ESAIM: COCV 24 (2018) 129–151
https://doi.org/10.1051/cocv/2017011

ESAIM: Control, Optimisation and Calculus of Variations
www.esaim-cocv.org

REDUCED BASIS APPROXIMATION OF LARGE SCALE PARAMETRIC ALGEBRAIC RICCATI EQUATIONS

Andreas Schmidt[1] and Bernard Haasdonk[1]

Abstract. The algebraic Riccati equation (ARE) is a matrix valued quadratic equation with many important applications in the field of control theory, such as feedback control, state estimation or $\mathcal{H}_\infty$-robust control. However, solving the ARE can get very expensive in applications that arise from semi-discretized partial differential equations. A further level of computational complexity is introduced by parameter dependent systems and the wish to obtain solutions rapidly for varying parameters. We thus propose the application of the reduced basis (RB) methodology to the parametric ARE by exploiting the well known low-rank structure of the solution matrices. We discuss a basis generation procedure and analyze the induced error by deriving a rigorous *a posteriori* error bound. We study the computational complexity of the whole procedure and give numerical examples that prove the efficiency of the approach in the context of linear quadratic (LQ) control.

Mathematics Subject Classification. 49N05, 34K35, 93B52, 65G99.

Received June 2, 2015. Revised October 5, 2016.

1. Introduction

The solution of many problems in the field of control theory lead to so-called algebraic Riccati equations (ARE), see for example [1] or [39]. A popular example where the ARE arises, is in the solution of feedback control problems for linear and time invariant (LTI) systems with quadratic cost functional – the so called linear quadratic regulator (LQR), see *e.g.* [37]. Other applications, like the optimal sensor placement, linear quadratic Gaussian balanced truncation, optimal state estimation and certain robust control problems also require the solution of AREs, see [26, 32, 33, 42, 66].

Many physical, biological or technical phenomena are modeled in terms of partial differential equations (PDEs). Often they depend on one or more parameters describing for instance material coefficients, geometric properties or quantifying uncertainties. For computational purposes one usually semi-discretizes the PDE in space in the first place, which typically results in large discrete systems. Formulating the AREs for those systems leads to quadratic matrix equations with the same dimension as the underlying discrete system. Hence, solving this equation can be a very challenging task, especially in multi-query scenarios like in real-time contexts, statistical analyses or on small computing devices such as micro controllers. Although in the recent decades many very efficient solvers for the ARE have been developed, the above mentioned scenarios can usually not

Keywords and phrases. Reduced basis method, optimal feedback control, algebraic riccati equation, low rank approximation.

[1] Institute for Applied Analysis and Numerical Simulation, University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany. schmidta@mathematik.uni-stuttgart.de; www.ians.uni-stuttgart.de/agh

Article published by EDP Sciences

be tackled, due to the very large dimension of the semi-discretized systems. We refer to [7] for a review of the state-of-the-art solvers. All large scale solvers are based on the assumption, that the solutions to the ARE are of low (numerical) rank, such that a so called low-rank factorization can be used to efficiently represent the solution. This assumption is also the key ingredient for the model reduction technique, which is developed in this paper.

The focus of this work is to apply parametric model reduction techniques to the ARE in order to obtain approximate solutions rapidly and with reliable error bounds. One reduction technique that has proved to work very well for parametric partial differential equations is the Reduced Basis (RB) method, *cf.* [14, 17, 20, 45]. Extensions to the open loop optimal control of PDEs have been carried out in [34] or [43], and a recent work [56] shows an application of the RB-technique for parametric Lyapunov equations, which is a special case of the parametric ARE.

The overall goal in RB methods is to divide the computation in an expensive offline step and a rapid and cheap online calculation. In the offline part of the procedure, a problem adapted low dimensional solution space is constructed from high dimensional solutions for suitably selected parameter values. The original problem is then projected onto this low dimensional space, which results in a small system of equations that can be solved rapidly online for different parameters. The expensive procedure pays off during the online phase where solutions to the problem can be obtained rapidly with a complexity ideally completely independent of the high dimension. A crucial ingredient in all certified RB methods are *a posteriori* error estimators that can be cheaply evaluated along with the reduced solution. Such error estimators are used in the online step for the verification of the approximation quality as well as in the offline step for choosing the worst approximated element in a greedy loop, *cf.* [14, 16]. Although we are not directly dealing with partial differential equations, we adopt the basic RB methodology for the parametric ARE scenario (RB-ARE).

This paper is organized as follows: we describe the basic setup and establish the notation used throughout this paper in Section 2. In Section 3 the reduction technique is explained and a new algorithm for the basis generation is presented, followed by a discussion of approximation error statements in Section 4, where we focus on a new *a posteriori* error estimator between the true and the approximated solution of the ARE. Furthermore, we discuss the calculation of the constants involved in the error estimation procedure. In particular, we show in Section 5 how the whole procedure can be decomposed in an offline/online scheme. Numerical examples in Section 6 show the performance of our approach. We conclude in Section 7 with final remarks.

2. Problem setting

Throughout this article we are considering the parametric Algebraic Riccati Equation for the unknown solution $P(\boldsymbol{\mu}) \in \mathbb{R}^{n\times n}$:

$$A(\boldsymbol{\mu})^T P(\boldsymbol{\mu})E(\boldsymbol{\mu}) + E(\boldsymbol{\mu})^T P(\boldsymbol{\mu})A(\boldsymbol{\mu}) - E(\boldsymbol{\mu})^T P(\boldsymbol{\mu})B(\boldsymbol{\mu})R(\boldsymbol{\mu})^{-1}B(\boldsymbol{\mu})^T P(\boldsymbol{\mu})E(\boldsymbol{\mu}) + C(\boldsymbol{\mu})^T Q(\boldsymbol{\mu})C(\boldsymbol{\mu}) = 0. \tag{2.1}$$

The matrices $E(\boldsymbol{\mu}), A(\boldsymbol{\mu}) \in \mathbb{R}^{n\times n}$, $B(\boldsymbol{\mu}) \in \mathbb{R}^{n\times m}$, $C(\boldsymbol{\mu}) \in \mathbb{R}^{p\times n}$, $Q(\boldsymbol{\mu}) \in \mathbb{R}^{p\times p}$ and $R(\boldsymbol{\mu}) \in \mathbb{R}^{m\times m}$ with $E(\boldsymbol{\mu})$ invertible are given data matrices and the solution matrix $P(\boldsymbol{\mu}) \in \mathbb{R}^{n\times n}$ is sought. We furthermore require that $Q(\boldsymbol{\mu})$ is symmetric positive semidefinite and $R(\boldsymbol{\mu})$ is symmetric positive definite, which in addition implies the invertibility of $R(\boldsymbol{\mu})$. We allow parameter dependency on all data matrices, where the parameter vector $\boldsymbol{\mu}$ stems from a bounded parameter set $\mathcal{P} \subset \mathbb{R}^q$. Due to the direct link of the data matrices in Equation (2.1) to dynamical systems, we call n the state dimension, m the number of inputs and p the number of outputs, see also Example 2.2 below. Throughout this paper, we will always consider the full parametric case, but for improved readability and sake of brevity we will often omit the explicit dependence of the matrices on the parameter $\boldsymbol{\mu}$ from now on.

The solvability of equation (2.1) is not clear unless additional assumptions and requirements on the data matrices are posed. Indeed, the equation can have none or multiple solutions. Among all possible solutions,

we are interested in the so called stabilizing solution, which is characterized by the following conditions:

$$P = P^T, \quad P \succeq 0, \quad (E, A - BR^{-1}B^T PE) \text{ is stable.} \tag{2.2}$$

Here, P^T is the transpose of P, the expression $P \succeq 0$ denotes positive semidefiniteness of the matrix P and a matrix pair (E, Y) is called stable when all real parts of all eigenvalues $\lambda_i(E, Y)$ of the generalized eigenvalue problem $Yx = \lambda Ex$ are strictly less than zero. Note that in this case, the matrix pair (E, Y) is also called Hurwitz. One possible assumption that guarantees the existence of a unique stabilizing solution to (2.1) is to require stabilizability and detectability of the matrices E, A, B, C: stabilizability of (E, A, B) means that $\mathrm{rk}[A - \lambda E, B] = n$ for all $\lambda \in \mathbb{C}$ with $\Re(\lambda) \geq 0$, where rk denotes the rank of a matrix and $\Re(\cdot)$ is the real part of a complex number. Detectability of (E, A, C) is equivalent to stabilizability of (E^T, A^T, C^T). As proven for example in [39], the stabilizability and detectability assumption guarantee the existence of a unique stabilizing solution $P \in \mathbb{R}^{n\times n}$.

Remark 2.1. We will assume that the problems under consideration are stable, *i.e.* (E, A) is a stable matrix pair. Unstable systems can be preprocessed by finding a suitable matrix H such that $(E, A - BH)$ is stable and by replacing A with $A - BH$ in the remainder of this article. This is always possible due to the stabilizability assumption. We refer to [4] for an example of this procedure.

If not otherwise stated, we will denote by $\|\cdot\| = \|\cdot\|_2$ the Euclidean 2-norm for vectors and induced operator norm of an operator. Further, for a matrix $A = (a_1 \ \dots \ a_n) = (a_{ij})_{i,j=1}^n \in \mathbb{R}^{n\times n}$ with column vectors $a_i \in \mathbb{R}^n$, we define the Frobenius norm $\|A\|_F := \sqrt{\mathrm{tr}\,(A^T A)}$, where $\mathrm{tr}(A) := \sum_{i=1}^n a_{ii}$ is the trace function. We denote by $\mathrm{colspan}(A) = \mathrm{span}(\{a_1, \dots, a_n\})$ the column span of the matrix A. By $|\cdot|$, we denote the number of elements of a set.

Another important property for matrices is the logarithmic norm, *cf.* [11], which is defined as $\nu[A] := \lim_{h\to 0^+} \frac{\|I_n + hA\| - 1}{h}$ for any induced norm $\|\cdot\|$ and which can be calculated as $\nu[A] = \max_i \lambda_i(\frac{1}{2}(A + A^T))$ for the Euclidean norm. The logarithmic norm of a matrix has the property, that it is directly related to the decay of the norm of the matrix exponential e^{At}, *i.e.* it holds

$$\|\mathrm{e}^{At}\| \leq \mathrm{e}^{\nu[A]t}, \quad t \geq 0. \tag{2.3}$$

A proof for this well known inequality can for example be found in [22]. We will make use of this property in Section 4.2, for proving the stability of the reduced controller in the full dimensional LTI system. A matrix A with $\nu[A] < 0$ is called dissipative, since it is related to the energy loss in the physical system. Usually we consider systems with mass matrices E. The logarithmic norm for those systems is defined as $\nu_E[A] := \nu[E^{-1}A]$. For symmetric E one can employ the Cholesky factorization to show that the calculation of $\nu_E[A]$ can be equivalently performed by finding the eigenvalue of the generalized eigenvalue problem $\frac{1}{2}(A + A^T)x = \lambda Ex$ with the largest real part, see also [44].

One of the main application for the ARE, which is also our main interest, is the feedback stabilization of linear and time-invariant (LTI) control systems.

Example 2.2. Consider the following linear quadratic regulator (LQR) problem:

$$\min_u \int_0^\infty (y(t)^T Q y(t) + u(t) R u(t))\mathrm{d}t \tag{2.4}$$

$$\text{s.t.} \begin{cases} E\dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) \\ x(0) = x_0. \end{cases} \tag{2.5}$$

As explained before, the matrices are given as $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$, where E is invertible. Systems of this form typically arise after the spatial discretization of time dependent linear partial differential

equations, see for example [15,31]. The function $u(t)$ is called "control" of the system and serves as an input variable that can be chosen to alter the dynamics in a specific way. The function $y(t)$ is called "output" and can model measurements of interest. As explained for the ARE, the weighting matrices $Q \in \mathbb{R}^{p\times p}$ and $R \in \mathbb{R}^{m\times m}$ in the cost functional are symmetric and Q must be positive semidefinite, whereas R must be positive definite. The minimization of the cost functional finds the control input of minimum energy, which steers the system to the origin $x = 0$. It can be shown (see for example [39]), that the problem of minimizing (2.4) with respect to (2.5) is solved by setting $u(t) = -R^{-1}B^T PEx(t)$ where P is the unique symmetric and positive semidefinite solution to the ARE

$$A^T PE + E^T PA - E^T PBR^{-1}B^T PE + C^T QC = 0. \tag{2.6}$$

As mentioned above P is also the stabilizing solution under the stabilizability and detectability assumptions. This reveals the close relation between AREs and feedback control problems: since the optimal control is given in the form $u(t) = \mathcal{F}(x(t))$, state information is fed back to the controller. This specific structure has the big advantage over classical open-loop control, that the feedback structure makes the closed-loop more robust to *a priori* unknown disturbances. Since spatial discretizations like the finite element method for linear PDEs lead to large LTI systems and hence to large AREs, a fast and reliable approximation of (2.6) is the key ingredient for solving parametric feedback control problems in real-time or multi-query scenarios.

Remark 2.3. Note that we are discussing the ARE in its generalized form (2.1), *i.e.* with explicit dependence on the mass matrix E. Since we assume invertibility of E we can substitute $P = E^{-T}\tilde{P}E^{-1}$ in equation (2.1), which results in

$$\tilde{A}^T\tilde{P} + \tilde{P}\tilde{A} - \tilde{P}\tilde{B}R^{-1}\tilde{B}^T\tilde{P} + C^T QC = 0, \tag{2.7}$$

with $\tilde{A} := E^{-1}A$ and $\tilde{B} := E^{-1}B$. Many solvers for AREs in the generalized form (2.6) internally transform the system to the standard form (2.7), and avoid the multiplication of A with E^{-1} by employing a LU-factorization and forward and backward substitutions, since the inversion would destroy important properties like sparsity or symmetry. We refer to the overview article [7] and the thesis [50] for further information.

3. Reduction approach

In cases where the number of inputs m and the number of outputs p is small compared to the number of degrees of freedom n of the underlying system, the solution matrix P to the ARE (2.1) often can be approximated by a low rank factorization of the form $P \approx ZZ^T$ with $Z \in \mathbb{R}^{n\times k}$ and $k \ll n$. This is possible due to a fast decay in the singular values of P, which means that P is of low numerical rank. In the contributions [3,40,47,60] theoretical justifications for this observation are carried out, especially for the Lyapunov equation with low rank right hand side. In [8], an upper bound for the decay of the singular values of P is presented, which can be used to explain the desired low-rank structure.

We further assume that the manifold that consists of the low-rank factors Z under parameter variation, can be approximated by linear spaces of much lower dimension than the original system dimension n. Hence, a projection of equation (2.1) on the relevant subspace might lead to a much smaller problem that can easily be solved by standard techniques. Projection-based methods for approximating matrix equations are not new: large scale Lyapunov equations can be solved numerically by projecting the equation on Krylov subspaces, see for example [28,49,53]. Extensions of those projection methods for the ARE can be found for instance in [24,29,30]. Very recently, for parametric problems, in [56] an RB method for large and sparse Lyapunov equations has been presented, whereas no previous work for the parametric ARE is known to the authors.

3.1. Projection based approximation of the ARE

A general class of projections can be described by a pair of biorthogonal matrices $V, W \in \mathbb{R}^{n\times N}$ with $W^T V = I_N$ (or $W^T EV = I_N$) and $N \ll n$. The columns of W span the N dimensional subspace $\mathcal{W} := \text{colspan}(W) \subset \mathbb{R}^n$.

We will approximate the ARE in the space of all matrices, whose columns stem from the space $\mathcal{W}$: For that purpose, we first note that any matrix $\hat{P} \in \mathbb{R}^{n\times n}$ with $\operatorname{colspan}(\hat{P}) \subset \mathcal{W}$ can be written as WG for some matrix $G \in \mathbb{R}^{N\times n}$. Since the stabilizing solution matrices of the ARE are always symmetric, one has the additional requirement $\hat{P} = WG = G^T W^T = \hat{P}^T$. Hence, the matrix $\hat{P}$ can also be written as $\hat{P} = W\bar{G}W^T$ for a symmetric matrix $\bar{G} \in \mathbb{R}^{N\times N}$. In the following, we will write $P_N := \bar{G}$, since this will be the N dimensional target quantity for our RB scheme. Substituting the solution P by its approximation

$$P \approx \hat{P} := WP_NW^T \tag{3.1}$$

in the left hand side of equation (2.1), we arrive at

$$\mathcal{R}(WP_NW^T) := A^TWP_NW^TE + E^TWP_NW^TA - E^TWP_NW^TBR^{-1}B^TWP_NW^TE + C^TQC,$$

where we introduced the matrix-valued residual $\mathcal{R}(\cdot)$ of the ARE. Requiring a so called Galerkin condition[2] $V^T\mathcal{R}(\hat{P})V = 0$ leads to

$$A_N^TP_NE_N + E_N^TP_NA_N - E_N^TP_NB_NR^{-1}B_N^TP_NE_N + C_N^TQC_N = 0, \tag{3.2}$$

where the reduced matrices are defined as

$$E_N = W^TEV, \quad A_N := W^TAV, \quad B_N = W^TB, \quad C_N = CV. \tag{3.3}$$

Equation (3.2) is again an ARE, but now of much lower dimension $N \times N$ and can thus be easily solved by using standard methods like a Hamiltonian approach, which is implemented in the function `care` in MATLAB.

The projection-based factorization naturally defines a low rank approximation for the full dimensional solution matrix: we can calculate the low rank factor $\hat{Z}$ of the approximation $\hat{P}$ by employing an eigenvalue decomposition of the small matrix $P_N = U_NS_NU_N^T$ and by setting

$$\hat{Z} := WU_NS_N^{1/2} \in \mathbb{R}^{n\times N}, \tag{3.4}$$

which yields $\hat{P} = \hat{Z}\hat{Z}^T$, since $\hat{Z}\hat{Z}^T = WU_NS_N^{1/2}S_N^{1/2}U_N^TW^T = WP_NW^T$. Instead of forming the whole and usually dense $n \times n$ matrix $\hat{P}$ through (3.1), the low rank factor (3.4) is used in places where the full matrix is required.

Remark 3.1. We want to briefly compare the presented RB-ARE reduction approach to the classical projection based model reduction techniques for dynamical systems such as Krylov-subspace methods, balanced truncation or proper orthogonal decomposition, *cf.* [2]. We will again consider LTI systems of the form (2.5). All these model reduction approaches are based on the projection of the state to a subspace, whose basis is defined by the columns of a projection matrix $V \in \mathbb{R}^{n\times N}$. If we now introduce a reduced state $x_N(t) \in \mathbb{R}^N$ and approximate $x(t) \approx Vx_N(t)$, we can substitute the high dimensional state in the system (2.5). Restricting the residual to be orthogonal to another test space, defined by a basis matrix $W \in \mathbb{R}^{n\times N}$, *e.g.* after introducing a Petrov-Galerkin condition, we end up with the reduced system

$$\begin{cases} W^TEV\dot{x}_N(t) = W^TAVx_N(t) + W^TBu(t) \\ \qquad\qquad y_N(t) = CVx_N. \end{cases} \tag{3.5}$$

Abbreviating $E_N := W^TEV$, $A_N := W^TAV$, $B_N = W^TB$ and $C_N := CV$ as in (3.3) reveals the link to the reduced ARE (3.2): if we define the minimization target

$$\min \int_0^\infty \left(y_N(t)^TQy_N(t) + u(t)^TRu(t)\right) \mathrm{d}t, \tag{3.6}$$

[2]See for example [49].

we see that equation (3.2) solves the reduced LQR problem (3.6), (3.5) when we set $u(t) = -R^{-1}B_N^T P_N E_N x(t)$. Although the reduced closed loop system $E_N \dot{x}_N = (A_N - B_N R^{-1} B_N^T P_N E_N) x_N$ is optimally controlled by choosing P_N as the stabilizing solution to the reduced ARE (3.2), the reconstruction $\hat{P} := W P_N W^T$ must not necessarily yield a good approximation to the solution P of the full dimensional ARE (2.1), if one of the above mentioned classical model order reduction techniques is used, as those reduced spaces are only capable of representing the state well, but not the closed-loop feedback structure of the system, see for example [9,51]. The procedure presented in this article is able to fill this gap and to allow the efficient approximation of closed-loop controllers in the LQR framework.

We finish this section with a useful property that allows verifying zero error whenever the projection matrix contains the whole (low rank) solution space of the ARE.

Proposition 3.2 (Reproduction of Solutions)**.** *Let $V, W \in \mathbb{R}^{n\times N}$ be given. Let the symmetric and positive semidefinite matrix $P \in \mathbb{R}^{n\times n}$ be the unique stabilizing solution to the ARE (2.1) and assume* $\operatorname{colspan}(P) \subset \operatorname{colspan}(W)$*. We assume that (E_N, A_N, B_N) is stabilizable and (E_N, A_N, C_N) from (3.3) is detectable. Then it follows*

$$P = W P_N W^T = \hat{P}$$

where P_N solves the reduced ARE (3.2).

Proof. Due to symmetry of P and the assumption that $\operatorname{colspan}(P) \subset \operatorname{colspan}(W)$, there exists a unique symmetric and positive semidefinite matrix $G \in \mathbb{R}^{N\times N}$ with $P = WGW^T$. Using the definition of the ARE, we get

$$A^T WGW^T E + E^T WGW^T A - E^T WGW^T BR^{-1} B^T WGW^T E + C^T QC = 0.$$

Multiplying this equation with V^T from left and V from right yields

$$A_N^T G E_N + E_N^T G A_N - E_N^T G B_N R^{-1} B_N^T G E_N + C_N^T Q C_N = 0, \tag{3.7}$$

where the reduced matrices are defined as in (3.3). Due to the assumption of stabilizability and detectability of the reduced matrices, the reduced ARE (3.7) has a unique positive semidefinite and symmetric solution G. As by construction $\hat{P} = WGW^T$ is a symmetric and positive semidefinite solution to the full dimensional ARE, we can conclude due to the uniqueness of the stabilizing solution that $P = \hat{P}$ and thus $G = P_N$, see [39]. □

3.2. Low rank factor greedy

Besides biorthogonality $W^T V = I_N$ (or $W^T E V = I_N$) we have not yet made any further assumptions on the matrices W, V. There are many possible ways how a basis can be constructed. In a parametric scenario, the reduced basis (RB) approach has proven to work well for the approximation of partial differential equations, see for example [14,45,48,61]. The goal in all RB methods is to split the calculation in an offline/online scheme, where the online computation ideally does not depend on the high dimension n. This is achieved by accepting a possibly expensive offline phase, where a problem adapted solution space is constructed which then allows the fast solution in the online phase. Rigorous error bounds that are likewise fast to evaluate are the key ingredient for justifying the approach online.

The standard approach for constructing the reduced basis works in a greedy fashion as it has been introduced for RB methods in [61]. The greedy procedure is an iterative algorithm that tries to uniformly minimize the approximation error on a finite, but possibly large, training set $\mathcal{P}_{\text{train}} \subset \mathcal{P}$ by subsequently adding vectors constructed from the worst approximated element to the basis. A crucial ingredient in all greedy approximations is the choice of a suitable error indicator for selecting the worst approximated element. There are several possible choices, each with its own advantages and disadvantages. One possibility would certainly be the true approximation error. However, this is often not possible, since the evaluation in each iteration would be too

expensive as the true solution for all training parameters would be required. A better approach makes use of error indicators that are cheaply computable for all parameters in the training set $\mathcal{P}_{\text{train}}$.

In this paper, we propose a new algorithm for the construction of a subspace for the RB-ARE method, which we denote Low-Rank Factor Greedy (LRFG) procedure. The pseudocode is listed in Algorithm 1. The basic structure of the algorithm follows the greedy-principle: in a loop the error indicator is evaluated for all elements $\boldsymbol{\mu}$ in the training set $\mathcal{P}_{\text{train}} \subset \mathcal{P}$ and for the current reduced basis, that is spanned by the columns of W. If the algorithm determines a maximum error indicator value beneath a given desired tolerance ε, the loop terminates and the basis construction for W is complete. The biorthogonal counterpart V can simply be chosen as $V := W$, but also an orthogonalization with respect to the mass matrix is possible, such that V is chosen such that $W^T E V = I_N$. If the tolerance is not yet reached, the parameter $\boldsymbol{\mu}^* \in \mathcal{P}_{\text{train}}$ of the worst approximated solution for the current basis is selected by finding the parameter $\boldsymbol{\mu}^*$ that maximizes the error indicator $\Delta(W, \boldsymbol{\mu})$. The next step requires an expensive calculation of the low rank factor $Z(\boldsymbol{\mu}^*)$ for the solution of the ARE $P(\boldsymbol{\mu}^*)$. The next step consists in finding the relevant information in the low rank factor $Z(\boldsymbol{\mu}^*)$. As a preprocessing step, only the part which is not yet captured in the current basis $Z_\perp := (I_n - WW^T)Z(\boldsymbol{\mu}^*)$ is considered. Afterwards the remaining information in the column space of $Z_\perp$ is compressed and added to the basis. This essentially is similar to the POD-Greedy procedure for parametric and time-dependent problems, where the low-rank factors $Z(\boldsymbol{\mu}^*)$ replace the solution trajectories, see [19].

For the compression we propose to use the proper orthogonal decomposition-technique (POD), which essentially is the principal component analysis. This has been successfully applied in the basis building procedure for time-dependent problems, see *e.g.* [21,62]. The idea behind POD is to find orthogonal vectors $\tilde{z}_1, \dots, \tilde{z}_l \in \mathbb{R}^n$ that capture the essential information in the columns of $(z_1, \dots, z_k) = Z_\perp \in \mathbb{R}^{n \times k}$. Mathematically speaking, we seek the solution of the following constrained maximization problem:

$$\max_{\tilde{z}_1,\dots,\tilde{z}_l \in \mathbb{R}^n} \sum_{i=1}^{l} \sum_{j=1}^{k} |\langle z_j, \tilde{z}_i \rangle|^2, \quad \text{subject to} \quad \langle \tilde{z}_i, \tilde{z}_j \rangle = \delta_{ij}, \text{ for } i, j = 1, \dots, l \tag{3.8}$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product in $\mathbb{R}^n$. Problem (3.8) can be solved by setting $\tilde{z}_i = u_i$ for $i = 1, \dots, l$, where $u_i \in \mathbb{R}^n$ are the first columns of the matrix U_1 in the thin singular value decomposition $Z = U_1 S U_2^T$, with $S = \operatorname{diag}(\sigma_1, \dots, \sigma_k) \in \mathbb{R}^{k \times k}$ and $U_1 \in \mathbb{R}^{n \times k}$ with orthogonal columns. In the following we will call $\tilde{z}_i$ POD modes. A natural approach would be to add the most dominant POD-mode in each greedy-iteration, but this can lead to large offline times as in (almost) each iteration an ARE has to be solved, see also the numerical examples in Section 6. A different approach is to add more elements by describing an inner tolerance $tol_i \in [0, 1]$, which correlates to the amount of information that should be captured from the low rank factor $Z_\perp$. To quantify this amount, we use the indicator function

$$\mathcal{E}(l) := \frac{\sum_{i=1}^{l} \sigma_i}{\sum_{i=1}^{r} \sigma_i}, \quad r := \operatorname{rk}(Z_\perp). \tag{3.9}$$

We find the smallest number $\tilde{l} \geq 1$, such that $\mathcal{E}(l) \geq tol_i$ for all $l \geq \tilde{l}$, meaning that we capture at least the fraction tol_i of all information. We abbreviate this POD-procedure by $(\tilde{z}_1, \dots, \tilde{z}_{\tilde{l}}) = \tilde{Z} = \text{POD}(\mathcal{Z}_\perp, tol_i)$ and refer to [62] for more information on POD in the context of model reduction. The add-one strategy described above, as well as a full inclusion of the low rank factor in the basis, is captured by this procedure by setting $tol_i = 0$ and $tol_i = 1$, respectively. There is a trade-off between the resulting basis size and the number of offline solves of the ARE. We refer to Section 6 for a numerical example.

In practice, the algorithm must certainly be extended at some points. For example it must be ensured that the algorithm terminates with a reasonably sized basis. This can be achieved by prescribing a maximum number of basis elements or by using techniques such as adaptive basis enrichment or $\mathcal{P}$-partitioning, *cf.* [18]. Furthermore, the calculated snapshots should be cached, since the same parameter could be chosen during the greedy-iterations. This is due to the fact that incluision of few modes does not necessarily imply exact

approximation of a given matrix snapshot. A similar phenomenon can be observed in RB approximation for unsteady problems, see [19].

Algorithm 1: Low-Rank Factor Greedy Algorithm (LRFG) for the basis generation.

Data: Initial basis matrix W_0, training set $\mathcal{P}_{\text{train}} \subset \mathcal{P}$, tolerance ε, inner tolerance $tol_i \in [0,1]$, error indicator $\Delta(W, \boldsymbol{\mu})$

Set $W := W_0$.

while $\max_{\boldsymbol{\mu} \in \mathcal{P}_{train}} \Delta(W, \boldsymbol{\mu}) > \varepsilon$ **do**

- $\boldsymbol{\mu}^* := \arg\max_{\boldsymbol{\mu} \in \mathcal{P}_{\text{train}}} \Delta(W, \boldsymbol{\mu})$
- Solve the full dimensional ARE for the low rank factor $Z(\boldsymbol{\mu}^*)$
- Set $Z_\perp := (I_n - WW^T)Z(\boldsymbol{\mu}^*)$
- $\tilde{Z} = \text{POD}(Z_\perp, tol_i)$
- Extend the current basis matrix $W = (W, \tilde{Z})$

Set $V := W$ (or choose V such that $W^T E V = I_n$)

Return V, W.

4. Error estimation

A crucial ingredient in certified RB methods is the validation of the procedure by means of error bounds. They should be rigorous and cheaply computable, which means that in the ideal case the evaluation complexity of the bound is independent of the high dimension n. The main result in this section is a computable *a posteriori* error bound for the error between the approximated and the true solution (potentially non low-rank) to the ARE. The application of the bound however requires the knowledge of several constants that are very expensive to calculate. As a remedy we state bounds for those constants and show how different techniques can be used to achieve fast (but possibly nonrigorous) approximations to those constants. A complete offline/online decomposition of the procedure allows the calculation of the approximation and the corresponding bounds in a complexity independent of the high dimension n and thus ensures a good online performance.

4.1. *A posteriori* error estimator for the parametrized ARE

In this section we present a new residual based error estimator for approximations to the generalized ARE (2.1). We recall the definition of the residual of the ARE for $P \in \mathbb{R}^{n \times n}$:

$$\mathcal{R}(P) := A^T PE + E^T PA - E^T PBR^{-1}B^T PE + C^T QC. \tag{4.1}$$

The ARE can be seen as a nonlinear problem defined on the Banach space $\mathbb{R}^{n \times n}$. We make use of a generic theorem that can be found in [10], and which can be used to derive *a posteriori* residual based error bounds for nonlinear problems. We begin with a rigorous statement and later relax the assumptions to allow online efficient computations.

Theorem 4.1 (Residual Based Error Bound)**.** *Let $\hat{P} \in \mathbb{R}^{n \times n}$ be an approximate solution to the ARE (2.1). Abbreviate $Y(\hat{P}) := A - BR^{-1}B^T\hat{P}E$ and define the linear operator $\mathcal{L}_{\hat{P}} := D\mathcal{R}|_{\hat{P}}$, i.e. $\mathcal{L}_{\hat{P}}(N) := Y(\hat{P})^T NE + E^T NY(\hat{P})$. Assume $\mathcal{L}_{\hat{P}}$ is invertible and define*

$$\gamma := \|\mathcal{L}_{\hat{P}}^{-1}\| = \sup_{\|S\|=1} \|\mathcal{L}_{\hat{P}}^{-1}(S)\|, \quad \varepsilon := \|\mathcal{R}(\hat{P})\|, \quad L := 2\|E\|^2\|BR^{-1}B^T\|.$$

Then: If the validity criterion

$$4\gamma^2\varepsilon L \leq 1 \tag{4.2}$$

holds true, there exists a unique solution $P^* \in B_{2\gamma\varepsilon}(\hat{P}) := \{X \in \mathbb{R}^{n\times n} | \, \|X - \hat{P}\| \leq 2\gamma\varepsilon\}$ *to the ARE, i.e.* $\mathcal{R}(P^*) = 0$*, and the following upper bound holds*

$$\|P^* - \hat{P}\| \leq \Delta_P := \frac{1}{2\gamma L}(1 - \sqrt{1 - 4\gamma^2 L\varepsilon}). \tag{4.3}$$

Proof. The proof is performed in two steps: We first prove the existence of the solution P^* by applying Banach's fixed-point theorem. This follows in large parts the proof of Theorem 2.1 in [10]. The second part shows how the bound (4.3) can be established.

Since the residual (4.1) is quadratic in P it is Fréchet differentiable with derivative $D\mathcal{R}|_{\hat{P}} = \mathcal{L}_{\hat{P}}$. In the sequel we need the following Lipschitz-property of the derivative with respect to the evaluation argument: let $S, P \in \mathbb{R}^{n\times n}$, then

$$\begin{aligned} \left\|(\mathcal{L}_{\hat{P}} - \mathcal{L}_P)(S)\right\| &= \left\|(BR^{-1}B^T(\hat{P} - P)E)^T SE + E^T S(BR^{-1}B^T(\hat{P} - P)E)\right\| \\ &\leq 2\|E\|^2\|BR^{-1}B^T\|\|P - \hat{P}\|\|S\| = L\|P - \hat{P}\| \, \|S\| \\ &\Rightarrow \left\|\mathcal{L}_{\hat{P}} - \mathcal{L}_P\right\| \leq L\left\|\hat{P} - P\right\|. \end{aligned} \tag{4.4}$$

Next, we define the mapping $H : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ by $H(P) := P - \mathcal{L}_{\hat{P}}^{-1}\mathcal{R}(P)$. Then it clearly holds $H(P) = P \Leftrightarrow \mathcal{R}(P) = 0$. We next prove the existence of a fixed-point of H, by showing that H is a contraction and self-mapping in a closed ball $B_\alpha := B_\alpha(\hat{P}) = \{P \in \mathbb{R}^{n\times n} | \|P - \hat{P}\| \leq \alpha\}$ around the approximate solution $\hat{P}$. We first show that H is a contraction in the set B_α, whenever the radius α is small enough. For this purpose we calculate with $P, P' \in B_\alpha$

$$\|H(P) - H(P')\| = \left\|P - P' - \mathcal{L}_{\hat{P}}^{-1}(\mathcal{R}(P) - \mathcal{R}(P'))\right\| = \left\|\mathcal{L}_{\hat{P}}^{-1}\left[\mathcal{L}_{\hat{P}}(P - P') - (\mathcal{R}(P) - \mathcal{R}(P'))\right]\right\|. \tag{4.5}$$

Using the fundamental theorem of calculus, we obtain the representation

$$\mathcal{R}(P) - \mathcal{R}(P') = \int_0^1 \mathcal{L}_{P' + t(P-P')}(P - P')\mathrm{d}t, \tag{4.6}$$

which can be combined with (4.5) to get the bound

$$\begin{aligned} \|H(P) - H(P')\| &= \left\|(\mathcal{L}_{\hat{P}})^{-1}\int_0^1 (\mathcal{L}_{\hat{P}} - \mathcal{L}_{\underbrace{P' + t(P-P')}_{\in B_\alpha}})(P - P')\mathrm{d}t\right\| \\ &\leq \gamma[\sup_{P\in B_\alpha}\left\|\mathcal{L}_{\hat{P}} - \mathcal{L}_P\right\|]\|P - P'\| \overset{(4.1)}{\leq} \gamma L\alpha\|P - P'\| \overset{!}{\leq} \tau\|P - P'\|, \quad \text{with } \tau < 1. \end{aligned} \tag{4.7}$$

Whenever the last inequality is satisfied, H is a contraction on B_α. For proving the self-mapping property, we find a condition on α such that $\|H(P) - \hat{P}\| \leq \alpha$, since then $H(P) \in B_\alpha$. Let $P \in B_\alpha$, then

$$\begin{aligned} \left\|H(P) - \hat{P}\right\| &= \left\|\mathcal{L}_{\hat{P}}^{-1}\left[\mathcal{L}_{\hat{P}}(P - \hat{P}) - (\mathcal{R}(P) - \mathcal{R}(\hat{P}))\right] - \mathcal{L}_{\hat{P}}^{-1}\mathcal{R}(\hat{P})\right\| \\ &\overset{(4.6)}{=} \left\|\mathcal{L}_{\hat{P}}^{-1}\int_0^1 (\mathcal{L}_{\hat{P}} - \mathcal{L}_{\hat{P}+t(P-\hat{P})})(P - \hat{P})\mathrm{d}t - \mathcal{L}_{\hat{P}}^{-1}\mathcal{R}(\hat{P})\right\| \\ &\leq \|\mathcal{L}_{\hat{P}} - \mathcal{L}_{\hat{P}+t(P-\hat{P})}\|\|P - \hat{P}\| + \gamma\varepsilon \leq \gamma\alpha^2 L + \gamma\varepsilon \overset{!}{\leq} \alpha. \end{aligned} \tag{4.8}$$

We first choose $\alpha = 2\gamma\varepsilon =: \hat{\alpha}$. Then from (4.7) we have $\|H(P) - H(P')\| \leq 2\gamma^2 L\varepsilon \|P - P'\| \leq \frac{1}{2}\|P - P'\|$, where we made use of the validity criterion (4.2). Furthermore, from (4.8) we obtain $\gamma 4\gamma^2\varepsilon^2 L + \gamma\varepsilon = \gamma\varepsilon(4\gamma^2\varepsilon L) + \gamma\varepsilon \leq 2\gamma\varepsilon = \hat{\alpha}$. Thus the mapping H is a contraction and a self-mapping in $B_{\hat{\alpha}} := B_{2\gamma\varepsilon}(\hat{P})$ and we can apply the Banach fixed-point theorem to prove the existence of $P^* \in B_{\hat{\alpha}}$ with $\mathcal{R}(P^*) = 0$. Note that this already gives an *a posteriori* error bound $\|\hat{P} - P^*\| \leq 2\gamma\varepsilon$. However this bound can be sharpened by rechecking the bounds (4.7), (4.8): the contraction property is valid for $\alpha \leq \hat{\alpha}$. Furthermore by explicitly solving the inequality $\gamma L\alpha^2 + \gamma\varepsilon \leq \alpha$ for the interval $[\alpha_-, \alpha_+]$ yields

$$\alpha_\pm = \frac{1}{2\gamma L}\left(1 \pm \sqrt{1 - 4\gamma^2\varepsilon L}\right).$$

Hence H is a self-mapping of B_α for all values $\alpha \in [\alpha_-, \hat{\alpha}]$ and $\|P^* - \hat{P}\| \leq \alpha$. Choosing $\alpha = \alpha_-$ yields the desired bound. □

Remark 4.2 (Comparison to existing bounds)**.** We want to provide a short comparison of existing error bounds for the ARE and want to highlight the difference to the bound stated in Proposition 4.1. Articles devoted to residual based error bounds for the ARE can for example be found in [13, 35, 36, 59]. All existing error bounds are formulated for the case where $E = I_n$, hence our bound generalizes the existing work. More importantly, the theorem can be used to *a posteriori* characterize the existence of a full solution in the neighborhood to the reduced solution, whereas all other bounds must assume the existence. Furthermore, it also works for non-stabilizing solutions of the ARE, as long as $\mathcal{L}_{\hat{P}}$ is an invertible operator.

Remark 4.3. The assumption of invertibility of $\mathcal{L}_{\hat{P}}$ is not a strong assumption. The theory for the Lyapunov equation states that $\mathcal{L}_{\hat{P}}$ is invertible when $\lambda_j + \bar{\lambda_k} \neq 0$ for all eigenvalues λ_j and λ_k of the generalized eigenvalue problem $(E, Y(\hat{P}))$, where $\bar{\lambda}$ denotes the complex conjugate of λ, see [46]. If $(E, Y(\hat{P}))$ is stable, which is what we expect from our approximation, this condition is guaranteed.

The application of Theorem 4.1 requires the knowledge of the constants γ, L and the norm of the residual ε, all depending on the n-dimensional data. In order to get an error bound that only depends on the RB dimension N, we state the following proposition, where the constants are replaced by computable upper bounds.

Proposition 4.4 (Computable Error Bound)**.** *Let the assumptions of Theorem 4.1 be valid. Assume further, that rapidly computable upper bounds $\gamma \leq \gamma_N$, $L \leq L_N$ and $\varepsilon \leq \varepsilon_N$ are available. Then it holds: if*

$$4\gamma_N^2 L_N \varepsilon_N \leq 1 \tag{4.9}$$

is valid, there exists a unique solution $P^* \in B_{2\gamma_N\varepsilon_N}(\hat{P})$ to the ARE and the following bound holds

$$\|P^* - \hat{P}\| \leq \Delta_{P,N} := \frac{1}{2\gamma_N L_N}\left(1 - \sqrt{1 - 4\gamma_N^2 L_N \varepsilon_N}\right). \tag{4.10}$$

Proof. The application of Theorem 4.1 requires the validity of criterion (4.2), which is satisfied due to $4\gamma^2\varepsilon L \leq 4\gamma_N^2\varepsilon_N L_N \leq 1$ by assumption. Hence Theorem 4.1 is applicable, where we used the corresponding bounds, and results in the upper bound (4.10). □

In the following sections we see how suitable upper bounds ε_N and γ_N can be obtained, such that an online complexity independence of n is guaranteed. We want to emphasize, that the above bound in Proposition 4.4 can theoretically be made arbitrarily sharp, as the residual norm ε_N is directly dependent on the quality of the reduced space. In particular, if the validity criterion (4.9) is not valid, the reduced space should be improved (*e.g.* by some additional greedy steps or by rerunning the greedy procedure with lower tolerances). A simple but important consequence of the Propositions 3.2 and 4.4 is the correct prediction of zero error when the reduced space contains the complete low rank factor for a given parameter:

Corollary 4.5 (Zero Error Prediction)**.** *Let the assumptions of Proposition 3.2 hold, i.e. in particular* $\operatorname{colspan}(P) \subset \operatorname{colspan}(W)$*. Assume that* $\varepsilon_N := \varepsilon$*. Then* $\Delta_{P,N} = 0$*.*

Proof. From Proposition 3.2 we can conclude $\hat{P} = P^*$ and hence $\mathcal{R}(\hat{P}) = \mathcal{R}(P^*) = 0$ resulting in $\varepsilon = 0$ and hence $\Delta_{P,N} = 0$ from Proposition 4.4. □

Note that Theorem 4.1 and Proposition 4.4 are valid for the approximation of any solution to the ARE. In the next section we establish a criterion that allows us to prove that P^* is indeed the stabilizing solution.

4.2. Stabilizing solution

In the sequel, we will often consider matrices of the form $E^{-1}(A - BR^{-1}B^TPE)$ and thus abbreviate $Y(P) := A - BR^{-1}B^TPE$, where $P \in \mathbb{R}^{n\times n}$ is an arbitrary matrix. We are usually only interested in finding the stabilizing solution P^* of the ARE, which is characterized by its symmetry, positive-semidefiniteness and stabilization property. The latter means, that the matrix $E^{-1}Y(P^*)$ is stable.

In order to make sure that the matrix P^*, which exists due to Theorem 4.1, is indeed the stabilizing solution, we need additional criteria and conditions. Symmetry and positive-semidefiniteness are always satisfied when we use the Galerkin-procedure as it is described in Section 3, as the reduced solution P_N is symmetric and positive semi-definite and those properties are inherited by the approximation WP_NW^T. Thus, it remains to prove the stability of $(E, Y(P^*))$ to get the desired result. One possible way how this can be done is to make use of the stability of the approximation $(E, Y(\hat{P}))$, which can be verified *a posteriori*. For this purpose, we begin with an important characterization of $\gamma = \|\mathcal{L}_{\hat{P}}^{-1}\|$:

Lemma 4.6. *Let* $Y(\hat{P})$ *be stable and let* $H \in \mathbb{R}^{n\times n}$ *solve* $Y(\hat{P})^THE + E^TY(\hat{P})E = -I_n$*. Then* $\gamma = \|H\|$*.*

Proof. See for example [58]. □

The next proposition provides a way to prove the stability of $E^{-1}Y(P^*)$. Note that a similar result in the context of Krylov-based approximations for the ARE has recently been formulated in [54] independently.

Proposition 4.7. *Let the assumptions of Theorem 4.1 be satisfied. Let* $P^* \in B_{2\gamma\varepsilon}(\hat{P})$ *be the ARE solution that is being approximated together with the error estimate* Δ_P *according to Theorem 4.1. Let* $(E, Y(\hat{P}))$ *be stable. Then: if* $2\gamma\|BR^{-1}B^T\|\frac{\|E\|}{\|E^{-1}\|}\Delta_P \leq 1$*, the matrix pair* $(E, Y(P^*))$ *is stable and* P^* *is the unique stabilizing solution of the ARE.*

Proof. The proof is based on Theorem 2.2 in [23], which we cite for convenience: "Let $\tilde{Y}$ be stable and let $\tilde{H}$ satisfy $\tilde{Y}^T\tilde{H} + \tilde{H}\tilde{Y} = -I_n$. Let $\Delta\tilde{Y}$ satisfy $\|\Delta\tilde{Y}\| \leq 1/(2\|\tilde{H}\|)$. Then $\tilde{Y} + \Delta\tilde{Y}$ is stable". In our case, we set $\tilde{Y} := E^{-1}Y(\hat{P})$ and $\Delta\tilde{Y} := E^{-1}BR^{-1}B^T(P^* - \hat{P})E$, then $E^{-1}Y(P^*) = E^{-1}Y(\hat{P}) + E^{-1}\Delta\tilde{Y}$. We apply the above cited theorem to this setting: Let $\tilde{H} \in \mathbb{R}^{n\times n}$ be the unique solution to

$$Y(\hat{P})^TE^{-T}\tilde{H} + \tilde{H}E^{-1}Y(\hat{P}) = -I_n.$$

We now substitute $\tilde{H} = E^THE$ and see that H solves $Y(\hat{P})^THE + E^THY(\hat{P}) = -I_n$. We thus get

$$\gamma = \|H\| = \|E^{-T}\tilde{H}E^{-1}\| \leq \|E^{-1}\|^2\|\tilde{H}\| \quad \Rightarrow \quad \frac{1}{\|\tilde{H}\|} \leq \frac{\|E^{-1}\|^2}{\gamma}. \tag{4.11}$$

We can now apply the theorem and get stability of $E^{-1}Y(P^*)$ when the following inequality holds true:

$$\|E^{-1}\Delta Y\| \leq \frac{1}{2\|\tilde{H}\|} \overset{(4.11)}{\leq} \frac{\|E^{-1}\|^2}{2\gamma} \quad \Leftrightarrow \quad 2\gamma\|E^{-1}\Delta Y\|\frac{1}{\|E^{-1}\|^2} \leq 1.$$

Using $\|E^{-1}\Delta Y\| = \|E^{-1}BR^{-1}B^T\|\Delta_P\|E\| \leq \|E^{-1}\|\|E\|\|BR^{-1}B^T\|\Delta_P$, and the uniqueness of the stabilizing solution due to the detectability and stabilizability assumption, we get the desired result. □

For the application of the above theorem, we need to verify the stability of $(E, Y(\hat{P}))$ *a posteriori*. This usually requires the calculation of the eigenvalue with the largest real part, which should be avoided in the RB setting to retain online efficieny. To avoid this, we propose a sufficient condition for proving the stability, which makes use of the properties of the logarithmic norm:

Proposition 4.8 (Sufficient Condition for Stability by Exponential Bounds)**.** *Let* (E, A) *be stable. Then* $(E, Y(\hat{P}))$ *is stable whenever one of the following requirements is valid*

1. $\nu_E[Y] < 0$
2. $\nu_E[A] + \nu_E[BR^{-1}B^T\hat{P}E] < 0$
3. $\nu_E[A] + \|E^{-1}BR^{-1}B^T\hat{P}E\|_F < 0.$

Proof. If $\nu_E[Y] < 0$, the stability follows from $\|\mathrm{e}^{E^{-1}Yt}\| \leq \mathrm{e}^{\nu[E^{-1}Y]t} \to 0$ as $t \to \infty$. The inequality $\nu_E[Y] = \nu[E^{-1}A - E^{-1}BR^{-1}B^T\hat{P}E] \leq \nu[E^{-1}A] + \nu[-E^{-1}BR^{-1}B^T\hat{P}E]$ implies the second statement. Finally, the bound $\nu[E^{-1}BR^{-1}B^T\hat{P}E] \leq \|E^{-1}BR^{-1}B^T\hat{P}E\| \leq \|E^{-1}BR^{-1}B^T\hat{P}E\|_F$ proves the last result. □

For more details and proofs of the used properties for the logarithmic norm $\nu[\cdot]$ we refer to [55]. Applying this Lemma requires a way to calculate or approximate the logarithmic norm $\nu_E[A]$. This can for example be implemented efficiently by employing the inverse power iteration. Furthermore, more sophisticated methods like the successive constraint method or the min-Θ approach allow the rapid calculation of lower bounds of stability factors with a complexity independent of n which could as well be applied to our situation, see for example [27, 45]. Especially the last criterion of Proposition 4.8 is easy to check *a posteriori* since it only involves the calculation of the norm of $\|E^{-1}BR^{-1}B^T\hat{P}E\|_F$ which can be efficiently performed due to a possible offline/online decomposition, see Section 5.

Note that Proposition 4.8 has only limited applicability, since the logarithmic norm of a stable matrix $(E, Y(\hat{P}))$ can be positive. A more detailed analysis indeed shows, that the value $\nu_E[Y(\hat{P})]$ is an indicator for the transient growth at time $t = 0$, since

$$\frac{\mathrm{d}}{\mathrm{d}t^+} \|\mathrm{e}^{E^{-1}Yt}\|\Big|_{t=0} = \nu_E[Y],$$

where $\frac{\mathrm{d}}{\mathrm{d}t^+}$ is the right-hand derivative, see for instance ([25], Prop. 5.5.8). Hence $\nu_E[Y] > 0$ only shows that the norm of the matrix exponential initially grows and it does not make any predictions for the long term behaviour of $\|\mathrm{e}^{E^{-1}Yt}\|$.

Proposition 4.8 furthermore indicates, that the magnitude of the matrix R could be a way to ensure stability for the closed loop system whenever the requirement $\nu_E[A] + \|E^{-1}BR^{-1}B^T\hat{P}E\|$ for the application of Proposition 4.8 is not fulfilled: scaling R to large values decreases the norm of $\|E^{-1}BR^{-1}B^T\hat{P}E\|$ and can finally ensure that one of the conditions of Lemma 4.8 holds true. However, if we relate the ARE again to the LQR problem in Example 2.2, we see that large values of R introduce a strong penalization of the control and thus lead to weaker control influence, which is not always desirable.

4.3. The constant γ

One of the main ingredients in the error estimation theory shown in the previous section is the constant γ. It is crucial to have a good upper bound or a suitable approximation γ_N as it enters the validity criterion (4.2) quadratically and also enters in the bound (4.3). We recall the definition of γ:

$$\gamma = \|\mathcal{L}_{\hat{P}}^{-1}\| = \|\, D\mathcal{R}|_{\hat{P}}^{-1}\,\|, \quad \text{with} \quad \mathcal{L}_{\hat{P}}(N) = Y(\hat{P})NE + E^TNY(\hat{P}).$$

We see that $\mathcal{L}_{\hat{P}}$ is a Lyapunov-operator, for which the theory has already been extensively studied, *cf.* [28, 40, 46, 47, 57, 58], however only few work has been dedicated to the efficient calculation of γ in the large scale

context. Recalling Lemma 4.6, we see that γ can be calculated directly by solving a Lyapunov equation of the type

$$Y(\hat{P})^T H E + E^T H Y(\hat{P}) = -I_n \tag{4.12}$$

and setting $\gamma = \|H\|$. As the right hand side has no low-rank structure, the solution H has very likely also no low-rank structure, which makes the Lyapunov equation extremely hard to solve, since no structure (except of potential sparsity of $Y(\hat{P})$ and E) is available. A direct calculation usually requires at least $\mathcal{O}(n^3)$ and iterative methods fail at latest when it comes to storing the dense n^2 dimensional solution matrix for large n. We thus need different approaches and upper bounds to be able to cope with these issues.

We begin with giving an explicit formula for the calculation of the solution to the Lyapunov equation:

Lemma 4.9. *Let $Y(\hat{P}), E \in \mathbb{R}^{n\times n}$ be such that $(E, Y(\hat{P}))$ is stable. Then the inverse of the operator $\mathcal{L}_{\hat{P}}$ is explicitly given by*

$$\mathcal{L}_{\hat{P}}^{-1}(X) = -E^{-T} \int_0^\infty \mathrm{e}^{(E^{-1}Y(\hat{P}))^T t} X \mathrm{e}^{(E^{-1}Y(\hat{P}))t} \mathrm{d}t E^{-1}, \quad X \in \mathbb{R}^{n\times n}. \tag{4.13}$$

Proof. The proof makes use of the equivalence of the generalized Lyapunov equation (4.12) and the Lyapunov equation in the case where E is nonsingular. The latter is defined as $\tilde{\mathcal{L}}(X) := (E^{-1}Y(\hat{P}))^T X + X(E^{-1}Y(\hat{P}))$. For this representation, a result from [38] shows, that the inverse operator is explicitly given by

$$\tilde{\mathcal{L}}^{-1}(X) = -\int_0^\infty \mathrm{e}^{(E^{-1}Y(\hat{P}))^T t} X \mathrm{e}^{(E^{-1}Y(\hat{P}))t} \mathrm{d}t.$$

On the other hand, we can easily verify $\mathcal{L}_{\hat{P}}(X) = \tilde{\mathcal{L}}(E^T X E)$ and hence we have $\mathcal{L}_{\hat{P}}^{-1}(X) = E^{-T}\tilde{\mathcal{L}}^{-1}(X)E^{-1}$ which proves the claimed result. □

Remark 4.10. If we again consider the case of the LQR Example 2.2, a more detailed discussion of γ reveals deeper insights to interesting properties of the closed loop LTI system with $E = I_n$. The closed loop system is defined as $\dot{x}(t; x_0) = (A - BR^{-1}B^T P)x(t; x_0)$ together with $x(0; x_0) = x_0$. The discussion in [35] shows that there is a close link between the "damping" of the closed-loop solution $x(t; x_0)$ and the constant γ:

$$\sqrt{\gamma} = \max_{\|x_0\|=1} \|x(\cdot, x_0)\|_{L^2([0,\infty),\mathbb{R}^n)}.$$

This equivalence shows that large γ values indicate a weak damping of the closed loop solution.

In the following we will discuss different possibilities to approximate γ. We will divide this in rigorous bounds and non-rigorous estimates, where the latter will lack an analytic justification but still may be accurate and are rapidly computable.

Rigorous Bounds for γ

The explicit formula (4.13) in Lemma 4.9 shows that an upper bound for γ can be obtained once a bound for the norm of the matrix exponential $\mathrm{e}^{E^{-1}Y(\hat{P})t}$ is available. We state one possibility which is based on the logarithmic norm.

Lemma 4.11. *Let $\nu := \nu_E[E^{-1}Y(\hat{P})] < 0$. Then the following upper bound for γ is valid:*

$$\gamma \le \gamma_{N,1} := -\frac{\|E^{-1}\|^2}{2\nu}. \tag{4.14}$$

Proof. After bounding the explicit solution formula (4.13), it follows

$$\gamma = \|\mathcal{L}^{-1}(-I_n)\| \leq \|E^{-1}\|^2 \int_0^\infty \|\mathrm{e}^{E^{-1}Yt}\|^2 \mathrm{d}t \leq \|E^{-1}\|^2 \int_0^\infty \mathrm{e}^{2\nu t}\, \mathrm{d}t = -\frac{\|E^{-1}\|^2}{2\nu},$$

which proves the claimed bound. □

The calculation of $\|E^{-1}\|$ can be done without an explicit formation of the inverse matrix, by using the equality $\|E^{-1}\| = \frac{1}{\sigma_{\min}}$ with $\sigma_{\min}$ denoting the smallest singular value of E. Note that the logarithmic norm $\nu_E[Y(\hat{P})]$ is not available in the online phase, unless the largest eigenvalue of an n-dimensional matrix is calculated, which can be infeasible. However, by using the same arguments as in the proof of Proposition 4.8, $\nu_E[Y(\hat{P})]$ can be replaced by the upper bound $\nu_E[A] + \|E^{-1}BR^{-1}B^T\hat{P}E\|_F$ as long as the latter is less than 0, which can be calculated in $\mathcal{O}(N)$:

Corollary 4.12. *If $\nu_E[A] + \|E^{-1}BR^{-1}B^T\hat{P}E\|_F < 0$, the upper bound*

$$\gamma \leq \gamma_{N,2} := -\frac{\|E^{-1}\|^2}{2(\nu_E[A] + \|E^{-1}BR^{-1}B^T\hat{P}E\|_F)} \tag{4.15}$$

is valid.

Although many problems that describe physically stable phenomena yield LTI systems with $\nu_E[A] < 0$, the closed loop system with the applied feedback control must not be dissipative. In this case the bounds in Lemma 4.11 and Corollary 4.12 cannot be applied and other approaches must be used.

Interpolation-Based Approximation of γ

Besides the rigorous error bounds presented above, one can also use interpolation methods that may not lead to rigorous bounds but provide very efficient and often quite accurate approximations.

The first method we are proposing is to use a kernel interpolation of the γ values over $\mathcal{P}$: let $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M\} = \mathcal{P}_{\text{train},\gamma} \subset \mathcal{P}$ be a finite set of training parameters with corresponding values $(\gamma(\boldsymbol{\mu}_i))_{i=1}^M$, computed by using the Lyapunov equation from Lemma 4.6. The goal is then to find a function $\Gamma(\boldsymbol{\mu}) : \mathcal{P} \to \mathbb{R}$ that interpolates between the calculated γ values. If the points in $\mathcal{P}_{\text{train},\gamma}$ are located on a structured grid in the parameter domain, techniques like linear, cubic or spline interpolation can be employed to construct $\Gamma(\boldsymbol{\mu})$. Large parameter dimensions or a fine grid resolution for $\mathcal{P}_{\text{train},\gamma}$ can result in a very large number of training points such that the procedure might become too expensive, as solving the corresponding Lyapunov equation is a very time- and memory-consuming task. A better and more flexible approach is to use scattered data interpolation techniques such as kernel interpolation. With those methods, the sampling points must not be located on a structured set but can be chosen for example as random elements from the parameter domain. The idea behind those techniques is to choose a kernel function $k(x, y)$ and compute weights $\boldsymbol{\alpha} = (\alpha_i)_{i=1}^M$, such that the interpolation takes the form

$$\Gamma(\boldsymbol{\mu}) := \sum_{i=1}^{M} \alpha_i k(\boldsymbol{\mu}, \boldsymbol{\mu}_i), \quad \text{with} \quad \boldsymbol{\mu}_i \in \mathcal{P}_{\text{train},\gamma}.$$

Once suitable weights α_i are calculated, the interpolation can be evaluated rapidly for new values $\boldsymbol{\mu}$. We use so called thin plate splines as kernel function $k(x, y) = \|x - y\|^2 \ln(\|x - y\|)$ as this turns out to provide good results. Other possible choices are the Gaussian function $k(x, y) = \mathrm{e}^{-\|x-y\|/\sigma^2}$ or multiquadratic radial basis functions $k(x, y) = \sqrt{1 + a\|x - y\|^2}$. The weights can be determined by solving the linear equation system $K\boldsymbol{\alpha} = \mathbf{g}$, where $K = (k(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j))_{i,j=1,\ldots M}$, $\mathbf{g} := (\gamma(\boldsymbol{\mu}_i))_{i=1,\ldots,M}$. We then define the kernel-based interpolation to γ as $\gamma_{N,\text{interpol}}(\boldsymbol{\mu}) := \Gamma(\boldsymbol{\mu})$. For more details concerning the scattered data approximation by using radial basis functions we refer to the monograph [63] and to [41,65].

Projection Based Calculation of γ

A rather different approach is to use a projection based technique for reducing the dimension of the Lyapunov equation $Y(\hat{P})^T HE + E^T HY(\hat{P}) = -I_n$ for the calculation of γ. As we are only interested in the calculation of the 2-norm of H, which is due to the symmetry of H equal to the largest eigenvalue of H, we can try to build a suitable linear subspace for approximating the eigenspace corresponding to its largest eigenvalue: let $V_{\mathcal{L}} \in \mathbb{R}^{n \times N_{\mathcal{L}}}$ be a matrix with orthonormal columns. We then define the approximation $\hat{H} := V_{\mathcal{L}} H_{N_{\mathcal{L}}} V_{\mathcal{L}}^T$, where $H_{N_{\mathcal{L}}} \in \mathbb{R}^{N_{\mathcal{L}} \times N_{\mathcal{L}}}$ solves the $N_{\mathcal{L}}$-dimensional Lyapunov equation

$$Y_{N_{\mathcal{L}}}(\hat{P})^T H_{N_{\mathcal{L}}} E_{N_{\mathcal{L}}} + E_{N_{\mathcal{L}}}^T H_{N_{\mathcal{L}}} Y_{N_{\mathcal{L}}}(\hat{P}) = -I_{N_{\mathcal{L}}},$$

where the reduced matrices are defined as $Y_{N_{\mathcal{L}}}(\hat{P}) := V_{\mathcal{L}}^T Y(\hat{P}) V_{\mathcal{L}}$ and $E_{N_{\mathcal{L}}} := V_{\mathcal{L}}^T E V_{\mathcal{L}}$. Note that this procedure is very similar to the approximation of the ARE, as discussed in Section 3. We can now define the approximation

$$\gamma_{N,\mathrm{proj}}(\boldsymbol{\mu}) := \|\hat{H}\| = \|H_{N_{\mathcal{L}}}\|. \tag{4.16}$$

For building the reduced space, we again consider a finite training set $\mathcal{P}_{\mathrm{train}} \subset \mathcal{P}$ with $|\mathcal{P}_{\mathrm{train}}| = T$, and construct a linear subspace consisting of suitable snapshots $\mathrm{colspan}(V_{\mathcal{L}}) \subset \mathrm{span}(\{v_1, \dots, v_T\})$. Since we want to approximate the dominant eigenvalue of H, a good choice for the reduced space is certainly to choose the snapshots v_i as the eigenvectors corresponding to the largest eigenvalues of H. However, this procedure can be very expensive, as the full matrices H for the training parameters must be available. This is infeasible in large scale scenarios, since the Lyapunov equation for H cannot be solved efficiently. We thus propose a different approach, which is motivated by the relationship between the weakest damping of the closed loop solution $\sqrt{\gamma} = \|x(t; x_0)\|_{L^2([0,\infty),\mathbb{R}^n}$ and γ, see also Remark 4.10. A heuristic approach to find the initial value with the slowest decay in $\|x(t; x_0)\|$, is to choose the initial value as the eigenvector corresponding to the eigenvalue with the smallest magnitude of the n-dimensional eigenvalue problem $(A - BR^{-1}B^T \hat{P}E)x = \lambda Ex$, which can be implemented quite efficiently. Thus, we extend the offline phase and precalculate T eigenvalue problems, collect all eigenvectors which correspond to the smallest magnitude and build the reduced basis $V_{\mathcal{L}}$ by applying a SVD to the snapshots $\{v_i\}_{i=1}^T$. In order to obtain a non-complex basis, we only take the real-parts of the eigenvectors in the SVD. We give a comparison of the presented methods in Section 6.

5. Offline/Online decomposition

The key to an efficient implementation of the whole procedure is a separation of expensive calculations in the offline phase, which then enables a rapid online phase. This is possible when we assume that all data matrices have a special structure that can be exploited to identify precomputable components, which can be reused in the online phase. The presented procedure relies on the parameter separability of the data and system matrices, this means we assume representations of the form

$$\begin{aligned} A(\boldsymbol{\mu}) &= \sum_{q_A}^{Q_A} \Theta_A^{q_A}(\boldsymbol{\mu}) A_{q_A}, \quad B(\boldsymbol{\mu}) = \sum_{q_B}^{Q_B} \Theta_B^{q_B}(\boldsymbol{\mu}) B_{q_B} \\ C(\boldsymbol{\mu}) &= \sum_{q_C}^{Q_C} \Theta_C^{q_C}(\boldsymbol{\mu}) C_{q_C}, \quad E(\boldsymbol{\mu}) = \sum_{q_E}^{Q_E} \Theta_E^{q_E}(\boldsymbol{\mu}) E_{q_E}. \end{aligned} \tag{5.1}$$

If the data matrices are not given in the above form, techniques like empirical interpolation can be employed to achieve such approximate decompositions, see [5,12] or [64] for a function, operator or matrix discrete empirical interpolation approach. This structure allows the precalculation of many parts of the residual norm and other quantities in an offline step. The online step then only requires matrix operations with a complexity depending on the quantities Q_A, Q_B, Q_C, Q_E and the dimension of the reduced system N. Hence, those numbers should be preferably small. Note that we do not pose any requirements on $R(\boldsymbol{\mu})$ and $Q(\boldsymbol{\mu})$. Since they are of low

dimension and also not touched by the reduction procedure, we can explicitly calculate them in the online phase. The calculation of the residual norm $\varepsilon(\boldsymbol{\mu}) = \|\mathcal{R}(\hat{P}(\boldsymbol{\mu}))\|$ can take an excessive amount of time for large n. This is due to the matrix multiplications that are necessary for forming the residual matrix and the calculation of the norm. We will now show how the residual norm can be obtained in a complexity completely independent of the high dimension.

Clearly the induced 2-norm of $\mathcal{R}(\hat{P}(\boldsymbol{\mu}))$ is not feasible for large scale applications, since this would require the knowledge of the largest eigenvalue of an $n \times n$ matrix. A better choice involves the Frobenius norm, which is an upper bound $\varepsilon_N(\boldsymbol{\mu})$ for the induced 2-norm of a matrix. Indeed, the norm equivalence inequality $\|A\| \leq \|A\|_F \leq \sqrt{n}\|A\|$ indicates, that it does not differ more than a factor $\sqrt{n}$ from the 2 norm of $A \in \mathbb{R}^{n\times n}$. Indeed, in practice we often observed a very accurate approximation of the Euclidean-norm. We use the trace representation of the Frobenius norm $\|\mathcal{R}(\hat{P})\|_F = \sqrt{\operatorname{tr}(\mathcal{R}(\hat{P})^T\mathcal{R}(\hat{P}))}$ and obtain the following decomposition of the residual:

$$\begin{aligned}\|\mathcal{R}(\hat{P}))\|_F^2 = {} & \operatorname{tr}(C^TQCC^TQC) + 4\operatorname{tr}(C^TQCA^T\hat{P}E) - 4\operatorname{tr}(E^T\hat{P}BR^{-1}B^T\hat{P}EA^T\hat{P}E) \\ & - 2\operatorname{tr}(E^T\hat{P}BR^{-1}B^T\hat{P}EC^TQC) + \operatorname{tr}(E^T\hat{P}BR^{-1}B^T\hat{P}EE^T\hat{P}BR^{-1}B^T\hat{P}E) \\ & + 2\operatorname{tr}(E^T\hat{P}AA^T\hat{P}E) + 2\operatorname{tr}(E^T\hat{P}AE^T\hat{P}A).\end{aligned} \tag{5.2}$$

We here made use of the identity $\operatorname{tr}(ABC) = \operatorname{tr}(CAB)$ and $\operatorname{tr}(A^TB) = \operatorname{tr}(AB^T)$. If we now use the definition of the reconstruction of the reduced solution $\hat{P} = WP_NW^T$, we can exploit the structure further and identify matrices that can be treated by an offline/online decomposition, for example

$$\begin{aligned}\operatorname{tr}(C^TQCA^T\hat{P}E) &= \operatorname{tr}(\underline{W^TEC^TQC}\underline{A^TW}P_N) \\ \operatorname{tr}(E^T\hat{P}BR^{-1}B^T\hat{P}EA^T\hat{P}E) &= \operatorname{tr}(\underline{W^TEE^TW}P_N\underline{W^TB}R^{-1}\underline{B^TW}P_N\underline{W^TEA^TW}P_N).\end{aligned}$$

All underlined matrices are low dimensional, *i.e.* the dimension only depends on N, m and p. By using the parameter separability, they can be assembled rapidly. For example the product $W^TEC^TQCA^TW$ can be treated in the following way[3]:

$$W^TE(\boldsymbol{\mu})C(\boldsymbol{\mu})^TQ(\boldsymbol{\mu})C(\boldsymbol{\mu})A(\boldsymbol{\mu})W = \left(\sum_{i=1}^{Q_E}\sum_{j=1}^{Q_C}(\Theta_E^i\Theta_C^j)(\boldsymbol{\mu})\underbrace{W^TE_iC_j^T}_{=:\mathbf{M}_1^{i,j}}\right)Q(\boldsymbol{\mu})\left(\sum_{k=1}^{Q_C}\sum_{l=1}^{Q_A}(\Theta_C^k\Theta_A^l)(\boldsymbol{\mu})\underbrace{C_kA_lW}_{=:\mathbf{M}_2^{k,l}}\right).$$

This sum only requires matrix operations on $N \times N$ matrices, once the parameter independent components $\mathbf{M}_1^{i,j}$ and $\mathbf{M}_2^{k,l}$ are calculated and stored in an offline step. All other parts in the Frobenius norm containing $\hat{P}$ can be treated in a similar way. The matrix $R(\boldsymbol{\mu})$ is inverted directly, since it is only a small m dimensional matrix. The overall complexity for the calculation of the Frobenius norm in the online step is

$$\mathcal{O}\left(Q_C^2 + p + N + N^2\left(Q_EQ_AQ_C^2 + Q_E^2 + Q_B^2 + Q_A^2 + Q_EQ_A + Q_E^2Q_C^2\right)\right). \tag{5.3}$$

Hence, if the parameter independent components are precalculated and stored in an offline phase, the Frobenius norm of the residual can be assembled in a complexity independent of n. The calculation time depends quartic on the number of terms in the expansion of $C(\boldsymbol{\mu})$ and quadratically on $C(\boldsymbol{\mu})$, $E(\boldsymbol{\mu})$ and $B(\boldsymbol{\mu})$.

Finally, we note that the computable *a posteriori* error bound for the ARE, as stated in Proposition 4.4, can be calculated online in a complexity independent of n: we have already seen that the Frobenius norm of the residual can be calculated online, which indeed gives an upper bound to the residual norm. Furthermore, we assume that a rapidly computable approximation $\gamma_N(\boldsymbol{\mu})$ is available, by using any of the methods presented in Section 4.3. An upper bound on the validity check $4L\gamma_N^2\varepsilon_N \leq 1$ can also be decomposed in an offline/online fashion, by using the same techniques for the norm calculations in L as for the residual. If $Q_E = 1$ or $Q_B = 1$, the direct calculation of $\|B(\boldsymbol{\mu})\|^2 = (\Theta_B^1(\boldsymbol{\mu}))^2\|B_1\|^2$ and $\|E(\boldsymbol{\mu})\|^2 = (\Theta_E^1(\boldsymbol{\mu}))^2\|E_1\|^2$ might yield sharper validity criteria, since the Frobenius norm can differ from the Euclidean norm by a factor of $\sqrt{n}$.

[3]We use the abbreviation $(\Theta_1\Theta_2)(\boldsymbol{\mu}) := \Theta_1(\boldsymbol{\mu})\Theta_2(\boldsymbol{\mu})$.

6. Numerical examples

We will now present two examples that show, that the RB-ARE approach together with the Low Rank Factor Greedy procedure indeed yield accurate approximations to the full solutions of the ARE. The examples stem from the solution of parametric linear quadratic optimal control problems, the so called linear quadratic regulator (LQR), see also Example 2.2 for a more detailed description. The first problem is a test model, intended to provide numerical examples for the theoretical results, that were obtained in Section 4. The second model is a real-world application and shows the great runtime benefit of the RB-ARE approach in multi-query scenarios.

All calculations were performed on a Debian Linux compute-server with eight Intel (R) Xeon CPU E7 CPUs, each with 2.13 Ghz, and 256 GB RAM. The algorithms were implemented and tested in MATLAB R2015b (8.6.0.267246), where we used parallel loops whenever possible. The examples and benchmarks are implemented within the MATLAB package RBmatlab, which can be freely obtained from the Morepas website[4]. All high dimensional AREs have been solved by using the package MESS[5].

We will study the basis generation and the error which is induced by the RB-ARE reduction technique. For the basis generation, we choose the normalized residual

$$\Delta(V,\boldsymbol{\mu}) := \frac{\|\mathcal{R}(\hat{P}(\boldsymbol{\mu}))\|_F}{\|C(\boldsymbol{\mu})^T Q(\boldsymbol{\mu}) C(\boldsymbol{\mu})\|_F}, \tag{6.1}$$

as error indicator for the LRFG-algorithm in all of the following examples. This measure is frequently used in numerical solution algorithms for large scale AREs to quantify the approximation quality and as a stopping criterion, see for example [7,50]. Furthermore, due to the offline/online decomposition, the error indicator (6.1) can be cheaply calculated, even in cases where the number of training elements is large. This is possible due to parameter separability, see also Section 5. For the initialization of the LRFG procedure, we choose the desired tolerance $\varepsilon := 10^{-6}$, and as inital basis we calculate the full solution for the first parameter in the training set, and choose the most-dominant POD-mode as initial basis.

6.1. LQ-feedback control of a 2D-heat transfer problem

The first example stems from a semi-discretized heat-transfer control problem on the unit square $\Omega = [0,1]^2$. The stationary version of this thermal-block model is often used as a standard example in the RB community, as its solution variety captures many interesting phenomena. We refer to [17] for details. The model has homogeneous Dirichlet boundary conditions (*i.e.* constant zero temperature) on $\Gamma_D[0,1]\times\{1\}$ and homogeneous Neumann boundary conditions (*i.e.* perfect insulation) on the remaining boundary. We introduce a distributed control $u(t)$ on $\Omega_B := [0.2,0.4]\times[0.4,0.6]$ and take an average temperature measurement $s(t;\mu)$ on the boundary $\Gamma_C = \{0\}\times[0,1]$. The partial differential equation for the temperature $w(\xi,t;\boldsymbol{\mu})$ with $\xi\in\Omega$, $t\geq 0$ and the output of interest thus reads as follows:

$$\begin{aligned}
w_t(\xi,t;\boldsymbol{\mu}) - a(\boldsymbol{\mu};\xi)\Delta w(\xi,t;\boldsymbol{\mu}) &= 5\cdot \mathbf{1}_{\Omega_B}(\xi)u(t) && \text{for } \xi\in\Omega, t>0, && \text{(6.2a)}\\
w(\xi,t;\boldsymbol{\mu}) &= 0 && \text{for } \xi\in\partial\Omega, t\geq 0, && \text{(6.2b)}\\
w(\xi,0;\boldsymbol{\mu}) &= w_0(\xi;\boldsymbol{\mu}), && \text{for } \xi\in\Omega && \text{(6.2c)}\\
s(t;\mu) &:= 5\int_{\Gamma_C} w(\cdot,t;\boldsymbol{\mu}), && \text{for } t\geq 0. && \text{(6.2d)}
\end{aligned}$$

We introduce parameter dependency by defining the diffusion coefficient as $a(\boldsymbol{\mu};\xi) := \mathbf{1}_{\Omega_1}(\xi)\mu_1 + \mathbf{1}_{\Omega_2}(\xi)\mu_2$, where $\mathbf{1}_X(\xi)$ denotes the characteristic function of the set X at the point ξ. This function provides different heat conductivities on the subdomains $\Omega_1 := [0,0.5]\times[0,1]$ and $\Omega_2 := (0.5,1]\times[0,1]$ with $\mu_1,\mu_2\in[1,5]$. We aim at minimizing the cost functional $\int_0^\infty(\mu_Q\|s(t)\|^2 + \mu_R\|u(t)\|^2)\mathrm{d}t$ with weights $\mu_Q\in[0.1,10]$ and $\mu_R\in[0.1,10]$.

[4] http://www.morepas.org

[5] MESS (Matrix Equation Sparse Solver) has been gratefully provided by J. Saak from the MPI Magdeburg.

TABLE 1. Basis sizes, number of high dimensional solves, calculation time t and average number of added basis vectors in each iteration.

tol_i	Basis size	#Solves	$t[s]$	Avg. added
0.000	51	14	286.18	1.00
0.500	51	12	151.82	2.22
0.800	56	10	103.31	4.67
0.900	59	9	84.70	6.56
0.950	60	7	72.57	7.50
0.960	59	7	70.04	8.43
0.970	64	7	75.02	9.14
0.990	66	6	62.13	11.00
0.999	67	5	53.26	13.40
1.000	349	4	761.47	87.25

Overall the model depends on four parameters: $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_Q, \mu_R) \subset \mathcal{P} := [1,5]^2 \times [0.1,1]^2$. The above PDE and the cost functional are semidiscretized in space by using linear finite elements on a regular triangular grid, resulting in an $n = 1.891$ dimensional LQR-control problem with $m = p = 1$:

$$\begin{aligned} &\min \int_0^\infty (x(t;\mu)^T (C^T \mu_Q C) x(t;\mu) + \mu_R u(t)^2) \mathrm{d}t \\ \text{s.t.} \quad & E\frac{\mathrm{d}}{\mathrm{d}t} x(t;\boldsymbol{\mu}) = A_1 x(t;\boldsymbol{\mu}) + \mu_1 A_2 x(t;\boldsymbol{\mu}) + \mu_2 A_3 x(t;\boldsymbol{\mu}) + Bu(t), \\ & Ex(0;\boldsymbol{\mu}) = x_0, \\ & y(t;\mu) = Cx(t;\mu). \end{aligned}$$

The Dirichlet boundary values are mapped by A_1 and the parameter dependency directly yields a parameter separated discrete representation. The training set $\mathcal{P}_{\text{train}}$ for this example is chosen as a set of 400 randomly choosen parameters from the parameter domain.

As a first insight to the basis generation, we investigate the basis sizes and number of iterations of the LRFG-algorithm for varying inner tolerances $\text{tol}_i \in [0,1]$. Table 1 gives an overview of the results. Setting $\text{tol}_i = 0$ means, that in each iteration of the algorithm only one element is added to the basis, what explains the large offline time since many iterations (and thus also high dimensional ARE solves) are required until the greedy procedure terminates. On the other hand, if we set $\text{tol}_i = 1$, we add the full low-rank factor to the basis. This requires less ARE solves to reach the desired tolerance, but results in a very large basis, as no additional compression is used. For values tol_i in between, we see that there is a tradeoff between the basis size, the number of full dimensional solutions and the offline time. An "optimal" value in our case with respect to computation time and basis size is obtained for $\text{tol}_i \approx 0.96$. We will use the corresponding basis with size $N = 59$ in all subsequent examples.

In Figure 1, the error indicator (6.1) is plotted together with the corridor of the true relative errors $\|P(\boldsymbol{\mu}) - \hat{P}(\boldsymbol{\mu})\|_F / \|P(\boldsymbol{\mu})\|_F$ and the mean relative true error for increasing basis sizes. We can conclude that the relative residual indicator (6.1) is a good choice for the construction of the basis. In this example, the greedy algorithm took eight iterations until the desired tolerance is achieved. However, only seven full AREs need to be solved, as the same parameter was revisited during the basis construction.

The crucial ingredient for the calculation of the error estimator is the constant γ. As it was shown in Lemma 4.6, it can be calculated by solving a generalized Lyapunov equation, which however, due to the lack of any low-rank structure, is very expensive to solve. For the 1.891-dimensional thermalblock-example a single solve using the `lyap` command from MATLAB takes about 8 min. We thus need to approximate it and therefore presented a rigorous bound (Cor. 4.12) and two approximation techniques (kernel approximation, projection-based calculation), which we will now compare. For the kernel-approach and the construction of the basis

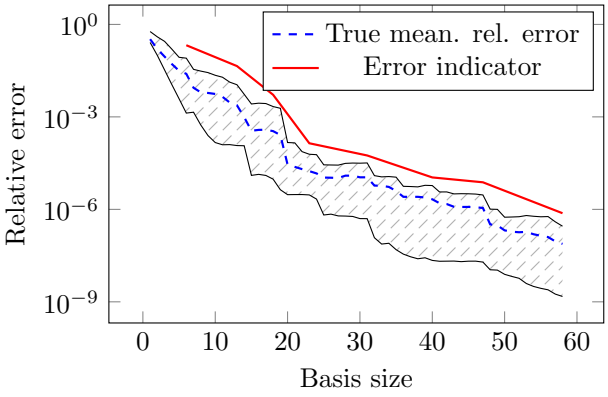

FIGURE 1. Error indicator and true relative error decay for increasing basis sizes. The true error lies within the dashed area.

TABLE 2. Comparison of relative errors for the approximation of γ through the kernel and projection approach. The maximum, mean and variance was taken over the test set $\mathcal{P}_{\text{test}}$.

		Train size				
		10	20	50	80	100
Projection	$\max_{\boldsymbol{\mu}} \eta(\boldsymbol{\mu})$	0.5486	6.2663	0.3740	0.0144	0.0076
	$\text{mean}_{\boldsymbol{\mu}} \eta(\boldsymbol{\mu})$	0.3433	0.4944	0.2280	0.0040	0.0018
	$\text{var}_{\boldsymbol{\mu}} \eta(\boldsymbol{\mu})$	0.0121	0.8974	0.0041	0.0000	0.0000
Kernel	$\max_{\boldsymbol{\mu}} \eta(\boldsymbol{\mu})$	1.3565	0.8120	0.2803	0.2493	0.2175
	$\text{mean}_{\boldsymbol{\mu}} \eta(\boldsymbol{\mu})$	0.7686	0.3420	0.0788	0.0540	0.0223
	$\text{var}_{\boldsymbol{\mu}} \eta(\boldsymbol{\mu})$	0.0727	0.0456	0.0059	0.0036	0.0036

for the projection-based approach, we choose a training set $\mathcal{P}_{\gamma,\text{train}}$ with $|\mathcal{P}_{\gamma,\text{train}}| = 100$ random elements from the parameter domain $\mathcal{P}$. For the kernel approach, we choose the thin-plate spline kernel $k(x, y) := \ln(\|x - y\|)\|x - y\|^2$ as radial basis function.

In Table 2 the maximum $\max_{\boldsymbol{\mu}\in\mathcal{P}_{\text{test}}} \eta(\boldsymbol{\mu})$ and mean $\text{mean}_{\boldsymbol{\mu}\in\mathcal{P}_{\text{test}}} \eta(\boldsymbol{\mu})$ relative approximation error $\eta(\boldsymbol{\mu}) := |\gamma(\boldsymbol{\mu}) - \hat{\gamma}(\boldsymbol{\mu})|/\gamma(\boldsymbol{\mu})$ for both approaches, as well as the variance $\text{var}_{\boldsymbol{\mu}\in\mathcal{P}_{\text{test}}} \eta(\boldsymbol{\mu})$ in the relative error is given. Here $\hat{\gamma}(\boldsymbol{\mu})$ is the corresponding approximation, *i.e.* either the kernel interpolation or the projection-based calculated value. For the calculations, we choose a random parameter test set $\mathcal{P}_{\text{test}} \subset \mathcal{P}$ with 40 random elements. We see that both approaches perform well. However, the overall average relative error of the projection approach is smaller, and especially the variance in the approximation error is significantly lower than the corresponding variance for the kernel approximation errors. A comparison of the red offline preparation time reveals, that the kernel approach suffers from the expensive calculation of the solution of the Lyapunov equation in Lemma 4.6: the offline time for the calculation of the training data takes almost an hour, whereas the calculation of the eigenvectors for the projection-based calculation only requires a couple of seconds. Hence, we can conclude that the projection based approach yields accurate approximations with relatively low computational complexity in the offline phase, but we observed that increasing the number of training parameters does not improve the results any further. On the other hand, the kernel interpolation can be made very precise by adding more sample values or doing a more sophisticated tuning of the kernel parameters, which however is far beyond the scope of this work. We refer to [63] for more details about this topic. The rigorous bound from Corollary 4.12 can be applied in certain parameter regions in this example, but the resulting bound overestimates the true error by a factor of $\approx$200, which renders the validity criterion for the application of the error bound invalid. However, if the system is first transformed to a system without mass matrix, *e.g.* by left-multiplying the state equation

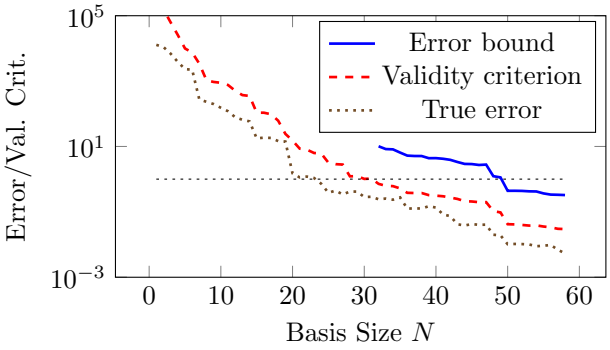

FIGURE 2. Error estimation results.

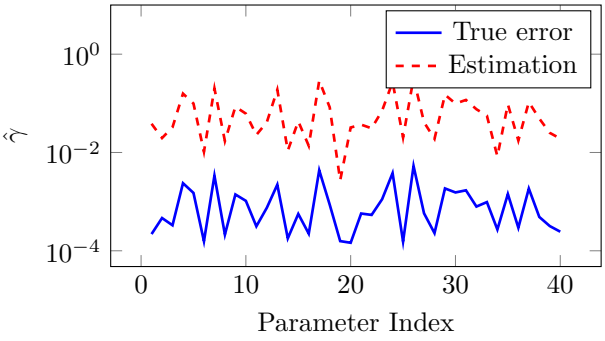

FIGURE 3. True absolute errors and error bound over a random parameter test set.

with E^{-1}, the bound yields sharp values, but important system properties like sparsity or symmetry are usually destroyed.

Finally, we examine the error bound in Proposition 4.4. For this purpose, we calculate the true maximum error, and the maximum error bound along with the maximum value of the validity criterion over the same parameter test set $\mathcal{P}_{\text{test}}$ as for the examples for γ. In order to get rigorous values for the bound, we use the true values for γ. However, when using one of the approximation techniques mentioned earlier, the values of the non-rigorous error estimator only marginally deviate from the rigorous results. The results are depicted in Figure 2. We see, that the error bound from Proposition 4.4 is valid for a basis size $N \geq 31$, which is reflected by the decay of the validity criterion. Note that the error estimation is valid whenever the validity criterion is below one. All reduced solutions in our examples result in stable matrix pairs $(E, Y(\hat{P}(\boldsymbol{\mu}))$ and Proposition 4.7 can be applied, to guarantee that the reduced solution approximates the desired unique stabilizing solution. Furthermore, the error bound nicely reflects the behaviour of the true error. This can be seen in Figure 3, where the true absolute error and the corresponding error bound for all test parameters in $\mathcal{P}_{\text{test}}$ is shown for the maximum basis size $N = 59$.

Finally we want to comment on the calculation time for this example. One high dimensional solve takes about 1.5 s in average. The reduced simulation, including the calculation of the residual norm and the calculation of the error bound, based on the projection approach, takes about 0.2 s. Thus we have an overall speed-up of approximately factor seven. A much larger runtime benefit can be achieved for examples with larger system dimension n, see also the next example.

Optimal Cooling of Rail Profiles

The second example stems from the optimal cooling of rail profiles. We refer to [6] for a detailed description of the model. The semi-discretization of the linearized heat-transfer results in a LTI system with dimension $n = 20.209$:

$$E\frac{\mathrm{d}}{\mathrm{d}t}x(t;\boldsymbol{\mu}) = Ax(t;\boldsymbol{\mu}) + Bu(t), \quad y(t;\mu) = Cx(t;\mu).$$

The system can be controlled through $m = 7$ inputs that model water which is sprayed on the rod, and $p = 6$ outputs are available for the calculation of the control signal and are measured in the cost functional. We parametrize the cost functional associated with this problem by introducing two weights $Q(\boldsymbol{\mu}) := I_6\mu_Q$ and $R(\boldsymbol{\mu}) := I_7\mu_R$ with $\boldsymbol{\mu} := (\mu_Q, \mu_R) \subset [10^{-3}, 10] \times [10^{-3}, 1]$. For the basis construction, we choose 961 logarithmically equidistant points in the parameter domain. One solution of the high dimensional ARE associated with this control problem takes about 105 s.

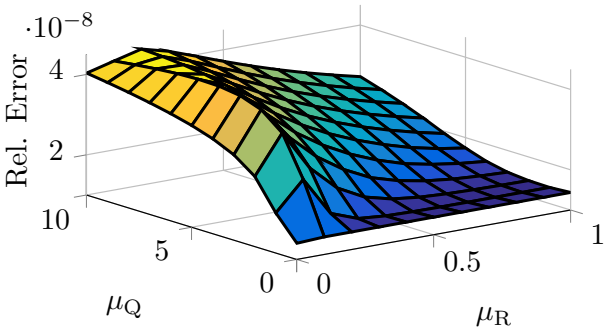

FIGURE 4. Relative error for the approximation of the ARE for the rail model.

In order to speed up the calculation, we first run the LRFG-algorithm for this problem and obtain a reduced basis of size $N = 112$ after approximately 4 min. The solution of the reduced ARE takes only 0.17 s in average, including the calculation of the residual norm, which is an enormous speed-up of about 500. We test the quality of the approximation by calculating the reduced and true solutions of the AREs on a fine grid in the parameter domain. In Figure 4 we see the relative error $\frac{\|P(\boldsymbol{\mu})-\hat{P}(\boldsymbol{\mu})\|_F}{\|P(\boldsymbol{\mu})\|_F}$ on the complete parameter domain, which is always below 10^{-7}. We can conclude, that the RB-ARE approach yields very accurate approximations for this example. We want to emphasize that this enormous speed-up only required two (expensive) offline solves for the construction of the basis.

7. Conclusion

In this paper we have studied the application of the RB approach for the approximation of large scale AREs in a parametric scenario. As it turned out, the state-space approximation with classical model reduction techniques for parametric problems does not capture the feedback information in the system, see [51]. This is the main motivation for the RB-ARE approach, which is developed in this paper. After defining a suitable reduced surrogate equation, the RB-ARE, a new algorithm called low-rank factor greedy was proposed for the basis construction. As the name suggests, the typical low rank structure of the solutions to the ARE has been used for the construction of the reduced basis. The quality of the approximation is certified by a rigorous *a posteriori* error estimation for the solution to the ARE. Splitting the procedure in an online/offline phase allows an online complexity completely independent of the dimension of the original problem. Experiments with LQ feedback control problems for heat conductions show the benefit of applying the RB-ARE technique in multi-query scenarios.

Interesting further potential applications of the RB-ARE technique can be found in many different fields of systems theory, for instance $\mathcal{H}_2$ or $\mathcal{H}_\infty$ optimal control. The numerical study [52] for the application of the RB-ARE approach for parametric $\mathcal{H}_2$ control problems shows that this indeed can yield accurate and fast approximations. This will be part of our future research.

Acknowledgements. The authors would like to thank the German Research Foundation (DFG) for financial support of the project within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart and the Baden-Württemberg Stiftung gGmbH.

References

[1] H. Abou-Kandil, G. Freiling, V. Ionescu and G. Jank, Matrix Riccati Equations in Control and Systems Theory. Springer Science + Business Media (2003).

[2] A.C. Antoulas, Approximation of Large-Scale Dynamical Systems. SIAM Publications, Philadelphia, PA (2005).

[3] A.C. Antoulas, D.C. Sorensen and Y. Zhou, On the decay rate of Hankel singular values and related issues. *Systems & Control Letters* **46** (2002) 323–342.

[4] E. Bänsch, P. Benner, J. Saak and H.K. Weichelt, Riccati-based boundary feedback stabilization of incompressible Navier-Stokes flows. *SIAM J. Sci. Comput.* **37** (2015) A832–A858.

[5] M. Barrault, Y. Maday, N.C. Nguyen and A.T. Patera, An "empirical interpolation" method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci., Series I* **339** (2004) 667–672.

[6] P. Benner and J. Saak, A semi-discretized heat transfer model for optimal cooling of steel profiles. In Dimension Reduction of Large-Scale Systems. Springer-Verlag (2005) 353–356.

[7] P. Benner and J. Saak, Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey. *GAMM-Mitteilungen* **36** (2013) 32–52.

[8] P. Benner and Z. Bujanovi, On the solution of large-scale algebraic Riccati equations by using low-dimensional invariant subspaces. *Linear Algebra Appl.* **488** (2016) 430–459.

[9] J. Borggaard, M. Stoyanov and L. Zietsman, Linear feedback control of a von Kármán street by cylinder rotation. In *Proc. of the 2010 American Control Conference*. IEEE (2010) 5674–5681.

[10] G. Caloz and J. Rappaz, Numerical analysis for nonlinear and bifurcation problems. In Techniques of Scientific Computing (Part 2), Vol. 5 of *Handbook of Numerical Analysis*. Elsevier (1997) 487–637.

[11] G. Dahlquist, Convergence and stability in the numerical integration of ordinary differential equations. *Math. Scand.* **4** (1956) 33–53.

[12] M. Drohmann, B. Haasdonk and M. Ohlberger, Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM J. Sci. Comput.* **34** (2012) A937–A969.

[13] A.R. Ghavimi and A.J. Laub, Backward error, sensitivity, and refinement of computed solutions of algebraic Riccati equations. *Numer. Lin. Algebra Appl.* **2** (1995) 29–49.

[14] M.A. Grepl and A.T. Patera, *A posteriori* error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: M2AN* **39** (2005) 157–181.

[15] C. Großmann and H.G. Roos, Numerische Behandlung partieller Differentialgleichungen. Vieweg+Teubner Verlag (2005).

[16] B. Haasdonk, Convergence rates of the POD–Greedy method. *ESAIM: M2AN* **47** (2013) 859–873.

[17] B. Haasdonk, Reduced basis methods for parametrized PDEs – a tutorial introduction for stationary and instationary problems. SimTech preprint, IANS, University of Stuttgart, Germany, 2014. Chapter in: Model Reduction and Approximation: Theory and Algorithms, edited by P. Benner, A. Cohen, M. Ohlberger and K. Willcox. SIAM, Philadelphia (2017).

[18] B. Haasdonk, M. Dihlmann and M. Ohlberger, A training set and multiple basis generation approach for parametrized model reduction based on adaptive grids in parameter space. *Math. Comp. Model. Dyn. Systems* **17** (2011) 423–442.

[19] B. Haasdonk and M. Ohlberger, Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: M2AN* **42** (2008) 277–302.

[20] B. Haasdonk and M. Ohlberger, Efficient reduced models and a posteriori error estimation for parametrized dynamical systems by offline/online decomposition. *Math. Comp. Model Dyn.* **17** (2011) 145–161.

[21] B. Haasdonk, M. Ohlberger and G. Rozza. A reduced basis method for evolution schemes with parameter-dependent explicit operators. *ETNA, Electron. Trans. Numer. Anal.* **32** (2008) 145–161.

[22] E. Heinz, Beiträge zur Störungstheorie der Spektralzerleung. *Math. Ann.* **123** (1951) 415–438.

[23] G. Hewer and C. Kenney, The sensitivity of the stable Lyapunov equation. *SIAM J. Control Optimiz.* **26** (1988) 321–344.

[24] M. Heyouni and K. Jbilou, An extended block Arnoldi algorithm for large-scale solutions of the continuous-time algebraic Riccati equation. *Electron. Trans. Numer. Anal.* **33** (2009) 53–62.

[25] D. Hinrichsen and A.J. Pritchard, Mathematical Systems Theory I: Modelling, State Space Analysis, Stability and Robustness. Springer (2005).

[26] K. Hiramoto, H. Doki and G. Obinata, Optimal sensor/actuator placement for active vibration control using explicit solution of algebraic Riccati equations. *J. Sound Vibration* **229** (2000) 1057–1075.

[27] D.B.P. Huynh, D.J. Knezevic, Y. Chen, J.S. Hesthaven and A.T. Patera, A natural-norm successive constraint method for inf-sup lower bounds. *Comput. Methods Appl. Mech. Eng.* **199** (2010) 1963–1975.

[28] I.M. Jaimoukha and E.M. Kasenally, Krylov subspace methods for solving large Lyapunov equations. *SIAM J. Numer. Anal.* **31** (1994) 227–251.

[29] K. Jbilou, Block Krylov subspace methods for large algebraic Riccati equations. *Numer. Algorithms* **34** (2003) 339–353.

[30] K. Jbilou, An Arnoldi based algorithm for large algebraic Riccati equations. *Appl. Math. Lett.* **19** (2006) 437–444.

[31] C. Johnson, Numerical solution of partial differential equations by the finite element method. Courier Corporation (1989).

[32] R.E. Kalman and R.S. Bucy, New results in linear filtering and prediction theory. *J. Basic Eng.* **83** (1961) 95.

[33] R.E. Kalman, Contributions to the theory of optimal control. *Bol. Soc. Mat. Mexicana* **5** (1960) 102–119.

[34] M. Kärcher and M.A. Grepl, A certified reduced basis method for parametrized elliptic optimal control problems. *ESAIM: COCV* **20** (2014) 416–441.

[35] C. Kenney and G. Hewer, The sensitivity of the algebraic and differential Riccati equations. *SIAM J. Control Optimiz.* **28** (1990) 50–69.

[36] C. Kenney, A.J. Laub and M. Wette, Error bounds for Newton refinement of solutions to algebraic Riccati equations. *Math. Control, Signals Systems* **3** (1990) 211–224.

[37] H. Kwakernaak and R. Sivan, Linear optimal control systems, Vol. 1. Wiley-interscience New York (1972).

[38] P. Lancaster, Explicit solutions of linear matrix equations. *SIAM Rev.* **12** (1970) 544–566.
[39] P. Lancaster and L. Rodman, *Algebr. Riccati Equ.* Oxford University Press (1995).
[40] J.-R. Li and J. White, Low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.* **46** (2004) 693–713.
[41] A. Manzoni and F. Negri, Heuristic strategies for the approximation of stability factors in quadratically nonlinear parametrized PDEs. Special Issue on Model Reduction of Parametrized Systems. *Advances in Computational Mathematics* (2015).
[42] D. Mustafa and K. Glover, Controller reduction by H_∞ balanced truncation. *IEEE Trans. Autom. Control* **36** (1991) 668–682.
[43] F. Negri, G. Rozza, A. Manzoni and A. Quarteroni, Reduced basis method for parametrized elliptic optimal control problems. *SIAM J. Scientific Comput.* **35** (2013) A2316–A2340.
[44] H. Panzer, T. Wolf and B. Lohmann, A strictly dissipative state space representation of second order systems. *at – Automatisierungstechnik* **60** (2012) 392–397.
[45] A.T. Patera and G. Rozza, Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Partial Differential Equations. To appear in (tentative) MIT Pappalardo Graduate Monographs in Mechanical Engineering. MIT (2007).
[46] T. Penzl, Numerical solution of generalized Lyapunov equations. *Adv. Comput. Math.* **8** (1998) 33–48.
[47] T. Penzl, Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case. *Syst. Control Lett.* **40** (2000) 139–144.
[48] 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: application to transport and continuum mechanics. *Arch. Comput. Meth. Eng.* **15** (2008) 229–275.
[49] Y. Saad and X.V. Gv, Numerical solution of large Lyapunov equations. In *Proc. of MTNS-89 Signal Processing, Scattering and Operator Theory, and Numerical Methods*. Birkhauser (1990) 503–511.
[50] J. Saak, *Efficient numerical solution of large scale algebraic matrix equations in PDE control and model order reduction*. Ph.D. thesis, Chemnitz University of Technology (2009).
[51] A. Schmidt, M. Dihlmann and B. Haasdonk, Basis generation approaches for a reduced basis linear quadratic regulator. In *Proc. MATHMOD 2015 – 8th Vienna International Conference on Mathematical Modelling* (2015) 713–718.
[52] A. Schmidt and B. Haasdonk, Reduced basis method for H2 optimal feedback control problems. *IFAC-PapersOnLine* **49** (2016) 327–332, 2nd IFAC Workshop on Control of Systems Governed by Partial Differential Equations CPDE (2016).
[53] V. Simoncini, A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. Sci. Comput.* **29** (2007) 1268–1288.
[54] V. Simoncini, Analysis of the rational krylov subspace projection method for large-scale algebraic Riccati equations. *SIAM J. Matrix Anal. Appl.* **37** (2016) 1655–1674.
[55] G. Söderlind, The logarithmic norm. history and modern theory. *BIT Numer. Math.* **46** (2006) 631–652.
[56] N.T. Son and T. Stykel, Solving parameter-dependent Lyapunov equations using reduced basis method with application to parametric model order reduction. Technical report, University of Augsburg (2015).
[57] T. Stykel, *Analysis and numerical solution of generalized Lyapunov equations*. Ph.D. thesis, Institut für Mathematik, Technische Universität, Berlin (2002).
[58] T. Stykel, Numerical solution and perturbation theory for generalized Lyapunov equations. *Lin. Algebr. Appl.* **349** (2002) 155–185.
[59] J.-G. Sun, Residual bounds of approximate solutions of the algebraic Riccati equation. *Numer. Math.* **76** (1997) 249–263.
[60] N. Truhar and K. Veselić, Bounds on the trace of a solution to the Lyapunov equation with a general stable matrix. *Syst. Control Lett.* **56** (2007) 493–503.
[61] K. Veroy, C. Prud'homme, D.V. Rovas and A.T. Patera, A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *16th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics (2003) 2003–3847.
[62] S. Volkwein, Model reduction using proper orthogonal decomposition. Lecture notes, University of Konstanz (2011).
[63] H. Wendland, Scattered Data Approximation. Cambridge University Press (2004). Cambridge Books Online.
[64] D. Wirtz, D.C. Sorensen and B. Haasdonk, A posteriori error estimation for DEIM reduced nonlinear dynamical systems. *SIAM J. Scientific Comput.* **36** (2014) A311–A338.
[65] Daniel Wirtz, Model Reduction for Nonlinear Systems: Kernel Methods and Error Estimation. epubli GmbH (2014).
[66] K. Zhou, J.C. Doyle and K. Glover, Robust and Optimal Control. Prentice Hall (1996).