01 Jan 2011

Balanced POD for Linear PDE Robust Control Computations

John R. Singler
Missouri University of Science and Technology, singlerj@mst.edu

Belinda A. Batten

Recommended Citation

J. R. Singler and B. A. Batten, "Balanced POD for Linear PDE Robust Control Computations," *Computational Optimization and Applications*, Springer-Verlag, Jan 2011.
The definitive version is available at https://doi.org/10.1007/s10589-011-9451-x

Balanced POD for Linear PDE Robust Control Computations

John R. Singler
Department of Mathematics and Statistics
Missouri University of Science and Technology
Rolla, MO 65409-0020
Email: singlerj@mst.edu

Belinda A. Batten
School of Mechanical, Industrial, and Manufacturing Engineering
Oregon State University
Corvallis, OR 97331–6001
Email: bbatten@engr.orst.edu

Abstract

A mathematical model of a physical system is never perfect; therefore, robust control laws are necessary for guaranteed stabilization of the nominal model and also "nearby" systems, including hopefully the actual physical system. We consider the computation of a robust control law for large-scale finite dimensional linear systems and a class of linear distributed parameter systems. The controller is robust with respect to left coprime factor perturbations of the nominal system. We present an algorithm based on balanced proper orthogonal decomposition to compute the nonstandard features of this robust control law. Convergence theory is given, and numerical results are presented for two partial differential equation systems.

1 Introduction

Since a mathematical model of a system is not a perfect description of the system, it is desirable for a control law to not only stabilize the mathematical model but also "nearby" systems. In this paper, we consider computing a control law for a distributed parameter system that stabilizes the nominal system and also left coprime factor perturbations of the system. The robustly stabilizing control, the central controller, is a solution to this problem and was given for finite dimensional systems by Glover and McFarlane in [19]; for infinite dimensional systems with bounded finite rank inputs and outputs, the solution can be found in [15]. For an example of the central controller applied to a PDE system, see [8].

Computing control laws for infinite dimensional systems often starts with discretizing the system with a convergent approximation scheme. Matrix approximations of the original system operators arise, and these matrices can be very large scale for many important applications, such as those in fluids. Much recent research has focused on developing algorithms to solve large-scale matrix Lyapunov and Riccati equations associated with control design for linear systems (among other applications); see Section 3 below for details and references. However, the authors are not aware of any work that computes the robust control law from the central controller for a large-scale system. To address these problems, we use balanced proper orthogonal decomposition (POD) methods.

Balanced POD is an algorithm introduced by Rowley [29] for approximate balanced model reduction of linear systems. The algorithm is also related to balanced model reduction algorithms

Preprint; published in: Comput. Optim. Appl. 53 (2012), no. 1, 227-248

proposed by Willcox and Peraire [41] and Lall, Marsden, and Glavaški [27]. Rowley's balanced POD algorithm is similar to the method of snapshots for standard POD computations [38], however it uses two datasets. In the present work, we require the balanced POD of time varying data taking values in a Hilbert space as described in [36, 31, 33]. We give a brief overview of the algorithm for this case in Section 4; for more information on the algorithm, see the above references. Also, for recent examples of the application of balanced POD for model reduction, see [1, 3, 7, 24]. Furthermore, we note that balanced POD can also be used to provide an optimal reconstruction of two general datasets [33] in an analogous way that standard POD can reconstruct a single dataset, e.g., [23].

In Section 5, we develop a balanced POD algorithm to compute the central controller for a class of linear partial differential equation (PDE) systems. The main computational cost of the algorithm is computing solution snapshots of linear PDEs. These computations can be performed with existing software and one can also take advantage of existing techniques such as special discretization schemes, domain decomposition methods, adaptive mesh refinement, and parallel algorithms. Also, since the algorithm is based on simulation data rather than matrix approximations, we bypass the potential difficulty of extracting matrices from existing simulation code. The algorithm described in this paper is new and is applicable to both large-scale finite dimensional systems and a class of infinite dimensional systems.

2 Robust Central Controller for Left Coprime Factor Perturbations

We consider robust feedback control design for a general infinite dimensional system

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t), \tag{1}$$

holding over a real Hilbert space X with inner product $(\cdot,\cdot)$ and corresponding norm $\|f\| = (f,f)^{1/2}$. Throughout this work, we assume the operator $A : D(A) \subset X \to X$ generates a C_0-semigroup, and the control input operator $B : \mathbb{R}^m \to X$ and the observation operator $C : X \to \mathbb{R}^p$ are both bounded and finite rank. The last assumption implies that the operators B and C must take the form

$$Bu = \sum_{j=1}^{m} u_j\, b_j, \quad Cx = [\,(x,c_1),\ldots,(x,c_p)\,]^T, \tag{2}$$

for some vectors $b_1,\ldots,b_m$ and $c_1,\ldots,c_p$ in X (see [40, Theorem 6.1]). We note that the algorithm presented here for the robust controller computations is more efficient when m and p are relatively small; this is true of most large-scale algorithms for control and model reduction computations.

We assume (A,B,C) is exponentially stabilizable and detectable so that the transfer function $G(s) = C(sI-A)^{-1}B$ has a normalized left coprime factorization $G = \tilde{M}^{-1}\tilde{N}$; see [15, Lemma 9.4.10]. We consider the following robust control problem: Given a robustness margin $\varepsilon > 0$, find a controller $K_c(s)$ stabilizing $G(s)$ and all "nearby" plants $G_\Delta(s)$ of the form

$$G_\Delta = (\tilde{M} + \Delta_M)^{-1}(\tilde{N} + \Delta_N),$$

where the perturbation $\Delta = [\Delta_M \;\; \Delta_N]$ satisfies $\|\Delta\|_\infty < \varepsilon$. Here, the applied norm is the $\mathcal{H}_\infty$ norm, which is the largest singular value of the function evaluated along the imaginary axis.

The solution to this problem can be found in Section 9.4 of Curtain and Zwart's book [15]. Let the operators $\Pi : X \to X$ and $P : X \to X$ be the unique self-adjoint, nonnegative solutions to the control and filter algebraic Riccati equations (AREs)

$$A^*\Pi + \Pi A - \Pi BB^*\Pi + C^*C = 0, \tag{3}$$

$$AP + PA^* - PC^*CP + BB^* = 0, \tag{4}$$

where the asterisk (*) denotes the Hilbert adjoint operator. There is a parameterized family of controllers solving the above problem; the central controller is given in state space form by [15, Theorem 9.4.16]

$$u(t) = -Kx_c(t), \quad \dot{x}_c(t) = A_c x_c(t) + \sigma^{-2}W^*Fy(t),$$

where

$$\begin{aligned} K &= B^*\Pi, & A_c &= A - BK - \sigma^{-2}W^*FC, \\ F &= PC^*, & W^* &= \left[I + (1-\sigma^{-2})\Pi P\right]^{-1}, \\ \sigma &= \left(1-\varepsilon^2\right)^{1/2}, & 0 &< \varepsilon < \varepsilon_{\max}. \end{aligned}$$

Here, the maximum robustness margin $\varepsilon_{\max}$ can be found exactly and is given by [15, Corollary 9.4.12]:

$$\varepsilon_{\max} = \left[1 + \lambda_{\max}(P\Pi)\right]^{-1/2}, \tag{5}$$

where $\lambda_{\max}(T)$ denotes the largest eigenvalue of the operator T. This extends the finite dimensional results of Glover and McFarlane [19] to an infinite dimensional case.

For finite dimensional systems, the central controller described above can be computed by using existing solvers for the algebraic Riccati equations (3) and (4). For infinite dimensional systems, a standard approach to control design is to approximate the infinite dimensional operators by matrices and design an approximate control law; see, e.g., [6, 12, 18]. However, for many infinite dimensional equations the approximating matrices are of high dimension and conventional computational approaches are difficult, if not impossible, to apply. For the central controller design above, we have the following computational challenges for large-scale systems:

1. Compute the solutions Π and P of the algebraic Riccati equations (3) and (4).
2. Compute the maximum robustness margin $\varepsilon_{\max}$ in (5).
3. Compute $W^*F = \left[I + (1-\sigma^{-2})\Pi P\right]^{-1}F$.

In this paper, we focus on using a balanced POD algorithm to treat the second and third challenges.

Before we describe the algorithm, we provide background on computing solutions of algebraic Riccati equations in Section 3 and balanced POD in Section 4.

3 Riccati Equation Algorithm

There are three main existing classes of algorithms for large-scale or infinite dimensional Riccati equations of the form (3).

Three Algorithm Classes for Riccati Equations

1. Apply Newton's method to the (quadratic) Riccati equation and solve the resulting (linear) Lyapunov equations using special techniques; see, e.g., [5, 28, 17].
2. Solve the related Chandrasekhar equations, a nonlinear system of differential equations that must be integrated to steady state; see, e.g., [5, 9, 11].

3. Reduce the infinite dimensional model and solve the resulting low order matrix Riccati equation; see, e.g., [2, 4, 25].

Although all three approaches can be successful, the first approach is convergent and is generally accepted to be the most accurate; the Chandrasekhar equations can lose accuracy when integrated to steady state [5], and the "reduce-then-design" approach can either have lower accuracy (see [10] for an example) or fail as it has no guarantees of accuracy or convergence.

In this work, we use the first approach coupled with a trapezoid snapshot algorithm for infinite dimensional Lyapunov equations developed in [34] (see also [32, 16, 35, 39]). The Lyapunov solver was first proposed for the matrix case by Saad in [30]. We use the standard Kleinman-Newton iteration [26, 13] as opposed to the modified Kleinman-Newton iteration proposed by Banks and Ito in [5]; Feitzinger, Hylla, and Sachs have recently shown in [17] that errors can accumulate in the modified iteration if the Lyapunov equations are solved inexactly (which is unavoidable except for simple problems).

3.1 Riccati Equation Snapshot Algorithm Details

We now describe our computational approach for the Riccati equations (3) and (4).

For this work, we do not require the entire solution operators $\Pi : X \to X$ and $P : X \to X$ of the Riccati equations; instead, we only require the computation of the feedback gain operators $K = B^*\Pi$ and $F = PC^*$. More specifically, we require the *functional gains* for the operators K and F. These are defined using the above representations (2) of B and C as follows. Since B takes the form $Bu = \sum_{j=1}^m u_j\, b_j$, it can be checked that $B^*x = [\,(x, b_1), \ldots, (x, b_m)\,]^T$. Therefore, for any $x \in X$,

$$\begin{aligned} Kx &= B^*\Pi x \\ &= [\,(\Pi\, x, b_1), \ldots, (\Pi\, x, b_m)\,]^T \\ &= [\,(x, \Pi\, b_1), \ldots, (x, \Pi\, b_m)\,]^T, \end{aligned}$$

since Π is self-adjoint. Thus, $Kx = [\,(x, k_1), \ldots, (x, k_m)\,]^T$, where $k_j = \Pi\, b_j \in X$ are the functional gains for K. Similarly, $Fy = \sum_{i=1}^p y_i\, f_i$, where $f_i = Pc_i$ are the functional gains for F.

The Kleinman-Newton iteration applied to the Riccati equation (3) yields the sequence of Lyapunov equations

$$(A - BK_\ell)^* S_\ell + S_\ell (A - BK_\ell) + K_\ell^* K_\ell + C^*C = 0. \tag{6}$$

The sequence $K_{\ell+1} = B^* S_\ell$ converges at a quadratic rate to K for any stabilizing initial guess K_0 [13].

We do not require the entire solution $S_\ell : X \to X$ of each Lyapunov equation. As above, it can be shown that

$$K_\ell\, x = [\,(x, S_\ell\, b_1), \ldots, (x, S_\ell\, b_m)\,]^T.$$

Therefore, at each iteration we need only compute $S_\ell\, b_j$ for $j = 1, \ldots, m$. We compute these products using a snapshot algorithm.

Consider a general infinite dimensional Lyapunov equation

$$A_0^* S + S A_0 + E^*E = 0, \tag{7}$$

where A_0 generates an exponentially stable C_0-semigroup $e^{A_0 t}$ and $E : X \to \mathbb{R}^q$ is given by $Ex = [\,(x, e_1),\, \ldots,\, (x, e_q)\,]^T$ with each $e_i \in X$. It is well known that the solution $S : X \to X$ is given by

$$Sx = \int_0^\infty e^{A_0^* t} E^* E e^{A_0 t} x \, dt.$$

Using the above representation of E, it can be shown [32, 34] that the solution may also be represented by

$$Sx = \int_0^\infty \sum_{i=1}^{q} \big(x, z_i(t)\big)\, z_i(t)\, dt, \tag{8}$$

where $z_i(t) = e^{A_0^* t} e_i$ is the solution of the infinite dimensional linear differential equation

$$\dot{z}_i(t) = A_0^* z_i(t), \quad z(0) = e_i. \tag{9}$$

This representation leads to the following snapshot algorithm.

Snapshot algorithm [32, 34] to approximate Sx, where S solves the Lyapunov equation (7)

1. For $i = 1, \ldots, q$, compute approximation $z_i^N(t)$ of the solutions $z_i(t)$ of the differential equations (9).

2. Replace $z_i(t)$ with $z_i^N(t)$ in the integral representation of Sx in (8) and approximate the integral (by quadrature or some other method).

If $\int_0^\infty \|z_i^N(t) - z_i(t)\|^2 \, dt \to 0$ for each i, then the resulting approximation converges to Sx [34].

The approximate solutions $z_i^N(t)$ of the differential equations (9) need not be stored to approximate Sx. Instead, a time stepping method can be used to approximate the differential equation and the approximation to the integral can be updated while simultaneously integrating the differential equations. For example, using a piecewise linear approximation to $z_i(t)$ in time leads to the trapezoid rule to time step the differential equation and the following approximation to the integral.

Trapezoid snapshot algorithm [34] to approximate Sx, where S solves the Lyapunov equation (7)

1. For $i = 1, \ldots, q$, approximate the solution of the differential equations (9) with the trapezoid rule:

   $$(I - \Delta t A_0^*/2) z_{i,n+1} = (I + \Delta t A_0^*/2) z_{i,n},$$

   where I is the identity operator.

2. Update the approximation to Sx:

   $$\begin{aligned} [Sx]_{i,n+1} &= [Sx]_{i,n} + \Delta t \big[(x, z_{i,n+1})/3 + (x, z_{i,n})/6\big] z_{i,n+1} \\ &\quad + \Delta t \big[(x, z_{i,n+1})/6 + (x, z_{i,n})/3\big] z_{i,n}. \\ [Sx]_{n+1} &= \sum_{i=1}^{q} [Sx]_{i,n+1}. \end{aligned}$$

This updating procedure can be stopped when the norm of the update to Sx (unscaled by Δt) is below a certain tolerance. We note that we used a constant time step for simplicity; this is not necessary in general.

For the Lyapunov equations arising in the modified Kleinman-Newton iterations (6), note that A_0^* in the Lyapunov equation (7) is replaced by $(A - BK_\ell)^*$. Thus, in the trapezoid snapshot algorithm, we must invert operators of the form $A_s - B_s K_s$, where $A_s = I - \Delta t A^*/2$, $B_s = -\Delta t K_i^*/2$ and $K_s = B^*$. To compute $(A_s - B_s K_s)^{-1} z$ we use the Sherman-Morrison-Woodbury formula (see, e.g., [21]):

$$(A_s - B_s K_s)^{-1} z = (I + A_s^{-1} B_s (I - K_s A_s^{-1} B_s)^{-1} K_s) A_s^{-1} z.$$

Also, the operator E in the Lyapunov equation (7) is replaced by $[K_\ell; C]$. Therefore, the vectors $\{e_1, \ldots, e_q\}$ in the definition of E should be replaced by $\{k_{\ell,1}, \ldots, k_{\ell,m}, c_1, \ldots, c_p\}$, where $K_\ell x = [(x, k_{\ell,1}), \ldots, (x, k_{\ell,m})]^T$.

4 Balanced POD

Recall that the goals of this work are to (a) compute the maximum robustness margin $\varepsilon_{\max}$, and (b) compute $W^* F$. Note that (a) requires the computation of the maximum eigenvalue of the product of the Riccati solutions, while (b) requires the inversion of an operator involving the product of the Riccati solutions.

Below, we perform these computations using the balanced POD of solution data for differential equations associated with the algebraic Riccati equations (3) and (4). The details of these computations are presented in Section 5. In this section, we describe balanced POD of time varying Hilbert space valued data in detail.

Let $\{z_i\}_{i=1}^q \subset L^2(I_z; X)$ and $\{w_j\}_{j=1}^s \subset L^2(I_w; X)$, where $I_z \subset (-\infty, \infty)$ and $I_w \subset (-\infty, \infty)$ are two possibly different intervals (finite or infinite). Here, $L^2(I; X)$ is the space of functions w such that $w(t) \in X$ for all $t \in I$ and whose X norm is square integrable, i.e.,

$$\|w\|_{L^2(I;X)} = \left(\int_I \|w(t)\|^2 \, dt \right)^{1/2} < \infty.$$

We define the balanced POD of the two datasets above in terms of the POD operator of each dataset.

Definition 1. *Let $\{z_i\}_{i=1}^q \subset L^2(I_z; X)$ and $\{w_j\}_{j=1}^s \subset L^2(I_w; X)$ be given datasets. Define the (compact) POD operators $Z_C : X \to X$ and $Z_B : X \to X$ by*

$$Z_C x = \int_{I_z} \sum_{i=1}^q \big(x, z_i(t) \big) \, z_i(t) \, dt, \quad Z_B x = \int_{I_w} \sum_{j=1}^s \big(x, w_j(t) \big) \, w_j(t) \, dt.$$

Let $\{\lambda_k\}$ and $\{\psi_k\} \subset X$ be the eigenvalues and eigenvectors of the product $Z_C Z_B$, and let $\{\lambda_k\}$ and $\{\varphi_k\} \subset X$ be the eigenvalues and eigenvectors of $Z_B Z_C$. We call $\{\lambda_k, \varphi_k, \psi_k\}$ the balanced POD eigenvalues and modes of the two datasets if the eigenvectors are scaled such that $(\varphi_j, \psi_i) = \delta_{ij}$.

It is shown in [33] that the eigenvalues of $Z_C Z_B$ and $Z_B Z_C$ are the same, and the eigenvalues can be ordered $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$.

Again, we note that balanced POD is most often used as a balanced truncation model reduction algorithm for an exponentially stable linear system of the form (1). In this case, the balanced POD modes yield the balancing transformation for the linear system [29, 36, 31]. For the application here, we do not reduce the linear system; instead, we use balanced POD to extract the eigenvalues and eigenvectors of the Riccati operator products ΠP and $P \Pi$. We then use these to approximate the maximum robustness margin and the product $W^* F$ needed for the robust central controller construction.

4.1 Approximating the Balanced POD

Next, we briefly review a quadrature method for computing the balanced POD eigenvalues and modes. This method was proposed by Rowley [29] for the finite dimensional Hilbert space $X = \mathbb{R}^n$ and extended to a general real Hilbert space in [36]. Convergence theory and an alternate approach for the balanced POD approximations can be found in [31].

The POD operators can be approximated using time snapshots of the states $w_j(t)$ and $z_i(t)$. Specifically, approximate the time integrals with the quadratures

$$Z_C x \approx Z_C^N x = \sum_{i=1}^{q} \sum_{k=1}^{n_z} \beta_k^2 \, z_i(t_k) \left(x, z_i(t_k)\right),$$
$$Z_B x \approx Z_B^N x = \sum_{j=1}^{s} \sum_{\ell=1}^{n_w} \gamma_\ell^2 \, w_j(t_\ell) \left(x, w_j(t_\ell)\right).$$

Here, $\{\beta_k^2\}$ and $\{\gamma_\ell^2\}$ are quadrature weights corresponding to the sets of quadrature points $\{t_k\}$ and $\{t_\ell\}$; different quadrature points and weights can be used for each w_j and z_i if desired.

Define "vectors" of weighted snapshots

$$\tilde{z} = [\,\beta_1 z_1(t_1), \, \ldots \, , \beta_{n_z} z_1(t_{n_z}), \, \ldots \, , \beta_1 z_q(t_1), \, \ldots \, , \beta_{n_z} z_q(t_{n_z})\,]^T \in X^{N_z},$$
$$\tilde{w} = [\,\gamma_1 w_1(t_1), \, \ldots \, , \gamma_{n_w} w_1(t_{n_w}), \, \ldots \, , \gamma_1 w_s(t_1), \, \ldots \, , \gamma_{n_w} w_s(t_{n_w})\,]^T \in X^{N_w},$$

where $N_z = q n_z$, $N_w = s n_w$, and $X^N = X \times \cdots \times X$ (N times). The approximate POD operators can be expressed as $Z_C^N = (R_C^N)^* R_C^N$ and $Z_B^N = R_B^N (R_B^N)^*$, where the operators $R_C^N : X \to \mathbb{R}^{N_z}$ and $R_B^N : \mathbb{R}^{N_w} \to X$ are defined by

$$R_C^N x = [\,(x, \tilde{z}_1), \, \ldots \, , (x, \tilde{z}_{N_z})\,]^T, \quad R_B^N a = \sum_{i=1}^{N_w} a_i \, \tilde{w}_i,$$

and their adjoint operators $(R_C^N)^* : \mathbb{R}^{N_z} \to X$ and $(R_B^N)^* : X \to \mathbb{R}^{N_w}$ are given by

$$(R_C^N)^* a = \sum_{i=1}^{N_z} a_i \, \tilde{z}_i, \quad (R_B^N)^* x = [\,(x, \tilde{w}_1), \, \ldots \, , (x, \tilde{w}_{N_w})\,]^T.$$

The eigenvalues and eigenvectors of the operator product $Z_C^N Z_B^N = (R_C^N)^* R_C^N R_B^N (R_B^N)^*$ can be found by computing the singular values and singular vectors of the bounded linear operator $R_C^N R_B^N : \mathbb{R}^{N_w} \to \mathbb{R}^{N_z}$. This operator can be represented by the $N_z \times N_w$ matrix Γ with entries $\Gamma_{ij} = (\tilde{w}_j, \tilde{z}_i)$. Let $\sigma_1^N \geq \sigma_2^N \cdots \geq 0$ be the singular values of Γ with corresponding orthonormal singular vectors $\{u_k^N\}$ and $\{v_k^N\}$ such that

$$\Gamma u_k^N = \sigma_k^N v_k^N, \quad \Gamma^T v_k^N = \sigma_k^N u_k^N.$$

Then (compare Proposition 1 below) for all k with σ_k^N nonzero, the eigenvalues $\{\lambda_k^N\}$ and eigenvectors $\{\psi_k^N\} \subset X$ and $\{\varphi_k^N\} \subset X$ of $Z_C^N Z_B^N$ and $Z_B^N Z_C^N$, respectively, are given by

$$\lambda_k^N = (\sigma_k^N)^2, \quad \varphi_k^N = (\sigma_k^N)^{-1/2} \, R_B^N \, u_k^N, \quad \psi_k^N = (\sigma_k^N)^{-1/2} \, (R_C^N)^* \, v_k^N.$$

5 Balanced POD Algorithm for the Central Controller

We now present the balanced POD algorithm to approximate the maximum robustness margin $\varepsilon_{\max} = \big[1+\lambda_{\max}(P\Pi)\big]^{-1/2}$, and also approximate $W^*F = \big[I+(1-\sigma^{-2})\Pi P\big]^{-1}F$. As noted earlier, both of these quantities involve the product of the solution operators to the Riccati equations (3) and (4). Throughout this work we use a superscript N on a quantity to denote an approximation of that quantity.

Step 1: The first step in the algorithm is to compute approximations $\{k_j^N\}_{j=1}^m$ and $\{f_i^N\}_{i=1}^p$ of the functional gains $\{k_j\}_{j=1}^m$ and $\{f_i\}_{i=1}^p$ for the operators $K = B^*\Pi$ and $F = PC^*$, where Π and P solve the Riccati equations (3) and (4). One approach for these computations was described in Section 3, but any algorithm may be used.

Step 2: Now rewrite the solutions of the Riccati equations (3) and (4) in terms of solutions of linear infinite dimensional differential equations. This is done as follows. First, as is well known, the Riccati equations can be rewritten as the Lyapunov equations

$$(A-BK)^*\Pi + \Pi(A-BK) + K^*K + C^*C = 0,$$
$$(A-FC)P + P(A-FC)^* + FF^* + BB^* = 0.$$

As mentioned in Section 3, the solutions of these Lyapunov equations (i.e., the Riccati operators Π and P) can be written

$$\Pi x = \int_0^\infty \sum_{i=1}^{m+p} \big(x, z_i(t)\big)\, z_i(t)\, dt,$$

$$Px = \int_0^\infty \sum_{j=1}^{m+p} \big(x, w_j(t)\big)\, w_j(t)\, dt,$$

where each $z_i(t)$ and $w_j(t)$ are the unique solutions of the linear evolution equations

$$\dot{z}_i(t) = (A-BK)^* z_i(t), \quad z_i(0) = z_i^0, \tag{10}$$
$$\dot{w}_j(t) = (A-FC) w_j(t), \quad w_j(0) = w_j^0, \tag{11}$$

with the initial conditions given by

$$z_i^0 = k_i, \;\; i = 1,\ldots,m, \qquad z_i^0 = c_i, \;\; i = m+1,\ldots,m+p,$$
$$w_j^0 = f_j, \;\; j = 1,\ldots,p, \qquad w_j^0 = b_j, \;\; i = p+1,\ldots,m+p.$$

Here, $\{b_j\}_{j=1}^m$ and $\{c_i\}_{i=1}^p$ are given in the representation (2) of the operators B and C.

Step 3: Compute approximations $\{z_i^N\}_{i=1}^{m+p}$ and $\{w_j^N\}_{j=1}^{m+p}$ of the solutions $\{z_i\}_{i=1}^{m+p}$ and $\{w_j\}_{j=1}^{m+p}$ to the above differential equations.

Note that these differential equations also appear in the snapshot approach of Section 3 for approximating the functional gains in Step 1; however, at each step in the Kleinman-Newton iteration the gains in the differential equations are updated. Therefore, if the snapshot approach is used in Step 1, then the differential equation solution data from the last Kleinman-Newton step can be used here (assuming the gains are sufficiently converged).

Step 4: Compute the balanced POD eigenvalues $\{\lambda_k^N\}$ and modes $\{\varphi_k^N, \psi_k^N\}$ of the above solution data. We outlined one approach to these computations in Section 4, however any method may be used. Recall that the balanced POD eigenvalues and modes for the exact solution data give the the eigenvalues $\{\lambda_k\}$ and eigenvectors $\{\varphi_k, \psi_k\}$ of the operator products ΠP and $P\Pi$.

Step 5: Approximate the maximum robustness margin $\varepsilon_{\max} = \big[1 + \lambda_{\max}(P\Pi)\big]^{-1/2}$ by $\varepsilon^N_{\max} = \big[1 + \lambda^N_1\big]^{-1/2}$.

Step 6: To approximate the operator product $W^* F$, recall that $Fy = \sum_{j=1}^{p} y_j f_j$, where each $f_j \in X$ is a functional gain for F. Thus, $W^* F y = \sum_{j=1}^{p} y_j (W^* f_j)$, and we need only approximate the products $W^* f_j$ for $j = 1, \ldots, p$ to form $W^* F$.

For $\alpha = -(1 - \sigma^{-2})$, where $\sigma = (1 - \varepsilon^2)^{1/2}$ and $0 < \varepsilon < \varepsilon^N_{\max}$, we approximate $W^* x = (I - \alpha \Pi P)^{-1} x$ by

$$W^* x \approx \xi^N_r = x + \sum_{k=1}^{r} \frac{\alpha \lambda^N_k}{1 - \alpha \lambda^N_k} (x, \varphi^N_k)\, \psi^N_k. \tag{12}$$

The origin of this approximation is discussed in the next section.

Notes:

- Using the definition of α, it can be shown that $0 < 1 - \alpha\lambda^N_k < 1$, and therefore $\alpha\lambda^N_k/(1 - \alpha\lambda^N_k)$ is never infinite. However, as $\varepsilon \to \varepsilon^N_{\max}$, $1 - \alpha\lambda^N_1$ approaches zero and therefore the first term in the expression for ξ^N_r becomes infinitely large.
- If the eigenvalues $\{\lambda^N_k\}$ decay quickly enough, a small value of r will give a good approximation to the entire series.
- Below, we show that if the approximate solutions of the differential equations in step 3 converge, then $\varepsilon^N_{\max} \to \varepsilon_{\max}$ and $\xi^N_r \to W^* x$ as $N, r \to \infty$.

6 Convergence Theory

Let $\{z_i\}_{i=1}^{q}$ and $\{w_j\}_{j=1}^{s}$ be finite collections of functions in $L^2(I_z; X)$ and $L^2(I_w; X)$, respectively. Let Π be the POD operator for $\{z_i\}$ and let P be the POD operator for $\{w_j\}$. These may be arbitrary square integrable functions, however our primary interest is with the functions $\{z_i\}$ and $\{w_j\}$ in Step 2 of the algorithm in Section 5 above; note that these functions are defined on the interval $I_z = I_w = (0, \infty)$.

Define $\mathcal{C} : X \to L^2(I_z; \mathbb{R}^q)$ and $\mathcal{B} : L^2(I_w; \mathbb{R}^s) \to X$ by

$$[\mathcal{C}x](t) = \big[\, \big(x, z_1(t)\big), \ldots, \big(x, z_q(t)\big) \,\big]^T, \quad \mathcal{B}u = \int_{I_w} \sum_{j=1}^{s} u_j(\tau)\, w_j(\tau)\, d\tau.$$

The adjoint operators $\mathcal{C}^* : L^2(I_z; \mathbb{R}^q) \to X$ and $\mathcal{B}^* : X \to L^2(I_w; \mathbb{R}^s)$ are given by

$$\mathcal{C}^* y = \int_{I_z} \sum_{j=1}^{q} y_j(\tau)\, z_j(\tau)\, d\tau, \quad [\mathcal{B}^* x](t) = \big[\, \big(x, w_1(t)\big), \ldots, \big(x, w_s(t)\big) \,\big]^T.$$

It is straightforward to check that the individual POD operators can be factored as $\Pi = \mathcal{C}^*\mathcal{C}$ and $P = \mathcal{B}\mathcal{B}^*$.

As in [33], define the balanced POD operator $\mathcal{H} : L^2(I_w; \mathbb{R}^s) \to L^2(I_z; \mathbb{R}^q)$ for $\{z_i, w_j\}$ by $\mathcal{H} = \mathcal{C}\mathcal{B}$. Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$ be the singular values of $\mathcal{H}$ with corresponding orthonormal singular vectors $\{u_k, v_k\}$ satisfying

$$\mathcal{H} u_k = \sigma_k v_k, \quad \mathcal{H}^* v_k = \sigma_k u_k. \tag{13}$$

We also have the singular value expansion

$$\mathcal{H}u = \sum_{k\geq 1} \sigma_k \, (u, u_k) \, v_k, \tag{14}$$

where we understand the sum to terminate if the singular values become zero. The following result [33, Proposition 2] relates the balanced POD eigenvalues and modes to the singular values and singular vectors of the balanced POD operator.

Proposition 1. *Let $\{z_i\}_{i=1}^q$ and $\{w_j\}_{j=1}^s$ be finite collections of functions in $L^2(I_z; X)$ and $L^2(I_w; X)$, respectively, with balanced POD eigenvalues $\{\lambda_k\}$ and balanced POD modes $\{\varphi_k, \psi_k\}$. For any k for which σ_k is nonzero, the balanced POD eigenvalues and corresponding balanced POD modes are*

$$\lambda_k = \sigma_k^2, \quad \varphi_k = \sigma_k^{-1/2} \, \mathcal{B} \, u_k, \quad \psi_k = \sigma_k^{-1/2} \, \mathcal{C}^* \, v_k. \tag{15}$$

Furthermore, the singular vectors are orthonormal and can be expressed as

$$u_k = \sigma_k^{-1/2} \, \mathcal{B}^* \, \psi_k, \quad v_k = \sigma_k^{-1/2} \, \mathcal{C} \, \varphi_k. \tag{16}$$

We are concerned with approximating the solution ξ of the equation

$$(I - \alpha \Pi P) \, \xi = x, \tag{17}$$

for a real number α and $x \in X$. Suppose α^{-1} is not an eigenvalue of ΠP. Then $(I - \alpha \Pi P)^{-1}$ is a bounded linear operator on X, and (17) has a unique solution $\xi \in X$ given by $\xi = (I - \alpha \Pi P)^{-1} x$. We express ξ in terms of the balanced POD eigenvalues and modes of the data below.

Lemma 1. *Let the above assumption hold and let $x \in X$. If α is a nonzero real number such that α^{-1} is not an eigenvalue of ΠP, then $\xi = (I - \alpha \Pi P)^{-1} x$ can be expressed as*

$$\xi = (I - \alpha \Pi P)^{-1} x = x + \sum_{k\geq 1} \frac{\alpha \lambda_k}{1 - \alpha \lambda_k} (x, \varphi_k) \, \psi_k. \tag{18}$$

Remark: This expression for ξ can be formally derived from the equation $(I - \alpha \Pi P) \, \xi = x$ by setting $\Pi P = \mathcal{C}^* \mathcal{H} \mathcal{B}^*$, replacing $\mathcal{H}$ with its singular value expansion (14), and finding that ψ must take the form $\xi = x + \sum_{k\geq 1} a_k \psi_k$ for certain constants a_k. Substituting this expression back into the equation for ξ yields values for a_k and the result (18). However, below we give a slightly simpler alternate proof.

Proof. First, we show the series converges. Let ξ_n be the right hand side of (18) with only the first n terms in the series. Using (15), we have

$$\xi_n - \xi_m = \sum_{k=m}^{n} \frac{\alpha \lambda_k}{1 - \alpha \lambda_k} (x, \varphi_k) \, \psi_k = \sum_{k=m}^{n} \frac{\alpha \, \sigma_k}{1 - \alpha \lambda_k} (x, \mathcal{B} \, u_k) \, \mathcal{C}^* \, v_k.$$

Since $\|u_k\| = 1$ and $\|v_k\| = 1$, this implies

$$\|\xi_n - \xi_m\| \leq \|\mathcal{B}\| \, \|\mathcal{C}^*\| \sum_{k=m}^{n} \left| \frac{\alpha \, \sigma_k}{1 - \alpha \lambda_k} \right| \leq \|\mathcal{B}\| \, \|\mathcal{C}^*\| \sup_{\ell \geq 1} \left| \frac{\alpha}{1 - \alpha \lambda_\ell} \right| \sum_{k=m}^{n} \sigma_k.$$

Recall the balanced POD eigenvalues $\{\lambda_k\}$ are the eigenvalues of ΠP. Therefore, since α^{-1} is not an eigenvalue of ΠP, we have $1 - \alpha \lambda_k$ is nonzero. Also, since Π and P are compact, the eigenvalues

$\{\lambda_\ell\}$ may only accumulate at zero; therefore, the supremum of $|\alpha/(1-\alpha\lambda_\ell)|$ is finite. Furthermore, the sum of the singular values of $\mathcal{H}$ is finite since $\mathcal{H}$ is trace class [33, Proposition 1]. Therefore, for m and n large enough, $\sum_{k=m}^{n} \sigma_k$ can be made as small as desired. This implies $\{\xi_n\}$ is a Cauchy series and therefore converges.

Let ξ equal the right hand side of (18). We will show $(I-\alpha\Pi P)\,\xi = x$ so that $\xi = (I-\alpha\Pi P)^{-1}x$, i.e., that (17) and (18) hold. Compute:

$$\begin{aligned}(I-\alpha\Pi P)\,\xi &= (I-\alpha\Pi P)\left[x + \sum_{k\geq 1} \frac{\alpha\lambda_k}{1-\alpha\lambda_k}(x,\varphi_k)\,\psi_k\right]\\ &= (I-\alpha\Pi P)\,x + \sum_{k\geq 1} \frac{\alpha\lambda_k}{1-\alpha\lambda_k}(x,\varphi_k)\,\psi_k\\ &\quad - \sum_{k\geq 1} \frac{\alpha^2\lambda_k}{1-\alpha\lambda_k}(x,\varphi_k)\,\Pi P\psi_k.\end{aligned}$$

Since $\Pi P\psi_k = \lambda_k\psi_k$, the two sums can be combined to give

$$\begin{aligned}(I-\alpha\Pi P)\,\xi &= (I-\alpha\Pi P)\,x + \sum_{k\geq 1}\left[\frac{\alpha\lambda_k}{1-\alpha\lambda_k} - \frac{\alpha^2\lambda_k^2}{1-\alpha\lambda_k}\right](x,\varphi_k)\,\psi_k\\ &= (I-\alpha\Pi P)\,x + \sum_{k\geq 1}\alpha\lambda_k\,(x,\varphi_k)\,\psi_k.\end{aligned}$$

Next, using $\Pi P = \mathcal{C}^*\mathcal{H}\mathcal{B}^*$, the expression (15) for the balanced POD eigenvalues and modes, and the singular value expansion of $\mathcal{H}$ (14) gives

$$\begin{aligned}\alpha\Pi P x &= \alpha\sum_{k\geq 1}\sigma_k\,(\mathcal{B}^*x, u_k)\,\mathcal{C}^*v_k\\ &= \alpha\sum_{k\geq 1}\sigma_k^2\,(x,\sigma_k^{-1/2}\,\mathcal{B}\,u_k)\,\sigma_k^{-1/2}\,\mathcal{C}^*\,v_k\\ &= \alpha\sum_{k\geq 1}\lambda_k\,(x,\varphi_k)\,\psi_k.\end{aligned}$$

(Here, as before we consider the above sum for $\alpha\Pi Px$ to terminate if the singular values become zero.) This combined with the above gives $(I-\alpha\Pi P)\,\xi = x$. □ □

Let $\{z_i^N\}_{i=1}^q$ and $\{w_j^N\}_{j=1}^s$ be another finite collection of functions in $L^2(I_z;X)$ and $L^2(I_w;X)$, respectively. Let $\Pi^N = (\mathcal{C}^N)^*\mathcal{C}^N$ be the POD operator for $\{z_i^N\}$ and let $P^N = \mathcal{B}^N(\mathcal{B}^N)^*$ be the POD operator for $\{w_j^N\}$, where $\mathcal{B}^N$ and $\mathcal{C}^N$ are defined analogously to $\mathcal{B}$ and $\mathcal{C}$ above. Furthermore, let $\{\lambda_k^N,\varphi_k^N,\psi_k^N\}$ be the balanced POD eigenvalues and modes for the data $\{z_i^N, w_j^N\}$. Again, these may be arbitrary square integrable functions, however our primary interest is with the functions $\{z_i^N\}$ and $\{w_j^N\}$ defined in Step 3 of the algorithm in Section 5 above.

For x^N given in X, define $\xi_r^N \in X$ by

$$\xi_r^N = x^N + \sum_{k=1}^{r}\frac{\alpha\lambda_k^N}{1-\alpha\lambda_k^N}(x^N,\varphi_k^N)\,\psi_k^N. \qquad (19)$$

The result below gives that ξ_r^N converges to $\xi = (I-\alpha\Pi P)^{-1}x$ as the approximate data $\{z_i^N, w_j^N\}$ converges to the data $\{z_i, w_j\}$. We also give the bound (20) below on the error $\|\xi - \xi_r^N\|$ that shows the effect of using approximate data to form ξ_r^N (the first three terms in the error bound) and the effect of only using r terms in the series for ξ_r^N as opposed to the full series (18) for ξ (the fourth term in the bound).

Theorem 1. *Let the above assumptions hold. If α^{-1} is not an eigenvalue of ΠP or $\Pi^N P^N$ for N sufficiently large, then there exist positive constants $C_{1,N}$, $C_{2,N}$, and $C_{3,N}$ such that*

$$\begin{aligned}\|\xi - \xi_r^N\| \le\, & C_{1,N}\left(\sum_{i=1}^{q}\int_{I_z}\|z_i(t) - z_i^N(t)\|^2\,dt\right)^{1/2}\\ &+ C_{2,N}\left(\sum_{j=1}^{s}\int_{I_w}\|w_j(t) - w_j^N(t)\|^2\,dt\right)^{1/2}\\ &+ C_{3,N}\,\|x - x^N\| + \|x^N\|\sum_{k>r}\left|\frac{\alpha\lambda_k^N}{1-\alpha\lambda_k^N}\right|\|\varphi_k^N\|\,\|\psi_k^N\| < \infty.\end{aligned} \tag{20}$$

If $x^N \to x$ in X, $z_i^N \to z_i$ in $L^2(I_z; X)$ for $i = 1, \ldots, q$, and $w_j^N \to w_j$ in $L^2(I_w; X)$ for $j = 1, \ldots, s$, then

$$\lim_{N,r\to\infty} \xi_r^N = \xi = (I - \alpha\Pi P)^{-1}x.$$

Furthermore, if the balanced POD eigenvalues $\lambda_1, \ldots, \lambda_r$ are distinct, then

$$\lim_{N\to\infty} \xi_r^N = \xi_r := x + \sum_{k=1}^{r}\frac{\alpha\lambda_k}{1-\alpha\lambda_k}(x, \varphi_k)\,\psi_k, \tag{21}$$

and the norm error between ξ and ξ_r is bounded by

$$\|\xi - \xi_r\| \le\ \|x\|\sum_{k>r}\left|\frac{\alpha\lambda_k}{1-\alpha\lambda_k}\right|\|\varphi_k\|\,\|\psi_k\| < \infty. \tag{22}$$

Proof. Let $\xi^N \in X$ be defined by $\xi^N = (I - \alpha\Pi^N P^N)^{-1}x^N$, where N must be large enough so that α^{-1} is not an eigenvalue of $\Pi^N P^N$. We have $\|\xi - \xi_r^N\| \le \|\xi - \xi^N\| + \|\xi^N - \xi_r^N\|$.

To bound the term $\|\xi - \xi^N\|$, subtract the equations $(I - \alpha\Pi P)\xi = x$ and $(I - \alpha\Pi^N P^N)\xi^N = x^N$ (see [22, Lemma 4.1.14, page 64]) to obtain

$$\xi - \xi^N = (I - \alpha\Pi^N P^N)^{-1}\Big[\alpha\,(\Pi P - \Pi^N P^N)\,\xi + (x - x^N)\Big].$$

Therefore,

$$\|\xi - \xi^N\| \le \|(I - \alpha\Pi^N P^N)^{-1}\|\Big[|\alpha|\,\|\Pi P - \Pi^N P^N\|\,\|\xi\| + \|x - x^N\|\Big].$$

Next, $\|\Pi P - \Pi^N P^N\| \le \|\Pi\|\|P - P^N\| + \|P^N\|\|\Pi - \Pi^N\|$. Factoring the POD operators gives

$$\|P - P^N\| \le \big(\|\mathcal{B}\| + \|\mathcal{B}^N\|\big)\,\|\mathcal{B} - \mathcal{B}^N\|, \quad \|\Pi - \Pi^N\| \le \big(\|\mathcal{C}\| + \|\mathcal{C}^N\|\big)\,\|\mathcal{C} - \mathcal{C}^N\|.$$

Using the definitions of the operators $\mathcal{B}$, $\mathcal{B}^N$, $\mathcal{C}$, and $\mathcal{C}^N$, it can be shown[1] that

$$\|\Pi - \Pi^N\| \le \big(\|\mathcal{C}\| + \|\mathcal{C}^N\|\big)\left(\sum_{i=1}^{q}\int_{I_z}\|z_i(t) - z_i^N(t)\|^2\,dt\right)^{1/2}, \tag{23}$$

$$\|P - P^N\| \le \big(\|\mathcal{B}\| + \|\mathcal{B}^N\|\big)\left(\sum_{j=1}^{s}\int_{I_w}\|w_j(t) - w_j^N(t)\|^2\,dt\right)^{1/2}. \tag{24}$$

[1] See [34] for more details and error bounds in the stronger trace norm.

This gives the first three terms in the error bound (20) for $\|\xi - \xi_r^N\|$ with

$$\begin{aligned} C_{1,N} &= |\alpha| \, \|(I - \alpha \Pi^N P^N)^{-1}\| \, \|P^N\| \, \|\xi\| \, \big(\|\mathcal{C}\| + \|\mathcal{C}^N\|\big), \\ C_{2,N} &= |\alpha| \, \|(I - \alpha \Pi^N P^N)^{-1}\| \, \|\Pi\| \, \|\xi\| \, \big(\|\mathcal{B}\| + \|\mathcal{B}^N\|\big), \\ C_{3,N} &= \|(I - \alpha \Pi^N P^N)^{-1}\|. \end{aligned}$$

To produce the fourth term in the error bound (20) for $\|\xi - \xi_r^N\|$, consider the second term in the bound $\|\xi - \xi_r^N\| \leq \|\xi - \xi^N\| + \|\xi^N - \xi_r^N\|$. By Lemma 1,

$$\xi^N = x^N + \sum_{k \geq 1} \frac{\alpha \lambda_k^N}{1 - \alpha \lambda_k^N} (x^N, \varphi_k^N) \, \psi_k^N.$$

Therefore,

$$\|\xi^N - \xi_r^N\| \leq \|x^N\| \sum_{k > r} \left| \frac{\alpha \lambda_k^N}{1 - \alpha \lambda_k^N} \right| \|\varphi_k^N\| \, \|\psi_k^N\|.$$

The same technique as in the proof of Lemma 1 shows that this bound is less than

$$\|\mathcal{B}^N\| \, \|(\mathcal{C}^N)^*\| \sup_{\ell > r} \left| \frac{\alpha}{1 - \alpha \lambda_\ell^N} \right| \sum_{k > r} \sigma_k^N, \tag{25}$$

where $\{\sigma_i^N\}$ are the singular values of $\mathcal{H}^N = \mathcal{C}^N \mathcal{B}^N$. As before, since α^{-1} is not an eigenvalue of the compact operator $\Pi^N P^N$ for N large enough, the quantity $\sup_{\ell > r} |\alpha/(1 - \alpha \lambda_\ell^N)|$ is finite. Also, since $\mathcal{H}^N$ is trace class [33, Proposition 1], the quantity $\sum_{k > r} \sigma_k^N$ is finite.

This proves the error bound (20).

Next, assume $x^N \to x$ and each $z_i^N \to z_i$ and $w_j^N \to w_j$ as $N \to \infty$. Again, using the definitions of the operators $\mathcal{B}$, $\mathcal{B}^N$, $\mathcal{C}$, and $\mathcal{C}^N$, it can be shown that $\mathcal{B}^N \to \mathcal{B}$ and $\mathcal{C}^N \to \mathcal{C}$ in the operator norm as $N \to \infty$. Therefore, $\Pi^N \to \Pi$ and $P^N \to P$, and [22, Theorem 4.1.13, page 63] gives $(I - \alpha \Pi^N P^N)^{-1} \to (I - \alpha \Pi P)^{-1}$ in the operator norm. This implies the constants $C_{1,N}$, $C_{2,N}$, and $C_{3,N}$ converge to finite values as $N \to \infty$, and the first two terms in the error bound (20) can be made as small as desired for N large enough. Clearly, the third term in the error bound can also be made as small as desired since $x^N \to x$ and $C_{3,N}$ converges. Lastly, consider the above bound (25) on the fourth term in the error bound (20). Again, since $\mathcal{H}^N$ is trace class, the quantity $\sum_{k > r} \sigma_k^N$ can be made as small as desired for r large enough. This proves $\xi_r^N \to \xi$ as $N, r \to \infty$.

Next, assume the balanced POD eigenvalues $\lambda_1, \ldots, \lambda_r$ are distinct. The theory in [31] gives that the convergence of each $z_i^N \to z_i$ in $L^2(I_z; X)$ and $w_j^N \to w_j$ in $L^2(I_w; X)$ implies that all of the approximate balanced POD eigenvalues and modes (suitably normalized) converge as $N \to \infty$, i.e., $\lambda_k^N \to \lambda_k$, $\varphi_k^N \to \varphi_k$, and $\psi_k^N \to \psi_k$ for $k = 1, \ldots, r$. This proves (21).

The error bound (22) follows directly from the representation of ξ in Lemma 1 and the definition of ξ_r. Also, it can be proved that the error bound is finite using the technique in the proof of Lemma 1. □ □

The following result on the convergence of the balanced POD eigenvalues can be found in [31]; for completeness, we include a proof here.

Proposition 2. *Let the above assumptions hold. If $z_i^N \to z_i$ in $L^2(I_z; X)$ for $i = 1, \ldots, q$, and $w_j^N \to w_j$ in $L^2(I_w; X)$ for $j = 1, \ldots, s$, then the balanced POD eigenvalues converge, i.e., $\lambda_k^N \to \lambda_k$ for all k, and also $\varepsilon_{\max}^N = \big[1 + \lambda_1^N\big]^{-1/2}$ converges to $\varepsilon_{\max} = \big[1 + \lambda_{\max}(P\Pi)\big]^{-1/2}$ as $N \to \infty$.*

Proof. Recall $\mathcal{H} = \mathcal{C}\mathcal{B}$, where $\mathcal{C}$ and $\mathcal{B}$ are defined above, and $\mathcal{H}^N$ is defined similarly. Again, using the definitions it can be shown that if each $z_i^N \to z_i$ in $L^2(I_z; X)$ and each $w_j^N \to w_j$ in $L^2(I_w; X)$, then $\mathcal{H}^N \to \mathcal{H}$ in the operator norm as $N \to \infty$. Convergence in norm implies the individual singular values converge since $|\sigma_k - \sigma_k^N| \leq \|\mathcal{H} - \mathcal{H}^N\|$ (see, e.g., [20, Corollary 2.3]). The definition of the balanced POD eigenvalues and Proposition 1 above give $\lambda_k = \lambda_k(P\Pi) = \sigma_k^2$ and $\lambda_k^N = \lambda_k(P^N\Pi^N) = (\sigma_k^N)^2$. Therefore, $\lambda_k^N \to \lambda_k$ for all k, and $\varepsilon_{\max}^N \to \varepsilon_{\max}$ as $N \to \infty$. □ □

7 Numerical Results

In this section we give numerical results for two example problems.

7.1 Example Problem 1

For the first example, consider a one dimensional convection diffusion equation

$$\begin{gathered} w_t(t,x) = \mu w_{xx}(t,x) - \kappa w_x(t,x) + b(x)u(t), \\ w(t,0) = 0, \quad w(t,1) = 0, \\ w(0,x) = w_0(x). \end{gathered}$$

System measurements are taken of the form

$$y(t) = \int_0^1 c(x) w(t,x)\, dx.$$

We assume the functions $b(x)$ and $c(x)$ are square integrable.

For the balanced POD algorithm, we require an abstract formulation of the problem. Briefly, this can be done as follows. Let X be the Hilbert space $L^2(0,1)$ of square integrable functions defined on the interval $(0,1)$ with standard inner product $(f,g) = \int_0^1 f(x)g(x)\, dx$ and norm $\|f\| = (f,f)^{1/2}$. Define the convection diffusion operator $A : D(A) \subset X \to X$ by

$$[Aw](x) = \mu w_{xx}(x) - \kappa w_x(x),$$

where functions in $D(A)$ are twice differentiable and satisfy the above boundary conditions. Define $B : \mathbb{R} \to X$ and $C : X \to \mathbb{R}$ by $[Bu](x) = b(x)u$ and $Cw = (w,c)$. In this way, the PDE system can be written as the infinite dimensional system

$$\dot{w}(t) = Aw(t) + Bu(t), \quad w(0) = w_0, \quad y(t) = Cw(t),$$

where the dot denotes a time derivative.

For this example, we can easily check for convergence of the algorithm and also compare results with computations using matrix approximations of the operators A, B, and C. We test the convergence of the algorithm with $\mu = 0.05$, $\kappa = 1$, control input function $b(x) = 4$ if $0 < x < 1/2$ and $b(x) = 0$ otherwise, and observation function $c(x) = 2$ if $1/2 < x < 1$ and $c(x) = 0$ otherwise.

For the snapshot algorithms, we used standard piecewise linear finite elements for the spatial discretization. For the functional gain computations, we used zero as the initial guess in the Kleinman-Newton iterations. We used a constant time step of $\Delta t = 0.01$ in the trapezoid Lyapunov solver. To approximate the solutions of the differential equations (10) and (11) in Step 2 of the algorithm, we used the trapezoid rule for the time integration.

The computed balanced POD eigenvalues are shown in Figure 1 for various numbers of equally spaced finite element nodes. The balanced POD eigenvalues decay very rapidly to zero, and they converge as the mesh is refined. Since the eigenvalues decay to zero rapidly, the error bound (20) of Theorem 1 leads us to expect that a small value of r for ξ_r^N will provide a good approximation to ξ in the computation of W^*F.

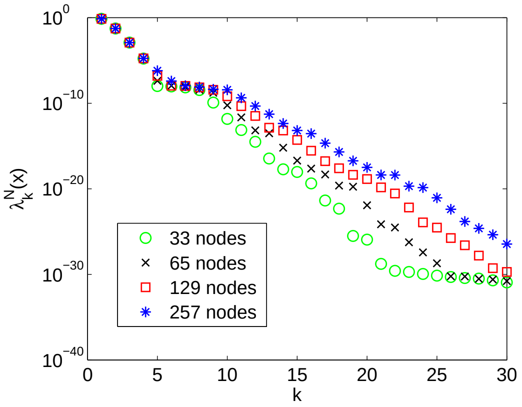

Figure 1: Approximate balanced POD eigenvalues λ_k^N computed using various numbers of equally spaced finite element nodes.

One main goal in the central controller computation is the approximation of the maximum robustness margin $\varepsilon_{\max} = \left[1+\lambda_{\max}(P\Pi)\right]^{-1/2}$. As discussed above, the first approximate balanced POD eigenvalue, λ_1^N, converges to $\lambda_{\max}(P\Pi)$ as the approximate solutions of the differential equations (10) and (11) converge. Therefore, we approximate $\varepsilon_{\max}$ by $\varepsilon_{\max}^N = \left[1+\lambda_1^N\right]^{-1/2}$. As shown above, λ_1^N converged very quickly as the mesh is refined. Table 1 shows the fast convergence of the approximated maximum robustness margin to approximately 0.7596 as the finite element mesh is refined.

Table 1: The maximum robustness margin $\varepsilon_{\max} = \left[1+\lambda_{\max}(P\Pi)\right]^{-1/2}$ approximated by $\varepsilon_{\max}^N = \left[1+\lambda_1^N\right]^{-1/2}$ using various numbers of equally spaced finite element nodes.

nodes	33	65	129	257
$\varepsilon_{\max}^N$	0.7599	0.7597	0.7597	0.7596

The second goal in the central controller computations is the approximation of $W^*F = (I - \alpha\Pi P)^{-1}F$, where $\alpha = -(1-\sigma^{-2})$, $\sigma = (1-\varepsilon^2)^{1/2}$, and $0 < \varepsilon < \varepsilon_{\max}$. For this model problem, $Fy = fy$, where $f \in X$ is the functional gain. As discussed above, to compute W^*F, we need only compute W^*f since $W^*Fy = (W^*f)y$. We approximate $\xi = W^*f$ with ξ_r^N defined in (12).

Figure 2 shows the function $\xi_r^N(x)$ with $r = 3$ and $\varepsilon = (0.9)\,\varepsilon_{\max}^N$ computed using 65 equally spaced finite element nodes. Refining the finite element mesh, decreasing the time step, or increasing r gave very little change in $\xi_r^N(x)$. For example, increasing r from 3 to 4 gave an L^2 norm difference in ξ_r^N of order 10^{-6}.

For all computations, the algorithm gave nearly identical results to computations using finite element matrix approximations of the operators A, B, and C. In particular, we found nearly identical maximum robustness margins and also ξ_r^N was very close to the function ξ computed using the matrix approximations.

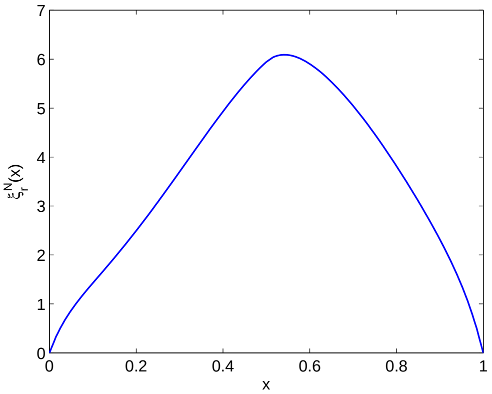

Figure 2: The function $\xi_r^N(x)$ for $r = 3$ and $\varepsilon = (0.9)\,\varepsilon_{\max}^N$ as computed using 65 finite element nodes.

In our computations, we found that using an adaptive solver (such as Matlab's `ode23s`) for the differential equations (10) and (11) in Step 2 of the algorithm gave similar results compared to integrating these differential equations using the trapezoid rule with a constant time step. We did notice that the smaller balanced POD eigenvalues were more accurately approximated using an adaptive solver; however, this did not affect the accuracy in the computation of ξ_r^N since ξ_r^N is constructed using only the largest balanced POD eigenvalues.[2]

7.2 Example Problem 2

For the second example, consider a two dimensional convection diffusion equation

$$w_t(t,x,y) = \mu\big(w_{xx}(t,x,y) + w_{yy}(t,x,y)\big) - x\,w_w(t,x,y) - y\,w_y(t,x,y) + b(x,y)\,u(t),$$

on the unit square $\Omega = [0,1] \times [0,1]$ with $w = 0$ on the bottom, right, and top walls (where $y = 0$, $x = 1$, and $y = 1$) and $w_x = 0$ on the left wall (where $x = 0$). System measurements are taken of the form

$$y(t) = \int_\Omega c(x,y)\,w(t,x,y)\,dx\,dy.$$

We assume the functions $b(x,y)$ and $c(x,y)$ are square integrable. The formulation of the A, B, and C operators are similar to the example above.

We chose $\mu = 0.05$, control input function $b(x,y) = 5\sin(\pi x)\sin(\pi y)$ if $x > 1/2$ and $b(x,y) = 0$ otherwise, and observation function $c(x,y) = 5$ for all x and y. We used standard piecewise bilinear finite elements for the spatial discretization, zero as the initial guess in the Kleinman-Newton iterations, and a constant time step of $\Delta t = 0.01$ in the trapezoid Lyapunov solver. To approximate the solutions of the differential equations (10) and (11) in Step 2 of the algorithm, we used the trapezoid rule for the time integration.

The computed balanced POD eigenvalues are shown in Figure 3 for various numbers of equally spaced finite element nodes in each coordinate direction. The balanced POD eigenvalues decay very rapidly to zero, and they converge as the mesh is refined. Again, we can expect that a small value of r for ξ_r^N will provide a good approximation to ξ in the computation of W^*F. Also, $\varepsilon_{\max}^N$

[2] In an earlier version of this paper [37], we erroneously reported that using an adaptive solver gave superior accuracy in the algorithm. However, our findings were due to an error in the computer implementation of the balanced POD computations for the trapezoid rule with constant time step.

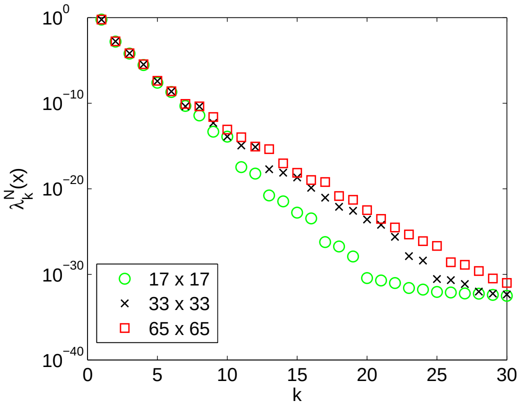

Figure 3: Approximate balanced POD eigenvalues λ_k^N computed using various numbers of equally spaced finite element nodes in each coordinate direction.

quickly converges to to approximately 0.793 as the mesh is refined.

Figure 4 shows the function $\xi_r^N(x,y)$ with $r = 3$ and $\varepsilon = (0.75)\,\varepsilon_{\max}^N$ computed using 33 equally spaced finite element nodes in each coordinate direction. Again, refining the finite element mesh, decreasing the time step, or increasing r gave very little change in $\xi_r^N(x)$.

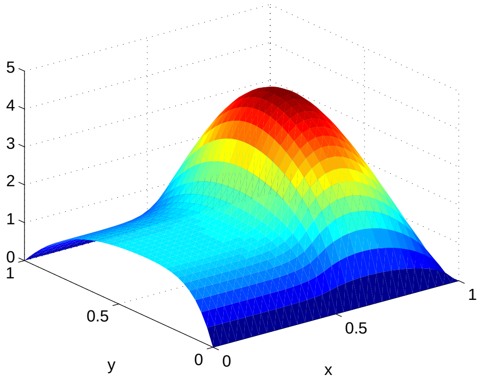

Figure 4: The function $\xi_r^N(x,y)$ for $r = 3$ and $\varepsilon = (0.75)\,\varepsilon_{\max}^N$ as computed using 33 finite element nodes in each coordinate direction.

8 Conclusions and Future Work

We presented an algorithm based on balanced POD for the computation of a robustly stabilizing control law for large-scale finite dimensional linear systems and a class of linear distributed parameter systems. Specifically, we considered the central controller, which is robust with respect to left coprime factor perturbations. The algorithm requires approximate solutions of specific linear differential equations; these computations can be performed accurately and efficiently with existing software. We proved convergence of the algorithm as the approximate solutions of the differential equations converge. The algorithm also does not require access to matrix approximations of system operators.

We believe the algorithm can be modified to compute the challenging features of other robust control laws, such as the standard $\mathcal{H}_\infty$ controller. Furthermore, although we used balanced POD in this work, it is likely that other balanced model reduction algorithms could be modified in a similar way for the robust control computations.

The computed central controller must be reduced in order to be implemented in real time. Also, in the reduction process, it would be desirable to retain as much robustness as possible. These topics will be considered in future work along with testing the algorithm on more challenging PDE systems.

We also note that Curtain [14] has extended the robustly stabilizing controller theory of [19, 15] to a very general class of PDE systems, including systems with unbounded input and output operators as often arise in applications. It is of interest to develop algorithms to compute robust controllers for such systems.

Acknowledgement

This work was supported in part by the Air Force Office of Scientific Research under grant FA9550-07-1-0540, and also by a grant from the University of Missouri Research Board.

References

[1] S. Ahuja and C. Rowley, *Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators*, Journal of Fluid Mechanics, 645 (2010), pp. 447–478. 1

[2] J. A. Atwell and B. B. King, *Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations*, Math. Comput. Modelling, 33 (2001), pp. 1–19. 3

[3] S. Bagheri, L. Brandt, and D. Henningson, *Input–output analysis, model reduction and control of the flat-plate boundary layer*, Journal of Fluid Mechanics, 620 (2009), pp. 263–298. 1

[4] H. T. Banks, R. C. H. del Rosario, and R. C. Smith, *Reduced-order model feedback control design: numerical implementation in a thin shell model*, IEEE Trans. Automat. Control, 45 (2000), pp. 1312–1324. 3

[5] H. T. Banks and K. Ito, *A numerical algorithm for optimal feedback gains in high-dimensional linear quadratic regulator problems*, SIAM J. Control Optim., 29 (1991), pp. 499–515. 1, 2, 3

[6] H. T. Banks, R. C. Smith, and Y. Wang, *Smart Material Structures*, Masson and Wiley, Paris, 1996. 2

[7] A. Barbagallo, D. Sipp, and P. Schmid, *Closed-loop control of an open cavity flow using reduced-order models*, Journal of Fluid Mechanics, 641 (2009), pp. 1–50. 1

[8] B. A. Batten and K. A. Evans, *Reduced-order compensators via balancing and central control design for a structural control problem*, Internat. J. Control, 83 (2010), pp. 563–574. 1

[9] J. Borggaard, J. Burns, and L. Zietsman, *Computational challenges in control of partial differential equations*, in Proceedings of the 2nd AIAA Flow Control Conference, 2004. AIAA 2004-2526. 2

[10] J. Borggaard, M. Stoyanov, and L. Zietsman, *Linear feedback control of a von Kármán street by cylinder rotation*, in Proceedings of American Control Conference, 2010, pp. 5674–5681. 3

[11] J. A. Burns and K. P. Hulsing, *Numerical methods for approximating functional gains in LQR boundary control problems*, Math. Comput. Modelling, 33 (2001), pp. 89–100. 2

[12] J. A. Burns, B. B. King, and D. Rubio, *Feedback control of a thermal fluid using state estimation*, Int. J. Comput. Fluid Dyn., 11 (1998), pp. 93–112. 2

[13] J. A. Burns, E. W. Sachs, and L. Zietsman, *Mesh independence of Kleinman–Newton iterations for Riccati equations in Hilbert space*, SIAM J. Control Optim., 47 (2008), pp. 2663–2692. 3, 3.1

[14] R. F. Curtain, *Robustly stabilizing controllers with respect to left-coprime factor perturbations for infinite-dimensional linear systems*, Systems Control Lett., 55 (2006), pp. 509–517. 8

[15] R. F. Curtain and H. J. Zwart, *An Introduction to Infinite-Dimensional Linear System Theory*, Springer-Verlag, New York, 1995. 1, 2, 2, 8

[16] B. T. Dickinson, J. R. Singler, and B. A. Batten, *A snapshot algorithm for linear feedback flow control design*, in Proceedings of the AIAA Infotech@Aerospace Conference and AIAA Unmanned...Unlimited Conference, 2009. AIAA paper number 2009-1961. 3

[17] F. Feitzinger, T. Hylla, and E. W. Sachs, *Inexact Kleinman-Newton method for Riccati equations*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 272–288. 1, 3

[18] J. S. Gibson and A. Adamian, *Approximation theory for linear-quadratic-Gaussian optimal control of flexible structures*, SIAM J. Control Optim., 29 (1991), pp. 1–37. 2

[19] K. Glover and D. McFarlane, *Robust stabilization of normalized coprime factor plant descriptions with H_∞-bounded uncertainty*, IEEE Trans. Automat. Control, 34 (1989), pp. 821–830. 1, 2, 8

[20] I. C. Gohberg and M. G. Kreĭn, *Introduction to the Theory of Linear Nonselfadjoint Operators*, American Mathematical Society, Providence, R.I., 1969. 6

[21] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1996. 3.1

[22] W. Hackbusch, *Integral Equations*, vol. 120 of International Series of Numerical Mathematics, Birkhäuser Verlag, Basel, 1995. 6, 6

[23] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, Cambridge, 1996. 1

[24] M. Ilak and C. Rowley, *Modeling of transitional channel flow using balanced proper orthogonal decomposition*, Physics of Fluids, 20 (2008). 1

[25] G. M. Kepler, H. T. Tran, and H. T. Banks, *Reduced order model compensator control of species transport in a CVD reactor*, Optimal Control Appl. Methods, 21 (2000), pp. 143–160. 3

[26] D. L. Kleinman, *On an iterative technique for Riccati equation computations*, IEEE Trans. Autom. Control, AC-13 (1968), pp. 114–115. 3

[27] S. Lall, J. E. Marsden, and S. Glavaški, *A subspace approach to balanced truncation for model reduction of nonlinear control systems*, Internat. J. Robust Nonlinear Control, 12 (2002), pp. 519–535. 1

[28] K. Morris and C. Navasca, *Approximation of low rank solutions for linear quadratic control of partial differential equations*, Comput. Optim. Appl., 46 (2010), pp. 93–111. 1

[29] C. W. Rowley, *Model reduction for fluids, using balanced proper orthogonal decomposition*, Internat. J. Bifur. Chaos Appl. Sci. Engrg., 15 (2005), pp. 997–1013. 1, 4, 4.1

[30] Y. Saad, *Numerical solution of large Lyapunov equations*, in Signal Processing, Scattering and Operator Theory, and Numerical Methods (Amsterdam, 1989), vol. 5 of Progr. Systems Control Theory, Birkhäuser Boston, Boston, MA, 1990, pp. 503–511. 3

[31] J. R. Singler, *Snapshot algorithm for balanced model reduction of linear parabolic systems: convergence theory.* to appear in Numerische Mathematik. 1, 4, 4.1, 6

[32] ——, *Approximate low rank solutions of Lyapunov equations via proper orthogonal decomposition*, in Proceedings of the 2008 American Control Conference, 2008, pp. 267–272. 3, 3.1, 3.1

[33] ——, *Optimality of balanced proper orthogonal decomposition for data reconstruction*, Numer. Funct. Anal. Optim., 31 (2010), pp. 852–869. 1, 4, 6, 6, 6, 6

[34] ——, *Convergent snapshot algorithms for infinite dimensional Lyapunov equations*, IMA Journal of Numerical Analysis, 31 (2011), pp. 1468–1496. 3, 3.1, 3.1, 3.1, 1

[35] J. R. Singler and B. A. Batten, *A comparison of balanced truncation methods for closed loop systems*, in Proceedings of the American Control Conference, 2009. 3

[36] ——, *A proper orthogonal decomposition approach to approximate balanced truncation of infinite dimensional linear systems*, Int. J. Comput. Math., 86 (2009), pp. 355–371. 1, 4, 4.1

[37] ——, *Balanced POD algorithm for robust control design for linear distributed parameter systems*, in Proceedings of the American Control Conference, 2010, pp. 4881–4886. 2

[38] L. Sirovich, *Turbulence and the dynamics of coherent structures. I. Coherent structures*, Quart. Appl. Math., 45 (1987), pp. 561–571. 1

[39] M. K. Stoyanov, *Reduced order methods for large scale Riccati equations*, PhD thesis, Virginia Polytechnic Institute and State University, 2009. 3

[40] J. Weidmann, *Linear Operators in Hilbert Spaces*, Springer-Verlag, New York, 1980. 2

[41] K. Willcox and J. Peraire, *Balanced model reduction via the proper orthogonal decomposition*, AIAA Journal, 40 (2002), pp. 2323–2330. 1