Power Module Heat Sink Design Optimization with Data-Driven Polynomial Chaos Surrogate Models

Dimitrios Loukrezis\textsuperscript{a,b,}\textsuperscript{*}, Herbert De Gersem\textsuperscript{a,b}

\textsuperscript{a}Institute for Accelerator Science and Electromagnetic Fields, Technische Universität Darmstadt, Darmstadt, Germany
\textsuperscript{b}Centre for Computational Engineering, Technische Universität Darmstadt, Darmstadt, Germany

Abstract

We consider the problem of optimizing the design of a heat sink used for cooling an insulated gate bipolar transistor (IGBT) power module. The thermal behavior of the heat sink is originally estimated using a high-fidelity computational fluid dynamics (CFD) simulation, which renders numerical optimization too computationally demanding. To enable optimization studies, we substitute the CFD simulation model with an inexpensive polynomial surrogate model that approximates the relation between the device’s design features and a relevant thermal quantity of interest. The surrogate model of choice is a data-driven polynomial chaos expansion (DD-PCE), which learns the aforementioned relation by means of polynomial regression. Advantages of the DD-PCE include its applicability in small-data regimes and its easily adaptable model structure. Once trained and tested in terms of prediction accuracy, the DD-PCE replaces the high-fidelity simulation model in optimization algorithms aiming to identify heat sink designs that optimize the thermal behavior of the IGBT under geometrical and operational constraints. Optimized heat sink designs are obtained for a computational cost much smaller than utilizing the original model in the optimization procedure. Comparisons against alternative surrogate modeling techniques illustrate why the DD-PCE should be preferred in the considered setting.

Keywords: data-driven modeling; design optimization; polynomial chaos expansion; power electronics; regression; supervised machine learning; surrogate modeling

1. Introduction

The widespread use of power electronic modules in diverse applications, ranging from common household devices to transportation systems and renewable energy sources, has resulted in high-quality standards in terms of operational reliability and robustness \cite{1}. To meet these demands, design processes of high precision are necessary. An essential challenge in power electronics design is restraining the temperature of an operating power module below an acceptable limit. To that end, it is necessary to complement the power module with a cooling system, typically a heat sink, such that the heat generated due to semiconductor
power losses is not accumulated within the module. Given specific operation features, e.g. power losses or ambient environment conditions, the heat sink must be designed such that the thermal stability and, thus, the operational reliability of the power module is guaranteed. At the same time, industrial demands often require that a suitable design must be delivered within strict time-frames, hence, the need for fast albeit still reliable design methods arises.

A common approach in heat sink design is to combine simulation models with numerical optimization methods in order to optimize the heat sink’s geometry with respect to a quantity of interest (QoI) related to the thermal behavior of the device [2–6]. This optimization is commonly conducted under cost- and size-related constraints. While unarguably successful, this design approach faces the bottleneck that the thermal or multi-physics simulations that are necessary to accurately approximate the thermal behavior of the device are usually time- and resource-demanding. Combined with the need to explore a possibly high-dimensional parameter space comprising operational and geometrical design features and run simulations for multiple configurations until an optimal set of feature values is identified, design optimization tasks often amount to an undesirable computational cost. This problem is especially relevant if stochastic optimization algorithms are employed [7, 8], which, despite typically being slower than deterministic optimization methods [9], are a popular choice in the context of engineering design due to their seamless, black-box application, as well as due to their ability to avoid local optima.

One way to accelerate optimization or other resource and time-intensive parametric studies is to replace the computationally demanding simulation with an inexpensive albeit sufficiently accurate surrogate model [10–14]. Among several surrogate modeling techniques available, an increasingly popular approach is to learn a model using existing design and operation data [15]. Owning to the ever improving data collection and storage capacities, many recent works have focused on machine learning (ML) methods [16], particularly on deep learning (DL) [17] and on the use of neural networks (NNs) as surrogate models [18, 19]. However, DL-based surrogate modeling typically requires a large amount of available data to train and test the NN model. This can be a crippling problem in cases where databases of sufficient size are not readily available and only a few data points can be generated within an acceptable time-frame. This is typically the case for numerous complex engineering applications, including heat sink design, e.g. considering data generated with the aforementioned computationally demanding simulations.

Motivated by the need for a data-driven surrogate modeling method which is applicable in small-data regimes, in this work we turn to classical statistical learning theory [20] and use polynomial regression to derive surrogate models known as polynomial chaos expansions (PCEs) [21–23]. PCEs were originally developed within the context of uncertainty quantification (UQ), where they are known to provide accurate and cost-efficient moment and sensitivity estimates for random QoIs [24, 25]. More recently, PCEs have been applied within a supervised ML context, i.e. for the purpose of providing purely data-driven predictive models [26]. The method was named data-driven polynomial chaos expansion (DD-PCE) and was found to perform comparably well against state-of-the-art ML methods, e.g. based on deep NNs and support vector
machines (SVMs). Besides model learning with small data, the choice of the DD-PCE method is further justified by the fact that, if combined with basis selection algorithms [27–41], the model structure of a DD-PCE can be easily adapted according to the influence of individual design features, desired approximation accuracy, and data availability. This is a significant advantage compared to ML models of fixed structure, e.g. deep NNs with a pre-determined architecture.

This work presents a framework based on surrogate modeling with DD-PCEs that enables otherwise intractable optimization studies with respect to the design of an air-cooled heat sink attached to an insulated gate bipolar transistor (IGBT). Our goal is to identify a heat sink design that results in the desired thermal behavior of the device, given application-dependent operational and geometrical constraints. The monitored QoI is the steady-state temperature of the heat sink, which is directly connected to the maximum temperature developed in the power module during its operation. For a given design, the heat sink’s temperature is originally estimated using a high-fidelity, 3D computational fluid dynamics (CFD) simulation. Due to the duration of a single simulation, using the 3D model in an optimization algorithm would result in an unacceptable computational cost, especially if a stochastic optimization algorithm is employed. Instead, we train a DD-PCE to approximate the dependency of the QoI on the design features using a relatively small set of simulation-generated data. A separate dataset is employed to test the predictive capability of the DD-PCE surrogate model. The verified DD-PCE then replaces the original model in the stochastic optimization algorithm, thus enabling the task of heat sink design optimization. Note that the training and test data are labeled, i.e. the datasets consist of temperature values for different design feature instances, hence, DD-PCE constitutes a supervised ML method.

The rest of this paper is organized as follows. Section 2 presents the setup of the device under investigation, consisting of an IGBT and an air-cooled heat sink, and discusses modeling, simulation, and design optimization aspects. Note that we consider a real-world IGBT as a test case, typical applications of which include motor control, uninterruptible power supply, and air-conditioning. Nevertheless, the framework suggested in this paper is generally applicable and not restricted to this particular power module. DD-PCEs are presented in Section 3. Numerical studies illustrating both the advantageous predictive capabilities of DD-PCEs compared to alternative data-driven surrogate modeling approaches and the utilization of DD-PCEs within stochastic optimization algorithms are presented in Section 4. Finally, our conclusions are presented in Section 5.

2. Power module heat sink design: modeling, simulation, and optimization

2.1. Device setup

An illustration of the IGBT module and the air-cooled heat sink attached to it is provided in Figure 1. The core elements of the IGBT module are power semiconductors, which generate heat due to conducting and switching losses. The semiconductors are connected to a direct copper bonded substrate, which acts as an isolation layer and consists of a copper layer, an epoxy-based dielectric layer, and another copper layer.
The lower copper layer is attached to a metallic base plate, which acts as the interface between the power module and the heat sink. A thin layer of thermal grease is inserted between the base plate and the heat sink to eliminate air gaps in the interface area. The heat sink consists of a root, which is the part attached to the base plate, thus accumulating heat due to conduction. This heat is then transferred towards the fins at the bottom of the heat sink, to be exchanged with the air that flows around them. The heat sink is painted black in order to maximize heat dissipation through radiation.

The heat accumulated within the power module during its operation depends on a number of factors, namely, the power losses $P$ of the semiconductors, the conditions of the surrounding environment, in particular the ambient temperature $T_a$ and the air flow velocity $v$, and the geometry of the heat sink. The latter is described by the length $l$, the base height $h_b$, the fin height $h_f$, the number of fins $N_f$, the fin width $w_f$, and the width of the gap between two fins $w_g$. All operational and geometrical design features are also shown in Figure 2 where a 3D simulation model of the power module and the heat sink is illustrated. Since the power module and the heat sink are designed for a certain application range, the design features take values within bounded intervals which are given in Table 1.

| Parameter               | Symbol | Unit     | Min. | Max. |
|-------------------------|--------|----------|------|------|
| Heat sink length        | $l$    | mm       | 50   | 200  |
| Fin gap                 | $g_f$  | mm       | 3    | 8    |
| Fin width               | $w_f$  | mm       | 1.4  | 4    |
| Fin height              | $h_f$  | mm       | 16   | 45   |
| Base height             | $h_b$  | mm       | 4    | 15   |
| Number of fins          | $N_f$  | –        | 5    | 25   |
| Air flow velocity       | $v$    | m s$^{-1}$ | 1    | 5    |
| Ambient temperature     | $T_a$  | °C       | 25   | 45   |
| Power loss              | $P$    | W        | 115  | 140  |
2.2. Thermal characterization of power modules

The thermal management of power electronic devices is typically based on monitoring the thermal impedance \( Z_{th} \) or resistance \( R_{th} \) between two locations of the device, upon which temperature measurements are taken. Note that, contrary to electric circuit design, the term “impedance” is used when considering transient heat dissipation, whereas the term “resistance” refers to stationary, steady-state heat dissipation. Assuming two locations on the device, respectively denoted with “1” and “2”, the thermal impedance is given by

\[
Z_{th,1\rightarrow2} = \frac{T_2(t) - T_1(t)}{P(t)}.
\] (1)

Removing the time dependence, i.e. for steady-state temperatures and power losses, the thermal resistance \( R_{th,1\rightarrow2} \) is obtained.

For investigating the thermal behavior of the IGBT module and the attached heat sink, the most commonly employed temperature positions are shown in Figure 1. These are: the junction temperature \( T_j \), measured on the semiconductor chip; the case temperature \( T_c \), measured at the bottom of the base plate; the heat sink’s temperature \( T_s \), measured exactly underneath the contacting surface with the power source, where the maximum temperature developed within the heat sink is observed; and the ambient temperature \( T_a \). The critical measure with respect to the reliable operation of the power module is the junction temperature \( T_j \), which must not exceed the threshold value given in the device’s specifications. For the particular IGBT under investigation, this threshold is 175°C. However, measuring the semiconductors’ temperature \( T_j \) is out of reach in most physical setups. Similar measurement difficulties arise for the case temperature \( T_c \).

Therefore, most IGBT specifications include a worst-case estimate for the thermal resistance of the power module \( R_{th,j\rightarrow c} \), which for the considered test case is 0.74 K W\(^{-1}\). Then, using thermal circuit model theory and taking into account that the steady-state temperature of the heat sink coincides with its maximum temperature during the operation of the IGBT, a worst-case estimate for the junction temperature \( T_j \) can be obtained using the formula

\[
T_j = T_a + (R_{th,j\rightarrow c} + R_{th,s\rightarrow a}) P,
\] (2)

where \( R_{th,s\rightarrow a} \) refers to the thermal resistance of the heat sink, which is given by

\[
R_{th,s\rightarrow a} = \frac{T_s - T_a}{P}.
\] (3)

Contrary to the junction and case temperatures, the heat sink’s temperature \( T_s \) can be easily measured, typically using a sensor drilled under the contact surface. The ambient temperature \( T_a \) is also easy to measure by placing sensors in the surrounding area of the device. As a result, to ensure the reliable operation of the power module, it suffices to monitor the heat sink’s temperature \( T_s \).

2.3. Simulation model

To accurately compute the heat sink’s temperature \( T_s \) and thus estimate the junction temperature \( T_j \) for a given set of operational and geometrical design feature values as described in Section 2.1, a 3D CFD
simulation is used to accurately resolve the heat transfer from the IGBT to the heat sink and its dissipation to the surrounding environment. The simulation model is implemented using the ANSYS Icepak software. Without delving into details, the simulation is based on the finite-volume discretization of the Reynolds-averaged Navier-Stokes (RANS) equations. Additionally, the Boussinesq approximation is chosen to model Reynolds stresses in the conservation of momentum equation and the \( k - \omega \) shear stress transport (SST) model is employed for turbulence modeling. The 3D simulation model is illustrated in Figure 2 and consists of three main components, namely the IGBT module, the heat sink, and a cubic box enforcing the necessary boundary conditions. In particular:

- The IGBT is not explicitly modeled, but represented only as a rectangular power source of fixed dimensions, shown with red in Figure 2. This modeling decision is connected to only considering the worst-case thermal resistance of the power module (see Section 2.2), hence, resolving the heat transfer inside the power module is not necessary. Moreover, the power losses are considered to be uniformly distributed on the area of the source. This assumption is justified by the fact that the considered IGBT module is equipped with a metallic base plate (see Section 2.1 and Figure 1).

- The heat sink is placed on the power source in order to dissipate the generated heat. The material of the heat sink is chosen such that its density and specific heat capacity remain constant over time, while its thermal conductivity is linearly dependent on its temperature.

- The cubic boundary box takes the form of an air duct, such that air flows in the positive \( x \)-direction from the left-side boundary to the right-side boundary. A uniform air flow of constant velocity \( v \) and at ambient temperature \( T_a \) is imposed on the left-side boundary, while the right-side boundary acts as a pressure outlet. The bottom boundary is an adiabatic wall, such that the heat generated by the power source is fully transferred to the heat sink. The remaining boundaries are open.

2.4. Heat sink design optimization

As already mentioned in Section 2.2, the critical measure regarding the reliable operation of the power module is the junction temperature \( T_j \), which must not exceed a critical threshold according to the device’s specifications. Hence, the main design goal is to identify operational and geometrical design features which satisfy this requirement. In principle, this can be accomplished by solving the optimization problem

\[
\min_{\mathbf{y}} T_j(\mathbf{y}),
\]

where \( \mathbf{y} = (l, g, w, h_f, h_b, N_f, v, T_a, P) \in \mathbb{R}^9 \) is the design feature vector and \( T_j(\mathbf{y}) \) denotes the dependency of the junction temperature on the design features. If the minimum junction temperature \( T_j^* \) obtained by solving problem [4] is below the given threshold, then a suitable heat sink design has been identified.

\[\text{https://www.ansys.com/products/electronics/ansys-icepak}\]
When solving the optimization problem (4), a number of constraints must be additionally taken into consideration. First, the design features take values within the bounded intervals given in Table 1. Second, the values of the ambient temperature $T_a$ and the power losses $P$ depend on the specific application of the IGBT and therefore lie outside of the design engineer’s control. On the contrary, the geometry of the heat sink can be fully controlled. The air flow velocity can also be controlled in most cases, e.g. using a fan. As a result, a constrained optimization problem must be solved, given as

$$\min_y T_j(y) \text{ subject to: } y \in \mathcal{D}, P = P_0, T_a = T_{a,0},$$

where $P_0, T_{a,0}$ denote the fixed power losses and ambient temperature, respectively, and $\mathcal{D} \subset \mathbb{R}^9$ is a set containing all admissible realizations of the parameter vector $y$.

However, solving the constrained optimization problem (5) will typically result in designs which maximize the size of the heat sink, thus also its material requirements and manufacturing cost. On the contrary, it is desirable that the heat sink remains as compact as possible, while still achieving a sufficiently low junction temperature. This requirement leads to the multi-objective optimization [50, 51] problem

$$\min_y (T_j(y), V_a(y)) \text{ subject to: } y \in \mathcal{D}, P = P_0, T_a = T_{a,0},$$

where $V_a(y)$ denotes the volume occupied by the heat sink for a given parameter vector. Typically, there is no unique solution that minimizes both objectives at the same time. Instead, there exist several solutions which cannot be further improved with respect to one objective without simultaneously worsening the other. These solutions are called Pareto optimal and form a so-called Pareto front. Once the Pareto front is available, the design engineer can choose the final heat sink designs according to his/her expertise.

Note that, as discussed in Section 2.2, the steady-state temperature of the heat sink $T_a$ is directly
connected to the maximum junction temperature $T_j$ via formulas (3) and (2). Therefore, all aforementioned optimization problems can be alternatively formulated such that a minimal value for $T_s$ is sought instead of $T_j$.

There exist numerous methods and algorithms for solving the aforementioned optimization problems. In the context of engineering design, stochastic optimization methods \[52\] are particularly popular due to the fact that they employ computational models in a black-box manner and avoid convergence problems due to local optima, which are often encountered when using deterministic optimization methods. On the downsides, stochastic optimization algorithms typically require a large number of model evaluations until converging to the sought optimal solutions, which can lead to severe computational costs, particularly if the employed model is time and resource demanding. Hence the necessity for an inexpensive albeit sufficiently accurate surrogate model, which is the focus of Section 3.

3. Data-driven polynomial chaos expansion

3.1. Polynomial chaos expansion

We assume that the functional dependence between a scalar QoI and an $N$-dimensional vector of features $y = (y_1, \ldots, y_N)$ is described by the deterministic map $S : \mathbb{R}^N \rightarrow \mathbb{R}$. In the context of this work, the feature vector refers to the operational and geometrical heat sink design features such that $y = (l, g_l, w_l, h_l, h_b, N_l, v, T_a, P)$, the map $S$ refers to evaluating and post-processing the RANS-based CFD simulation model (see Section 2.3), and the QoI is the heat sink’s temperature $T_s$, which can then be used to infer the junction temperature $T_j$ using formulas (3) and (2).

A PCE is a global polynomial approximation of the form

$$S(y) \approx \tilde{S}(y) = \sum_{k=1}^{K} s_k \Psi_k(y),$$

(7)

where $s_k$ are scalar coefficients and $\Psi_k$ are multivariate orthonormal polynomials, such that

$$\int_{\mathbb{R}^N} \Psi_k(y) \Psi_l(y) \varrho(y) \, dy = \delta_{kl},$$

(8)

with $\delta_{kl}$ being the Kronecker delta and $\varrho(y)$ the joint probability density function (PDF) that collectively characterizes the design features. Assuming that feature realizations occur independently from one another, the joint PDF is given as

$$\varrho(y) = \prod_{n=1}^{N} \varrho_n(y_n),$$

(9)

where $\varrho_n(y_n), n = 1, \ldots, N$, are univariate PDFs corresponding to single features. We note that the independence assumption is not crucial, i.e. dependent parameters can be addressed with suitable transformations \[53\]. Nevertheless, it has been shown that the independence assumption often leads to improved results in the context of data-driven approximation via PCEs \[26\]. For the purpose of this study, the design features
are modeled as uniformly distributed within the corresponding bounds given in Table 1, i.e. they may take any value within these bounds with equal probability. The multivariate polynomials are then given by

$$\Psi_k(y) = \Psi_p(y) = N \prod_{n=1}^{N} \psi_{p_n}(y_n),$$

where $\psi_{p_n}$ are univariate Legendre polynomials in $y_n$ with degree $p_n$. The univariate Legendre polynomials are orthonormal with respect to the uniform univariate PDFs $g_n(y_n)$ [23]. The multi-index $p = (p_1, \ldots, p_N)$ collects the polynomial degrees of the univariate polynomials that comprise the multivariate polynomial $\Psi_p$ in (10). There exists a one-to-one relation between the multi-indices $p$ and the global indices $k = 1, \ldots, K$ in (7), such that

$$\tilde{S}(y) = \sum_{k=1}^{K} s_k \Psi_k(y) = \sum_{p \in \Lambda} s_p \Psi_p(y),$$

where $\Lambda$ is a multi-index set with cardinality $\#\Lambda = K$.

3.2. Polynomial basis and model structure

There exist numerous options regarding the model structure of a PCE, equivalently, for choosing the multi-indices that comprise the multi-index set $\Lambda$ and thus form the polynomial basis $\{\Psi_p\}_{p \in \Lambda}$. One obvious choice is to construct the multivariate polynomial basis as a tensor product of $N$ univariate polynomial bases, such that

$$\{\Psi_p\}_{p \in \Lambda} = \{\psi_{p_1}^1\}_{p_1=1}^{P_1} \otimes \cdots \otimes \{\psi_{p_N}^N\}_{p_N=1}^{P_N},$$

where $\otimes$ denotes a tensor product and $P_n \in \mathbb{Z}_{\geq 0}$, $n = 1, \ldots, N$. Assuming that $P_n = P$, $n = 1, \ldots, N$, the corresponding multi-index set is given by

$$\Lambda_{\text{TP}}^P = \left\{ p : \max_{n=1,\ldots,N} \{p_n\} \leq P \right\},$$

and the cardinality of the basis is equal to $\#\Lambda_{\text{TP}}^P = P^N$. Due to the exponential dependence of the basis size on the number of features, tensor-product bases become intractable even for relatively small values of $P$ and $N$.

Motivated by the so-called “sparsity of effects” principle [29], i.e. the assumption that only few feature interactions have a non-negligible impact on the QoI, additional constraints on the model structure can be placed. A popular choice is to use a so-called total-degree PCE basis, which comprises polynomials corresponding to the multi-index set

$$\Lambda_{\text{TD}}^P = \left\{ p : \sum_{n=1}^{N} p_n \leq P \right\},$$

where $P \in \mathbb{Z}_{\geq 0}$, in which case the cardinality of the PCE basis is reduced to $\#\Lambda_{\text{TD}}^P = (N+P)!^N P!$. The number of basis terms can be reduced even further if a hyperbolic basis [29] is used, corresponding to the multi-index set

$$\Lambda_{\text{H}}^P = \left\{ p : \left( \sum_{n=1}^{N} (p_n)^q \right)^{1/q} \leq P \right\},$$

9
where $0 < q < 1$. Note that for $q = 1$, the total-degree basis given in (14) is retrieved.

PCEs based on tensor-product, total-degree, or hyperbolic polynomial bases are models of fixed structure. Equivalently, the multi-index set $\Lambda$ is a priori chosen. This can be a limiting factor due to the fact that a fixed model may contain terms that do not affect significantly the QoI and could therefore be omitted. Accordingly, a fixed model may neglect influential terms which could improve its predictive capability if added. Additionally, these bases are isotropic, i.e. they handle the input features in a similar manner, while in most practical cases the features have a different impact on the QoI which can be taken into consideration to form an anisotropic basis. Therefore, numerous approaches for constructing so-called sparse PCEs that neglect non-influential basis terms have been proposed, where the PCE coefficients are computed using either numerical quadrature \cite{54–57} or regression \cite{27–41}. The latter approach is essential for DD-PCEs, as it recasts the PCE as a supervised ML method based on polynomial regression, as discussed next.

3.3. Polynomial chaos regression

We now assume that a fixed polynomial basis $\{\Psi_p\}_{p \in \Lambda}$, $\#\Lambda = K$, and a dataset $D = \{\bm{y}(m), S(\bm{y}(m))\}_{m=1}^{M}$ containing $M$ realizations of the feature vector and the corresponding QoI are available. This dataset may consist of measurement or simulation-generated data, e.g. by evaluating the model presented in Section 2.3 for different design configurations. The PCE coefficients $s = (s_p)_{p \in \Lambda}$ can then be computed by means of regression \cite{20, 58}, as follows.

If $M > K$, the coefficient vector $s$ is obtained by solving the discrete least squares (LS) minimization problem

$$
\min_{\hat{s} \in \mathbb{R}^K} \sum_{m=1}^{M} \left( S(\bm{y}(m)) - \sum_{p \in \Lambda} \hat{s}_p \Psi_p(\bm{y}(m)) \right)^2.
$$

(16)

To solve (16), we introduce the so-called design matrix $D \in \mathbb{R}^{M \times K}$ with elements $d_{mk} = \Psi_k(\bm{y}(m))$ and the QoI value vector $b = (S(\bm{y}(1)), \ldots, S(\bm{y}(M)))$. The discrete minimization problem (16) can then be written in the equivalent algebraic form

$$
\min_{\hat{s} \in \mathbb{R}^K} \| D\hat{s} - b \|_2^2.
$$

(17)

Applying the necessary conditions for a minimum, we obtain the solution

$$
\hat{s} = (D^\top D)^{-1} D^\top b.
$$

(18)

To avoid conditioning problems, the solution is more commonly obtained with a QR decomposition of the design matrix $D$ \cite{59}. The conditioning of the LS problem can be further improved with the use of optimal experimental design methods \cite{31, 57, 60–64}.

If $M < K$, the LS problem (16) is ill-posed. Moreover, even if $M > K$, but with $M \approx K$, the LS problem will most probably suffer from conditioning issues leading to an overfitted solution \cite{65}. In that case, a
regularized LS problem is solved instead, given as

\[ \mathbf{s} = \arg \min_{\mathbf{s} \in \mathbb{R}^K} \left\{ \sum_{m=1}^{M} \left( S(y^{(m)}) - \sum_{p \in \Lambda} \hat{s}_p \Psi_p(y^{(m)}) \right)^2 + \lambda R(\mathbf{s}) \right\}, \tag{19} \]

where the loss function of (16) is modified with the addition of a regularization function \( R(\cdot) \), which imposes some form of constraint upon the coefficient vector, and a weighting parameter \( \lambda \) which controls the importance of the regularization in the minimization. Specifically for computing PCEs, it is common to use \( \ell_1 \)-regularization, such that \( R(\hat{s}) = \|\hat{s}\|_1 \), a technique closely connected to the least absolute shrinkage and selection operator (LASSO) method in statistics and to compressed sensing methods in signal processing. The \( \ell_1 \)-regularization forces multiple coefficients to zero, thus effectively reducing the model parameters and resulting in a sparse PCE. The \( \ell_1 \)-regularized LS problem is commonly solved with linear programming algorithms or with the least angle regression (LAR) algorithm. The latter approach can additionally be complemented with optimal experimental design and active learning methods.

### 3.4. Model uncertainty and ensemble modeling

As previously noted, DD-PCE is a supervised ML method. As is common in the context of supervised ML, to train and assess a data-driven model, an initial dataset of feature realizations and corresponding QoI observations is separated into a training dataset \( \mathcal{D}_{\text{train}} = \{ y^{(m)}, S(y^{(m)}) \}_{m=1}^{M} \) and a test dataset \( \mathcal{D}_{\text{test}} = \{ y^{(j)}, S(y^{(j)}) \}_{j=1}^{J} \), such that \( \mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{test}} = \emptyset \). The training dataset is here employed to compute the PCE coefficients by means of regression, i.e. for solving the problem (16), respectively (19). The test dataset is then used to evaluate the prediction accuracy of the trained model on previously unseen data, e.g. using a suitable error norm. However, the performance of data-driven surrogate models is crucially dependent on which subsets of the full dataset are selected as training or test data. That is, a specific partition of the dataset to training and test data might lead to results that are not representative of the actual predictive capability of the model. Exemplarily, in the case of DD-PCEs, different training datasets result in variations in the PCE coefficients, thus affecting the model’s predictions. For sparse PCEs in particular, different training datasets can lead to variations in the number of non-zero coefficients, subsequently to the number of expansion terms.

To avoid this pitfall, in this work we resort to the solution of ensemble modeling. That is, the original dataset is randomly re-ordered \( I \) times, thus generating training and test datasets \( \mathcal{D}_{\text{train}}^{(i)}, \mathcal{D}_{\text{test}}^{(i)} \), \( i = 1, \ldots, I \), accordingly, an ensemble of surrogate models \( \{ \tilde{S}^{(i)}(y) \}_{i=1}^{I} \). Aggregating the results over the \( I \) dataset combinations and corresponding surrogate models, one can then compute mean values and standard deviations for the error metrics which are used to evaluate the surrogate model’s accuracy. In that way, we are not only able to assess the performance of a surrogate model with respect to a specific accuracy metric on average, but also evaluate its robustness against variations in the training and test data. Once available, all surrogate models \( \tilde{S}(y), i = 1, \ldots, I \), can be employed for predicting the QoI for a given feature.
combination, such that the QoI prediction takes the form of a mean value along with a standard deviation or a confidence interval, thus quantifying the prediction uncertainty attributed to the uncertainty in the model. It is expected that model uncertainty and the corresponding uncertainty in the data-driven model’s predictions will be reduced as the training dataset is increased in size, as also verified by the numerical results presented in Section 4.

4. Application and Numerical Results

4.1. Predictive capability of data-driven polynomial chaos expansions

The numerical experiments presented in this section aim to assess the predictive capabilities of DD-PCEs, as well as compare them against other data-driven surrogate models, specifically Gaussian processes (GPs) and NNs. Surrogate models based on SVMs were also tested, but their predictive capability was found to be significantly worse than the other surrogate models, therefore, SVM-based results have been omitted. The compared surrogate models are described next.

• Both total-degree and sparse DD-PCE surrogate models are employed. The polynomial basis of the total-degree DD-PCE is given as in [14] with \( P = 3 \), which yielded superior results compared to total-degree bases with different maximum polynomial orders. The sparse DD-PCE is obtained with a sensitivity-adaptive algorithm that additionally incorporates an optimal experimental design criterion [37]. In particular, the polynomial basis is adaptively expanded by including polynomial terms corresponding to large PCE coefficients, which in turn are connected to variance-based sensitivity indices [24, 25], as long as the condition number of the design matrix \( D \) remains smaller than 10. This optimal experimental design criterion was found to perform better than other values used for the maximum allowed condition number.

• The GP surrogate model employs a Matérn covariance function, also referred to as the “kernel”, with a Matérn coefficient \( \nu = 5/2 \). This choice was found to outperform other popular kernel options, e.g. (squared) exponential kernels or Matérn kernels with different coefficient values.

• The NN surrogate model consists of 2 hidden layers with 10 neurons per layer. Hyperbolic tangent activation functions are employed and a limited memory BFGS optimizer is used for minimizing the loss function. These options with respect to the architecture of the NN and its training were found to outperform several other tested alternatives.

To train and assess the accuracy of the surrogate models, we use a dataset with 935 readily available feature value combinations and the corresponding heat sink temperatures \( T_s \). These results have been obtained using the ANSYS Iicepak simulation model presented in Section 2.3 for random realizations of the design features within the value ranges given in Table 1. Note that the average duration of a single simulation on an up-to-date machine is 35 minutes. We compute surrogate models using training datasets of increasing
size, i.e. \( M = 100, 200, \ldots, 800 \). The test dataset has a fixed size of \( J = 135 \) data points, which are not included in the training data. For the ensemble modeling procedure described in Section 3.4 the original dataset is randomly re-ordered \( I = 100 \) times.

The performance of a surrogate model in terms of predictive capability is assessed using the following error metrics. Denoting the output of the original model with \( S(y) \) and that of a surrogate model with \( \tilde{S}(y) \), the relative prediction error on the \( j \)-th test data point is

\[
\epsilon_{rel,j} = \left| \frac{S(y^{(j)}) - \tilde{S}(y^{(j)})}{S(y^{(j)})} \right|.
\]

The average performance of the surrogate model in terms of prediction accuracy is quantified using the mean absolute percentage error

\[
\epsilon_{mean} = \frac{1}{J} \sum_{j=1}^{J} \epsilon_{rel,j}.
\]

To obtain a worst-case performance estimate, we compute the maximum absolute percentage error

\[
\epsilon_{max} = \max_{j=1, \ldots, J} \{ \epsilon_{rel,j} \}.
\]

Aggregating the error results over the \( I = 100 \) dataset combinations and corresponding surrogate models, we compute the mean values \( \mu(\cdot) \) and the standard deviations \( \sigma(\cdot) \) of the errors (21) and (22). Moreover, an absolute worst-case prediction estimate is computed as \( \max_{i} \{ \epsilon_{(i)}^{max} \} \), \( i = 1, \ldots, 100 \).

The corresponding numerical results are presented in Figure 3. It can be observed that sparse-adaptive DD-PCEs outperform all other surrogate models for all training dataset sizes, both in terms of accuracy and robustness, as they yield a superior predictive accuracy both on average and in the worst-case, while the corresponding error standard deviation values are significantly smaller. Comparing the errors of sparse-adaptive and total-degree DD-PCEs, it is clear that the flexibility in terms of model structure offered by the former plays a crucial role in both prediction accuracy and robustness. Of particular interest is the observation that sparse-adaptive DD-PCEs trained with as few as \( M = 200 \) data points reach error values that cannot be reached by the other surrogate models even if trained with datasets four times larger. We can therefore conclude that, in the considered setting, sparse-adaptive DD-PCEs are particularly suitable for computing accurate surrogate models using only small training datasets.

4.2. Surrogate-based stochastic optimization

In the following, we consider that the considered power module operates at a fixed ambient temperature \( T_a = 45 \)°C and with constant power losses \( P = 140 \) W, i.e. the maximum allowed values for these two features (see Table 1). In the first optimization study, shown in Section 4.2.1 we aim to identify heat sink designs that minimize the heat sink temperature \( T_s \), thus the junction temperature \( T_j \) as well, see Sections 2.2 and 2.4. To that end, the single-objective optimization problem (4) is solved using the particle swarm optimization (PSO) [78, 79] and differential evolution (DE) [80] algorithms. In the second optimization study, shown in Section 4.2.2 two design objectives are considered as in the multi-objective optimization problem (6), such
that the junction temperature $T_j$ and the heat sink’s volume $V_s$ are simultaneously minimized. In this case we use a non-dominated sorting genetic algorithm (NSGA) algorithm [81, 82]. All optimization algorithms are implemented in the Python package pymoo [83]. For both optimization problems, the sparse-adaptive DD-PCE surrogate models are employed within the optimization algorithms, as they were found to outperform the competitive surrogate models in Section 4.1.

### 4.2.1. Single-objective surrogate-based optimization

By using the sparse-adaptive DD-PCE surrogate models \( \{ \hat{S}^{(i)}_M(y) \}_{i=1}^{100} \), trained with datasets of increasing size $M = 100, 200, \ldots, 800$, (see Section 4.1), the optimization problem (4) is effectively solved $I$ times per training dataset size, i.e. once with each corresponding surrogate. Then, by aggregating the $I$ optimization results, the minimum heat sink temperature $T_{s,\text{min}}$ is given in the form of a mean value and a corresponding standard deviation. The same holds for the optimal design parameter values, denoted in the following as $l^*, \ g_f^*, \ w_f^*, \ h_f^*, \ N_f^*, \ \text{and} \ \sigma^*$. Denoting the mean and standard deviation values of $T_{s,\text{min}}$ with $\mu (T_{s,\text{min}})$ and $\sigma (T_{s,\text{min}})$, respectively, a worst-case estimate for the minimum junction temperature $T_{j,\text{min}}$ is obtained by setting $T_{s,\text{min}} = \mu (T_{s,\text{min}}) + 6\sigma (T_{s,\text{min}})$, as is often done in the context of reliability engineering to ensure that the design will comply to the requirements with a very high probability. Such designs are commonly referred to as six-sigma ($6\sigma$) [84].

Tables 2 and 3 present the optimization results and the performance of the PSO and DE algorithms using DD-PCE surrogate models of increasing accuracy, equivalently, DD-PCEs trained with datasets of increasing size. A first observation is that the two optimization algorithms yield very similar results, however, the
computational cost of the PSO algorithm, as measured by the number of model evaluations until an optimal design is identified, is significantly lower than the one of the DE algorithm. Nevertheless, even in the case of the PSO algorithm, the computational gains due to using the surrogate model instead of the original 3D CFD heat sink model are tremendous. Exemplarily, the (on average) 1734 model evaluations with each of the 100 DD-PCE surrogate models computed with a training data set of size $M = 800$ (see Table 2, last column), amount to 4.8 CPU-hours of computation. This computation time is exceeded by merely 9 evaluations of the original heat sink model.

Considering the optimization results, it can be observed that using surrogate models of reduced accuracy results in an underestimation of the minimum heat sink temperature. Additionally, there is a much greater uncertainty attached to these results, as indicated by the corresponding standard deviation values. There is a certain tendency to be observed, that is, as the size of the training dataset increases, along with the accuracy of the surrogate model, the estimations for $T_{s,\text{min}}$ become more pessimistic. At the same time, the optimization results become increasingly more robust, as indicated by the progressively reduced standard deviation values. In the particular case of the least accurate surrogate model, i.e. for $M = 100$, the uncertainty regarding the value of $T_{s,\text{min}}$ leads to a six-sigma value for $T_{j,\text{min}}$ which significantly exceeds the threshold of 175 °C (also see Section 2.2). This constraint is respected for the remaining surrogate models and the margin to the threshold becomes more pronounced as the surrogate models increase in accuracy. The corresponding optimal design parameter values, again given in the form of mean value and standard deviation, are shown in Tables 3 and 5 for the PSO and DE algorithms, respectively. Once again, the PSO and DE algorithms yield very similar results, particularly for DD-PCEs of increased accuracy.

**Table 2: Single-objective constrained optimization with the PSO algorithm: minimum temperature values and computational cost.**

| Training dataset size $M$ | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
|---------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Average model evaluations | 1639| 1696| 1681| 1598| 1542| 1600| 1603| 1734|
| $T_{s,\text{min}}$ (°C), mean | 51.9| 55.5| 58.1| 58.6| 59.1| 59.7| 60.1| 60.5 |
| $T_{s,\text{min}}$ (°C), st. dev. | 5.4 | 2.6 | 1.7 | 1.7 | 1.2 | 1.2 | 0.9 | 0.7 |
| $T_{j,\text{min}}$ (°C), six-sigma | 187.9| 174.7| 171.9| 172.4| 169.9| 170.5| 169.1| 168.3 |

**Table 3: Single-objective constrained optimization with the DE algorithm: minimum temperature values and computational cost.**

| Training dataset size $M$ | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
|---------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Average model evaluations | 6734| 7672| 7849| 7505| 7452| 7299| 7275| 7090|
| $T_{s,\text{min}}$ (°C), mean | 51.8| 55.3| 58.1| 58.7| 59.1| 59.7| 60.2| 60.5 |
| $T_{s,\text{min}}$ (°C), st. dev. | 5.4 | 2.6 | 1.7 | 1.6 | 1.2 | 1.1 | 0.8 | 0.7 |
| $T_{j,\text{min}}$ (°C), six-sigma | 187.8| 174.8| 171.9| 171.9| 169.9| 169.9| 168.6| 168.3 |

15
Table 4: Single-objective constrained optimization with the PSO algorithm: optimal design parameters.

| Training dataset size $M$ | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
|---------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| $h^*_b$ (mm), mean        | 14.2| 14.8| 15.0| 14.9| 15.0| 15.0| 14.8| 14.8|
| $h^*_b$, st. dev.         | 1.4 | 1.0 | 0.2 | 0.2 | 0.2 | 0.3 | 1.0 | 0.8 |
| $N^*_f$, mean             | 24.0| 22.6| 23.4| 24.2| 24.6| 24.2| 24.4| 23.7|
| $N^*_f$, st. dev.         | 2.8 | 2.5 | 2.4 | 1.8 | 1.0 | 1.1 | 0.8 | 1.1 |
| $h^*_f$ (mm), mean        | 33.6| 43.6| 40.5| 39.1| 41.3| 41.8| 42.1| 42.5|
| $h^*_f$, st. dev.         | 8.8 | 6.1 | 7.9 | 7.9 | 4.7 | 4.1 | 3.0 | 2.8 |
| $l^*$ (mm), mean          | 183.2| 185.9| 170.1| 184.2| 185.5| 189.4| 189.5| 192.8|
| $l^*$ (mm), st. dev.      | 31.4| 23.6| 25.2| 22.5| 22.9| 19.4| 19.3| 15.5|
| $v^*$ (m s$^{-1}$), mean  | 4.0 | 4.8 | 4.7 | 4.7 | 4.6 | 4.7 | 4.7 | 4.5 |
| $v^*$ (m s$^{-1}$), st. dev| 1.2 | 0.6 | 0.5 | 0.5 | 0.5 | 0.6 | 0.6 | 0.6 |
| $g^*_t$ (mm), mean        | 5.1 | 3.2 | 3.1 | 3.1 | 3.2 | 3.0 | 3.0 | 3.0 |
| $g^*_t$, st. dev.         | 2.3 | 1.1 | 0.7 | 0.5 | 0.5 | 0.8 | 0.05| 0.05|
| $w^*_t$ (mm), mean        | 3.1 | 3.2 | 3.9 | 4.0 | 3.9 | 4.0 | 4.0 | 3.9 |
| $w^*_t$, st. dev.         | 1.0 | 0.7 | 0.3 | 0.2 | 0.2 | 0.3 | 0.3 | 0.4 |

Table 5: Single-objective constrained optimization with the DE algorithm: optimal design parameters.

| Training dataset size $M$ | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
|---------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| $h^*_b$ (mm), mean        | 13.8| 14.7| 15.0| 14.9| 14.9| 14.9| 14.9| 14.9|
| $h^*_b$, st. dev.         | 1.6 | 0.9 | 0.2 | 0.3 | 0.1 | 0.5 | 0.4 | 0.3 |
| $N^*_f$, mean             | 24.2| 21.9| 22.7| 23.9| 24.6| 24.2| 24.3| 23.7|
| $N^*_f$, st. dev.         | 2.2 | 2.5 | 2.6 | 2.3 | 0.9 | 1.0 | 0.8 | 1.0 |
| $h^*_f$ (mm), mean        | 33.1| 44.3| 42.2| 40.4| 41.0| 42.0| 41.9| 42.0|
| $h^*_f$, st. dev.         | 8.9 | 4.0 | 6.1 | 6.4 | 4.8 | 3.0 | 3.0 | 2.6 |
| $l^*$ (mm), mean          | 183.2| 185.9| 170.1| 184.2| 185.5| 189.4| 189.5| 192.8|
| $l^*$ (mm), st. dev.      | 31.1| 13.8| 21.4| 21.9| 17.5| 17.6| 13.6| 12.3|
| $v^*$ (m s$^{-1}$), mean  | 4.2 | 4.9 | 4.7 | 4.7 | 4.7 | 4.7 | 4.7 | 4.7 |
| $v^*$ (m s$^{-1}$), st. dev| 1.0 | 0.2 | 0.3 | 0.4 | 0.5 | 0.5 | 0.5 | 0.4 |
| $g^*_t$ (mm), mean        | 4.5 | 3.6 | 3.1 | 3.0 | 3.1 | 3.1 | 3.0 | 3.0 |
| $g^*_t$, st. dev.         | 2.1 | 1.5 | 0.6 | 0.2 | 0.7 | 0.6 | 0.02| 0.03|
| $w^*_t$ (mm), mean        | 3.1 | 3.0 | 3.9 | 4.0 | 3.9 | 3.9 | 3.9 | 3.8 |
| $w^*_t$, st. dev.         | 1.0 | 0.8 | 0.4 | 0.2 | 0.3 | 0.2 | 0.3 | 0.4 |
Figure 4: Pareto front solutions for each of the $I = 100$ sparse-adaptive DD-PCE surrogate models computed with training datasets of size $M = 100, 200, \ldots, 800$.

4.2.2. Multi-objective surrogate-based optimization

Similar to Section 4.2.1, we again use the sparse-adaptive PCE models $\{\tilde{S}_M(i)\}_{i=1}^{I=100}$ to compute $I$ Pareto fronts for each training dataset size $M = 100, 200, \ldots, 800$. The corresponding results are presented in Figure 4 where each panel corresponds to a different value for $M$ and all $I$ Pareto fronts are plotted within a panel. Note that the Pareto fronts cannot be aggregated and averaged in a simple manner, e.g. as done with the single-objective optimization results presented in Tables 2 and 3, as each Pareto front comprises significantly different $T_j$-$V_s$ pairs, equivalently, a different Pareto front “trajectory” is obtained for each $i = 1, \ldots, 100$. For the same reason, we do not provide optimal parameter values as in Tables 4 and 5. The task of choosing an optimal parameter configuration given the Pareto front solutions is typically left to the expertise of the design engineer.

As expected, increasing the volume of the heat sink results in lower junction temperatures and this trend is captured by all surrogate models. Nevertheless, it is now even more obvious that surrogate model accuracy plays a crucial role in the robustness of the optimization results, as can be observed from Figure 4. In particular, the overall variation in the Pareto front solutions obtained with surrogate models of lower accuracy is very large, especially concerning heat sink design realizations corresponding to small junction temperature values. Increasing the accuracy of the surrogate models results in significantly more conservative Pareto fronts, which at the same time do not present as strong variations and the corresponding results can be considered to be robust. It is however still the case, that variations in the Pareto fronts are greater for the heat sink designs that correspond to the lowest junction temperature values, while a robustification of the Pareto fronts can be observed as the junction temperature increases.

Concerning the computational cost of the multi-objective optimization by means of the NSGA algorithm, the average number of surrogate model evaluations for each training dataset size $M = 100, 200, \ldots, 800$, is
given in Table 6. In comparison to the single-objective optimization costs given in Tables 2 and 3 for the PSO and DE algorithms, respectively, it is obvious that multi-objective optimization is significantly more expensive. In this case, using a surrogate model in the place of the original 3D CFD model is even more advantageous in terms of computational demand.

5. Conclusions

This work presented a data-driven surrogate modeling framework based on DD-PCEs, which has been applied to alleviate the costs of single- and multi-objective heat sink design optimization. An ensemble modeling technique was additionally applied to avoid pitfalls related to partitioning datasets to training and test data and enhance the surrogate model’s predictions with statistical metrics such as mean and variance. Of particular interest is the performance of the data-driven surrogate modeling approach in the context of small data learning, as the employed training datasets have relatively small sizes of $M = 100, 200, \ldots, 800$ data points.

Based on the numerical results presented in Section 4.1, surrogate models based on DD-PCEs, in particular based on sparse-adaptive PCEs, are significantly more accurate and robust compared to competitive surrogate modeling methods based on GPs and NNs, at least for the specific test case considered in this work. Sparse-adaptive DD-PCEs were then used within stochastic single- and multi-objective optimization algorithms in Section 4.2, with the goal to identify suitable heat sink designs. Therein, it is observed that surrogate model accuracy plays a crucial role not only in the accuracy, but also in the robustness of the optimization results. In terms of computational cost, surrogate-based optimization yields tremendous savings compared to the case where the original heat sink model would be used within the optimization algorithms.

While already seemingly successful, a number of improvements to the DD-PCE-based surrogate modeling framework considered in this work can be pursued in follow-up works. Exemplarily, active learning techniques [73] can be employed to create more informative training datasets of reduced size, thus leading to improved surrogate model accuracy at comparatively lower computational cost. Further improvement could be accomplished by using multi-element PCE methods [85] to accommodate cases where the functional relation between design features and QoI is not sufficiently smooth for a global polynomial approximation. The use of multi-fidelity surrogate modeling [10] could also accelerate approximation convergence and further reduce computational costs. Last, it would be of interest to examine whether the proposed surrogate-based heat sink design optimization approach extends beyond power electronics cooling, e.g. for heat sinks designed for microprocessors, optoelectronic devices, or various types of integrated circuits.

Table 6: Average surrogate model evaluations for multi-objective optimization with the NSGA algorithm.

| Training dataset size $M$ | 100  | 200  | 300  | 400  | 500  | 600  | 700  | 800  |
|---------------------------|------|------|------|------|------|------|------|------|
| Average model evaluations | 13705| 14305| 14555| 15080| 14140| 14480| 13795| 14560|
Acknowledgments

Both authors are supported by the Graduate School Computational Engineering within the Centre for Computational Engineering at the Technische Universität Darmstadt. The authors would like to thank our former M.Sc. student, Robin Scheich, for generating the original dataset of design feature combinations and corresponding model evaluations.

References

[1] M. Rashid, Power Electronics Handbook: Devices, Circuits and Applications, Elsevier, 2010.

[2] R. Bornoff, B. Blackmore, J. Parry, Heat sink design optimization using the thermal bottleneck concept, 2011 27th Annual IEEE Semiconductor Thermal Measurement and Management Symposium, IEEE, 2011, pp. 76–80.

[3] E. M. Dede, S. N. Joshi, F. Zhou, Topology optimization, additive layer manufacturing, and experimental testing of an air-cooled heat sink, Journal of Mechanical Design 137 (11).

[4] G. Maranzana, I. Perry, D. Maillet, S. Raël, Design optimization of a spreader heat sink for power electronics, International Journal of Thermal Sciences 43 (1) (2004) 21–29.

[5] R. Pakrouh, M. Hosseini, A. Ranjbar, R. Bahrampoury, A numerical method for pcm-based pin fin heat sinks optimization, Energy Conversion and Management 103 (2015) 542–552.

[6] C. Shih, G. Liu, Optimal design methodology of plate-fin heat sinks for electronic cooling using entropy generation strategy, IEEE Transactions on Components and Packaging Technologies 27 (3) (2004) 551–559.

[7] M. Alrasheed, C. de Silva, M. Gadala, A modified particle swarm optimization scheme and its application in electronic heat sink design, Vol. ASME 2007 InterPACK Conference, Volume 1 of International Electronic Packaging Technical Conference and Exhibition, 2007, pp. 627–636. doi: 10.1115/IPACK2007-33256.

[8] M. Alrasheed, C. de Silva, M. Gadala, Evolutionary optimization in the design, Mechatronic Systems: Devices, Design, Control, Operation and Monitoring.

[9] I. G. Ion, Z. Bontinck, D. Loukreizis, U. Römer, O. Lass, S. Ulbrich, S. Schöps, H. De Gersem, Robust shape optimization of electric devices based on deterministic optimization methods and finite-element analysis with affine parametrization and design elements, Electrical Engineering 100 (4) (2018) 2635–2647.
[10] A. I. J. Forrester, A. Sóbester, A. J. Keane, Multi-fidelity optimization via surrogate modelling, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 463 (2088) (2007) 3251–3269.

[11] A. I. J. Forrester, A. Sobester, A. J. Keane, Engineering design via surrogate modelling: a practical guide, John Wiley & Sons, 2008.

[12] A. I. J. Forrester, A. J. Keane, Recent advances in surrogate-based optimization, Progress in Aerospace Sciences 45 (1-3) (2009) 50–79.

[13] S. Koziel, L. Leifsson, Surrogate-based modeling and optimization, Springer, 2013.

[14] Y. S. Ong, P. B. Nair, A. J. Keane, Evolutionary optimization of computationally expensive problems via surrogate modeling, AIAA Journal 41 (4) (2003) 687–696.

[15] A. Cozad, N. V. Sahinidis, D. C. Miller, Learning surrogate models for simulation-based optimization, AIChE Journal 60 (6) (2014) 2211–2227.

[16] K.-Z. Huang, H. Yang, I. King, M. R. Lyu, Machine learning: modeling data locally and globally, Springer Science & Business Media, 2008.

[17] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press, 2016.

[18] R. K. Tripathy, I. Bilionis, Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification, Journal of Computational Physics 375 (2018) 565–588.

[19] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, Journal of Computational Physics 394 (2019) 56–81.

[20] V. Vapnik, The nature of statistical learning theory, Springer Science & Business Media, 2013.

[21] R. Ghanem, P. D. Spanos, Polynomial Chaos in Stochastic Finite Elements, Journal of Applied Mechanics 57 (1) (1990) 197–202. doi:10.1115/1.2888303.

[22] N. Wiener, The homogeneous chaos, American Journal of Mathematics 60 (4) (1938) 897–936. doi:10.2307/2371268.

[23] D. Xiu, G. E. Karniadakis, The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations, SIAM Journal on Scientific Computing 24 (2) (2002) 619–644.

[24] T. Crestaux, O. Le Mautre, J.-M. Martinez, Polynomial chaos expansion for sensitivity analysis, Reliability Engineering & System Safety 94 (7) (2009) 1161–1172.
[25] B. Sudret, Global sensitivity analysis using polynomial chaos expansions, Reliability Engineering & System Safety 93 (7) (2008) 964–979.

[26] E. Torre, S. Marelli, P. Embrechts, B. Sudret, Data-driven polynomial chaos expansion for machine learning regression, Journal of Computational Physics 388 (2019) 601–623.

[27] S. Abraham, M. Raisee, G. Ghorbaniasl, F. Contino, C. Lacor, A robust and efficient stepwise regression method for building sparse polynomial chaos expansions, Journal of Computational Physics 332 (2017) 461–474.

[28] G. Blatman, B. Sudret, An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis, Probabilistic Engineering Mechanics 25 (2) (2010) 183–197. doi:10.1016/j.probengmech.2009.10.003

URL http://www.sciencedirect.com/science/article/pii/S0266892009000666

[29] G. Blatman, B. Sudret, Adaptive Sparse Polynomial Chaos Expansion Based on Least Angle Regression, Journal of Computational Physics 230 (6) (2011) 2345–2367.

[30] K. Cheng, Z. Lu, Sparse polynomial chaos expansion based on d-morph regression, Applied Mathematics and Computation 323 (2018) 17–30.

[31] P. Diaz, A. Doostan, J. Hampton, Sparse polynomial chaos expansions via compressed sensing and d-optimal design, Computer Methods in Applied Mechanics and Engineering 336 (2018) 640–666.

[32] A. Doostan, H. Owhadi, A non-adapted sparse approximation of pdes with stochastic inputs, Journal of Computational Physics 230 (8) (2011) 3015–3034.

[33] J. Hampton, A. Doostan, Basis adaptive sample efficient polynomial chaos (base-pc), Journal of Computational Physics 371 (2018) 20–49.

[34] W. He, Y. Zeng, G. Li, An adaptive polynomial chaos expansion for high-dimensional reliability analysis, Structural and Multidisciplinary Optimization 62 (4) (2020) 2051–2067.

[35] C. Hu, B. D. Youn, Adaptive-sparse polynomial chaos expansion for reliability analysis and design of complex engineering systems, Structural and Multidisciplinary Optimization 43 (3) (2011) 419–442.

[36] X. Huan, C. Safta, K. Sargsyan, Z. P. Vane, G. Lacaze, J. C. Oefelein, H. N. Najm, Compressive sensing with cross-validation and stop-sampling for sparse polynomial chaos expansions, SIAM/ASA Journal on Uncertainty Quantification 6 (2) (2018) 907–936.

[37] D. Loukrezis, A. Galetzka, H. De Gersem, Robust adaptive least squares polynomial chaos expansions in high-frequency applications, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields n/a (n/a) (2020) e2725. doi:10.1002/jnm.2725

21
[38] G. Migliorati, Adaptive Polynomial Approximation by Means of Random Discrete Least Squares in: A. Abdulle, S. Deparis, D. Kressner, F. Nobile, M. Picasso (Eds.), ENUMATH, Vol. 103 of Lecture Notes in Computational Science and Engineering, Springer, 2013, pp. 547–554. URL http://dblp.uni-trier.de/db/conf/enumath/enumath2013.htmlMigliorati13

[39] J. Peng, J. Hampton, A. Doostan, A weighted l1-minimization approach for sparse polynomial chaos expansions, Journal of Computational Physics 267 (2014) 92–111.

[40] P. Tsilifis, X. Huan, C. Safta, K. Sargsyan, G. Lacaze, J. C. Oefelein, H. N. Najm, R. G. Ghanem, Compressive sensing adaptation for polynomial chaos expansions, Journal of Computational Physics 380 (2019) 29–47.

[41] H. Zhao, Z. Gao, F. Xu, Y. Zhang, J. Huang, An efficient adaptive forward–backward selection method for sparse polynomial chaos expansion, Computer Methods in Applied Mechanics and Engineering 355 (2019) 456–491.

[42] Y. Gerstenmaier, W. Kiffe, G. Wachutka, Combination of thermal subsystems by use of rapid circuit transformation and extended two-port theory, Microelectronics Journal 40 (1) (2009) 26–34.

[43] K. Ma, Electro-thermal model of power semiconductors dedicated for both case and junction temperature estimation, in: Power electronics for the next generation wind turbine system, Springer, 2015, pp. 139–143.

[44] R. Wu, H. Wang, K. B. Pedersen, K. Ma, P. Ghimire, F. Iannuzzo, F. Blaabjerg, A temperature-dependent thermal model of igbt modules suitable for circuit-level simulations, IEEE Transactions on Industry Applications 52 (4) (2016) 3306–3314.

[45] G. Alfonsi, Reynolds-averaged navier-stokes equations for turbulence modeling, Applied Mechanics Reviews 62 (4).

[46] H. Chen, V. Patel, S. Ju, Solutions of reynolds-averaged navier-stokes equations for three-dimensional incompressible flows, Journal of Computational Physics 88 (2) (1990) 305–336.

[47] R. Teman, Navier-Stokes equations: theory and numerical analysis, Vol. 343, American Mathematical Society, 2001.

[48] D. D. Gray, A. Giorgini, The validity of the boussinesq approximation for liquids and gases, International Journal of Heat and Mass Transfer 19 (5) (1976) 545–551.

[49] F. R. Menter, Influence of freestream values on k-omega turbulence model predictions, AIAA Journal 30 (6) (1992) 1657–1659.

[50] J. Branke, J. Branke, K. Deb, K. Miettinen, R. Słowiński, Multiobjective optimization: Interactive and evolutionary approaches, Vol. 5252, Springer Science & Business Media, 2008.
[51] N. Gunantara, A review of multi-objective optimization: Methods and its applications, Cogent Engineering 5 (1) (2018) 1502242.

[52] J. Schneider, S. Kirkpatrick, Stochastic optimization, Springer Science & Business Media, 2007.

[53] J. Feinberg, V. Eck, H. Langtangen, Multivariate polynomial chaos expansions with dependent variables, SIAM Journal on Scientific Computing 40 (1) (2018) A199–A223. doi:10.1137/15M1020447

[54] R. Ahlfeld, B. Belkouchi, F. Montomoli, Samba: Sparse approximation of moment-based arbitrary polynomial chaos, Journal of Computational Physics 320 (2016) 1–16.

[55] P. R. Conrad, Y. M. Marzouk, Adaptive smolyak pseudospectral approximations, SIAM Journal on Scientific Computing 35 (6) (2013) A2643–A2670.

[56] P. G. Constantine, M. S. Eldred, E. T. Phipps, Sparse pseudospectral approximation method, Computer Methods in Applied Mechanics and Engineering 229 (2012) 1–12.

[57] J. Winokur, D. Kim, F. Bisetti, O. P. Le Maître, O. M. Knio, Sparse pseudo spectral projection methods with directional adaptation for uncertainty quantification, Journal of Scientific Computing 68 (2) (2016) 596–623.

[58] B. Ratner, Statistical and Machine-Learning Data Mining: Techniques for Better Predictive Modeling and Analysis of Big Data, CRC Press, 2017.

[59] N. J. Higham, Accuracy and Stability of Numerical Algorithms, 2nd Edition, Society for Industrial and Applied Mathematics, 2002. doi:10.1137/1.9780898718027

[60] D. Loukrezis, Adaptive approximations for high-dimensional uncertainty quantification in stochastic parametric electromagnetic field simulations, Ph.D. thesis, Technische Universität Darmstadt (2019). URL http://tuprints.ulb.tu-darmstadt.de/8485/

[61] M. Hadigol, A. Doostan, Least squares polynomial chaos expansion: A review of sampling strategies, Computer Methods in Applied Mechanics and Engineering 332 (2018) 382–407.

[62] J. Hampton, A. Doostan, Coherence motivated sampling and convergence analysis of least squares polynomial chaos regression, Computer Methods in Applied Mechanics and Engineering 290 (2015) 73–97.

[63] N. Fajraoui, S. Marelli, B. Sudret, Sequential design of experiment for sparse polynomial chaos expansions, SIAM/ASA Journal on Uncertainty Quantification 5 (1) (2017) 1061–1085.

[64] S. Zein, B. Colson, F. Glineur, An efficient sampling method for regression-based polynomial chaos expansion, Communications in computational physics 13 (4) (2013) 1173–1188.
[65] G. Migliorati, F. Nobile, E. Von Schwerin, R. Tempone, Analysis of discrete l2 projection on polynomial spaces with random evaluations, Foundations of Computational Mathematics 14 (3) (2014) 419–456.

[66] J. Hampton, A. Doostan, Compressive sampling of polynomial chaos expansions: Convergence analysis and sampling strategies, Journal of Computational Physics 280 (2015) 363–386.

[67] R. Tibshirani, Regression shrinkage and selection via the LASSO, Journal of the Royal Statistical Society: Series B (Methodological) 58 (1) (1996) 267–288.

[68] D. L. Donoho, Compressed sensing, IEEE Transactions on Information Theory 52 (4) (2006) 1289–1306.

[69] S. Becker, J. Bobin, E. J. Candès, Nesta: A fast and accurate first-order method for sparse recovery, SIAM Journal on Imaging Sciences 4 (1) (2011) 1–39.

[70] E. J. Candes, J. K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences 59 (8) (2006) 1207–1223.

[71] E. Van Den Berg, M. P. Friedlander, Probing the pareto frontier for basis pursuit solutions, SIAM Journal on Scientific Computing 31 (2) (2009) 890–912.

[72] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, The Annals of statistics 32 (2) (2004) 407–499.

[73] S. Marelli, B. Sudret, An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis, Structural Safety 75 (2018) 67–74.

[74] T. Goel, R. T. Haftka, W. Shyy, N. V. Queipo, Ensemble of surrogates, Structural and Multidisciplinary Optimization 33 (3) (2007) 199–216.

[75] C. Ren, Y. Aoues, D. Lemosse, E. S. De Cursi, Ensemble of surrogates combining kriging and artificial neural networks for reliability analysis with local goodness measurement, Structural Safety 96 (2022) 102186.

[76] C. K. Williams, C. E. Rasmussen, Gaussian processes for machine learning, Vol. 2, MIT press Cambridge, MA, 2006.

[77] D. Graupe, Principles of artificial neural networks, Vol. 7, World Scientific, 2013.

[78] J. Kennedy, R. Eberhart, Particle swarm optimization, Vol. 4 of Proceedings of ICNN’95-International Conference on Neural Networks, IEEE, 1995, pp. 1942–1948.

[79] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, Swarm intelligence 1 (1) (2007) 33–57.
[80] K. Price, R. M. Storn, J. A. Lampinen, Differential evolution: a practical approach to global optimization, Springer Science & Business Media, 2006.

[81] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.

[82] K. Deb, J. Sundar, Reference point based multi-objective optimization using evolutionary algorithms, Proceedings of the 8th annual conference on Genetic and evolutionary computation, 2006, pp. 635–642.

[83] J. Blank, K. Deb, pymoo: Multi-objective optimization in python, IEEE Access 8 (2020) 89497–89509.

[84] P. N. Koch, R.-J. Yang, L. Gu, Design for six sigma through robust optimization, Structural and Multidisciplinary Optimization 26 (3) (2004) 235–248.

[85] B. Chouvion, E. Sarrouy, Development of error criteria for adaptive multi-element polynomial chaos approaches, Mechanical Systems and Signal Processing 66 (2016) 201–222.