Proceedings of ALGORITMY 2024
pp. 66–75

APPLICATION OF AN ADAPTIVE MODEL HIERARCHY TO PARAMETRIZED OPTIMAL CONTROL PROBLEMS*

HENDRIK KLEIKAMP †

Abstract. In this contribution we apply an adaptive model hierarchy, consisting of a full-order model, a reduced basis reduced order model, and a machine learning surrogate, to parametrized linear-quadratic optimal control problems. The involved reduced order models are constructed adaptively and are called in such a way that the model hierarchy returns an approximate solution of given accuracy for every parameter value. At the same time, the fastest model of the hierarchy is evaluated whenever possible and slower models are only queried if the faster ones are not sufficiently accurate. The performance of the model hierarchy is studied for a parametrized heat equation example with boundary value control.

Key words. Parametrized optimal control problems, adaptive model hierarchy, reduced order models, machine learning, a posteriori error estimation

AMS subject classifications. 49N10, 46E22, 65M06

1. Introduction. Optimal control problems with parameter-dependent system components typically require an enormous computational effort when considered in a multi-query or real time scenario. Solving these kinds of problems exactly for many different values of the parameter is computationally demanding and often prohibitively costly. In [10], a greedy procedure to construct a reduced order model for parametrized optimal control problems has been proposed. To further speed up the online computations of the reduced model, machine learning algorithms have been used in a certified manner in [9]. In this work, we combine the aforementioned ideas with an adaptive and certified model hierarchy for parametrized problems which was introduced in [6] and further applied in [12]. This model hierarchy allows for an adaptive construction and improvement of reduced order models and machine learning surrogates while already querying the model hierarchy for different parameter values. Hence, no costly offline phase is required, while the results provided by the model hierarchy still fulfill a prescribed error tolerance. The parametrized optimal control problem is only solved exactly using the underlying full-order model if necessary. Whenever possible, cheaper reduced order models are used, which are built, trained and improved on the fly using data from more accurate but at the same time more costly models.

The adaptive model hierarchy for optimal control problems introduced in this contribution could for instance be applied in conjunction with Monte Carlo estimation of derived quantities. In addition, parameter optimization problems with optimal control problems as constraint could be another possible field of usage.

Applications of reduced order models to optimal control problems can for instance be found in [1, 3, 10]. A combination with tools from machine learning has been proposed in [2]. In [7], an approach using deep neural networks for the solution of

*Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2044 –390685587, Mathematics Münster: Dynamics–Geometry–Structure.

†Institute for Analysis and Numerics, Mathematics Münster, University of Münster, Einsteinstrasse 62, 48149 Münster, Germany (hendrik.kleikamp@uni-muenster.de).

parametrized partial differential equations combined with reduced basis reduced order models constructed via proper orthogonal decomposition has been discussed.

The paper is organized as follows: In Section 2, we introduce the problem considered in this contribution and present the associated optimality system. Afterwards, in Section 3, the two reduced order models and an a posteriori error estimator will be discussed. Section 4 presents the adaptive model hierarchy in a general formulation which is then applied to the parametrized optimal control setting in Section 5. A numerical example showing the performance of the devised algorithms is performed and evaluated in Section 6. The paper ends in Section 7 with some concluding remarks and an outlook to future research directions.

2. Linear-quadratic parametrized optimal control problems. First, we introduce the parametrized optimal control problems considered in this work. For simplicity, we state the optimal control problem in a finite-dimensional setting. The more general formulation for infinite-dimensional parameter, state and control spaces can be found in [9]. Afterwards, the optimality system using an adjoint variable and a linear system of equations for the optimal adjoint at final time are presented.

2.1. Problem formulation. Let $\mathcal{P} \subset \mathbb{R}^p$ be a compact parameter set for some $p \in \mathbb{N}$. For a parameter $\mu \in \mathcal{P}$, the state system is given as

$$\begin{aligned} \dot{x}_\mu(t) &= A_\mu x_\mu(t) + B_\mu u(t), \qquad t \in [0,T], \\ x_\mu(0) &= x_\mu^0, \end{aligned} \tag{2.1}$$

where $x_\mu \colon [0,T] \to \mathbb{R}^n$ is the state trajectory, $u \colon [0,T] \to \mathbb{R}^m$ denotes the control, $A_\mu \in \mathbb{R}^{n\times n}$ is the state operator, $B_\mu \in \mathbb{R}^{n\times m}$ the control operator, $T > 0$ is the final time and $x_\mu^0 \in \mathbb{R}^n$ denotes the initial state. In the examples we have in mind, for instance discretizations of time-dependent partial differential equations (PDEs), the dimension $n \in \mathbb{N}$ of the state space is typically quite large whereas the number of controls $m \in \mathbb{N}$ is of moderate size. For each parameter $\mu \in \mathcal{P}$, we aim to steer the system state $x_\mu(T)$ at time T close to a given target state $x_\mu^T \in \mathbb{R}^n$ while not spending too much control energy. We hence aim to minimize the following functional $\mathcal{J}_\mu$ defined for a control $u \colon [0,T] \to \mathbb{R}^m$ as

$$\mathcal{J}_\mu(u) \coloneqq \frac{1}{2}\left[(x_\mu(T) - x_\mu^T)^\top M (x_\mu(T) - x_\mu^T) + \int_0^T u(t)^\top R u(t) \,\mathrm{d}t \right],$$

where $x_\mu \colon [0,T] \to \mathbb{R}^n$ solves the state equation (2.1) for the control u, the matrix $M \in \mathbb{R}^{n\times n}$ is symmetric and positive-semidefinite and the matrix $R \in \mathbb{R}^{m\times m}$ is symmetric and positive-definite. The matrices M and R allow for individual weights for different state and control components. To summarize, given a parameter $\mu \in \mathcal{P}$, we would like to solve the optimal control problem

$$\min_u \mathcal{J}_\mu(u), \quad \text{s.t. } \dot{x}_\mu(t) = A_\mu x_\mu(t) + B_\mu u(t) \text{ for } t \in [0,T], \quad x_\mu(0) = x_\mu^0. \tag{2.2}$$

2.2. Optimality system. The following optimality system characterizes the optimal state trajectory $x_\mu^* \colon [0,T] \to \mathbb{R}^n$, the optimal control $u_\mu^* \colon [0,T] \to \mathbb{R}^m$ and the optimal adjoint trajectory $\varphi_\mu^* \colon [0,T] \to \mathbb{R}^n$ (see [9, Theorem 2.4] for more details)

that solve (2.2):

$$\begin{aligned} \dot{x}_\mu^*(t) &= A_\mu x_\mu^*(t) + B_\mu u_\mu^*(t), \\ -\dot{\varphi}_\mu^*(t) &= A_\mu^\top \varphi_\mu^*(t), \\ u_\mu^*(t) &= -R^{-1} B_\mu^\top \varphi_\mu^*(t), \end{aligned} \tag{2.3a}$$

for $t \in [0, T]$ with initial respectively terminal conditions

$$x_\mu^*(0) = x_\mu^0, \qquad \varphi_\mu^*(T) = M\left(x_\mu^*(T) - x_\mu^T\right). \tag{2.3b}$$

Using the exponential function, we define the *weighted controllability Gramian* $\Lambda_\mu^R \in \mathbb{R}^{n\times n}$ as

$$\Lambda_\mu^R \coloneqq \int_0^T e^{A_\mu(T-s)} B_\mu R^{-1} B_\mu^\top e^{A_\mu^\top (T-s)} \,\mathrm{d}s.$$

With this definition, the optimal state at final time $x_\mu^*(T)$ can be expressed as

$$x_\mu^*(T) = e^{A_\mu T} x_\mu^0 - \int_0^T e^{A_\mu(T-s)} B_\mu R^{-1} B_\mu^\top e^{A_\mu^\top (T-s)} \dot{s} \cdot \varphi_\mu^*(T) \,\mathrm{d}s = e^{A_\mu T} x_\mu^0 - \Lambda_\mu^R \varphi_\mu^*(T).$$

Taking into account the terminal condition of the optimality system in (2.3b), we have

$$\varphi_\mu^*(T) = M\left(x_\mu^*(T) - x_\mu^T\right) = M\left(e^{A_\mu T} x_\mu^0 - \Lambda_\mu^R \varphi_\mu^*(T) - x_\mu^T\right).$$

By rearranging this equation we thus obtain a linear system for the optimal final time adjoint $\varphi_\mu^*(T)$ (see [9, Lemma 2.5]), which is given as

$$\left(I + M\Lambda_\mu^R\right) \varphi_\mu^*(T) = M\left(e^{A_\mu T} x_\mu^0 - x_\mu^T\right). \tag{2.4}$$

The solution of the optimality system (2.3) is already uniquely determined by the optimal final time adjoint $\varphi_\mu^*(T)$ (in the following we assume that the product $M\Lambda_\mu^R$ is positive-semidefinite for all parameters $\mu \in \mathcal{P}$). It is therefore sufficient to first solve the linear system in (2.4) for $\varphi_\mu^*(T)$ and afterwards solve the ordinary differential equation system in (2.3a) to obtain the optimal control, state and adjoint trajectories. The linear system can be solved using iterative methods since applying the Gramian matrix Λ_μ^R to a vector $p \in \mathbb{R}^n$ is (up to a minus sign) equivalent to solving the optimality system (2.3) for terminal condition $\varphi_\mu(T) = p$ and initial condition $x_\mu(0) = 0$. In the following, solving the linear system in (2.4) and the optimality system in (2.3a) exactly will be referred to as solving the full-order model (FOM). The main idea of the reduced order model introduced in the next section is to approximate the final time adjoint $\varphi_\mu^*(T)$ by an element from a low-dimensional subspace of $\mathbb{R}^n$.

3. Reduced order models for parametrized optimal control problems. In this section we first introduce a reduced order model (ROM) based on a reduced basis approximation of the manifold $\mathcal{M} \coloneqq \{\varphi_\mu^*(T) : \mu \in \mathcal{P}\}$ of optimal final time adjoint states. Afterwards, we show how to further accelerate the online phase by applying machine learning algorithms. We finally discuss an a posteriori error estimator for both reduced models.

3.1. Reduced basis ROM. Assume that we are given a reduced basis $\Phi^N = \{\varphi_1, \dots, \varphi_N\}$ for some $\varphi_1, \dots, \varphi_N \in \mathbb{R}^n$ and the respective reduced subspace $X^N = \operatorname{span}(\Phi^N) \subset \mathbb{R}^n$. In the adaptive model hierarchy described below, the reduced basis is built iteratively by starting with an empty basis and adding optimal final time adjoint states for certain parameters. A greedy procedure (see [4] for the theoretical background) to determine a reduced basis for the optimal control problem in (2.2) was discussed in [9, Section 3]. Given the reduced space X^N and a parameter $\mu \in \mathcal{P}$, we compute the approximate final time adjoint $\tilde{\varphi}_\mu^N \in X^N$ as

$$\tilde{\varphi}_\mu^N \coloneqq \operatorname*{arg\,min}_{\tilde{\varphi} \in X^N} \left\| M \left(e^{A_\mu T} x_\mu^0 - x_\mu^T \right) - \left(I + M \Lambda_\mu^R \right) \tilde{\varphi} \right\|. \tag{3.1}$$

In other words, we choose $\tilde{\varphi}_\mu^N$ such that

$$\left(I + M \Lambda_\mu^R \right) \tilde{\varphi}_\mu^N = P_{Y_\mu^N} \Big((I + M \Lambda_\mu^R) \varphi_\mu^*(T) \Big),$$

where $P_{Y_\mu^N}$ denotes the orthogonal projection onto the space $Y_\mu^N \coloneqq \left(I + M \Lambda_\mu^R \right) X^N$. This choice is motivated by the least squares solution of the linear system (2.4) over the space X^N.

To compute the approximation $\tilde{\varphi}_\mu^N \approx \varphi_\mu^*(T)$ in practice, one first computes the states $x_i^\mu = (I + M \Lambda_\mu^R) \varphi_i$ for $i = 1, \dots, N$ (which essentially means to solve the optimality system in (2.3)). Afterwards, the matrix $\bar{X}_\mu = [x_1^\mu \cdots x_N^\mu] \in \mathbb{R}^{n \times N}$ can be assembled and the coefficients $\alpha^\mu = (\alpha_1^\mu, \dots, \alpha_N^\mu)^\top \in \mathbb{R}^N$ are derived as solutions of the linear system

$$\bar{X}_\mu^\top \bar{X}_\mu \alpha^\mu = \bar{X}_\mu^\top M \left(e^{A_\mu T} x_\mu^0 - x_\mu^T \right).$$

Having the coefficients at hand, the approximate final time adjoint is given as

$$\tilde{\varphi}_\mu^N = \sum_{i=1}^N \alpha_i^\mu \varphi_i.$$

The corresponding approximate optimal control $\tilde{u}_\mu^N$ is computed by solving the optimality system in (2.3). The reduced order model described in this section will be called reduced basis ROM (RB-ROM) in the following.

3.2. Machine learning ROM. The RB-ROM introduced in the previous subsection still involves several steps whose computational effort depends on the dimension n of the state space. In particular, computing $x_i^\mu \in \mathbb{R}^n$ for $i = 1, \dots, N$ corresponds to solving the adjoint and the state equation N times for every new parameter $\mu \in \mathcal{P}$. However, these computations are solely required to solve for the coefficients $\alpha^\mu \in \mathbb{R}^N$ with respect to the reduced basis. In the machine learning ROM (ML-ROM) proposed in [9], instead of solving a linear system of equations for the coefficients, the map $\pi_N \colon \mathcal{P} \to \mathbb{R}^N$, $\pi_N(\mu) \coloneqq \alpha^\mu$, from parameter to coefficients is approximated using machine learning algorithms. This idea is motivated by the approach first introduced in [7]. Given an approximation $\hat{\pi}_N \colon \mathcal{P} \to \mathbb{R}^N$ of π_N, the machine learning approximation $\hat{\varphi}_\mu^N \in \mathbb{R}^n$ of the optimal final time adjoint is defined as

$$\hat{\varphi}_\mu^N \coloneqq \sum_{i=1}^N [\hat{\pi}_N(\mu)]_i \varphi_i.$$

Similar to the RB-ROM in the previous section, the approximate optimal control $\hat{u}_\mu^N$ is derived according to the optimality system from Section 2.2.

The machine learning surrogate is trained in a supervised manner, i.e. by means of training data consisting of parameters and corresponding coefficients. Several different machine learning algorithms are applicable in this scenario, the only requirement is that vector-valued functions can be approximated using training data (see [9, Section 4]). In our numerical experiment below we apply kernel methods (as discussed in more detail in [9, Section 4.3.2]) and in particular the *vectorial kernel orthogonal greedy algorithm* (VKOGA) as introduced in [11].

3.3. Residual based a posteriori error estimation. To estimate the error of the RB-ROM and the ML-ROM in an a posteriori manner, we consider the norm of the residual of the linear system in (2.4). To be more precise, given a parameter $\mu \in \mathcal{P}$ and an approximate final time adjoint $p \in \mathbb{R}^n$, the error estimate $\eta_\mu(p)$ is defined as

$$\eta_\mu(p) \coloneqq \left\| M \left(e^{A_\mu T} x_\mu^0 - x_\mu^T \right) - (I + M\Lambda_\mu^R) p \right\|. \tag{3.2}$$

The reduced solution $\tilde{\varphi}_\mu^N$ is thus, by combining (3.1) and (3.2), defined as the element from the reduced space X^N that minimizes the error estimator η_μ, i.e. it holds $\tilde{\varphi}_\mu^N = \arg\min_{\tilde{\varphi} \in X^N} \eta_\mu(\tilde{\varphi})$. As proven in [9, Theorem 3.1], it is possible to show that η_μ is an efficient and reliable error estimator for the true error, i.e. it holds

$$\left\| \varphi_\mu^*(T) - p \right\| \quad \leq \quad \eta_\mu(p) \quad \leq \quad \left\| I + M\Lambda_\mu^R \right\| \left\| \varphi_\mu^*(T) - p \right\|. \tag{3.3}$$

This error estimator can be applied to both, the solution $\tilde{\varphi}_\mu^N$ of the RB-ROM and the solution $\hat{\varphi}_\mu^N$ of the ML-ROM. Evaluating the error estimator for some $p \in \mathbb{R}^n$ requires solving the adjoint equation once backwards in time with terminal condition p, computing the corresponding control, and finally solving the state equation forward in time with zero initial condition for the state.

4. Adaptive model hierarchy. In [6], an adaptive model hierarchy for parametrized PDEs was presented. The model hierarchy consists of a FOM, an RB-ROM and an ML-ROM. Both ROMs can be evaluated in terms of their a posteriori error using a residual-based error estimator. When the model hierarchy is queried for a new parameter $\mu \in \mathcal{P}$, first the ML-ROM is evaluated and its error compared to a prescribed error tolerance $\varepsilon > 0$. If the ML-ROM, which is the fastest of the three involved models, is sufficiently accurate, the machine learning approximation is returned. Otherwise, the RB-ROM, which takes more time to solve than the ML-ROM but is still faster than the FOM, is called. Also for the RB-ROM, the a posteriori error estimator is evaluated and the estimated error is compared to the tolerance ε. If the RB-ROM is accurate enough, the reduced basis approximation is returned, if not, the solution provided by the FOM is computed. The FOM is typically much slower than both of the reduced models. The model hierarchy is constructed in such a way that, whenever possible, i.e. their accuracy is sufficient, the faster to evaluate reduced order models are used and calls to the slower models are avoided.

Furthermore, the model hierarchy is also adaptive in the sense that the reduced models are built while already querying the model hierarchy for different parameters. To be more precise, instead of using pre-trained reduced order models, one starts with an empty reduced basis such that the reduced order models can only return zero as solution. Whenever the FOM is called in the hierarchy, new training data for the RB-ROM is generated. Hence, calling the FOM improves the RB-ROM. Similarly,

if the ML-ROM is not sufficiently accurate but the RB-ROM is, new training data for the ML-ROM is obtained by calling the RB-ROM. This way, the performance of both reduced models can be improved by solving the more expensive models in the hierarchy and thus benefit from more accurate solutions obtained by the slower models.

5. Application of the model hierarchy to optimal control problems. In the following we transfer the adaptive model hierarchy described in Section 4 to parametrized optimal control problems by making use of the ROMs presented in Sections 3.1 and 3.2 together with the error estimator from Section 3.3. To this end, we assume that a desired accuracy $\varepsilon > 0$ of the approximate final time adjoint with respect to the optimal final time adjoint obtained by solving the FOM is prescribed.

As a first step, the ML-ROM as introduced in Section 3.2 is called to obtain the approximate final time adjoint $\hat{\varphi}_\mu^N$. Afterwards, the error of the ML-ROM is estimated by evaluating $\eta_\mu(\hat{\varphi}_\mu^N)$ where the error estimator η_μ is defined in (3.2). If the estimated error is smaller or equal to the tolerance ε, the approximate optimal control $\hat{u}_\mu^N$ is returned. If instead the ML-ROM was not accurate enough, the RB-ROM is solved for $\tilde{\varphi}_\mu^N$. Similarly to the ML-ROM, the error estimate $\eta_\mu(\tilde{\varphi}_\mu^N)$ is computed and compared to the prescribed tolerance ε. If the RB-ROM is sufficiently accurate, i.e. it holds $\eta_\mu(\tilde{\varphi}_\mu^N) \leq \varepsilon$, the control $\tilde{u}_\mu^N$ is returned. If even the RB-ROM is not accurate enough, the FOM is called and the optimal control u_μ^* is returned.

Due to the reliability of the error estimator η_μ from (3.2), i.e. the first estimate in (3.3), the result of the model hierarchy comes with a guaranteed accuracy. To be more precise, the error of the (approximate) final time adjoint used to compute the (approximate) optimal control that is returned by the model hierarchy is at most ε with respect to the optimal final time adjoint derived from the FOM.

The way the adaptive model hierarchy determines a control when it is evaluated for a parameter $\mu \in \mathcal{P}$ is visualized in Figure 5.1.

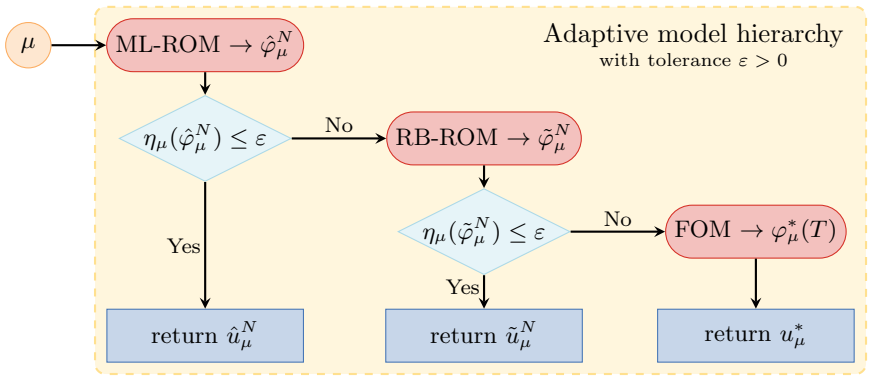

FIG. 5.1. *Visualization of the adaptive model hierarchy applied to the parametrized optimal control setting.*

At this point, we recall again that every call to the RB-ROM in the model hierarchy generates new training data for the ML-ROM. Similarly, evaluating the FOM results in a new function for the reduced basis in the RB-ROM. Since extending the reduced basis results in a larger dimension of the reduced space, we also have to extend the previous ML-ROM which was created for a smaller reduced basis. One

possibility to do so is to extend the previous training data by adding zeros for the new components as done in [6]. However, this strategy might result in an undesired bias towards zero for the ML-ROM training. To circumvent this issue, we follow a different policy in which we instead train a new machine learning surrogate for each individual coefficient in the reduced basis expansion. This means in particular that in the machine learning training no training data consisting of artificial zeros is used.

We also emphasize that the main computational effort in the ML-ROM is spent for evaluating the error estimator. Computing the error estimate $\eta_\mu(p)$ according to (3.2) for a given parameter $\mu \in \mathcal{P}$ and approximate final time adjoint $p \in \mathbb{R}^n$ still requires the solution of the adjoint equation and the primal state equation and therefore depends on the (large) dimension n of the state space. This computational effort in particular limits the efficiency of the ML-ROM when compared to the RB-ROM and will be discussed in more detail when investigating the numerical test case in the next section.

6. Numerical experiment. As a numerical example we consider a parametrized heat equation where the parameter determines the heat conductivity of the underlying material as well as the target state. The two components of the control act on the Dirichlet boundary of the one-dimensional domain. The setting is similar to the one presented in [9, Section 6.2] and will be recalled briefly in the following (see [9] for more details).

Given a parameter $\mu = [\mu_1, \mu_2] \in \mathcal{P} \coloneqq [1, 2] \times [0.5, 1.5] \subset \mathbb{R}^2$, the parametrized heat equation is given as

$$\begin{aligned} \partial_t v_\mu(t,y) - \mu_1 \Delta v_\mu(t,y) &= 0 && \text{for } t \in [0,T], y \in \Omega, \\ v_\mu(t,0) &= u_{\mu,1}(t) && \text{for } t \in [0,T], \\ v_\mu(t,1) &= u_{\mu,2}(t) && \text{for } t \in [0,T], \\ v_\mu(0,y) &= v_\mu^0(y) = \sin(\pi y) && \text{for } y \in \Omega, \end{aligned}$$

where $v_\mu \colon [0,T] \times \Omega \to \mathbb{R}$ with $T = 0.1$ denotes the state of the system. Furthermore, we denote by $u_\mu(t) = \left[u_{\mu,1}(t), u_{\mu,2}(t)\right]^\top \in \mathbb{R}^2$ for $t \in [0,T]$ the control acting as Dirichlet boundary values on both ends of the one-dimensional domain $\Omega = [0,1]$. In the optimal control problem we consider, the goal is to steer the system state at final time T close to the target state given by $v_\mu^T(y) = \mu_2 y$ for $y \in \Omega$. In this setting, the first component of the parameter determines the heat conductivity of the underlying material and the second component changes the slope of the target state. For the discretization of the system above, we use a second-order central finite difference scheme with $n = 200$ inner points in space and the Crank-Nicolson method with 6000 time steps for the time discretization. The system matrices are thus given as

$$A_\mu = \frac{\mu_1}{h^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} \in \mathbb{R}^{n \times n} \quad \text{and} \quad B_\mu = \frac{\mu_1}{h^2} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{n \times 2},$$

where $h = 1/(n+1)$. We choose the matrix $M = I \in \mathbb{R}^{n \times n}$ as the identity matrix and $R = \begin{bmatrix} 0.125 & 0 \\ 0 & 0.25 \end{bmatrix} \in \mathbb{R}^{2 \times 2}$. The initial and target states x_μ^0 and x_μ^T result from evaluating v_μ^0 and v_μ^T on the uniform grid also used in the finite difference scheme.

The adaptive model hierarchy is queried for 10,000 parameters from a uniformly distributed grid in the parameter set $\mathcal{P}$ that were randomly shuffled. The model hierarchy is applied to this system with a fixed tolerance of $\varepsilon = 10^{-4}$. Furthermore, the ML-ROM is trained whenever the reduced basis is extended or when five new training samples from the RB-ROM are collected. The experiment was performed on a dual socket compute server with two Intel(R) Xeon(R) Gold 6254 CPUs running at 3.10GHz and 36 cores in each CPU. The `Python` code for the experiment is available in [8] and can be used to reproduce the results shown below[1].

Table 6.1 summarizes the number of solves, number of error estimates, and timings for the FOM, RB-ROM and the ML-ROM within the adaptive model. We observe that the ML-ROM is sufficiently accurate in more than 99% of the calls to the adaptive model hierarchy. The FOM was called 4 times in total which corresponds to a final reduced basis of size $N = 4$. We should remark at this point that the required amount of time for extending the reduced basis of the RB-ROM and for training the ML-ROM is negligibly small and was therefore omitted in the table. The advantage of using the adaptive model hierarchy instead of solely the FOM or the RB-ROM is reflected in the average time per solve of the three models (last column of Table 6.1). The ML-ROM is about three times faster than the RB-ROM and about seventeen times faster than the FOM. This additional speedup of the ML-ROM pays off in view of the large number of evaluations of the model hierarchy. In particular, due to the error certification, the results obtained by the RB-ROM and the ML-ROM come with a guaranteed accuracy. However, we also observe that the speedup of the ML-ROM compared to the RB-ROM is only moderate. The reason for this observation is the relatively costly error estimation also for the ML-ROM as already discussed at the end of Section 5. The speedup of the ML-ROM depends on the size of the reduced basis and becomes more pronounced for larger reduced bases.

Model	Number of solves	Number of error estimates	Total time for error est. and solving (s)	Average time for error est. and solving per solve (s)
FOM	4	–	112.24	28.06
RB-ROM	65	69	299.26	4.60
ML-ROM	9,931	10,000	16,655.78	1.68

TABLE 6.1
Results of the numerical experiment for the heat equation using the adaptive model hierarchy.

The calls to the models together with their runtimes over the queried parameters are shown in Figure 6.1. We observe that the FOM is called only for some of the first parameters and afterwards either the RB-ROM or the ML-ROM is sufficiently accurate and the FOM is never called again. In contrast, the model hierarchy falls back to the RB-ROM due to an insufficiently accurate ML-ROM for some parameters everywhere in the set of 10,000 parameters. Due to the a posteriori error estimation, such cases are detected and handled properly. The figure further shows that the ML-ROM is already used after a small number of parameters, i.e. with a limited amount of training data an ML-ROM that is sufficiently accurate for several parameter values can be trained. In the last 2,000 calls to the model hierarchy, only the ML-ROM was used. As before, the runtimes for extending the reduced basis and training the ML-

[1] The corresponding `GitHub`-repository containing the source code is available at `https://github.com/HenKlei/ADAPTIVE-ML-OPT-CONTROL`

ROM are not shown in the plot since they are negligibly small.

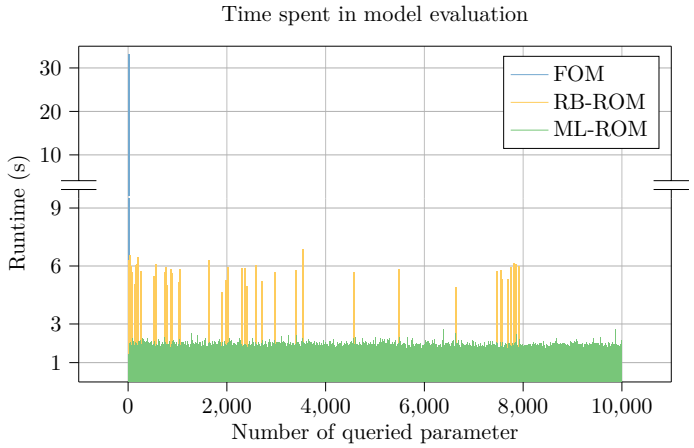

FIG. 6.1. *Performance of the adaptive model hierarchy in terms of the required times for error estimation and evaluation of the involved models when applied to a parametrized heat equation problem.*

In Figure 6.2 we further present the estimated errors of the RB-ROM and the ML-ROM over the queried set of parameters. As expected, for the first couple of parameters, the estimated error for both reduced models is above the desired tolerance ε. After the four evaluations of the FOM, the RB-ROM is always sufficiently accurate. The estimated errors of the ML-ROM vary relatively strongly between about 10^{-4} and 10^{-6}. For some parameters, the ML-ROM is not accurate enough and the RB-ROM is evaluated instead, but for several parameters the error of the ML-ROM is even about two orders of magnitude smaller than the prescribed tolerance.

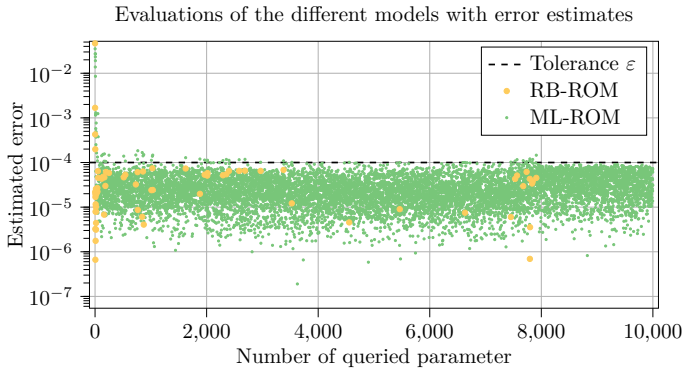

FIG. 6.2. *Error estimation of the RB-ROM and the ML-ROM in the adaptive model hierarchy when applied to a parametrized heat equation problem.*

Altogether, applying the adaptive model hierarchy enables the certified approximate solution of the optimal control problem for 10,000 different parameters, where in the same time span only about 608 calls to the FOM or 3,710 evaluations of the RB-ROM would have been possible.

7. Conclusion and outlook. This work combines an adaptive model hierarchy introduced in [6] with reduced basis and machine learning ROMs for parametrized optimal control problems presented in [9]. The main ingredient for the model hierarchy is the a posteriori error estimator available for both ROMs. By means of this error estimator, the model hierarchy is capable of providing certified results with guaranteed accuracy for every parameter, while only using the fastest models whenever possible.

The numerical example of a parametrized heat equation with controls acting on the Dirichlet boundary values shows how a combination of several layers of reduced order models can provide an additional speedup. At the same time, the error in the result of each query of the model hierarchy can be bounded by the prescribed tolerance due to the error certification. Hence, applying the model hierarchy results in a speedup while still maintaining the accuracy of the outputs.

As future research directions, it might be of interest to investigate larger test cases and apply the model hierarchy in practical applications. Furthermore, one could investigate different choices of machine learning algorithms also in the setting of the adaptive model hierarchy. To further improve the performance of the model hierarchy, the evaluation of the error estimator should be sped up, since this constitutes the main remaining bottleneck in terms of computational efficiency of the overall procedure. The authors in [5] describe an approach based on a reduced basis ROM for the state and the adjoint equations which also accelerates the evaluation of the error estimator. However, using a reduction for the state and adjoint equations results in an additional error, which, to the best of our knowledge, has not been investigated so far.

REFERENCES

[1] F. Ballarin, G. Rozza, and M. Strazzullo. Chapter 9 - Space-time POD-Galerkin approach for parametric flow control. In Emmanuel Trélat and Enrique Zuazua, editors, *Numerical Control: Part A*, volume 23 of *Handbook of Numerical Analysis*, pages 307–338, Elsevier, Amsterdam, The Netherlands, 2022.

[2] T. Daniel, F. Casenave, N. Akkari, and D. Ryckelynck. Model order reduction assisted by deep neural networks (ROM-net). *Advanced Modeling and Simulation in Engineering Sciences*, 7(1):16, 2020.

[3] L. Dedè. Reduced basis method and error estimation for parametrized optimal control problems with control constraints. *SIAM Journal of Scientific Computing*, 50(2):287–305, 2012.

[4] R. DeVore, G. Petrova, and P. Wojtaszczyk. Greedy algorithms for reduced bases in Banach spaces. *Constructive Approximation*, 37(3):455–466, 2013.

[5] G. Fabrini, L. Iapichino, and S. Volkwein. Reduced-Order Greedy Controllability of Finite Dimensional Linear Systems. *IFAC-PapersOnLine: Proceedings of the 9th Vienna International Conference on Mathematical Modelling*, 51(2):296–301, 2018.

[6] B. Haasdonk, H. Kleikamp, M. Ohlberger, F. Schindler, and T. Wenzel. A new certified hierarchical and adaptive RB-ML-ROM surrogate model for parametrized PDEs. *SIAM Journal on Scientific Computing*, 45(3):A1039–A1065, 2023.

[7] J. S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.

[8] H. Kleikamp. Python code for "Application of an adaptive model hierarchy to parametrized optimal control problems", 2024. `https://doi.org/10.5281/zenodo.10669854`.

[9] H. Kleikamp, M. Lazar, and C. Molinari. Be greedy and learn: efficient and certified algorithms for parametrized optimal control problems. 2023. `https://arxiv.org/abs/2307.15590`.

[10] M. Lazar and E. Zuazua. Greedy controllability of finite dimensional linear systems. *Automatica*, 74:327–340, 2016.

[11] G. Santin and B. Haasdonk. Kernel methods for surrogate modeling. In P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, editors, *Model Order Reduction*, volume 2, De Gruyter, Berlin, Boston, 2021.

[12] T. Wenzel, B. Haasdonk, H. Kleikamp, M. Ohlberger, and F. Schindler. Application of deep kernel models for certified and adaptive RB-ML-ROM surrogate modeling. *Lecture Notes in Computer Science*, pages 117–125, Springer Nature Switzerland, 2024.