MODEL ORDER REDUCTION TECHNIQUES WITH A POSTERIORI ERROR CONTROL FOR NONLINEAR ROBUST OPTIMIZATION GOVERNED BY PARTIAL DIFFERENTIAL EQUATIONS[*]

OLIVER LASS[†] AND STEFAN ULBRICH[†]

Abstract. We consider a nonlinear optimization problem governed by partial differential equations (PDE) with uncertain parameters. It is addressed by a robust worst case formulation. The resulting optimization problem is of bi-level structure and is difficult to treat numerically. We propose an approximate robust formulation that employs linear and quadratic approximations. To speed up the computation, reduced order models based on proper orthogonal decomposition (POD) in combination with a posteriori error estimators are developed. The proposed strategy is then applied to the optimal placement of a permanent magnet in the rotor of a synchronous machine with moving rotor. Numerical results are presented to validate the presented approach.

Key words. model order reduction, PDE constrained optimization, robust optimization, proper orthogonal decomposition, parametrized optimal control problem

AMS subject classifications. 35Q93, 49J20, 49K20, 90C31

1. Introduction.

Optimization with partial differential equations (PDE) has become an important research field in applied mathematics. It is driven by the engineering disciplines and the desire to optimize physical phenomena without performing real physical experiments. In most cases, these real world applications are subject to uncertainties which enter the optimization through parameters, coefficients, or boundary conditions of the PDE model. These quantities are usually measurements or estimations based on expert opinions and hence carry some amount of error. The effect of these uncertain parameters on optimization has long been a focus of the mathematical programming community. In particular it was observed that even very small perturbations in the parameters can lead to large changes in the optimal solution [6] and exhibit remarkable sensitivities. Thus, slightly modified optimal solutions can become infeasible, seriously suboptimal, or both. Hence, a nominal solution of an optimization problem with uncertain parameters might be absolutely irrelevant from a practical point of view whenever uncertainty comes into play. Consequently, it is a desirable goal to generate solutions that are insensitive against the influence of uncertainty.

There are two main approaches to incorporate uncertainty in the framework of mathematical optimization. Stochastic optimization and robust optimization. The fundamental assumption in stochastic optimization is that the uncertainty can be described probabilistically. Then the uncertainty can be incorporated into the problem by means of probabilistic measures like the standard deviation or some risk measure [12, 56]. There are some works investigating these methods theoretically and numerically in the context of PDE constrained optimization [37, 46, 61]. While [61] uses stochastic collocation, the recent paper [37] considers risk-averse optimization based on the conditional value-at-risk in combination with Monte Carlo sampling. Our focus is on the robust optimization. In this case no probabilistic model of the uncertainty is required. Instead, a deterministic approach is applied by assuming

[*] This work is supported by the German BMBF in the context of the SIMUROM project (grant number 05M2013) and the DFG within the SFB 805.

[†] Technische Universität Darmstadt, Fachbereich Mathematik, Dolivostraße 15, D-64295 Darmstadt, Germany (lass@mathematik.tu-datmstadt.de, ulbrich@mathematik.tu-darmstadt.de).

that the uncertainty is restricted to a bounded uncertainty set. Using the notion of a robust counterpart the associated original uncertain optimization problem is reformulated. The solution obtained in this way stays feasible for all realizations from the uncertainty set and at the same time yields the best worst case value of the objective function. The resulting problem is of bi-level structure and challenging to solve. There has been a lot of work on this topic and for an overview we refer to [8, 9, 10, 34]. The research focus in robust optimization however remains mainly within the framework of convex optimization, where a lot of progress has been made. There only exist isolated works for the case of more general nonlinear programs. The common idea is to replace the robust counterpart by a tractable approximation using linearization [20, 65]. First steps in the direction of higher order approximation with the focus on PDE constrained optimization have been done in [57].

In this work we will develop a new approximation for the robust counterpart by utilizing a quadratic model. To account for approximation deficiencies an adaptive strategy is proposed to enhance the approximation quality. To minimize computational costs we combine quadratic and linear approximations whenever suited. This gives a numerically feasible problem. The proposed strategy is novel and numerical results are very promising. It provides a huge improvement over strategies utilizing only linear approximations [20, 65]. The advantage of the proposed strategy is that it is not limited to convex problems, which is a limitation of many other approaches [8, 9, 10, 34]. It is a fact that not all optimization problems of practical interest can be modeled appropriately as convex programs. This new approximated robust optimization problem has the structure of an MPCC. We will utilize results from [27, 35, 41] and develop an algorithm based on the sequential quadratic programming (SQP) method.

When applying the robust optimization framework to optimization problems with PDE constraints we face huge computational costs. Large linear systems arising from the discretization of the PDE have to be solved numerically. We further need to compute derivatives with respect to the uncertain parameter and optimization variables which significantly increase the computational cost since further solutions to PDEs are required. Finally, since we are looking at a nonlinear optimization problem which is solved iteratively, these computations have to be repeated for different configurations. Hence, it is desirable to employ methods that can speed up this process. In recent years, the development of reduced order models (ROM) has been a successful field of research. In the context of optimization, remarkable performance improvements were achieved [2, 3, 23, 24, 29, 39, 44]. We will develop an adaptive method that combines the idea of a greedy algorithm from the reduced basis method [19, 48] and the method of proper orthogonal decomposition (POD) [16, 29].

The proposed methods are applied to a shape optimization problem, where the geometry of a permanent magnet synchronous machine is optimized. The goal is to optimize the volume and position of the permanent magnet while maintaining a given performance level.

The paper is organized in the following way. In Section 2 we introduce the model problem under investigation. We then introduce in Section 3 the robust optimization framework together with the approximation strategies. In the model problem, we will consider uncertainty in material parameters as well as in the optimization variables. Also the numerical methods for solving the arising problems are outlined. Section 4 is devoted to model order reduction. Numerical results for the problem introduced in Section 2 are presented in Section 5. Finally a conclusion is drawn in Section 6.

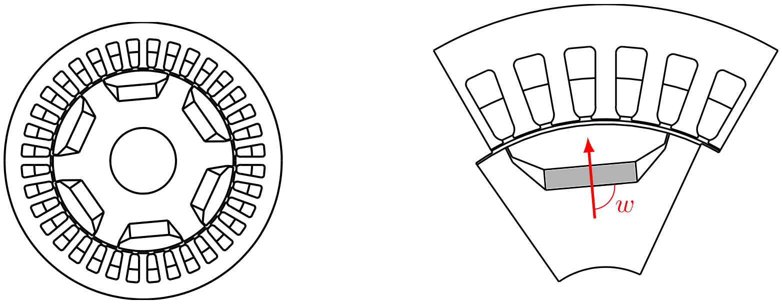

FIG. 1. *Geometry of the design problem presented by a 2D planar cross section. Full engine to the left and detailed view on one pole on the right. The parts of the engine are the stator (outer ring) and the rotor holding the permanent magnets (inner part). In the detailed view the permanent magnet is indicated in gray and the magnetic field angle w in red.*

2. The model problem. We consider a three-phase 6-pole permanent magnet synchronous machine (PMSM). The geometry is shown in Figure 1. The windings in the stator are double layered with two slots per pole per phase resulting in 36 slots for the whole machine. Each pole contains one buried permanent magnet (gray). The machine is operated at 50Hz. Using the relation between speed and frequency, $S = 120F/N$, where F is the frequency in Hz (cycles per second), N is the number of poles and S the rotational speed in revolutions per minute (RPM), we get an operation of 1000 RPM. The goal is to optimize the geometry of the PMSM.

PMSMs are described with sufficient accuracy by the magnetostatic approximation of Maxwell's equations, i.e., displacement and eddy currents are disregarded with respect to the source current. One retrieves the elliptic partial differential equation (PDE) in terms of the magnetic vector potential A

$$\nabla \times (\nu \nabla \times A(\vartheta)) = J_{src} - \nabla \times H_{pm} \quad \text{in} \quad \Omega \tag{1}$$

with homogeneous Dirichlet boundary condition $A = 0$ on the outer surface, where $\nu(\mathbf{x})$ is the magnetic permeability, J_{src} is the source current density and H_{pm} is the field of the permanent magnet. The rotor and the stator of a PMSM are made out of iron. In this work we consider a linear magnetic permeability, which for the considered machine is adequate. Further we account for the rotation which is an important feature when studying rotating electrical machines, indicated by ϑ. The machine is divided into two parts, the static part (stator) and the moving part (rotor). The position of the moving part is given by the angle ϑ with respect to the static part.

In our setting the 2D planar symmetry is taken into account with the symmetry plane xy. By $u(\vartheta)$ we denote the only nonzero component of the vector potential $A(\vartheta) = (0, 0, u(\vartheta))^\top$ which corresponds to the z direction. Utilizing this formulation, the magnetic induction is expressed by

$$B(\vartheta) = \nabla \times A(\vartheta) = \left(\frac{\partial u(\vartheta)}{\partial y}, -\frac{\partial u(\vartheta)}{\partial x}, 0 \right)^\top .$$

Hence the model under consideration in this work is given by the linear elliptic equation

$$-\nabla\cdot(\nu\nabla u(\vartheta)) = f(\mathbf{x}) \quad \text{in} \quad \Omega \tag{2a}$$

together with Dirichlet boundaries on the outer surface

$$[u(\vartheta)]\Big|_{\Gamma_D} = 0. \tag{2b}$$

The gray area in the rotor is the magnet and will be subject to our optimization problem (Figure 1). The size of the permanent magnet can be described by the width and height. Additionally we optimize the location of the permanent magnet in the rotor. We introduce the parameter $\mu \in \mathbb{R}^3$, where μ_1 corresponds to the width, μ_2 to the height and μ_3 to describe the central perpendicular distance between the permanent magnet and the surface of the rotor.

Let $\Omega(\mu)$ be the parameter dependent domain. We formulate (2) in the weak form as

$$a(u(\vartheta), v; \mu) = f(v;\mu), \quad \forall v \in H_0^1(\Omega(\mu))$$

with

$$a(u(\vartheta), v; \mu) = \int_{\Omega(\mu)} \nu\nabla u(\vartheta)\cdot\nabla v\,\mathrm{d}\mathbf{x} \quad \text{and} \quad f(v;\mu) = \int_{\Omega(\mu)} \mathcal{F} v\,\mathrm{d}\mathbf{x}.$$

The goal is to perform the computation on a reference domain $\hat{\Omega}$. For this we decompose the region around the permanent magnet into L non-overlapping triangular domains $\hat{\Omega}_k$. In each domain we consider an affine mapping of the form

$$\begin{array}{rccc} T_k(\hat{\mathbf{x}},\mu): & \hat{\Omega}_k & \rightarrow & \Omega(\mu) \\ & \hat{\mathbf{x}} & \mapsto & C_k(\mu)\hat{\mathbf{x}} + d_k(\mu) \end{array}$$

for $k = 1, \ldots, L$, where $C_k(\mu) \in \mathbb{R}^{2\times 2}$ and $d_k(\mu) \in \mathbb{R}^2$. Integrating by substitution we obtain

$$\begin{aligned} a(u,v;\mu) &= \int_{\Omega(\mu)} \nu\nabla u(\vartheta)\cdot\nabla v \mathrm{d}\mathbf{x} \\ &= \sum_{k=1}^{L}\sum_{i,j=1}^{2} \underbrace{\left[C_k(\mu)^{-1}\nu_k C_k(\mu)^{-\top}\right]_{ij} |\det C_k(\mu)|}_{\Theta_k^{ij}(\mu)} \underbrace{\int_{\hat{\Omega}_k} \frac{\partial \hat{u}(\vartheta)}{\partial \hat{\mathbf{x}}_i}\frac{\partial \hat{v}}{\partial \hat{\mathbf{x}}_j}\mathrm{d}\hat{\mathbf{x}}}_{\hat{a}_k^{ij}(\hat{u}(\vartheta),\hat{v})} \\ &= \sum_{k=1}^{L}\sum_{i,j=1}^{2} \Theta_k^{ij}(\mu) a_k^{ij}(\hat{u}(\vartheta),\hat{v}). \end{aligned}$$

The same can be done for the right hand side resulting in

$$f(v;\mu) = \sum_{k=1}^{L} \underbrace{|\det C_k(\mu)|}_{\Theta_k^f(\mu)} \int_{\hat{\Omega}_k} f\hat{v}\mathrm{d}\hat{\mathbf{x}} = \sum_{k=1}^{L} \Theta_k^f(\mu)\hat{f}_k(\hat{v}).$$

This parametrized geometry description allows us to perform the computations on a reference domain. In the numerical realization the mesh does not have to be deformed and the components Θ_k^{ij}, $\hat{a}^{ij}$ and Θ_k^f, $\hat{f}_k$ can be precomputed. Further, the transformation of the PDE into a parametrized PDE eases the treatment in the optimization process since the transformation can be computed analytically and derivatives with respect to μ can be computed exactly. For more details on the presented transformation we refer the reader to [52], where this approach is utilized to generate an affine decomposition for the efficient realization of model order reduction. Let us note at this place that the specific choice of the described parametrization does not influence the following results. In our specific application this choice seemed the most reasonable. There are other possibilities to parametrize the geometry. The advantage of the chosen methodology is that the parameters have a physical meaning opposed to some other popular choices like, e.g., the Free-form deformation [54]. An overview of other popular choices is given in [53] and with the focus on optimization in [30, 40].

After introducing the model and the parametrization of the geometry let us proceed to the formulation of the optimization problem. For this we introduce a cost function. The goal is to minimize the volume of the permanent magnet while maintaining a prescribed performance. The permanent magnets contain rear-earth and hence contribute significantly to the financial and ecological cost [11]. To obtain the performance of the machine we have to solve the magnetostatic approximation of Maxwell's equation (2) for one full rotation to account for eventual non-symmetries. Then the performance can be extracted by the power balance method by means of the torque $\tau(u, \mu)$. For this we compute the voltage of the stranded conductors

$$u_{\mathrm{str}}(\vartheta) = R_{\mathrm{str}} i_{str}(\vartheta) + \frac{\mathrm{d}}{\mathrm{d}\vartheta}(X_{str}^{\top} u(\vartheta)),$$

where R_{str} is the resistance, i_{str} the current and X_{str} the winding function of the stranded conductor. The total electrical energy and losses are given by

$$P_{\mathrm{elec}} = \frac{1}{2\pi}\int_0^{2\pi} u_{\mathrm{str}}^{\top}(\vartheta) i_{str}(\vartheta)\mathrm{d}\vartheta \quad \text{and} \quad P_{\mathrm{loss}} = \frac{1}{2\pi}\int_0^{2\pi} R_{\mathrm{str}} i_{\mathrm{str}}^{\top}(\vartheta) i_{str}(\vartheta)\mathrm{d}\vartheta.$$

The torque is then given by $\tau(u, \mu) = N(P_{elec} - P_{\mathrm{loss}})/(4\pi F)$. For more details we refer the reader to [13, 58, 59].

Having stated the quantities of interest, we define the cost function to be minimized by

$$\hat{g}_0(\mu, u) := \mu_1 \mu_2 \tag{3a}$$

governed by the constraint

$$\hat{g}_1(\mu, u) := \tau^d - \tau(u, \mu) \leq 0 \tag{3b}$$

with $\hat{g}_i : \mathbb{R}^3 \times H_0^1(\Omega) \to \mathbb{R}$, $i = 1, 2$. The cost function represents the volume of the permanent magnet which in the 2D case corresponds to the area while the constraint ensures that a prescribed torque τ^d is met. Recall that u is the solution to the magnetostatic Maxwell equation, hence we have a PDE constrained optimization problem. Cost functions of this type are of interest in application [47] although further quantities of interest (e.g., torque ripple) could be added.

3. Robust optimization framework. In this section we introduce a general framework for solving a robust nonlinear optimization problem. For this we will first review the nominal optimization problem. To take uncertainties into account we then formulate the robust counterpart and develop approximation techniques.

3.1. Nominal optimization problem. Let V and Z be a Banach spaces, we consider the nominal optimization problem of the form

$$\min_{\mu\in\mathbb{R}^{n_\mu},u\in V} \hat{g}_0(\mu,u,w) \quad \text{subject to (s.t.)} \begin{cases} \hat{g}_i(\mu,u,w)\leq 0, & i=1,\dots,n_g,\\ e(\mu,u,w)=0, \end{cases} \tag{4}$$

where $\hat{g}_i : \mathbb{R}^{n_\mu}\times V\times\mathbb{R}^{n_w}\to\mathbb{R}$ is the objective function for $i=0$ and inequality constraints for $i=1,\dots,n_g$ and $e:\mathbb{R}^{n_\mu}\times V\times\mathbb{R}^{n_w}\to Z$ an equality constraint given by a parametrized PDE. Further, let $\hat{g}_i$ and e be continuously differentiable and let the Jacobian $\frac{\partial}{\partial u}e(\mu,u,w)\in\mathcal{L}(V,Z)$ be continuously invertible. Note that we introduced an additional parameter $w\in\mathbb{R}^{n_w}$ which is not part of the optimization problem. In the nominal optimization this parameter is fixed and known. Later we will assume uncertainty in this parameter which leads us to the robust optimization. The optimization problem introduced in Section 2 fits exactly in this setting, where the equality constraint is given by (2) and the objective by (3). Additionally we will later introduce some inequality constraints accounting for design constraints.

Since the equality constraint is given by a PDE, we are in the PDE constrained optimization [32, 63] context. Let $e(\mu,u,w)=0$ have a unique solution $u=u(\mu,w)$ for every admissible w and μ. Then we can introduce the reduced objective and inequality constraint functions

$$g_i(\mu,w):=\hat{g}_i(\mu,u,w), \quad \text{for} \quad i=0,\dots,n_g.$$

Consequently, the reduced optimization problem associated with (4) reads

$$\min_{\mu\in\mathbb{R}^{n_\mu}} g_0(\mu,w) \quad \text{s.t.} \quad g_i(\mu,w)\leq 0, \quad i=1,\dots,n_g. \tag{5}$$

Since e is continuously differentiable with invertible Jacobian $\frac{\partial}{\partial u}e(\mu,u,w)$, the implicit function theorem guarantees that also u is continuously differentiable. Hence the reduced objective and reduced constraints $g_i(\mu,w)$, $i=0,\dots,n_g$, are continuously differentiable.

3.2. Robust optimization problem. Now let us introduce the robust optimization problem. Here the previously introduced parameter w will come into play. It is assumed that this parameter is not known exactly and is subject to uncertainty. In our case the parameter w describes model parameters and primarily enters the problem formulation through the equality constraint. The goal is to formulate the robust version of (5). For this we assume some prior knowledge about the uncertain parameter. We assume that the uncertain parameter remains in a predefined neighbourhood of some nominal parameter $\hat{w}\in\mathbb{R}^{n_w}$. We define the *uncertainty set*

$$\begin{aligned} U_{w,k} &= \{\, w\in\mathbb{R}^{n_w} \mid \|D_w^{-1}(w-\hat{w})\|_k\leq 1\,\}\\ &= \{\, w\in\mathbb{R}^{n_w} \mid w=\hat{w}+\delta_w,\ \|D_w^{-1}\delta_w\|_k\leq 1\,\}, \end{aligned}$$

where $D_w\in\mathbb{R}^{n_w\times n_w}$ is an invertible matrix and $1\leq k\leq\infty$. In this work we only consider the cases $k\in\{\,2,\infty\,\}$ which are the two most prominent cases in application.

Remark 1. Choosing $D_w=\operatorname{diag}((\overline{w}-\underline{w})/2)$, $\hat{w}=(\underline{w}+\overline{w})/2$ and $k=\infty$ we get as uncertainty set the box constraints

$$U_{w,\infty}=\{\, w\in\mathbb{R}^{n_w} \mid \underline{w}\leq w\leq\overline{w}\,\}.$$

Having introduced the uncertainty set we can now formulate the robust counterpart. The goal is to formulate an optimization problem that is robust with respect to the uncertain parameter w. For this we choose the worst case formulation [8, 20, 65]. We define the *worst case function* as

$$\varphi_i(\mu) = \max_{w \in U_{w,k}} g_i(\mu, w) \quad \text{for} \quad i = 0, \dots, n_g.$$

The function $\varphi_i : U_{w,k} \mapsto \mathbb{R}$ gives for every fixed parameter μ the worst case value of the function g_i under the condition that w lies in the uncertainty set. Utilizing the worst case function we arrive at the *robust counterpart*, or worst case formulation, for our optimization problem (5):

$$\min_{\mu \in \mathbb{R}^{n_\mu}} \varphi_0(\mu) \quad \text{s.t.} \quad \varphi_i(\mu) \le 0 \quad \text{for} \quad i = 1, \dots, n_g. \tag{6}$$

A solution μ that satisfies (6) is referred to as *robust optimal solution*. This optimal solution is robust against uncertainty since it is feasible for (5) no matter what the value of $w \in U_{w,k}$ is and optimal with respect to the objective function g_0.

These types of problems are difficult to solve due to its bi-level structure. This has been investigated in [4] for a variety of problems. In [7] for general nonlinear problems it was proposed to replace the robust counterpart (6) by an approximation

$$\min_{\mu \in \mathbb{R}^{n_\mu}} \hat{\varphi}_0(\mu) \quad \text{s.t.} \quad \hat{\varphi}_i(\mu) \le 0 \quad \text{for} \quad i = 1, \dots, n_g, \tag{7}$$

where the *approximated worst case function* $\hat{\varphi}_i$ can be computed more efficiently compared to the original worst case function φ_i. The new problem is then referred to as the *approximated robust counterpart* of (5). Alternatively, in [34] it was investigated to solve the robust counterpart by sequential convex bi-level programming. The drawback of that approach is that there are strong assumptions made on g_i which do not allow the application to general problems.

In the following we investigate first and second order approximations of φ_i. While the first order approximation has already been considered for the general nonlinear case [20, 65], the quadratic approximation is new. The second order approximation utilized in this work is a modification and generalization of the approach presented in [57]. In this paper we will consider the case that the constraints and objective function are twice differentiable. Otherwise, instead of first or second order Taylor expansions one could use our approach in connection with quadratic models generated from sampled data by similar techniques as in derivative free optimization, see for example [49].

3.2.1. Linear approximation of the robust counterpart. Let us start by recalling the approach of [20, 65] and rewrite it using the introduced notation. The idea is to replace g_i in the worst case function by a linearization, i.e., a first-order Taylor expansion at the nominal parameter $\hat{w}$ is performed. For this we utilize the linear approximation

$$g_i^l(\mu, \hat{w}, \delta_w^i) = g_i(\mu, \hat{w}) + \nabla_w g_i(\mu, \hat{w})^\top \delta_w^i \quad \text{for} \quad i = 0, \dots, n_g,$$

where $\nabla_w g_i(\mu, w) \in \mathbb{R}^{n_w}$ is the gradient of g_i with respect to w. Then the approximated worst case function is given as

$$\hat{\varphi}_i^l(\mu) = \max_{\delta_w^i \in \mathbb{R}^{n_w}} g_i^l(\mu, \hat{w}, \delta_w^i) \quad \text{s.t.} \quad \|D_w^{-1} \delta_w^i\|_k \le 1 \quad \text{for} \quad i = 0, \dots, n_g.$$

This convex optimization problem can now be solved analytically and the solution is given by

$$\varphi_i^l(\mu) = g_i(\mu,\hat{w}) + \|D_w^\top \nabla_w g_i(\mu,\hat{w})\|_{k^*} \quad \text{for} \quad i = 0,\dots,n_g,$$

where $\|D_w^\top \cdot \|_{k^*}$ is the dual norm of $\|D_w^{-1} \cdot \|_k$, with $k^* = k/(k-1)$ and we define $k^* = \infty$ for $k = 1$ and $k^* = 1$ for $k = \infty$. This result follows directly from inserting $g_i^l(\mu, \hat{w}+\delta_w^i)$ and the definition of the dual norm. Note that this is a standard result for Hölder norms. We can now write the *linear approximation of the robust counterpart* as

$$\begin{aligned} &\min_{\mu\in\mathbb{R}^{n_\mu}} g_0(\mu,\hat{w}) + \|D_w^\top \nabla_w g_0(\mu,\hat{w})\|_{k^*} \\ &\text{s.t.} \quad g_i(\mu,\hat{w}) + \|D_w^\top \nabla_w g_i(\mu,\hat{w})\|_{k^*} \le 0, \quad i = 1,\dots,n_g. \end{aligned} \tag{8}$$

Note that the objective function and the inequality constraints are non-differentiable if the term inside the norm becomes zero. For the case $k = \infty$ this can be circumvented by introducing slack variables, i.e.,

$$\begin{aligned} \min_{\mu\in\mathbb{R}^{n_\mu},\zeta_0,\dots,\zeta_{n_g}\in\mathbb{R}^{n_w}} \; & g_0(\mu,\hat{w}) + \mathbb{E}^\top\zeta_0 \\ \text{s.t.} \quad & g_i(\mu,\hat{w}) + \mathbb{E}^\top\zeta_i \le 0, && i = 1,\dots,n_g, \\ & -\zeta_i \le D_w^\top \nabla_w g_i(\mu,\hat{w}) \le \zeta_i, && i = 0,\dots,n_g. \end{aligned}$$

with $\mathbb{E} = (1,\dots,1)^\top \in \mathbb{R}^{n_w}$. Similarly, one can reformulate the problem in a smooth way using slack variables in the case $k = 1$. For $1 < k < \infty$ there is no such reformulation [20, 57]. Note that (8) is a nonlinear problem with nonlinear second-order-cone constraints, which can be solved efficiently by interior-point methods [1] or semismooth methods [36].

The computation of the derivatives can be carried out using two different approaches. By the sensitivity approach we get

$$\nabla_w g_i(\mu,w)^\top := \frac{\partial}{\partial u}\hat{g}_i(\mu,u,w)s + \frac{\partial}{\partial w}\hat{g}_i(\mu,u,w), \tag{9a}$$

where $s \in L(\mathbb{R}^{n_w}, Y)$ solves the first order sensitivity equation

$$\frac{\partial}{\partial u}e(\mu,u,w)s + \frac{\partial}{\partial w}e(\mu,u,w) = 0. \tag{9b}$$

When using the adjoint approach we get

$$\nabla_w g_i(\mu,w)^\top := \left(\frac{\partial}{\partial w}e(\mu,u,w)\right)^* p + \frac{\partial}{\partial w}\hat{g}_i(\mu,u,w), \tag{10a}$$

where $p \in Z^*$ solves the adjoint equation

$$\left(\frac{\partial}{\partial u}e(\mu,u,w)\right)^* p + \frac{\partial}{\partial u}\hat{g}_i(\mu,u,w) = 0. \tag{10b}$$

Here $\big(\frac{\partial}{\partial w}e(\mu,u,w)\big)^* \in L(Z^*, (\mathbb{R}^{n_w})^*)$, and $\big(\frac{\partial}{\partial u}e(\mu,u,w)\big)^* \in L(Z^*, Y^*)$ denote the adjoint operators. For a detailed discussion about the different approaches we refer the reader to [20, 32]. In the numerical experiments we will utilize the sensitivity approach, since n_w is assumed to be small. It can be seen that when n_w is large the adjoint approach is preferable.

This approximation suffers from poor accuracy in the case of nonlinear problems since the linear approximation is used and $\hat{w}$ is chosen fixed at the beginning. To increase the approximation quality we have a look at a higher approximation order for the robust counterpart.

3.2.2. Quadratic approximation of the robust counterpart. While the first order approach is numerically very efficient it can suffer from bad approximation and hence might describe the influence of the uncertain parameter w inaccurately. This was already observed in [21]. Hence we seek a higher order approximation. A similar approach was already investigated in [57], where the objective function and inequality constraints are approximated by quadratic Taylor expansions while the PDE constraint is linearized. In the presented work we eliminate the PDE constraint in (4) by introducing the reduced problem (5) which is the basis for further investigation. Hence, by applying the second order approximation to (5) we implicitly include curvature information of the PDE constraint. Let us next outline the construction of the robust counterpart. First we start by introducing the quadratic approximation

$$g_i^q(\mu, \hat{w}, \delta_w^i) = g_i(\mu, w) + \nabla_w g_i(\mu, \hat{w})^\top \delta_w^i + \frac{1}{2}(\delta_w^i)^\top \nabla_{ww} g_i(\mu, \hat{w})\delta_w^i$$

given by the second order Taylor expansion, where $\nabla_{ww} g_i$ denotes the Hessian matrix with respect to w. We will only consider the case $k = 2$ when developing the quadratic approximation. The worst case function then reads as

$$\hat{\varphi}_i^q(\mu) = \max_{\delta_w^i \in \mathbb{R}^{n_w}} g_i^q(\mu, \hat{w}, \delta_w^i) \quad \text{s.t.} \quad \|D_w^{-1}\delta_w^i\|_2 \leq 1 \quad \text{for} \quad i = 0, \dots, n_g. \tag{11}$$

Compared to the first order approximation, it is not possible to write down a closed form expression for $\hat{w}_i^q(\mu)$. However, the quadratic approximation of the worst case function corresponds to a trust region problem. The solution to this problem is then characterized by the following theorem [18, 45].

Theorem 2. *Let the H_i be a symmetric matrix. Then the trust region problem (11) possesses at least one global solution. Moreover, δ_w^i is a global solution if and only if there exists a Lagrange multiplier $\lambda_w^i \in \mathbb{R}$ such that*

$$\begin{aligned}(-H_i + \lambda_w^i \mathbb{D}_w)\delta_w^i - \nabla_w g_i(\mu, \hat{w}) &= 0,\\ \lambda_w^i(\|D_w^{-1}\delta_w^i\|_2 - 1) &= 0,\\ \|D_w^{-1}\delta_w^i\|_2 &\leq 1,\\ \lambda_w^i &\geq 0,\\ H_i - \lambda_w^i \mathbb{D}_w &\preceq 0\end{aligned}$$

holds for $i = 0, \dots, n_g$, where $\mathbb{D}_w = D_w^{-\top} D_w^{-1}$ and $A \preceq 0$ denotes that A is a negative semidefinite matrix.

The proof to this theorem is standard and we refer the reader to [18]. The optimality conditions define the Lagrange multiplier uniquely while the solution δ_w^i might not be unique. A detailed discussion can be found in [18]. In our application we set $H_i = \nabla_{ww} g_i(\mu, \hat{w})$ since the Hessian is symmetric. For a non-symmetric Hessian one can use without restriction its symmetric part in the quadratic model.

We reformulate the problem by applying a square to the norm. In this way we obtain differentiable constraints in our robust counterpart. The constraint then reads as $\|D^{-1}\delta_w^i\|_2^2 \leq 1$. Using the results from the theorem we can now formulate the robust counterpart by

$$\tag{12}\begin{aligned}
\min_{\substack{\mu\in\mathbb{R}^{n_\mu},\\ \delta_w^0,\dots,\delta_w^{n_g}\in\mathbb{R}^{n_w},\\ \lambda_w^0,\dots,\lambda_w^{n_g}\in\mathbb{R}}} \quad & g_0(\mu,\hat{w}) + \nabla_w g_0(\mu,\hat{w})^\top \delta_w^0 + \frac{1}{2}(\delta_w^0)^\top \nabla_{ww} g_0(\mu,\hat{w})\delta_w^0 \\
g_i(\mu,\hat{w}) + \nabla_w g_i(\mu,\hat{w})^\top \delta_w^i + \frac{1}{2}(\delta_w^i)^\top \nabla_{ww} g_i(\mu,\hat{w})\delta_w^i &\leq 0, \quad i = 1,\dots,n_g, \\
-\nabla_w g_i(\mu,\hat{w}) - \nabla_{ww} g_i(\mu,\hat{w})\delta_w^i + \lambda_w^i \mathbb{D}_w \delta_w^i &= 0, \quad i = 0,\dots,n_g, \\
\lambda_w^i(\|D_w^{-1}\delta_w^i\|_2^2 - 1) &= 0, \quad i = 0,\dots,n_g, \\
\|D_w^{-1}\delta_w^i\|_2^2 - 1 &\leq 0, \quad i = 0,\dots,n_g, \\
-\lambda_w^i &\leq 0, \quad i = 0,\dots,n_g, \\
\nabla_{ww} g_i(\mu,\hat{w}) - \lambda_w^i \mathbb{D}_w &\preceq 0, \quad i = 0,\dots,n_g.
\end{aligned}$$

The semidefinite constraint can be reformulated using the smallest eigenvalues. Since the dimension n_w of the uncertain parameters will be of moderate size (6 in our particular instance), this is a feasible strategy. This approach was outlined in [57]. Note that (12) is a *mathematical program with complementarity constraints* (MPCC). These types of problems can be solved efficiently by a sequential quadratic programming (SQP) method under relatively mild assumptions [28, 41]. We will provide a summary of the utilized strategy in a later section.

The second derivative in the quadratic approximation can be derived in two different ways, similar as for the first derivative in the linear approximation. Let us shortly provide the necessary equations. For the sensitivity approach we get

$$\begin{aligned}
\nabla_{ww} g_i(\mu,w) := {} & \frac{\partial}{\partial u}\hat{g}_i(\mu,u,w)\bar{s} + s^* \frac{\partial^2}{\partial u^2}\hat{g}_i(\mu,u,w)s \\
& + s^* \frac{\partial^2}{\partial u \partial w}\hat{g}_i(\mu,u,w) + \frac{\partial^2}{\partial w \partial u}\hat{g}_i(\mu,u,w)s + \frac{\partial^2}{\partial w^2}\hat{g}_i(\mu,u,w),
\end{aligned}$$

where s solves (9b) and $\bar{s}$ is given by the second order sensitivity equation

$$\tag{13}\begin{aligned}
& \frac{\partial}{\partial u}e(\mu,u,w)\bar{s} + s^* \frac{\partial^2}{\partial u^2}e(\mu,u,w)s \\
& \qquad + s^* \frac{\partial^2}{\partial u \partial w}e(\mu,u,w) + \frac{\partial^2}{\partial w \partial u}e(\mu,u,w)s + \frac{\partial^2}{\partial w^2}e(\mu,u,w) = 0.
\end{aligned}$$

Next we have a look at the more efficient adjoint approach. This approach does not require the solution of the second order sensitivity equation (13). We define the Lagrange function $L_i : \mathbb{R}^{n_\mu} \times V \times \mathbb{R}^{n_w} \times Z^* \to \mathbb{R}$

$$L_i(\mu,u,w,p) := \hat{g}_i(\mu,u,w) + \langle p, e(\mu,u,w)\rangle_{Z^*,Z},$$

where $\langle\cdot,\cdot\rangle_{Z^*,Z}$ denotes the dual pairing of Z^* and Z. Then the second derivative of

g_i using the adjoint approach is given by

$$\begin{aligned}\nabla_{ww} g(\mu, w) := s^* \frac{\partial^2}{\partial u^2} L_i(\mu, u, w, p) s + s^* \frac{\partial^2}{\partial u \partial w} L_i(\mu, u, w, p) \\ + \frac{\partial^2}{\partial w \partial u} L_i(\mu, u, w, p) s + \frac{\partial^2}{\partial w^2} L_i(\mu, u, w, p),\end{aligned}$$

where s and p solve (9b) and (10b), respectively. Usually the computation of the second partial derivatives of the Lagrangian is cheap and thus the adjoint approach is already for a moderate number of parameters more efficient than using the second order sensitivity equation (13). Details on the derivation can be found in [32]. In this work we assume that the number of parameters is small and we use the sensitivity based approach. Let us note that an adjoint approach is preferable since it does not require the computation of second order sensitivities but only one adjoint equation and first order sensitivities.

3.2.3. Moving expansion point. While the quadratic approximation can be a significant improvement to the accuracy of the approximation of the robust counterpart it might still suffer from bad approximation. This is mostly due to the choice of the expansion point in the second order Taylor series. The default choice of using the nominal value $\hat{w}$ does not necessarily lead to a good quadratic approximation on the entire region defined by the uncertainty set. To overcome this problem we propose a strategy using a moving expansion point. The idea is to check the approximation accuracy and if required move the expansion point. Let us outline the strategy in more details.

We introduce the variables $\bar{w}_i$ and $\bar{\delta}_i$ which are initialized with $\hat{w}$ and 0, for $i = 0, \dots, n_f$, respectively. Then the quadratic approximation of $g_i(\mu, w)$ with respect to w can be written as

$$g_i^q(\mu, \bar{w}_i, \delta_i^w) = g_i(\mu, \bar{w}_i) + \nabla_w g_i(\mu, \bar{w})^\top \delta_i + \frac{1}{2} \delta_i^\top \nabla_{ww} g_i(\mu, \bar{w}) \delta_i, \tag{14}$$

where $\delta_i = \delta_i^w - \bar{\delta}_i$. In the initial case, i.e. $\bar{w}_i = \hat{w}$ and $\bar{\delta}_i = 0$ for $i = 0, \dots, n_g$ this is equivalent to the previously introduced quadratic approximation. If the approximation point gets shifted,

$$\bar{w}_i = \hat{w} + \delta_i^w \quad \text{and} \quad \bar{\delta}_i = \delta_i^w, \quad i = 0, \dots, n_g \tag{15}$$

we obtain a quadratic approximation at the new expansion point $\bar{w}_i$. Note that for each g_i we can obtain a different expansion point. Further, the constraints on δ_i^w do not have to be adjusted since we perform the shift by $\bar{\delta}_i$. This formulation makes it easy to incorporate the proposed strategy into existing codes. The decision to update the expansion point for the quadratic model is made by looking at the approximation error. For this we compare the exact evaluation of g_i with the quadratic approximation g_i^q. If the difference is too large we perform the update. This strategy is embedded into the optimization method and is performed before the computation of an update. This guarantees that locally, at each iteration of the optimization algorithm, the quadratic approximation is of good quality. The procedure is summarized in Algorithm 1.

Let us note that we choose to update the expansion point by $\bar{w}_i = \hat{w} + \delta_i^w$. It can be of advantage to add a scaling factor, i.e. $\bar{w}_i = \hat{w} + \alpha \delta_i^w$ with $\alpha \in (0, 1]$. This factor can be interpreted as a step size strategy and can for example be determined by an Armijo-Backtracking strategy [45]. This can lead to a faster convergence although

Algorithm 1 Moving expansion point algorithm

```
Require: $\hat{w}$, $\bar{w}_i$, $\bar{\delta}_i$ $\delta_i^w$ for $i = 0, \dots, n_g$, and tolerance $\epsilon_{mov}$
for $i = 0$ to $n_g$ do
    Set $\delta_i = \delta_i^w - \bar{\delta}_i$
    Evaluate $g_i^q = g_i(\mu, \bar{w}_i) + \nabla_w g_i(\mu, \bar{w}_i)^\top \delta_i + \frac{1}{2}\delta_i^\top \nabla_{ww} g_i(\mu, \bar{w}_i)\delta_i$
    if $|g_i(\mu, \hat{w}_i + \delta_i^w) - g_i^q| > \epsilon_{mov}$ then
        Update expansion point $\bar{w}_i = \hat{w} + \delta_i^w$ and $\bar{\delta}_i = \delta_i^w$
    end if
end for
return $\bar{w}_i$ and $\bar{\delta}_i$ for $i = 0, \dots, n_g$
```

in our numerical examples the choice $\alpha = 1$ worked well. We did not perform a convergence analysis on the proposed strategy, but in the numerical experiments we observe fast convergence. After a few iteration in the optimization algorithm the expansion point does not get updated any more.

Remark 3. The moving expansion point strategy is not applied to the linear approximation of the robust counterpart. In case that the maximum is within the uncertainty set and not on the boundary, an update of the expansion point will not improve the approximation but will lead to oscillations.

3.2.4. Extension to the optimization variable. While we have so far only considered uncertainty in the model parameter w we will briefly outline the robustification of optimization variables in this section. With this we want to account for uncertainty in the realization of the optimal solution. This can be due to manufacturing tolerances during the production process of the work piece or due to errors introduced by numerical simulations. The approach chosen is analogous to the previous sections. Hence we will only shortly outline it and give the final results. To incorporate the uncertainty into the optimization process we introduce an uncertain perturbation δ_μ for the optimization variable. The uncertainty set for the perturbation is defined as

$$U_{\mu,k} = \{\, \delta_\mu \in \mathbb{R}^{n_\mu} \mid \|D_\mu^{-1}\delta_\mu\|_k \le 1 \,\}$$

with $D_\mu \in \mathbb{R}^{n_\mu \times n_\mu}$. The worst case function then reads

$$\varphi_i(\mu) := \max_{\delta_\mu \in U_{\mu,k}} g_i(\mu + \delta_\mu, w). \tag{16}$$

Hence the robust counterpart associated to our optimization problem (5) can be written as (6). To obtain a numerical feasible problem we apply an approximation of the worst case function. In this case we only consider the linear approximation. This choice can be justified by the fact that the linearization is carried out around the parameter μ. This means that during the optimization procedure the linearization is updated whenever μ changes. This is similar to the moving expansion point strategy for the quadratic approximation. Hence, if the uncertainty set is of moderate size the linear approximation will be of good quality. Utilizing the result (8) we can write the linear approximation of the robust counterpart as

$$\min_{\mu \in \mathbb{R}^{n_\mu}} g_0(\mu, w) + \|D_\mu^\top \nabla_\mu g_0(\mu, w)\|_{k^*} \quad \text{s.t.} \quad g_i(\mu, w) + \|D_\mu^\top \nabla_\mu g_i(\mu, w)\|_{k^*} \le 0. \tag{17}$$

A smooth reformulation in the case $k = \{\, 1, \infty \,\}$ can again be obtained by introducing slack variables. For the case $k = 2$ the same arguments apply as previously outlined.

As a last step we will now combine the results. The goal is to formulate a robust optimization problem that accounts for both, uncertainties in model parameters and optimization variables. We will right away opt for the quadratic approximation in the model parameter and leave the other cases to the reader. For the uncertainty sets we consider the setting $w \in U_{w,2}$ and $\mu \in U_{\mu,\infty}$. Then the mixed linear quadratic approximated robust counterpart reads:

$$\tag{18} \begin{aligned} &\min_{\substack{\mu,\zeta_\mu^0,\dots,\zeta_\mu^{n_g} \in \mathbb{R}^{n_\mu} \\ \delta_w^0,\dots,\delta_w^{n_g} \in \mathbb{R}^{n_w} \\ \lambda_w^0,\dots,\lambda_w^{n_g} \in \mathbb{R}}} g_0(\mu,\hat{w}) + \mathbb{E}^\top \zeta_\mu^0 + \nabla_w f_0(\mu,\hat{w})^\top \delta_w^0 + \frac{1}{2}(\delta_w^0)^\top \nabla_{ww} g_0(\mu,\hat{w})\delta_w^0 \\ & g_i(\mu,\hat{w}) + \mathbb{E}^\top \zeta_\mu^i + \nabla_w g_i(\mu,\hat{w})^\top \delta_w^i + \frac{1}{2}(\delta_w^i)^\top \nabla_{ww} g_i(\mu,\hat{w})\delta_w^i \le 0, \quad i = 1,\dots,n_g, \\ & -\zeta_\mu^i \le D_\mu^\top \nabla_\mu g_i(\mu,\hat{w}) \le \zeta_\mu^i, \quad i = 0,\dots,n_g, \\ & -\nabla_w g_i(\mu,\hat{w}) - \nabla_{ww} g_i(\mu,\hat{w})\delta_w^i + \lambda_w^i \mathbb{D}_w \delta_w^i = 0, \quad i = 0,\dots,n_g, \\ & \lambda_w^i(\|D_w^{-1}\delta_w^i\|_2^2 - 1) = 0, \quad i = 0,\dots,n_g, \\ & \|D_w^{-1}\delta_w^i\|_2^2 - 1 \le 0, \quad i = 0,\dots,n_g, \\ & -\lambda_w^i \le 0, \quad i = 0,\dots,n_g, \\ & \nabla_{ww} g_i(\mu,\hat{w}) - \lambda_w^i \mathbb{D}_w \preceq 0, \quad i = 0,\dots,n_g \end{aligned}$$

with $\mathbb{E} = (1,\dots,1)^\top \in \mathbb{R}^{n_\mu}$. The additional variables and indices for the moving average strategy for the quadratic approximation is omitted in order to avoid a notation overload.

3.3. Numerical methods. The introduced problem (18) is a MPCC and is hard to solve. In [17] it was shown that problems of this type violate the Mangasarian-Fromovitz constraint qualification (MFCQ) at any feasible point. As a consequence the multiplier set is unbounded, the active constraint normals are linearly dependent, the central path fails to exist, and linearizations of the nonlinear program can be inconsistent arbitrarily close to the solution. Fortunately, many advances have been made with respect to the numerical treatment of this type of problems using the SQP solvers with some slight modifications [26, 28, 35, 41] and the interior-point methods [42, 51]. It turns out that a large class of MPCCs, written as nonlinear programs can be solved reliably and efficiently [27]. We adapt the SQP solver presented in [26] and replace the complementarity constraint in (18) by a nonlinear complementarity problem (NCP) function. This gives rise to

$$\tag{19} \begin{aligned} &\min_{\substack{\mu,\zeta_\mu^0,\dots,\zeta_\mu^{n_g} \in \mathbb{R}^{n_\mu} \\ \delta_w^0,\dots,\delta_w^{n_g} \in \mathbb{R}^{n_w} \\ \lambda_w^0,\dots,\lambda_w^{n_g} \in \mathbb{R} \\ \xi^0,\dots,\xi^{n_g} \in \mathbb{R}}} g_0(\mu,\hat{w}) + \mathbb{E}^\top \zeta_\mu^0 + \nabla_w g_0(\mu,\hat{w})^\top \delta_w^0 + \frac{1}{2}(\delta_w^0)^\top \nabla_{ww} g_0(\mu,\hat{w})\delta_w^0 \\ & g_i(\mu,\hat{w}) + \mathbb{E}^\top \zeta_\mu^i + \nabla_w g_i(\mu,\hat{w})^\top \delta_w^i + \frac{1}{2}(\delta_w^i)^\top \nabla_{ww} g_i(\mu,\hat{w})\delta_w^i \le 0, \quad i = 1,\dots,n_g, \\ & -\zeta_\mu^i \le D_\mu^\top \nabla_\mu g_i(\mu,\hat{w}) \le \zeta_\mu^i, \quad i = 0,\dots,n_g, \\ & -\nabla_w g_i(\mu,\hat{w}) - \nabla_{ww} g_i(\mu,\hat{w})\delta_w^i + \lambda_w^i \mathbb{D}_w \delta_w^i = 0, \quad i = 0,\dots,n_g, \\ & \|D_w^{-1}\delta_w^i\|_2^2 - 1 + \xi^i = 0, \quad i = 0,\dots,n_g, \\ & \Phi(\xi^i,\lambda_w^i) \le 0, \quad i = 0,\dots,n_g, \\ & -\xi^i \le 0, \quad i = 0,\dots,n_g, \\ & -\lambda_w^i \le 0, \quad i = 0,\dots,n_g, \\ & \nabla_{ww} g_i(\mu,\hat{w}) - \lambda_w^i \mathbb{D}_w \preceq 0, \quad i = 0,\dots,n_g, \end{aligned}$$

where $\Phi(a, b)$ is the NCP function. Let us next introduce the NCP functions under consideration in this work.

An NCP function is defined as $\Phi : \mathbb{R}^2 \to \mathbb{R}$ such that $\Phi(a, b) = 0$ if and only if a, $b \geq 0$, and $ab \leq 0$. The NCP function utilized in this work is the extension of the smoothed residual function introduced in [41]

$$\Phi(a, b) = \frac{1}{2}\left(a + b - \sqrt{(a-b)^2 + \frac{ab}{\sigma}}\right). \tag{20}$$

For a fixed $\sigma = 1/2$ we get the well known Fischer-Burmeister function [25]. It can easily be seen that for the chosen NCP function the condition a, $b \geq 0$ and $\Phi(a, b) \leq 0$ is equivalent to the complementarity condition a, $b \geq 0$ and $ab = 0$. The gradient is given by

$$\nabla\Phi(a, b) = \frac{1}{2}\left(\begin{array}{c} 1 - \dfrac{a - b + \frac{b}{2\sigma}}{\sqrt{a^2 + b^2} + \frac{ab}{\sigma}} \\ 1 - \dfrac{a - b + \frac{a}{2\sigma}}{\sqrt{a^2 + b^2} + \frac{ab}{\sigma}} \end{array}\right).$$

Further, for $a > 0$ and $b > 0$ we have

$$\nabla\Phi(a, 0) = \left(\begin{array}{c} 0 \\ 1 - \frac{1}{4\sigma} \end{array}\right) \quad \text{and} \quad \nabla\Phi(0, b) = \left(\begin{array}{c} 1 - \frac{1}{4\sigma} \\ 0 \end{array}\right)$$

Note that by introducing the NCP function, problem (19) becomes non-smooth at the origin. Fortunately, 0 is a generalized gradient of the NCP function at the origin. Further, we relax the linearization of the NCP function as introduced in [41] given by

$$a \geq 0,\ b \geq 0,\ \Phi(\hat{a}, \hat{b}) + \nabla\Phi(\hat{a}, \hat{b})^\top \left(\begin{array}{c} a - \hat{a} \\ b - \hat{b} \end{array}\right) \leq \delta\left(\min(1, \Phi(\hat{a}, \hat{b}))\right)^{\kappa}$$

with $0 \leq \delta \leq 1$, and $0 < \kappa \leq 1$ chosen constant. Recommended choices for the parameters are $\sigma = 32$, $\delta = 0.1$ and $\kappa = 1$.

Having introduced the NCP function and its derivatives, we will have a short look at some implementation details. We will consider an SQP method with line search utilizing an ℓ^1-penalization function [45]. In the numerical experiments we will only compute first order derivatives for the optimization procedure. The Hessian is approximated using damped BFGS updates. If the quadratic program (QP) of the SQP method at any point should become infeasible we enter a restoration phase. For this we utilize a combination of the ideas presented in [26, 43]. We define the function

$$h(z) = \sum_{i \in \mathcal{E}} |c_i(z)| + \sum_{i \in \mathcal{I}} \max(0, c_i),$$

where $c_i(z)$, $i \in \mathcal{E}$ are the equality constraints and $c_i(z)$, $i \in \mathcal{I}$ the inequality constraints. Then the strategy in the restoration phase is to solve the minimization problem

$$\min_z h(z).$$

The optimization procedure is summarized in Algorithm 2. Furthermore, in the algorithm the integration of the moving expansion point strategy is outlined. A stopping criterion based on relative optimality is used. Additionally, the approximation error of the moving expansion point algorithm is included to the stopping criterion to ensure that a good approximation is achieved.

Algorithm 2 Solving strategy (SQP Method)

```
Require: Initial value $z = (\mu^0, \delta_w^0, \dots, \delta_w^{n_g}, \lambda_w^0, \dots, \lambda_w^{n_g})$
Set $\bar{w}_i = \hat{w}$ and $\bar{\delta}_i = 0$ for $i = 0, \dots, n_g$
Initialize counter $k = 0$
while stopping criterion not satisfied do
    Check approximation quality (Algorithm 1)
    Compute update $d$ solving QP
    if QP is feasible then
        Update variable $z^{k+1} = z^k + \alpha d$ with suited $\alpha \in (0, 1]$
    else
        $z^{k+1} = \arg\min h(z)$     (Restoration Phase)
    end if
    Update counter $k = k + 1$
end while
return $z^k$
```

Remark 4. The robust counterparts (6) and (7) are semi-infinite optimization problems and our approach can be considered as a solution method for their MPCC formulation. We note that there exist also solution approaches by using an equality constraint based on smoothed NCP functions Φ_τ, $\tau > 0$. This leads to an outer approximation of the original problem and the original problem is obtained for $\tau \searrow 0$, see e.g. [60]. Since our method based on [41] turns out to be very efficient, we prefer not to invoke smoothing of the NCP function.

4. Model order reduction.

In this section we look at a method to accelerate the computation during the optimization procedure. As can be seen in (18), for the approximated robust counterpart several derivatives are required. Since the presented problem originates from a PDE constrained optimization problem, additional PDE solves are required for each derivative. This can be computationally expensive since the discretization of PDEs leads to very large linear systems. Hence we require an efficient strategy to overcome this problem. In the past, model order reduction methods based on proper orthogonal decomposition (POD) [16, 33, 38], balanced truncation [3, 31] and the reduced basis method [19, 44, 48, 52] have been developed to speed up the computation in PDE constrained optimization problems. In this work we consider a combination of POD and the reduced basis method. We will perform the model order reduction with respect to the rotation parameter ϑ, but not with respect to the geometry parameter μ. Let us start by introducing a discrete version of (2). In a next step we will then apply the model order reduction to the discrete version of the PDE.

4.1. Finite element discretization.

To obtain a discrete version of (2) we utilize the finite element method (FEM). We choose the approximation

$$u^N(\vartheta; \mu) = \sum_{i=1}^{N} \mathbf{u}_i(\vartheta; \mu) \phi_i(\mathbf{x}, \vartheta),$$

where $\phi_i(\mathbf{x},\vartheta)$ are linear ansatz functions in 2D [14, 15]. Denoting the finite element coefficients by the vector $\mathbf{u} \in \mathbb{R}^N$ and using the introduced parametrization we can write the discrete version of (2) as a N dimensional linear system

$$\mathbf{K}(\vartheta;\mu)\mathbf{u} = \mathbf{f}(\vartheta;\mu), \tag{21}$$

where the stiffness matrix and right hand side are given by

$$\mathbf{K}(\vartheta;\mu) = \sum_{k=1}^{L}\sum_{i,j=1}^{2} \Theta_k^{ij}(\mu)\mathbf{K}_k^{ij}(\vartheta) \quad \text{and} \quad \mathbf{f}(\vartheta;\mu) = \sum_{k=1}^{L} \Theta_k^{f}(\mu)\mathbf{f}_k(\vartheta), \tag{22}$$

respectively. Note that the $\mathbf{K}_k^{ij}(\vartheta)$ and $\mathbf{f}_k(\vartheta)$ are the local stiffness matrices and right hand sides introduced by the affine decomposition.

To take the movement of the rotor into account we use the locked step method [50, 55]. This method is widely used in literature to study electrical machines. For this, the circular interface between the static and moving part is introduced. This interface is subdivided by an equidistant mesh. The rotation of the rotor is then discretized by fixed angles corresponding to a rotation by one mesh cell of the equidistant interface mesh. The nodes on the interface of the stator and the rotor are then reconnected leading to the mesh for the next computation. Let us define $\Delta\vartheta$ as the angular step between two positions of the moving part. Then the rotation angle ϑ is given by $\vartheta_k = k\Delta\vartheta$, $k \in \mathbb{N}_0$. In the discrete setting this can be realized by partitioning $\mathbf{u}$ into a static part, a rotating part and an interface. This idea is directly related to the method of domain decomposition [62]. Then the system (21) can be rewritten as

$$\begin{bmatrix} \mathbf{K}_{ss} & 0 & \mathbf{K}_{sI} \\ 0 & \mathbf{K}_{rr}(\mu) & \mathbf{K}_{rI}(\vartheta) \\ \mathbf{K}_{sI}^{\top} & \mathbf{K}_{rI}^{\top}(\vartheta) & \mathbf{K}_{II}(\vartheta) \end{bmatrix} \begin{bmatrix} \mathbf{u}_s \\ \mathbf{u}_r \\ \mathbf{u}_I \end{bmatrix} = \begin{bmatrix} \mathbf{f}_s \\ \mathbf{f}_r(\mu) \\ \mathbf{f}_I(\vartheta) \end{bmatrix}, \tag{23}$$

where $\mathbf{K}_{ss}$, $\mathbf{K}_{rr}(\mu)$, $\mathbf{f}_s$ and $\mathbf{f}_r(\mu)$ are the stiffness matrices and right hand sides on the static and moving part and do no longer depend on the angle ϑ. For the points on the interface there are two cases. The interface of the static part is independent of the angle ϑ and hence we get the corresponding stiffness matrix $\mathbf{K}_{sI}$. For the rotor side we have to perform the shift, this is indicated with ϑ in the corresponding stiffness matrix $\mathbf{K}_{rI}$. On the interface also a shift has to be performed hence also here the corresponding stiffness matrix $\mathbf{K}_{II}$ and right hand side $\mathbf{f}_I$ are dependent on ϑ. Let us note that it is not required to reassemble matrices. All of these shifts can be performed by index shift and hence allow a very efficient implementation. The size of the system does not change, i.e., we have $N = N_s + N_r + N_I$. The dependency of the matrix $\mathbf{K}$ on the parameter μ is limited to the rotating part, which is also indicated. This is not a limitation but rather to avoid a notation overload.

4.2. Proper orthogonal decomposition. Let us recall the POD method so we can then develop the extensions used in this work. We will assume that the solution to the PDE is given in the discrete form, i.e., let $\mathbf{u}(\vartheta;\mu) \in \mathbb{R}^N$ be the solution to the discretized version of our PDE. From a simulation we obtain the snapshots $\mathbb{R}^N \ni \mathbf{u}^k(\mu) \approx \mathbf{u}(k\Delta\vartheta;\mu)$ for $k \in \mathcal{K}$, where $\mathcal{K}$ is an index set with elements in $\mathbb{N}_0$ for which (21) was solved. Then, a POD basis $\{\psi_i\}_{i=1}^{\ell}$ computed from these snapshots is given by the solution to

$$\left\{\begin{aligned} &\min_{\psi_1,\ldots,\psi_\ell\in\mathbb{R}^N}\sum_{k\in\mathcal{K}}\Big|\mathbf{u}^k(\mu)-\sum_{i=1}^{\ell}\langle\mathbf{u}^k(\mu),\psi_i\rangle_{\mathrm{W}}\,\psi_i\Big|_{\mathrm{W}}^2\\ &\qquad\text{s.t. } \langle\psi_i,\psi_j\rangle_{\mathrm{W}}=\delta_{ij}\ \text{for } 1\le i,j\le\ell,\end{aligned}\right. \tag{24}$$

where $\langle\cdot\,,\cdot\rangle_{\mathrm{W}}$ stands for the weighted inner product in $\mathbb{R}^N$ with a positive definite, symmetric matrix $\mathrm{W}\in\mathbb{R}^{N\times N}$. Let us introduce the matrix $\mathbf{U}_{\mathcal{K}}$ as the collection of the snapshots $\mathbf{u}^k(\mu)$ with $k\in\mathcal{K}$. Then we can write the operator $\mathcal{R}$ arising from the optimization problem (24) as

$$\mathcal{R}\psi=\sum_{k\in\mathcal{K}}\langle\mathbf{u}^k(\mu),\psi\rangle_{\mathrm{W}}\mathbf{u}^k=\mathbf{U}_{\mathcal{K}}\big(\mathbf{U}_{\mathcal{K}}\big)^{\top}\mathrm{W}\psi\quad\text{for }\psi\in\mathbb{R}^N.$$

The unique solution to (24) is then given by the eigenvectors corresponding to the ℓ largest eigenvalues of $\mathcal{R}$, i.e., $\mathcal{R}\psi_i=\lambda_i\psi_i$ with $\lambda_i>0$ [29]. In many cases the operator $\mathcal{R}$ is of large dimension and it might be better to set up and solve the eigenvalue problem

$$\mathbf{U}_{\mathcal{K}}^{\top}\mathrm{W}\mathbf{U}_{\mathcal{K}}v_i=\lambda_i v_i,\quad i=1,\ldots,\ell$$

and obtain the POD basis by $\psi_i=1/\sqrt{\lambda_i}\mathbf{U}_{\mathcal{K}}v_i$. This approach is computationally more efficient. Both approaches are related by the singular value decomposition of the matrix $\mathrm{W}^{1/2}\mathbf{U}_{\mathcal{K}}$. For completeness let us state the POD approximation error given by

$$\sum_{k\in\mathcal{K}}\Big|\mathbf{u}^k(\mu)-\sum_{i=1}^{\ell}\langle\mathbf{u}^k(\mu),\psi_i\rangle_{\mathrm{W}}\,\psi_i\Big|_{\mathrm{W}}^2=\sum_{i=\ell+1}^{d}\lambda_i, \tag{25}$$

where d is the rank of $\mathbf{U}_{\mathcal{K}}$. We will collect the POD basis ψ_i in the matrix $\Psi=[\psi_1,\ldots,\psi_l]\in\mathbb{R}^{N\times\ell}$. This will ease notation later on.

Having introduced the general way of computing a POD basis let us now lay out the details of the approach utilized in this work. To speed up the computation we want to minimize the number of solves involving FEM and push the computations onto the reduced order models. It is crucial however to have accurate reduced order models in order to have accurate results. The approach we are presenting will not require precomputation as for example in the reduced basis method. Let us start by outlining how we apply the POD basis to (23) and in a second step give the details on how to obtain the basis efficiently.

We generate a POD basis for each of the parts. One basis for the static part and one basis for the moving part. We do not perform a reduction on the interface, i.e., we work with the FEM ansatz space on the interface. This is motivated by the observation that the decay of the eigenvalues is very slow, which would result in a large POD basis. Since the FEM space for the interface is usually of moderate dimension the gain of using POD would be negligible. We compute the POD basis as solution to (24) utilizing the snapshots $\mathbf{u}_s^k$ and $\mathbf{u}_r^k$ to obtain Ψ^s and Ψ^r, respectively. We then make the ansatz

$$\mathbf{u}_s^{\ell}=\sum_{i=1}^{\ell^s}\psi_i^s\bar{\mathbf{u}}_{s,i}=\Psi^s\bar{\mathbf{u}}_s\quad\text{and}\quad\mathbf{u}_r^{\ell}=\sum_{i=1}^{\ell^r}\psi_i^r\bar{\mathbf{u}}_{r,i}=\Psi^r\bar{\mathbf{u}}_r,$$

where the POD coefficients are indicated with a bar. The reduced order model is then generated by projecting (23) onto the subspace spanned by the POD basis and we obtain the system

$$\begin{bmatrix} (\Psi^s)^\top \mathbf{K}_{ss}\Psi^s & 0 & (\Psi^s)^\top \mathbf{K}_{sI} \\ 0 & (\Psi^r)^\top \mathbf{K}_{rr}(\mu)\Psi^r & (\Psi^r)^\top \mathbf{K}_{rI}(\vartheta) \\ \mathbf{K}_{sI}^\top \Psi^s & \mathbf{K}_{rI}^\top(\vartheta)\Psi^r & \mathbf{K}_{II}(\vartheta) \end{bmatrix} \begin{bmatrix} \bar{\mathbf{u}}_s \\ \bar{\mathbf{u}}_r \\ \mathbf{u}_I \end{bmatrix} = \begin{bmatrix} (\Psi^s)^\top \mathbf{f}_s \\ (\Psi^r)^\top \mathbf{f}_r(\mu) \\ \mathbf{f}_I(\vartheta) \end{bmatrix}.$$

In short the system will be written as

$$\mathbf{K}^\ell(\vartheta;\mu)\bar{\mathbf{u}} = \mathbf{f}^\ell(\vartheta;\mu) \tag{26}$$

with $\bar{\mathbf{u}}$ the vector with the POD coefficients. This system is of dimension $\ell_s + \ell_r + N_I$ and of much smaller dimension as the original system (23) which was of dimension N. Note that we did not exploit the affine decomposition introduced in the parametrization for the geometry deformation. As previously mentioned, the choice for this method was made because the transformation has physical meaning and is known analytically and eases the computation of derivatives for the optimization. We do not require it for the model order reduction.

Let us next have a look at how to determine the POD basis. The goal is to reduce the computational cost with respect to the rotation. One full rotation requires N_I solves of the system (23), i.e., for all ϑ_k with $k \in \mathcal{K} := \{0,\dots,N_I-1\}$. In the symmetric case it is not required to solve a full rotation but only for angles that cover one pole, for our particular example this means one sixth, i.e., $N_I/6$ solutions are needed. Note, we assume that N_I is divisible by N_p, where N_p is the number of poles of the machine. In the non-symmetric case this is not possible. The idea is to partition $\mathcal{K}$ into disjoint sets, $\mathcal{K}_i := \{ i(N_I/N_p), \dots (i+1)(N_I/N_p) - 1 \}$ with $i = 0,\dots,N_p-1$. Additionally we define a sequence of index sets $\mathcal{S}_j$, $j = 1,\dots,N_s$ with the property $\mathcal{S}_j \subset \mathcal{S}_{j+1}$. A possible choice is for example

$$\mathcal{S}_1 = \{0, 10, \dots, N_I/N_p\}, \mathcal{S}_2 = \{0, 5, \dots, N_I/N_p\}, \mathcal{S}_3 = \{0, 1, \dots, N_I/N_p\}.$$

Note that the sets can also be chosen arbitrarily as long as they fulfill the subset property. This is required to be able to reuse the already computed snapshots and hence minimize computational overhead. The strategy is then as follows. We start at the first partition ($\mathcal{K}_1$) and evaluate (23) for ϑ_k and $k \in \mathcal{K}_1(\mathcal{S}_1)$. From the obtained solutions a POD basis is computed. Then an error estimator $\Delta\mathbf{u}(\vartheta_k)$, $k = 1,\dots,N_I$, is evaluated to determine in which partition the largest error is obtained. If the partition changes we repeat the step for the new partition and enlarge the snapshot set by adding the new solutions to the old ones. When the error is in the same partition, we increase j, i.e., increase the number of computed snapshots in the partition. The same is done if a partition is revisited during the process. The strategy is summarized in Algorithm 3. This sampling of the partitions is similar to the greedy algorithm. We decided to add more than one PDE solution at a time to the snapshot set to minimize the overhead of evaluating the error estimator and generating the Roms too many times since all the computations are done during the optimization process and are not in a preprocessing. In our application the proposed strategy converges very fast. In the nominal case, where the machine is symmetric usually only one partition is visited and in some cases a refinement is applied. In the robust optimization, where non-symmetries are introduced, additional partitions are visited. The dimension of the POD basis is chosen such that

$$\frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \leq \varepsilon_{rel}$$

holds. This is a popular choice, where a typical value is $\varepsilon_{rel} = 0.9999$. Note that the denominator can be computed by $\operatorname{trace}(\mathbf{U}_{\mathcal{K}}^{\top} \mathrm{W} \mathbf{U}_{\mathcal{K}})$ and hence not all d eigenvalues have to be computed.

Algorithm 3 Adaptive POD

```
Require: $N_p$, $\mathcal{K}_{i=1}^{N_p}$, $\mathcal{S}_{j=1}^{N_s}$ and $\varepsilon$ (tolerance)
Build history vector $h = (1, \dots, 1)^\top \in \mathbb{N}^{N_p}$
Choose first section $i = 1$ and snapshot location index $j = h(i)$
Solve $\mathbf{u}^k(\mu)$ for $k \in \mathcal{K}_i(S_j)$
Compute POD basis using (24)
Evaluate error estimator $\Delta\mathbf{u}(\vartheta_k)$ for $k = 1, \dots, N_I$
if $\max_k \Delta\mathbf{u}(\vartheta_k) > \varepsilon$ is in same partition then
    Increase snapshot location index, i.e., $h(i) = h(i) + 1$ and $j = h(i)$
    GOTO 3
else if $\max_k \Delta\mathbf{u}(\vartheta_k) > \varepsilon$ is in different partition then
    Move to partition $i$ containing $k$ and choose snapshot location index $j = h(i)$
    GOTO 3
else
    return POD basis and reduced solution $\mathbf{u}^\ell$
end if
```

We recompute a POD basis for each geometric configuration. Alternatively, a strategy can be used that keeps the POD basis and adaptively updates in the course of the optimization process [2, 39, 64].

Let us now shortly have a look at the error estimator. For this let us recall some basic quantities required. For a Hilbert space X we define the coercivity and continuity constant

$$\alpha(\vartheta, \mu) = \inf_{v \in X \setminus \{0\}} \frac{a(v, v; \vartheta, \mu)}{\|v\|_X^2} \quad \text{and} \quad \gamma(\vartheta, \mu) = \sup_{v \in X \setminus \{0\}} \sup_{u \in X \setminus \{0\}} \frac{a(u, v; \vartheta, \mu)}{\|u\|_X \|v\|_X},$$

respectively. Further, we define the residual $r^0(\mathbf{u}^\ell; \vartheta, \mu) = \mathbf{f}(\vartheta; \mu) - \mathbf{K}(\vartheta; \mu)\mathbf{u}^\ell$. Then the error introduced by the ROM in the variable $\mathbf{u}$ can be characterized by

$$\|\mathbf{u} - \mathbf{u}^\ell\|_X \leq \Delta\mathbf{u}(\vartheta) := \frac{\|r^0(\mathbf{u}^\ell; \vartheta, \mu)\|_{X'}}{\alpha(\vartheta, \mu)}, \tag{27}$$

where $\| \cdot \|_{X'}$ is the dual norm. This is a standard result and can be found in [52]. Note that the error is measured with respect to the finite element solution. At this point it is assumed that the finite element solution is accurate enough to approximate the solution of the continuous problem.

Since we started from a reference geometry $\hat{\Omega}$, we assume at this point that the coercivity constant $\alpha(\vartheta, \hat{\mu})$ is known. Further, we consider isotropic material, i.e., $\nu_k^{ii} = \nu_k^{jj}$ and $\nu_k^{ij} = \nu_k^{ji} = 0$ for $i, j = 1, 2$, $i \neq j$ and $k = 1, \dots, L$. Then we can compute the lower bound $0 < \alpha_{LB}(\mu, \vartheta) \leq \alpha(\mu, \vartheta)$ by using the introduced affine decomposition by the *Min-Theta* approach [48]:

$$\alpha_{LB}(\vartheta,\mu) := \alpha(\vartheta,\hat{\mu}) \min_{\substack{k=1,\dots,L\\ i=1,2}} \frac{\Theta_k^{ii}(\mu)}{\Theta_k^{ii}(\hat{\mu})}.$$

Note that the rotation as introduced in this work does not influence the coercivity constant and can be omitted. Let us remark that the computation of the residual norms can be performed very efficiently using the introduced affine decomposition [24].

Since we are considering an optimization problem also the accuracy of the derivatives is crucial. As previously outlined we consider an approach using the sensitivities. Therefore, additional PDEs have to be solved which contribute to the computational expenses in the optimization process. Hence, model order reduction is of great interest. Let us assume that $\mu \in \mathbb{R}$, i.e., $n_\mu = 1$. This is to simplify the following equations by avoiding mixed partial derivatives.

Proposition 5. *Let the coefficient functions $\Theta_k^{ij}(\mu)$ and $\Theta_k^{f}(\mu)$ be n times differentiable with respect to μ. Then $\mathbf{u}$ is differentiable with respect to μ and the derivative $\mathbf{u}^n$ is given by the sensitivity equation*

$$\mathbf{K}(\vartheta;\mu)\mathbf{u}^n = \mathbf{f}^n(\vartheta;\mu) - \sum_{k=1}^{n} \binom{n}{k} \mathbf{K}^k(\vartheta;\mu)\mathbf{u}^{n-k}, \tag{28}$$

where the super-indices denote derivatives and $\binom{n}{k} = \frac{n!}{k!(n-k)1}$ the binomial coefficient. The derivatives of $\mathbf{K}$ and $\mathbf{f}$ are given through the derivatives of $\Theta_k^{ij}(\mu)$ and $\Theta_k^{f}(\mu)$, see (22).

This result follows directly from applying the general Leibniz rule for the n-th derivative of a product to (21). Note that for $n = 0$ we obtain our discretized PDE (21). We define the residuals for the sensitivities by

$$r^n(\mathbf{u}^\ell;\vartheta,\mu) = \mathbf{f}^n(\vartheta;\mu) - \mathbf{K}(\vartheta;\mu)\mathbf{u}^n - \sum_{k=1}^{n} \binom{n}{k} \mathbf{K}^k(\vartheta;\mu)\mathbf{u}^{n-k}.$$

Then an upper bound for the error $\|\mathbf{u}^n(\vartheta,\mu) - \mathbf{u}^{\ell,n}(\vartheta,\mu)\|_X$ is given by the following theorem:

Theorem 6. *Let $\mathbf{u}^n$ be a solution to (28). Further, let $\mathbf{u}^{\ell,n}$ be the corresponding solution obtained by the reduced order model. Then an upper bound $\Delta\mathbf{u}^n(\vartheta)$ for the error $\|\mathbf{u}^n(\vartheta,\mu) - \mathbf{u}^{\ell,n}(\vartheta,\mu)\|_X$ is given by*

$$\Delta\mathbf{u}^n(\vartheta) := \frac{\|r^n(\mathbf{u}^\ell;\vartheta,\mu)\|_{X'}}{\alpha(\vartheta,\mu)} + \sum_{k=1}^{n} \binom{n}{k} \frac{\gamma^k(\vartheta,\mu)}{\alpha(\vartheta,\mu)} \Delta\mathbf{u}^{n-k}, \tag{29}$$

where $\gamma^k(\vartheta,\mu)$ are continuity constants of $\frac{\partial^k}{\partial\mu^k} a(w,v;\vartheta,\mu)$.

The proof is analogous to the proof of the first order sensitivity presented in [22] and is left to the reader. The key modification is to use the definition for the n-th sensitivity (28). Let us remark at this point that for $k = 0$ we get the error estimator for the variable $\mathbf{u}$ given by (27). Equation (29) is a generalization of the results presented in [22, 23, 24], where first and second order sensitivities for higher dimensional parameters were stated. The continuity constant γ^k can be computed with the same approach as the coercivity constant.

We have only shown the computation of the sensitivity and the error estimator for the variable μ. The steps for the uncertain variable w are analogous. In our case the influence of the uncertain variable can be written as

$$\mathbf{f}(\vartheta;\mu,w) = \sum_{k=1}^{L} h_k(w)\Theta_k^f(\mu)\hat{\mathbf{f}}_k(\vartheta;\mu),$$

where $h_k(w)$ are nonlinear differentiable functions depending on the uncertain parameter. The sensitivity equations and error estimators are analogous to (28) - (29) and are left to the reader.

In the application we generate a POD basis for each equation independently [24]. This means running Algorithm 3 for each sensitivity equation. It is also possible to combine the snapshots and compute one basis for all equations [2]. We opted for the separated approach since we can directly replace the FEM solve by Algorithm 3 in the implementation. This simplifies the modification of existing codes significantly. If an adjoint approach is used to compute the derivatives it can be beneficial to use combined POD bases for the corresponding state and adjoints to guarantee stability [44]. This can be achieved by minor modifications of the proposed algorithm.

5. Numerical results.

In this section we present numerical results utilizing the introduced methods. To start, we will describe the underlying problem in more detail. We start by restating the optimization problem introduced in Section 2 in the discrete form. The nominal optimization problem then reads as

$$\min_{\mu\in\mathbb{R}^3} g_0(\mu) := \mu_1\mu_2 \quad \text{s.t.} \quad g_{1,\dots,7}(\mu) := \begin{pmatrix} 3\mu_1 - 2\mu_3 - 50 \\ \mu_2 + \mu_3 - 15 \\ 1-\mu_1 \\ 1-\mu_2 \\ 5-\mu_3 \\ \mu_3 - 14 \\ \tau^d - \tau(\mathbf{u},\mu) \end{pmatrix} \leq 0,$$

where the desired torque τ^d is computed by the initial configuration, i.e., $\tau^d = \tau(\mathbf{u},\mu^0)$ which in our case is set to $\tau^d = 4.06217$ with $\mu^0 = (19,7,7)^\top$. The constraints g_1 to g_6 are simple design constraints to ensure that we obtain a feasible location of the permanent magnet. We utilize the reduced formulation of the cost and constraints as given in (5), hence we do not have to state the equality constraint. Nevertheless, for the evaluation of the torque $\tau(\mathbf{u},\mu)$ in g_7, for each configuration μ the linear system (23) has to be solved.

Before we go to the discretization details let us have a look at the robust optimization. We will not rewrite the approximated robust counterpart at this point but refer to (18), or (19) where the NCP function is utilized. As the uncertain model parameters w_i, $i=1,\dots,6$, we consider the magnetic field angle of the permanent magnet, see Figure 1. In the ideal case they are aligned perfectly, i.e., the field angle is 90°. This perfect alignment is assumed in the nominal optimization. In practice this can not be met and deviations are to be expected. This can be due to manufacturing imperfection. We introduce the uncertainty set

$$U_{w,2} = \{\, w\in\mathbb{R}^6 \mid w = \hat{w} + \delta_w,\ \|D_w^{-1}\delta_w\|_2 \leq 1 \,\}$$

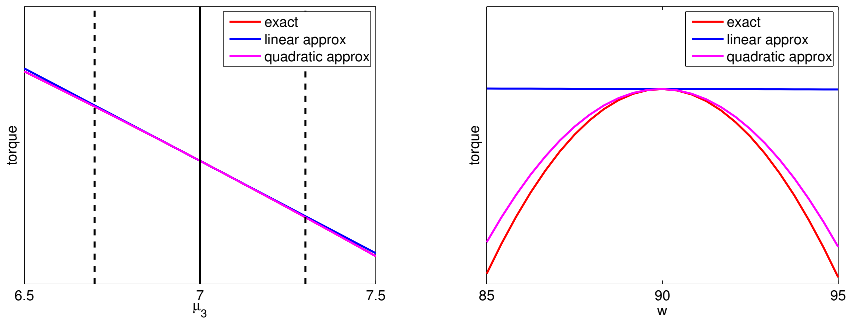

FIG. 2. *First and second order approximation of the torque with respect to the optimization variable μ_3 (left) and model parameter w (right).*

with $\hat{w} = [90°, 90°, 90°, 90°, 90°, 90°]^\top$ and $D_w = \mathrm{diag}([5°, 5°, 5°, 5°, 5°, 5°]^\top)$. The model parameter w enters (23) by the right hand side $\mathbf{f}$ in a nonlinear manner through trigonometric functions. For updating the expansion point a tolerance of $\epsilon_{mov} = 10^{-3}$ is used in Algorithm 1. Next, let us define the uncertainty in the optimization variable. Here, we try to account for production inaccuracies. The uncertainty set under consideration is

$$U_{\mu,\infty} = \{\, \delta_\mu \in \mathbb{R}^3 \mid \|D_\mu^{-1}\delta_\mu\|_\infty \le 1 \,\}$$

with $D_\mu = \mathrm{diag}([0.3, 0.3, 0.3]^\top)$. As already previously outlined we will consider a mixed approach in the approximated robust counterpart, i.e., linear approximation in the optimization variable μ and quadratic approximation in the model parameter w. To illustrate and justify our particular choice we look at the first and second order approximation of g_7, where μ and w enter nonlinearly. It is sufficient to look at the torque $\tau(\mathbf{u}, \mu)$ which is the main component. In Figure 2 the approximation quality is shown. It can be seen that for the approximation in the variable μ (left figure) the linear approximation is very accurate, especially when looking at the interval of interest indicated by the dashed lines. The reason is that the approximation is updated for every μ in the interval, where the approximation is considered is small. We only show the behaviour for μ_3 since it is the most prominent. For the model parameter w we get a different picture (right figure). Since it is hard to display the six dimensional case, for this demonstration we assume that w is a scalar parameter. Due to the symmetric nature of problem, the linearization at the nominal parameter $\hat{w}$ provides a very bad approximation. The robust optimization utilizing a linear approximation would lead to the same results as the nominal optimization. On the other hand the quadratic approximation provides a very good approximation. Further, by shifting the expansion point, one will obtain an even better approximation during the optimization.

The introduced parametrization is realized in a box around the permanent magnet, see Figure 3 (left) red dashed lines. In blue lines the decomposition into the triangular subdomains is indicated. For each pole we obtain 14 triangular domains. The constraints $g_{2,\ldots,6}$ in the optimization problem guarantee that we stay within the red dashed box and hence always have a feasible problem.

Let us next turn to the discretization. For the finite element discretization we use the standard linear ansatz functions. We use a mesh with 42061 nodes. After removing

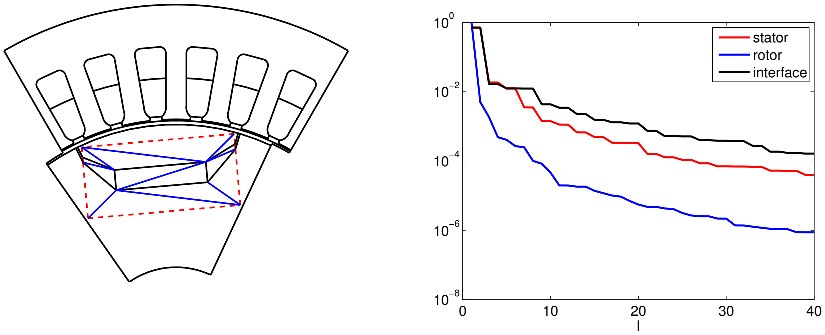

Fig. 3. *(left) Geometry of the PMSM (one pole) with details of the affine decomposition. The red dashed lines indicate the region where the affine decomposition is performed and the blue lines indicate the affine decomposition into the triangular subdomains. (right) Decay of the normalized eigenvalues for the stator, rotor and interface .*

points associated to the Dirichlet boundary and adding them to the right hand side we end up with a system of $N = 41880$ unknowns. These are then partitioned into stator and rotor with $N_s = 30320$ and $N_r = 10660$ degrees of freedom, respectively. The interface is discretized by $N_I = 900$ equidistant points. Hence for a full rotation, the linear system (23) of dimension 41880 has to be solved 900 times giving us an overall number of unknowns of 37692000. After introducing the finite element discretization let us give some details on the model order reduction. The decay of the eigenvalues for the stator, rotor and interface is shown in Figure 3 (right). In particular it can be seen that the decay of the eigenvalues associated with the interface is very slow which motivated the proposed strategy. Using Algorithm 3 we generate POD bases for the stator and rotor independently. In Figure 4 the first three POD basis vectors are shown for the stator and rotor. Also here it can be observed that the third basis vector for the rotor contributes very little information since it is mostly zero (green) with some contributions on the boundary to the interface. As a tolerance for the error estimator we use $\epsilon = 10^{-2}$. Since we use an adaptive strategy we can not give the exact number for the dimension of the ROM. It was observed that the ROM for the stator is of dimension $\ell_s \leq 20$ and for the rotor $\ell_r \leq 10$. Hence the maximum dimension of the ROM is $\ell_s + \ell_r + N_I = 930$ for each position of the rotation. Overall we get 837000 unknowns in the ROM. It can be seen that this is a significant reduction compared to the original setting. This reduction not only reduces computational cost but also the storage cost. The implementation was carried out in `MATLAB` and performed on a desktop PC in single thread mode.

To compare the effectiveness of the introduced methods we will perform the numerical test utilizing the FEM and ROM. Then the results can be directly compared and the efficiency of the proposed strategy can be interpreted. The SQP algorithm is stopped when optimality is reached with a relative tolerance of 10^{-6}. In Table 1 and Table 2 we state the results obtained by FEM and ROM, respectively. We provide the quantities volume V, the optimal parameter μ, the torque τ and the torque in the worst case τ^{wc} for the initial configuration 'Init.', the nominal optimization 'Nom.', the robust optimization 'Rob.' and the robust optimization using the moving expansion point strategy 'Rob. mov.'. Looking at the obtained results it can be seen that the nominal optimization manages to reduce the volume of the permanent magnet

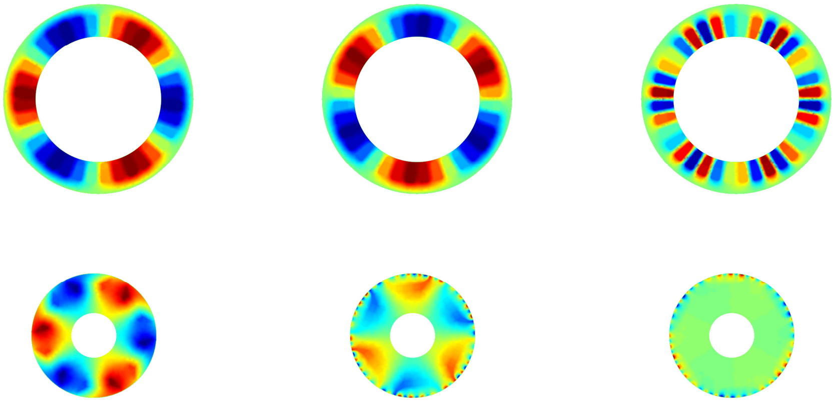

FIG. 4. *First three POD basis vectors for the stator (top) and rotor (bottom).*

TABLE 1
Numerical results for the nominal and robust optimization using the FEM.

	V	μ	τ	τ^{wc}
Init.	133.00	(19.0000, 7.0000, 7.0000)	4.062170	3.864719
Nom.	62.84	(21.0735, 2.9820, 6.6102)	4.062170	3.804459
Rob.	101.14	(20.7582, 4.8722, 6.9023)	4.279193	4.057369
Rob. mov.	102.29	(20.7538, 4.9287, 6.8807)	4.286731	4.064957

significantly, by more than 50%, which is very desirable. On the other hand we can also see that in the case of inaccuracy in manufacturing or material the performance drops well below the desired torque τ^d given by the initial configuration. Even worse, the solution obtained by the nominal optimization is outperformed by the initial configuration. This is a very undesirable result. The robust optimization on the other hand return a result that is robust with respect to the uncertainty. The price to pay in this case is that the volume of the permanent magnet is not reduced as much as in the nominal optimization. Still a reduction of more than 20% is achieved. It can be seen that the robust optimization utilizing the moving expansion point strategy provides a better approximation and hence performs better and the worst case value is within the prescribed tolerance. Comparing the results obtained by FEM and ROM it can be seen that the difference is very small. Considering that a tolerance of 10^{-2} is used in the basis generation of the ROM, the obtained results are very satisfying.

It is a natural question whether there exists an instance for a fixed parameter $w \in U_{w,2}$ that realizes the objective function value of the robust counterpart. For the case of uncertain linear programs with constraint-wise uncertainty it was shown in [5] that under a boundedness assumption on the feasible sets there exists a worst case instance that yields the optimal value of the robust counterpart. For the nonlinear nonconvex case such a result can be expected only in particular cases.

To illustrate that in our case the optimal value of the robust counterpart is in fact attained by an instance, we proceed as follows. For the optimal solution obtained by the robust optimization with adaptive expansion point (see Rob. mov. in Table 1) we compute the worst case w^{WC} corresponding to inequality constraint g_7 (the only

Table 2
Numerical results for the nominal and robust optimization using the ROM.

	V	μ	τ	τ^{wc}
Init.	133.00	(19.0000, 7.0000, 7.0000)	4.062170	3.940564
Nom.	62.86	(21.0730, 2.9829, 6.6105)	4.062271	3.804873
Rob.	101.09	(20.7580, 4.8699, 6.9042)	4.278691	4.056902
Rob. mov.	102.26	(20.7532, 4.9275, 6.8823)	4.286302	4.064836

constraint that depends on w).

Then we fix the uncertain parameter w to w^{WC} and perform a robust optimization for the design μ. Table 3 shows the result. We see that the optimal design μ is very close to the robust solution in Table 1. As a consequence, also the torque τ corresponding to the optimal solution μ and the nominal parameter $\hat{w}$ as well as the worst case value τ^{wc} are very close to the values in Table 1.

Table 3
Solution of the robust optimization problem with respect to μ for fixed parameter $w = w^{wc}$.

	V	μ	τ	τ^{wc}
Rob., $w = w^{wc}$	101.58	(20.7532, 4.8947, 6.8798)	4.283122	4.061077

Let us now have a look at the computational time required for the optimization. The performance is summarized in Table 4, where CPU time in seconds (wall clock) and number of iterations ‘iter.’ are given. The computation time includes everything except the mesh generation and the computation of the affine decomposition. It can be seen that there is a huge increase in computational time for the robust optimization. This is mainly due to the fact that a considerable number of PDEs have to be solved. In the nominal case we require one PDE solve for the magnetic vector potential $\mathbf{u}$ and three additional PDE solves for the sensitivities with respect to the variable μ for the gradient computation. In the robust case on the other hand, we require additional 54 sensitivities, i.e., PDE solves. This is due to the formulation we chose. If a formulation utilizing the adjoint method [20, 57] is used a reduction can be expected. Further, the computational cost can be reduced by exploiting parallel structures in modern hardware which are neglected in this work. Comparing the performance of the FEM and ROM we can observe a significant speed up. For both the nominal and robust optimization a factor of more than 15 is achieved. When putting this in relation with the number of iterations we see a significant speed up. The optimization using then ROM is then done within one iteration of the optimization using FEM. Further, the robust optimization using ROM is significantly faster than the nominal optimization using FEM. These are very satisfying results and underline

Table 4
Performance comparison of FEM and ROM in the optimization.

	FEM		ROM		
	iter.	CPU time	iter.	CPU time	Factor
Nom.	14	41928	13	2508	16.72
Rob.	9	300820	7	15385	19.55
Rob. mov.	9	304875	7	14885	20.48

that the proposed strategy allows for a very efficient realization of robust optimization problems with PDE constraints.

6. Conclusion.

In this work we have developed a robust optimization framework utilizing quadratic approximation of the worst case function. We combined the new approach with existing techniques utilizing linear approximations. In order to apply the strategies to PDE constrained optimization problems efficiently, we introduced a modified POD strategy which utilizes a greedy type approach to generate reduced order models. An adaptive strategy which utilizes a-posteriori error estimators is proposed. The techniques are then applied to an optimal control problem governed by a parametrized elliptic PDE with application to PMSM. It turns out that the approximation techniques for the robust optimization combined with the introduced model order reduction present a very attractive framework to solve this type of problems.

REFERENCES

[1] F. Alizadeh and D. Goldfarb, *Second-order cone programming*, Mathematical Programming, 95 (2003), pp. 3–51.

[2] A. Alla, M. Hinzeand, O. Lass, and S. Ulbrich, *Model order reduction approach for the optimal design of permanent magnets in electro-magnetic machine*, IFAC-PapersOnLin, 43 (2015), pp. 242–247.

[3] H. Antil, M. Heinkenschloss, and R. H. W. Hoppe, *Domain decomposition and balanced truncation model reduction for shape optimization of the stokes system*, Optimization Methods and Software, 26 (2011), pp. 643–669.

[4] A. Ben-Tal, A. Goryashko, and A. Nemirovski, *Robust optimization*, Princton University Press, 2009.

[5] A. Ben-Tal and A. Nemirovski, *Robust solutions of uncertain linear programs*, Operations Research Letters, 25 (1999), pp. 1–13.

[6] A. Ben-Tal and A. Nemirovski, *Robust solution of linear programming problems contaminated with uncertain data*, Mathematical Programming, 88 (2000), pp. 411–421.

[7] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MPS-SIAM Series on Optimization, MPS-SIAM, 2002.

[8] A. Ben-Tal and A. Nemirovski, *Robust optimization – methodology and application*, Mathematical Programming, 92 (2002), pp. 453–480.

[9] A. Ben-Tal and A. Nemirovski, *Selected topics in robust convex optimization*, Mathematical Programming, 112 (2008), pp. 125–158.

[10] D. Bertsimas, D. B. Brown, and C. Caramanis, *Theory and application of robust optimization*, SIAM Review, 53 (2011), pp. 464–501.

[11] K. Binnemans, P. T. Jones, B. Blanpain, T. Van Gerven, Y. Yang, A. Walton, and M. Buchert, *Recycling of rare earths: a critical review*, Journal on Cleaner Production, 51 (2013), pp. 1–22.

[12] J. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer, 1997.

[13] Z. Bontinck, O. Lass, H. De Gersem, and S. Schöps, *Uncertainty quantification of a PMSM with dynamic rotor eccentricity and shape optimization of its magnet*, in Proceedings of EPNC 2016, XXIV Symposium on Electromagnetic Phenomena in Nonlinear Circuits, Helsinki, Finland, 2016.

[14] D. Braess, *Finite Elements*, Cambridge University Press, 3rd ed., 2007.

[15] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, vol. 15 of Texts in Applied Mathematics, Springer, 2008.

[16] A. Chatterjee, *An introduction to the proper orthogonal decomposition*, Current Science, 78 (2000), pp. 539–575.

[17] Y. Chen and M. Florian, *The nonlinear bilevel programming problem: Formulations, regularity and optimality condition*, Optimization, 32 (1995), pp. 193–209.

[18] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, Society for Industrial and Applies Mathematics, 2000.

[19] L. Dede, *Reduced basis method and a posteriori error estimation for parametrized linear-quadratic optimal control problems*, SIAM Journal on Scietific Computing, 32 (2010),

pp. 997–1019.

[20] M. Diehl, H. G. Bock, and E. Kostina, *An approximation technique for robust nonlinear optimization*, Mathematical Programming, 107 (2006), pp. 213–230.

[21] M. Diehl, J. Gerhard, W. Marquardt, and M. Mönnigmann, *Numerical solution approaches for robust nonlinear optimal control problems*, Computer & Chemical Engineering, 32 (2008), pp. 1287–1300.

[22] M. Dihlmann, *Adaptive Reduced Basis Methods for Parameterized Evolution Problems with Application in Optimization and State Estimation*, PhD thesis, Universit ät Stuttgart, 2014.

[23] M. Dihlmann and B. Haasdonk, *Certified nonlinear parameter optimization with reduced basis surrogate models*, Proceedings in Applied Mathematics and Mechanics, 13 (2013), pp. 3–6.

[24] M. Dihlmann and B. Haasdonk, *Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems*, Computational Optimization and Applications, 60 (2015), pp. 753–787.

[25] A. Fischer, *A newton-type method for positive semi-definite linear complementarity problems*, Journal of Optimization Theory and Applications, 86 (1995), pp. 585–608.

[26] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Mathematical Programming, 91 (2002), pp. 239–270.

[27] R. Fletcher and S. Leyffer, *Solving mathematical program with complementarity constraints as nonlinear programs*, Optimization Methods and Software, 19 (2004), pp. 15–40.

[28] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes, *Local convergance of SQP methods for mathematical programs with equilibrium constraints*, SIAM Journal on Optimization, 17 (2006), pp. 259–286.

[29] M. Gubisch and S. Volkwein, *Proper orthogonal decomposition for linear-quadratic optimal control*, in Model Reduction and Approximation: Theory and Algorithms, P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, eds., SIAM, 2016.

[30] J. Haslinger and R. A. E. Mäkinen, *Introduction to shape optimization: theory, approxiamtion, and computation*, SIAM, 2003.

[31] M. Heinkenschloss, D. C. Sorensen, and K. Sun, *Balanced truncation model reduction for a class of descriptor systems with application to the Oseen eqaution*, SIAM Journal on Scientific Computing, 30 (2008), pp. 1038–1063.

[32] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, vol. 23 of Mathematical Modelling: Theory and Applications, Springer, 2009.

[33] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge Monographs on Mechanics, Cambridge University Press, 2nd ed., 2012.

[34] B. Houska and M. Diehl, *Nonlinear robust optimization via sequential convex bilevel programming*, Mathematical Programming, 142 (2013), pp. 539–577.

[35] H. Jiang and D. Ralph, *Smooth SQP methods for mathematical programs with nonlinear complementarity constraints*, SIAM Journal on Optimization, 10 (2000), pp. 779 – 808.

[36] C. Kanzow, I. Ferenczi, and M. Fukushima, *On the local convergence of semismooth newton methods for linear and nonlinear second-order cone programs without strict complementarity*, SIAM Journal on Optimization, 20 (2009), pp. 297–320.

[37] D. P. Kouri and T. M. Surowiec, *Risk-averse PDE-constrained optimization using the conditional value-at-risk*, SIAM Journal on Optimization, 26 (2016), pp. 365–396.

[38] K. Kunisch and S. Volkwein, *Galerkin proper orthogonal decomposition methods for parabolic problems*, Numerische Mathematik, 90 (2001), pp. 117–148.

[39] O. Lass and S. Volkwein, *Parameter identification for nonlinear elliptic-parabolic systems with application in lithium-ion battery modeling*, Computational Optimization and Applications, 62 (2015), pp. 217–239.

[40] T. Lehnhäuser and M. Schfer, *A numerical approach for shape optimization of fluid flow domains*, Computer Methods in Applied Mechanics and Engineering, 194 (2005), pp. 5221–5241.

[41] S. Leyffer, *Complementarity constraints as nonlinear equations: Theory and numerical experience*, in Optimization with Multivalued Mappings: Theory, Applications, and Algorithms, S. Dempe and V. Kalashnikov, eds., vol. 2 of Springer Series in Optimization and Its Applications, Springer, 2006, pp. 169–208.

[42] S. Leyffer, G. Løpez-Calva, and J. Nocedal, *Interior methods for mathematical programs with complementarity constraints*, SIAM Journal on Optimization, 17 (2006), pp. 52–77.

[43] M. Liu, X. Li, and D. Pu, *A feasible flter SQP algorithm with global and local convergence*, Journal of Applied Mathematics and Computing, 40 (2012), pp. 261–275.

[44] F. Negri, G. Rozza, A. Manzoni, and A. Quateroni, *Reduced basis method for parametrized elliptic optimal control problems*, SIAM Journal on Scientific Computing, 35 (2013), p. A2316A2340.

[45] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operation Research and Financial Engineering, Springer, 2ed ed., 2006.

[46] B. Øksendal, *Optimal control of stochastic partial differential equations*, Stochastic Analysis and Applications, 23 (2005), pp. 165–179.

[47] U. Pahner, *A general design tool for terical optimization of electromagnetic energy transducers*, PhD thesis, KU Leuven, 1998.

[48] A. T. Patera and G. Rozza, *Reduced Basis Approximation and A Posteriori Error Estimator for Parameterrametrized Partial Differential Equations*, MIT Pappalardo Graduate Monographs in Mechanical Engineering, MIT, 2006.

[49] M. J. D. Powell, *The BOBYQA algorithm for bound constrained optimization without derivatives*, technical report NA2009/06, Department of Applied Mathematics and Theoretical Physics, Cambridge, England, 2009.

[50] T. W. Preston, A. B. J. Reece, and P. S. Sangha, *Inuction motor analysis by time-stepping techniques*, IEEE Transactions on Magnetics, 24 (1988), pp. 471–474.

[51] A. U. Raghunathan and L. T. Biegler, *An interior point method for mathematical programs with complementarity constraints (mpccs)*, SIAM Journal on Optimization, 15 (2005), pp. 720–750.

[52] G. Rozza, D. B. P. Huynh, and A. T. Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations*, Archives of Computational Methods in Engineering, 15 (2008), pp. 229–275.

[53] J. A. Samareh, *A survey of shape parametrization techniques*, Tech. Report NASA/CP-1999-2009136, NASA, 1999.

[54] T. W. Sederberg and S. R. Parry, *Free-form deformation of solid geometric models*, Computer Graphics, 20 (1986), pp. 151–160.

[55] X. Shi, Y. Le Menach, J. P. Ducreux, and F. Piriou, *Comparison of slip surface and moving band techniques for modelling movement in 3D with FEM*, The international journal for computation and mathematics in electrical and electronic engineering, 25 (2006), pp. 17–30.

[56] A. Shpiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*, SIAM, 2009.

[57] A. Sichau, *Robust Nonlinear Programming with Discretized PDE Constraints using Second-order Approximations*, PhD thesis, Technische Universität Darmstadt, 2013.

[58] B. Silwal, P. Rasilo, L. Perkkiö, A. Hannukainen, T. Eirola, and A. Arkkio, *Numerical analysis of the power balance of an electrical machine with rotor eccentricity*, IEEE Transactions on Magnetics, 52 (2016), pp. 1–4.

[59] B. Silwal, P. Rasilo, L. Perkkiö, M. Oksman, A. Hannukainen, T. Eirola, and A. Arkkio, *Computation of torque of an electrical machine with different types of finite element mesh in the air gap*, IEEE Transactions on Magnetics, 50 (2014), pp. 1–9.

[60] O. Stein, *How to solve a semi-infinite optimization problem*, European Journal of Operational Research, 223 (2012), pp. 312–320.

[61] H. Tiesler, R. M. Kirby, D. Xiu, and T. Preusser, *Stochastic collocation for optimal control problems with stochastic pde constraints*, SIAM Journal on Control and Optimization, 50 (2012), pp. 2659–2682.

[62] A. Toselli and O. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Springer Series in Computational Mathematics, Springer, 2005.

[63] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Application*, American Mathematical Society, 2010.

[64] M. J. Zahr and C. Farhat, *Progressive construction of a parametric reduced-order model for PDE-constrained optimization*, International Journal for Numerical Methods in Engineering, 102 (2015), pp. 1111–1135.

[65] Y. Zhang, *General robust-optimization formulation for nonlinear programming*, Journal of Optimization Theory and Applications, 132 (2007), pp. 111–124.