accuracy of the surrogate model depends on:

- the number of input parameters \mathbf{x} ,
- the correlation between input parameters and
- the degree of nonlinearity of the surrogate model.

The adaptive Gaussian process is very efficient for meta modeling. It requires less support points when compared with other design of experiment (DoE) methods in order to achieve a comparable accuracy.

3 Code-Extraction with OptiY[®]

For system modeling purposes, commercial software systems for CAD/CAE, FEA, CFD, electronic circuit simulation, electro-magnetics, etc. are available. It is better to use tailored programs for the different components of a complex model. The advantages are:

- easy usage and quick handling of the software,
- availability of expert knowledge,
- detailed and accurate component behavior modeling,
- small number of model parameters, which have to be identified.

For the system simulation, different component models created by different software programs have to be coupled. In general, this is difficult to arrange. System models contain component models having a large number of degrees of freedom. This results in a high computing effort for system simulation. Using surrogate models in form of Modelica codes instead of the underlying component models reduces computation time and cost drastically. Such fast system models allow robust design optimization (RDO), e. g. design for six sigma, and reliability based design optimization (RBDO) which both require a large number of runs of system models.

The multitude of software tools normally used for component modeling makes automatic computation of the meta models desirable. The multidisciplinary design software OptiY[®] supports the automatic generation of meta models in form of Modelica code used in system models (*Fig. 2*) [7]. This software provides generic and direct interfaces to many commercial software systems for CAD, FEA, electronic circuit simulation, CFD, multi body simulation, in-house codes etc. Furthermore, user can easily self-integrate external CAD/CAE-software for ease of use later with a predefined user element and script

template using Visual Basic or C# based on the .NET Framework[®] technology.

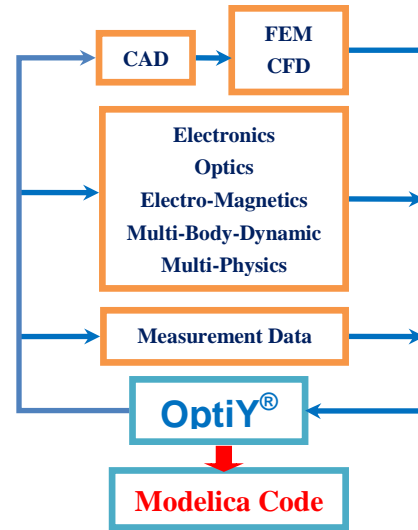


Fig 2: Auto-extraction of Modelica code with OptiY

The numerical algorithms of the adaptive Gaussian process presented in last section are also implemented in OptiY. For this reason an easy and quick-to-use connection to external component models is supported. Combined with the adaptive Gaussian process, this allows the automatic extraction of Modelica codes from external models or data.

With other implemented numerical algorithms in OptiY, following valuable possibilities are also available for the design process:

- Sensitivity Study
- Probabilistic Analysis
- Reliability Analysis
- Robustness Evaluation
- Six Sigma Design
- Robust Design
- Design Optimization
- Data Mining
- Parameter Identification

4 Electromagnetic Actuator

We use a Braille printer with an electromagnetic actuator in order to demonstrate the system simulation with surrogate models [8] (*Fig. 3*).

In the first step, the system model of the printer consists only of network elements which are taken from the model library of SimulationX [9]. The resulting network schematic is shown in *Fig. 4*, left side. The

model computes the dynamic behavior of the actuator.

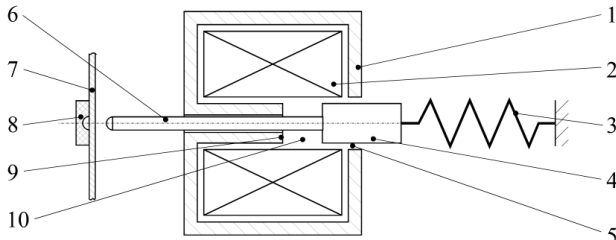


Fig. 3: Braille printer with electromagnetic actuator; 1 back iron, 2 coil, 3 return spring, 4 armature, 5 guiding air gap, 6 needle, 7 paper sheet, 8 die, 9 yoke, 10 working air gap

In the second step, we replace the magnetic network elements by a finite element model applying the software FEMM [10] (Fig 4).

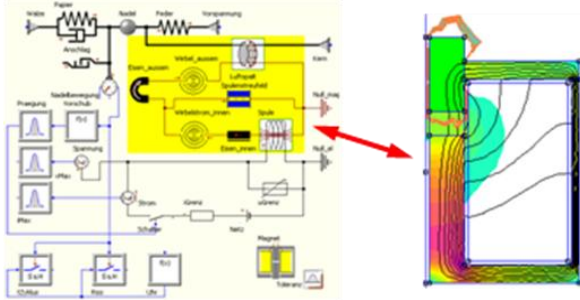


Fig 4: System model of the Braille printer; network model (left) and finite-element model (right) of the magnetic parts of the actuator

OptiY controls the FEMM tool, provides the input vector of the sampling points and collects the simulation results. After the simulation is finished, the surrogate model of the magnetic part of the actuator is computed by the adaptive Gaussian process and exported as Modelica code. All magnetic network elements (yellow block in Fig. 4) are replaced by surrogate Modelica codes of the magnetic force $F_m(x, i)$ and the flux density linkage $\Psi(x, i)$ (Fig. 5). x is the working air gap, i is the coil current. These codes are used to build a new system model in SimulationX.

These modeling steps have to be performed manually and only once. Each update of the surrogate Modelica code can be performed automatically inside of OptiY using the graphical workflow editor.

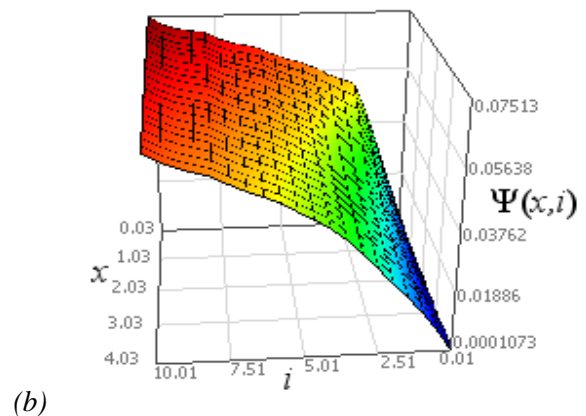
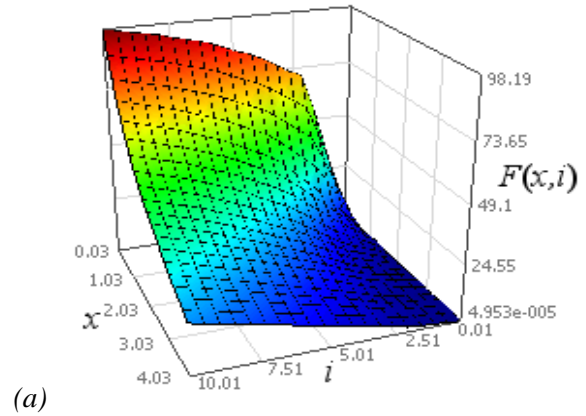


Fig. 5: Surrogate models of the magnetic force $F_m(x, i)$ (a) and the flux linkage $\Psi(x, i)$ (b)

Fig. 6 compares the dynamic behavior of the Braille printer computed with the network model and the surrogate model. The diagram reveals slight differences between the models due to the idealized representation of the coil in the magnetic network element. These differences are particularly notable when the back iron comes into saturation. Therefore, the system model using surrogate Modelica code yields better results of the printer behavior because the underlying finite element model considers spatial inhomogeneity of the magnetic field. However, eddy currents and magnetic hysteresis are neither covered by the network model nor the finite element model. A more detailed description of the example of the Braille printer can be found in [8].

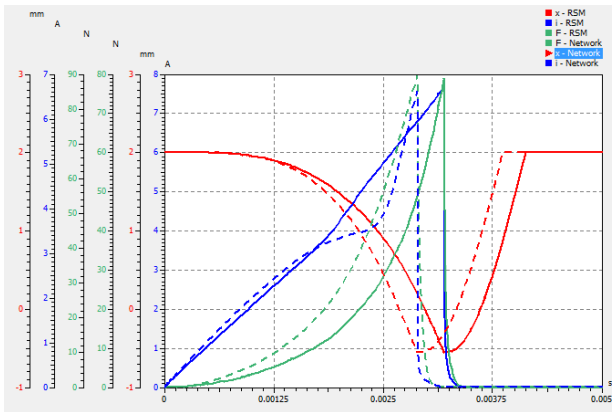


Fig. 6: Dynamic behavior of the Braille printer simulated by a network model (dashed) and a surrogate model (solid); x - printer needle displacement, i - coil current, F - magnetic force

5 Conclusions

Modeling technical systems with network elements is an adequate approach, however, challenging and time-consuming if performed manually. The adaptive Gaussian process is an approach that allows an efficient and automatic generation of precise component models for system modeling. It requires only few support points of the black box system. With additional support points, the accuracy of the computed meta model can be improved step by step if necessary. The mathematical meta model can be written as Modelica code. The algorithms which are needed for this procedure are implemented in the multidisciplinary design software OptiY[®]. It provides generic and direct interfaces to many specialized commercial CAD/CAE-software tools and also in-house codes. Within, user can easily create fast surrogate models and export them as Modelica code automatically.

The study case shows that the meta modeling process is very fast and useful. The amount of identified parameters is smaller in comparison to the network model. The system behavior is more accurate. The application of a Braille printer with an electromagnetic actuator has been demonstrated. Simulation results of a network model and a surrogate model have been compared. The use of fast meta models allows computationally intensive optimization and test procedures, e. g. robust design optimization.

References

- [1] Rasmussen C. E., Williams C. K. I.: Gaussian Process for Machine Learning. MIT Press 2006.
- [2] Santner, T. J., Williams, B. J., Notz, W. I.: The Design and Analysis of Computer Experiment. Springer New York 2003
- [3] Sacks J., Welch W. J., Mitchell T. J., Wynn H. P.: Design and Analysis of Computer Experiments. Statistical Science 4, pp. 409-435, 1989
- [4] Jones, R. D.: A Taxonomy of Global Optimization Methods Based on Response Surfaces. Journal of Global Optimization 21: 345-383, 2001
- [5] Xiong, Y., Chen, W, and Tsui, K.: A New Variable Fidelity Optimization Framework Based on Model Fusion and Objective-Oriented Sequential Sampling. ASME Journal of Mechanical Design , 130 (11), 2008
- [6] Antuolas, A. C.: Approximation of Large-Scale Dynamical Systems. SIAM 2005
- [7] OptiY Software and Documentation. www.optiy.eu
- [8] http://www.optiyummy.de/index.php/Software:_FEM_-_Tutorial_-_Magnetfeld, see Kennfeld-Export als Modelica-Code
- [9] SimulationX Software and Documentation. www.iti.de
- [10] FEMM Software and Documentation. www.femm.info