

Universität Stuttgart



D. Wirtz and B. Haasdonk

Efficient a-posteriori error estimation for nonlinear kernel-based reduced systems

Stuttgart, December 2010

Institute of Applied Analysis and Numerical Simulation,
University of Stuttgart,
Pfaffenwaldring 57
D-70569 Stuttgart, Germany
{daniel.wirtz,bernard.haasdonk}@mathematik.uni-stuttgart.de
www.agh.ians.uni-stuttgart.de

Abstract In this paper we present a model reduction technique for nonlinear dynamical systems based on kernel expansions and derive an a-posteriori state-space error estimator for the reduction error of kernel systems. Our approach allows for a full offline/online decomposition and rapid online computation of the reduced model. A key ingredient within the error estimation is a local Lipschitz constant estimation that enables rigorous but efficient a-posteriori error estimation. The computation of the error estimator is realized by solving a small auxiliary ODE system during online simulations and estimation iterations can be performed that allow a balancing between estimation sharpness and computation time. Finally, numerical experiments demonstrate the rigour of the error bounds.

Keywords nonlinear dynamical systems · model reduction · kernel methods · a-posteriori error estimates · offline/online decomposition · proper orthogonal decomposition

1 Introduction

Today modelling of real world processes such as biochemical reactions or electric circuits naturally lead to a formulation as dynamical systems with inputs and outputs. Mathematically they are described by (systems of) ODEs and can be roughly categorized into linear and nonlinear types. Even though computational power has been greatly increasing over the last years high-resolution models can result in large-scale dynamical systems that are expensive to simulate. In this context the need for fast simulation results from many-query scenarios like parameter studies or inverse problems where multiple simulations have to be performed for different inputs or initial state configurations. Additionally, some dynamical systems model processes in a real-time setting like interactive simulation environments or control components of cars, robots etc and thus also require fast computation and low hardware requirements. Finally, within all those contexts fast and rigorous error estimation procedures are required to quantify the model errors introduced by the reduction process.

Thus, model reduction techniques for dynamical systems are nowadays subject to intensive research. In particular linear dynamical systems has been getting great attention for the time invariant (LTI) [1–4] or time-variant [5, 6] case in the last few years. A common method for model reduction is the projection of the state variable into some lower dimensional subspace which is ought to capture the system’s main dynamics. As many dynamical systems in fact “live” in a lower dimensional subspace this is a justified assumption which also is made frequently for nonlinear systems. However, there are numerous ways to compute such subspaces: SVD-based approaches like balanced truncation [7], Hankel norm approximation [1, 8], POD/KLT [9] or POD-Greedy [10]. Another class of methods are Krylov-based methods like moment matching [11], for an overview we refer the reader to [12].

Investigation of reduction techniques for nonlinear dynamical systems has been paid less attention to not at least because of the arising difficulties, but yet there are different approaches. One well known attempt for nonlinear model reduction is the trajectory piecewise linear approach [13], which is extended by moment matching techniques in [14] or to a piecewise-polynomial approach in [15]. Recently some effort has been taken to extend the balanced truncation-like approach to nonlinear systems [16], an “approximate reduction” approach is introduced in [17] and model reduction for weakly nonlinear systems by bilinearization has been discussed in [18], for example. In addition, the special class of bilinear quadratic nonlinearities has been investigated in [19]. To name a completely different approach which is based on the empirical interpolation method frequently used for reduced basis methods [20, 21], discrete empirical interpolation (DEIM) for dynamical systems [22] is a promising technique for nonlinear model reduction. Finally, computing a projection subspace using the data-driven ansatz of performing a SVD of presumably statistically representative trajectories or *snapshots* [23–25] is an expensive but well-established method which we will also apply.

The structure of our paper is as follows: In Section 2 we introduce our base dynamical system and discuss the reduction technique thereof. Section 3 is concerned with a-posteriori error estimates and introduces our local Lipschitz constant estimation method. Next, Section 4 presents numerical experiments for synthetic dynamical systems and we conclude with Section 5.

2 Reduction of kernel based systems

Starting point for our investigations is the projection technique introduced in [26], which combines subspace projection with kernel methods. The latter have been getting great attention in the field of machine learning, i.e. classification or regression tasks [27] and also have huge potential in the field of model order reduction of nonlinear dynamical systems. Here, kernel methods can be used to approximate the nonlinearities of dynamical systems with corresponding theoretical foundation [28–30]. However, the approximation techniques are out of the scope of this article. Instead we focus on the projection behaviour of nonlinear systems whose inner dynamics are already given by a kernel expansion and the resulting a-posteriori error estimators.

2.1 The base dynamical system

As motivated above the base system we will consider throughout this paper is of the form

$$x'(t) = f(x) + Bu(t) \quad (1)$$

$$x(0) = x_0 \quad (2)$$

$$y(t) = Cx(t) \quad (3)$$

where $x(t) \in \mathbb{R}^d$ denotes the state of the system at times $t \in [0, T]$, $0 \leq T < \infty$, $x_0 \in \mathbb{R}^d$ the initial condition, $u : [0, T] \rightarrow \mathbb{R}^m$ an input function with input mapping matrix $B \in \mathbb{R}^{d \times m}$ and, if applicable, some output mapping $C \in \mathbb{R}^{k \times d}$ for $y(t) \in \mathbb{R}^k$. Further our central assumption is to have a nonlinear kernel expansion

$$\begin{aligned} f : \mathbb{R}^d &\longrightarrow \mathbb{R}^d \\ x &\longmapsto f(x) = \sum_{i=1}^N \alpha_i \Phi(x, x_i), \end{aligned} \quad (4)$$

where $X := \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ are the centers or *support vectors* of the expansion and $\alpha_i \in \mathbb{R}^d$ the coefficient vectors. The function Φ is a *kernel*, which is most generally viewed simply a symmetric function $\Phi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$. In applications, however, there are used more specific kernels where the most important property positive definiteness is given by

$$\forall n \in \mathbb{N}, \{x_1, \dots, x_n\} \subset \mathbb{R}^d : \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j \Phi(x_i, x_j) \geq 0 \quad \forall \alpha \in \mathbb{R}^n.$$

The key analytical property is that for some open $\Omega \subset \mathbb{R}^d$ each symmetric positive definite kernel spans a unique Hilbert space $\mathcal{N}_\Phi(\Omega)$ of functions. Those Hilbert spaces have a special reproducing property and are commonly referred to as reproducing kernel Hilbert spaces (RKHS). For more details on kernels, RKHS and their applications we refer to [28, 31, 32], for example. In our context those RKHS serve as a base class for the dynamical systems functions f , so in particular we assume with (4) that $f \in \mathcal{N}_\Phi(\Omega)$.

Having introduced the underlying dynamical system (1)-(3) we now can detail the reduction approach using subspace projection. Therefore we assume to have two biorthogonal projection matrices

$$V, W \in \mathbb{R}^{d \times r}, \quad W^t V = I_r \quad (5)$$

with $r \ll d$ denoting the reduced system's dimension. Biorthogonality (5) relaxes the common requirement of having an orthogonal projection matrix V with $V^t V = I_r$. Now applying a Petrov-Galerkin projection on the base system (1)-(3) we obtain the reduced system

$$z'(t) = W^t f(Vz(t)) + W^t Bu(t) \quad (6)$$

$$z(0) = W^t x_0 \quad (7)$$

$$y^r(t) = CVz(t), \quad (8)$$

where $z(t) \in \mathbb{R}^r$ now denotes the reduced system's state. Consequently, the reconstructed approximate solution and output is then given by

$$x^r(t) := Vz(t) \in \mathbf{V}, \quad y^r(t) := Cx^r(t), \quad t \in [0, T], \quad (9)$$

respectively, where $\mathbf{V} := \text{span}\{V\}$ is the subspace of \mathbb{R}^d in which the reconstructed state variable resides. Now the reduction aim is to find matrices V, W such that $x^r(t) \approx x(t)$ and thus $y^r(t) \approx y(t)$ for different initial values and inputs. Of course each dimension of the reduced state $z(t)$ cannot be interpreted directly like the full $x(t)$ state. But one could interpret the reduced vector entries as contribution factors of the corresponding vectors of V to the full solution $x(t)$. As there are many possibilities (with different advantages) to compute such matrices we will assume their computation method as a blackbox. Important here is that our method is applicable using any basis/projection matrix generation method. However, computing a POD of snapshots gained from simulations for different training

input/initial value configurations and choosing a subspace using the most significant singular values turns out to be a practical method which we will also use within our numerical experiments in Section 4.

Unfortunately, the reduced system (6) is still expensive to simulate since the reduced state variable is reconstructed in \mathbb{R}^d , f evaluated at $Vz(t)$ and the result projected via W^t to \mathbb{R}^r again. As will be shown in Sections 2.3.1 and 2.3.2 the structure of the kernel expansion (4) allows for dramatic reduction of the computation costs for certain kernels.

2.2 Coefficient projection

In the following sections we discuss the different projection aspects of the reduction technique for kernel-based dynamical systems of the form (1)-(3). At first, the projection into \mathbb{R}^r by W^t can be applied straightforward to the coefficient vectors α_i via $\beta_i := W^t \alpha_i$, $i = 1 \dots N$ resulting in a reduced function f^r defined by

$$\begin{aligned} f^r : \mathbb{R}^r &\longrightarrow \mathbb{R}^r \\ z &\longmapsto f^r(z) := \sum_{i=1}^N \beta_i \Phi(Vz, x_i). \end{aligned} \quad (10)$$

The new coefficient vectors β_i obviously have reduced dimension \mathbb{R}^r compared to $\alpha_i \in \mathbb{R}^d$.

2.3 Kernel projection

Next, in order to avoid \mathbb{R}^d as input argument dimension, we consider two special classes of kernels. Before we get into details on these, we recall that for any error statements suitable norms must be chosen. Hence, let $G \in \mathbb{R}^{d \times d}$ be a symmetric positive definite matrix defining a scalar product $\langle x, y \rangle_G := x^t G y$ on \mathbb{R}^d with induced norm $\|x\|_G := \sqrt{\langle x, x \rangle_G}$. Choosing $G := I_d$ will yield the standard Euclidean inner product $\langle \cdot, \cdot \rangle$ and 2-norm $\|\cdot\|$.

2.3.1 Inner product kernels

The first class of kernels that allow efficient argument evaluations are the inner product kernels which also have been mentioned in [26]. So, assume Φ to be such a kernel, i.e. $\Phi(x, y) = \phi(\langle x, y \rangle_G)$ for some scalar function $\phi : \mathbb{R} \rightarrow \mathbb{R}$. Then we deduce

$$\Phi(Vz, x_i) = \phi(\langle Vz, x_i \rangle_G) = \phi(\langle z, z_i \rangle) =: \Phi^r(z, z_i) \quad (11)$$

for $z_i := V^t G x_i \in \mathbb{R}^r$, $i = 1 \dots N$. So all we need to do here is project the centers x_i into the low-dimensional subspace and the evaluations can be computed rapidly and *lossless* during the reduced simulation. This special type of kernel inherits the property of scalar products to ignore components of vectors which lie in the orthogonal complement of \mathbf{V} in \mathbb{R}^d and thus those parts can be dropped beforehand. Some examples for those kernels are the standard linear kernel $\Phi(x, y) := \langle x, y \rangle_G$ or the polynomial kernels $\Phi(x, y) := (1 + \langle x, y \rangle_G)^p$ of degree $p \in \mathbb{N}$.

2.3.2 Translation- and rotation invariant kernels

In addition to [26] we present here a further class of kernels that allow for efficient *lossless* argument evaluations: The translation and rotation-invariant kernels which are defined by

$$\Phi(x, y) := \phi(\|x - y\|_G) \quad (12)$$

for some scalar function $\phi : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$. Those kernels are also commonly referred to as *radial basis functions*. We recall (5) and additionally assume

$$x_i \in \mathbf{V}, \text{ i.e. } x_i = Vz_i \quad i = 1 \dots N, \quad (13)$$

for some $z_i \in \mathbb{R}^r$. Then we obtain

$$\Phi(Vz, x_i) = \phi(\|Vz - Vz_i\|_G) = \phi(\|z - z_i\|_{V^tGV}) =: \Phi^r(z, z_i), \quad (14)$$

with V^tGV being a small $\mathbb{R}^{r \times r}$ matrix inducing a new norm on \mathbb{R}^r . An example of a translation and rotation-invariant kernel is the Gaussian kernel

$$\Phi(x, y) := e^{-\frac{\|x-y\|^2}{\beta}}, \quad (15)$$

for a $\beta > 0$. For further examples and characterizations of the above mentioned kernels and others we again refer to [27, 32].

So, for any of the specific kernels above and using (10) we obtain from (6)-(8) an equivalent reduced system whos evaluation is independent from d and has only low dimensional components:

$$z'(t) = \sum_{i=1}^N \beta_i \Phi^r(z, z_i) + B^r u(t) \quad (16)$$

$$z(0) = z_0 \quad (17)$$

$$y^r(t) = C^r z(t), \quad (18)$$

with $\beta_i := W^t \alpha_i$, $z_0 := W^t x_0$, $B^r := W^t B$, $C^r := CV$.

Remark 1 The assumption (13) is of a technical nature. We either extend the subspace \mathbf{V} by the span of the x_i or, as is also the case for our later applications, the kernel expansion is created after the subspace \mathbf{V} is computed which allows to choose $x_i \in \mathbf{V}$ via a transformation $x_i = VW^t x_i$ in the first place.

3 A-posteriori error estimates

When dealing reduced dynamical systems it is essential to have means to compute or at least estimate the error that is made during simulations. For the reduced systems (16)-(18) we derive rigorous and efficient error estimators. In the field of error estimations for nonlinear dynamical systems little work has been done so far, error estimations for POD and Ritz projection can be found in [33], a-posteriori error analysis for reduced non-parametric dynamical systems is considered in [34] and error analysis of POD reduction schemes for optimal control can be found in [35]. Recent work on a-posteriori error estimation has also been done for nonlinear parametrized evolution equations [36].

The main ingredients to our estimators are an application of Gronwall's Lemma and local Lipschitz constant estimations. Note that the Gronwall's Lemma is frequently used to bound ODE solution-/errors *a-priori*, whereas we will use it to provide rigorous *a-posteriori* error bounds which can be computed along with the reduced system for any initial value x_0 and input $u(t)$. Gronwall's Lemma has also been applied for a-posteriori error estimations in the context of non-stable and time-dependent parameterized linear systems [2], for example. Our estimators bound the error in the state-space for finite time intervals instead of giving frequency domain errors for long time simulations.

3.1 A first a-posteriori error estimator

For the reconstructed full state variable $x^r(t)$ as given in (9) we can define the state space and output reduction error

$$\begin{aligned} e(t) &:= x(t) - x^r(t) = x(t) - Vz(t), \quad t \in [0, T] \\ e_y(t) &:= y(t) - y^r(t) = Cx(t) - CVz(t), \quad t \in [0, T]. \end{aligned} \quad (19)$$

In order to derive an error estimator we have a look at the error system

$$e'(t) = f(x(t)) - VW^t f(Vz(t)) + Bu(t) - VW^t Bu(t) \quad (20)$$

$$e(0) = x_0 - VW^t x_0. \quad (21)$$

The analysis of the equation (20) will be in our main focus, since (21) can be readily computed. Once a bound for the state error is available the output error can be bounded using (19) which gives

$$\|e_y(t)\| \leq \|C\|_G \|e(t)\|_G, \quad t \in [0, T].$$

With (21) the error $e(t)$ satisfies

$$e(t) = \int_0^t e'(s) ds + e(0) = \int_0^t e'(s) ds + (I - VW^t)x_0. \quad (22)$$

Having the representation (22), our goal is to derive an estimator of the form

$$\|e(t)\|_G \leq \Delta^{V,W}(t)$$

with rapidly computable bound $\Delta^{V,W}(t)$. A central part for the following analysis will be Gronwall's Lemma which we use here in a tailored version in difference to the original [37].

Lemma 1 (Gronwall's Lemma) *Assume $u(t), \alpha(t), \beta(t)$ to be real valued continuous functions and further let*

$$u(t) \leq \alpha(t) + \int_0^t \beta(s)u(s) ds \quad \forall t \in \mathbb{R}_0^+.$$

Then we have

$$u(t) \leq \alpha(t) + \int_0^t \alpha(s)\beta(s)e^{\int_s^t \beta(r) dr} ds \quad \forall t \in \mathbb{R}_0^+.$$

Now we are ready to state our first estimation theorem:

Theorem 1 (Global Lipschitz constant a-posteriori error estimator (GLE)) *Let an error system be given as defined in (20)-(21). Let Φ be Lipschitz w.r.t. the first variable, i.e. for some $c_\Phi > 0$ we have*

$$|\Phi(x_1, y) - \Phi(x_2, y)| \leq c_\Phi \|x_1 - x_2\|_G \quad \forall x_1, x_2, y \in \mathbb{R}^d.$$

Then the following continuous, monotonously increasing error estimator can be obtained:

$$\begin{aligned} \|e(t)\|_G &\leq \Delta^{V,W}(t) \quad \forall t \in [0, T] \\ \Delta^{V,W}(t) &:= e^{c_f t} \left(\int_0^t \|(I - VW^s)(f(Vz(s)) + Bu(s))\|_G ds + E_{x_0} \right) \\ c_f &:= c_\Phi \sum_{i=1}^N \|\alpha_i\|. \end{aligned}$$

Here $\Delta^{V,W}(t)$ stands for an error estimator that rigorously bounds the error introduced by the subspace projection.

Proof Recall that f is a kernel expansion (4). Given $x_1, x_2 \in \mathbb{R}^d$ we obtain a global Lipschitz constant via

$$\begin{aligned} \|f(x_1) - f(x_2)\|_G &= \left\| \sum_{i=1}^N \alpha_i \Phi(x_1, x_i) - \sum_{i=1}^N \alpha_i \Phi(x_2, x_i) \right\|_G \\ &\leq \sum_{i=1}^N \|\alpha_i\| c_\Phi \|x_1 - x_2\|_G = c_f \|x_1 - x_2\|_G. \end{aligned} \quad (23)$$

Reordering of equation (20) gives

$$e'(t) = f(x(t)) - f(Vz(t)) + (I - VW^t) (f(Vz(t)) + Bu(t)). \quad (24)$$

Using the abbreviation

$$E_{x_0} := \|(I - VW^t)x_0\|_G$$

and equations (23),(24) on the error representation (22) we get

$$\begin{aligned} \|e(t)\|_G &\leq \left\| \int_0^t f(x(s)) - f(Vz(s)) + (I - VW^t) (f(Vz(s)) + Bu(s)) ds \right\|_G + E_{x_0} \\ &\leq \int_0^t \|(I - VW^t) (f(Vz(s)) + Bu(s))\|_G ds + E_{x_0} + \int_0^t c_f \|e(s)\|_G ds \end{aligned} \quad (25)$$

Finally, applying Lemma 1 for $\alpha(s)$ equal to the first two summands of (25) and $\beta(s) = c_f$ yields the result.

3.2 Local Lipschitz constants

Having this first error estimator from Theorem 1 one can easily see that if the Lipschitz constant c_Φ is large the estimator will grow exponentially. For example, the Gaussian kernel (15) is Lipschitz with $c_\Phi = \sqrt{2/\beta}$. As to expect, experiments confirm that a small β renders the estimator practically useless and thus an improvement is necessary. Therefore, we propose a *local Lipschitz constant* estimation that utilizes the current position $Vz(t)$ of the reduced state variable and optionally a coarse a-priori bound. The key to the local Lipschitz estimations is to use radial basis functions (12) which are induced by a certain class of scalar functions ϕ . So at first we introduce a new class of functions which we will use in our results.

Definition 1 (Bell functions) A function $\phi : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ is called a *bell function*, if

$$\begin{aligned} i) \quad &\phi \in C^2(\mathbb{R}_0^+), \\ ii) \quad &\exists B > 0 : \phi(x) < B \quad \forall x \in \mathbb{R}^+, \\ iii) \quad &\phi'(0) \leq 0, \end{aligned} \quad (26)$$

$$iv) \quad \exists x_m > 0 : \phi''(x) \begin{cases} > 0, & x > x_m \\ < 0, & x < x_m \\ = 0, & x = x_m \end{cases} \quad \forall x \in \mathbb{R}^+. \quad (27)$$

Condition *iv)* means that ϕ is strictly concave on the left side of x_m and strictly convex on the right side of x_m . Further *iii)* implies that $\phi'(x_m) < \phi'(x) \quad \forall x \in \mathbb{R}^+$. Let us further denote the set of all bell functions with \mathcal{B} .

For example the Gaussian kernel (15) is induced by a bell function $\phi(r) = e^{-\frac{r^2}{\beta}}$ with $x_m = \sqrt{\frac{\beta}{2}}$. Now for this class of functions we can obtain Lipschitz constant estimations that utilize locality characterized by two aspects. Given a fixed $y \in \mathbb{R}_0^+$ and a maximum distance C for any points $x \in \mathbb{R}_0^+$ from y the bell function properties will allow to obtain a smaller Lipschitz constant than the global maximum absolute derivative. In fact, the absolute maximum local *secant* gradient is sufficient as Lipschitz constant as we will see. Those aspects will later relate to the current state space position of the reduced system and a coarse a-priori bound, respectively.

But before we can come back to the multidimensional setting we need some closer analysis of the scalar bell functions \mathcal{B} which will yield the base theory for our error estimators.

Lemma 2 (Local Lipschitz estimation using gradients (LGL)) *Let $\phi \in \mathcal{B}$. Then for any $y \in \mathbb{R}_0^+$, $C > 0$ and $\Omega := B_C(y) \cap \mathbb{R}_0^+$ we have*

$$|\phi(x) - \phi(y)| \leq L_{\phi,C}^g(y) |x - y| \quad \forall x \in \Omega,$$

with

$$L_{\phi,C}^g(y) := |\phi'(p_C(y))|$$

and

$$p_C(y) := \begin{cases} y - C, & y - C \geq x_m \\ x_m, & x_m \in \Omega \\ y + C, & y + C \leq x_m \end{cases}. \quad (28)$$

Proof From the bell function conditions (26)-(27) we know that

$$\|\phi'\|_{L^\infty(\mathbb{R}_0^+)} = |\phi'(x_m)| \quad (29)$$

By condition (27) we further see that ϕ' is nonpositive and strictly monotoneously decreasing/increasing on $[0, x_m]/[x_m, \infty[$ which means the reverse for $|\phi'|$. Thus, if $x_m \notin B_C(y)$, $|\phi'|$ takes its maximum value at the border of Ω that is closest to x_m , yielding the choice of $p_C(y)$ and $\|\phi'\|_{L^\infty(\Omega)} = |\phi'(p_C(y))|$. The result follows directly by an application of the mean value theorem.

Note that if C is large enough we simply obtain $L_{\phi,C}^g(y) = |\phi'(x_m)| \quad \forall y \in \mathbb{R}_0^+$ by (28). This results in a nonlocal estimation similar to Theorem 1 since then we have (29). Hence, an estimator that avoids falling back to $|\phi'(x_m)|$ for large C would be desirable. This can be achieved when switching to secant gradients instead of derivative evaluations. Therefore, for arbitrary $y \in \mathbb{R}_0^+$, we introduce a functional for the secant gradient by

$$S_y : \mathcal{B} \longrightarrow C^1(\mathbb{R}_0^+) \\ \phi \longmapsto S_y(\phi)(x) := \begin{cases} \frac{\phi(x) - \phi(y)}{x - y}, & x \neq y \\ \phi'(x), & x = y \end{cases} \quad x \in \mathbb{R}_0^+. \quad (30)$$

One can easily see that S_y is well-defined by basic analysis. Now, using (30) we can improve the estimation in Lemma 2 when finding the *maximum* local secant

$$\max_{x \in \mathbb{R}_0^+} |S_y(\phi)(x)|, \quad (31)$$

for a given $y \in \mathbb{R}_0^+$. Before we formulate our improved estimator see that (31) in fact has a unique solution and some interesting properties regarding x_m :

Lemma 3 (Maximum absolute secant gradient position) *For any $\phi \in \mathcal{B}$ the map*

$$\varphi_\phi : \mathbb{R}_0^+ \longrightarrow \mathbb{R}_0^+ \\ y \longmapsto \arg \max_{x \in \mathbb{R}_0^+} |S_y(\phi)(x)| \quad (32)$$

is well-defined and exactly one of the following cases holds:

$$y < x_m \leq \varphi_\phi(y) \quad (33)$$

$$\varphi_\phi(y) \leq x_m < y \quad (34)$$

$$y = x_m = \varphi_\phi(x_m), \quad (35)$$

where x_m is given from the bell function property (27).

The cases (33)-(35) mean, unless $y = x_m$, that $\varphi_\phi(y)$ must be located on the "other" side of x_m seen from y . This result will also be useful at a later stage and we omit the proof here as it is of technical but elemental nature and would only distract from our main focus. Using Lemma 3 and similar ideas as in the proof of Lemma 2 we obtain an improved local Lipschitz estimation using secant gradient information:

Lemma 4 (Local Lipschitz estimation using secants (LSL)) *Let $\phi \in \mathcal{B}$. Then for any $C > 0$, $y \in \mathbb{R}^+$ and $\Omega := B_C(y) \cap \mathbb{R}_0^+$ we have*

$$|\phi(x) - \phi(y)| \leq L_{\phi,C}^s(y) |x - y| \quad \forall x \in \Omega,$$

with

$$L_{\phi,C}^s(y) := |S_y(\phi)(p_C(y))|,$$

and

$$p_C(y) := \begin{cases} y - C & y - C \geq \varphi_\phi(y) \\ \varphi_\phi(y) & \varphi_\phi(y) \in \Omega \\ y + C & y + C \leq \varphi_\phi(y) \end{cases}. \quad (36)$$

Remark 2 (Connection of estimators) For $\phi \in \mathcal{B}$ we actually can show that

$$\begin{aligned} L_{\phi,C}^s(y) &< L_{\phi,C}^g(y) \quad \forall y \in \mathbb{R}_0^+, y \neq x_m \\ L_{\phi,C}^s(x_m) &= L_{\phi,C}^g(x_m). \end{aligned}$$

This means the secant estimation is always the better estimation except for x_m , for the price of having to solve the maximization problem (31). An efficient numerical implementation will be given in Section 3.5.

3.3 Improved error estimator

Having those local Lipschitz constant estimations for bell functions we can generalize them to our multidimensional kernel setting via the following Corollary.

Corollary 1 (Local Lipschitz constants for bell function kernels) *For any $\phi \in \mathcal{B}$ the function*

$$\Phi(x, z) = \phi(\|x - z\|_G) \quad (37)$$

is a rotation- and translation invariant kernel and $\forall y, z \in \mathbb{R}^d, C > 0$ we have a local Lipschitz constant estimation of the form

$$|\Phi(x, z) - \Phi(y, z)| \leq L_{\phi,C}^\alpha(\|y - z\|_G) \|x - y\|_G \quad \forall x \in B_C(y), \alpha \in \{g, s\}.$$

Here $L_{\phi,C}^\alpha$ is one of the local Lipschitz estimations $L_{\phi,C}^g, L_{\phi,C}^s$ as stated in Lemma 2 or 3.

Proof Fix $y, z \in \mathbb{R}^d, C > 0$ and let $L_{\phi,C}^\alpha(r)$ be one of the local Lipschitz estimators from 2 or 3. Further define

$$R := \{x \in \mathbb{R}^d \mid \|x - z\|_G \in B_C(\|y - z\|_G)\},$$

which is a ring around z with distance $\|y - z\|_G$ and thickness C . Knowing that

$$\left| \|x - z\|_G - \|y - z\|_G \right| \leq \|x - y\|_G \quad \forall x, y, z \in \mathbb{R}^d,$$

it is easy to see that $B_C(y) \subset R$. Using this and (37) gives the desired result.

With this additional insight we can create the link to our reduced system error estimation. As shortly mentioned in the beginning of Section 3.2, optionally a given a-priori coarse error bound

$$\|e(t)\|_G \leq C(t) \quad \forall t \in [0, T] \quad (38)$$

can be used to improve the error estimation. So, assuming we have some upper bound (38) for the reduction error, we can reorder $e(t) = x(t) - Vz(t)$ to $x(t) = Vz(t) + e(t)$ which means nothing else but

$$x(t) \in B_{C(t)}(Vz(t)) \quad \forall t \in [0, T].$$

Now we can identify the x, y and $z \in \mathbb{R}^d$ from Corollary 1 with $x(t), Vz(t)$ of the full/reduced solution and the kernel expansion centers x_i , respectively. Moreover, it does not matter if we assume the

constant C in Corollary 1 to be time-dependent or not since $Vz(t)$ is time-dependent anyway. For any $i \in \{1 \dots N\}$ this leads to time-dependent choices (28)/(36) with $y \equiv \|Vz(t) - x_i\|$ in the context of Lemma 2/4. Hence, we are able to obtain local Lipschitz constant estimations at each time t using the current reduced simulation's state $Vz(t)$ as "center" of locality.

With this in mind we can state our main estimator result using Gronwall's Lemma in a similar way as in Theorem 1, but this time we have a time-dependent $\beta(t)$ due to the previously introduced local Lipschitz constant estimations.

Theorem 2 (A-posteriori error estimators using local Lipschitz constant estimations) *Let the error system be given as in (20)-(21), $\phi \in \mathcal{B}$ and Φ its induced rotation- and translation invariant kernel. Further assume that we have an a-priori error bound*

$$\|e(t)\| \leq C(t), \quad C(t) \in \mathbb{R}_0^+ \cup \{\infty\}, \quad t \in [0, T]$$

and $\alpha \in \{d, s\}$. Then the following continuous, monotonously increasing a-posteriori error estimator can be obtained:

$$\|e(t)\| \leq \Delta^{V,W}(t) := \alpha(t) + \int_0^t \alpha(s) \beta(s) e^{\int_s^t \beta(r) dr} ds \quad \forall t \in [0, T] \quad (39)$$

with

$$\alpha(t) := \int_0^t \|(I - VW^t)(f(Vz(s)) + Bu(s))\|_G ds + E_{x_0} \quad (40)$$

$$\beta(t) := \sum_{i=1}^N \|\alpha_i\| L_{\phi, C(t)}^\alpha (\|Vz(t) - x_i\|_G). \quad (41)$$

To complete the estimation results we will look into the applications of the coarse bound (38) and some computational aspects regarding the error estimators. Before we start we introduce the notation

$$d_i(t) := \|Vz(t) - x_i\|, \quad i = 1 \dots N. \quad (42)$$

for the distance of the reduced state $Vz(t)$ to each center x_i during a reduced simulation for a given system configuration.

3.4 Utilization of coarse bound $C(t)$

We still need to justify the assumption (38) since it means having an a-priori bound that is used to derive an a-posteriori error estimator. The core idea is now to use this coarse bound in an iterative scheme. Starting with a coarse initial error estimation we can reuse an a-posteriori estimation as a-priori bound $C(t)$ for another iteration of the error estimation procedure. By the next theorem we also see that actual progress can be made under reasonable assumptions and the iterative scheme reproduces the previous estimates in the worst case.

Theorem 3 (Convergence of error estimator iteration) *Let the conditions of Theorem 2 hold using the LGL estimation method from Lemma 2 and define*

$$C_0(t) := \alpha(t) + \int_0^t \alpha(s) \beta_0 e^{(t-s)\beta_0} ds \quad (43)$$

with $\alpha(t)$ as in (40) and

$$\beta_0 := \sum_{i=1}^N \|\alpha_i\| |\phi'(x_m)|. \quad (44)$$

Define the sequence of estimation iterates as

$$C_n(t) := \Delta_{C_{n-1}}^{V,W}(t) \quad n \in \mathbb{N}, \quad (45)$$

where the subscript denotes the function used as a-priori bound in the estimation procedure. Recall the distances (42), x_m from (27) and let

$$\Gamma_0 := \{t \in [0, T] \mid \exists i \in \{1 \dots N\} : x_m \notin B_{C_0(t)}(d_i(t))\}. \quad (46)$$

If $\Gamma_0 \neq \emptyset$ then

$$\left\| \Delta_{C_n}^{V,W} \right\|_{L^1([0, T])} < \|C_n\|_{L^1([0, T])} \quad \forall n \in \mathbb{N}_0 \quad (47)$$

or else $\Delta_{C_n}^{V,W}(t) = C_0(t) \quad \forall n \in \mathbb{N}_0$. Additionally, the sequence (45) converges to a lower error bound

$$\Delta_\infty^{V,W}(t) := \lim_{n \rightarrow \infty} \Delta_{C_n}^{V,W}(t), \quad t \in [0, T]. \quad (48)$$

Proof We prove (47) by induction. It actually suffices to show that

$$\Delta_{C_n}^{V,W}(t) \begin{cases} < C_n(t) & t \in \Gamma_0 \\ \leq C_n(t) & t \notin \Gamma_0 \end{cases} \quad \forall n \in \mathbb{N}_0 \quad (49)$$

as this directly implies (47). Moreover, from (43) or more precisely (44), and (28) we immediately see that $\Delta_{C_n}^{V,W}(t) \leq C_0(t) \quad \forall t \in [0, T]$, as C_0 is the estimator obtained for the worst case choice in (28). From this we can derive two conclusions: First, since $0 \leq C_n(t) \leq C_0(t)$ for any $t \in [0, T]$ we also have the pointwise convergence (48) (up to a subsequence). Second, we see that if we used $C_n(t)$ instead of $C_0(t)$ in the definition of (46) and called the resulting set Γ_n we would have $\Gamma_0 \subset \dots \subset \Gamma_{n-1} \subset \Gamma_n \quad \forall n \in \mathbb{N}$ as the bounds C_n only get tighter and the trajectory $Vz(t)$ of course stays the same over all iterations. Due to this inclusion it is sufficient to restrict on the times in Γ_0 in (49) as any augmentation of Γ_0 would only improve the estimates at more timesteps.

In order to show (49) we recall (28) and first define $p_n^i(t) := p_{C_n(t)}(d_i(t))$, $n \in \mathbb{N}_0$ as a shorthand. Now, for $n = 0$ we see that

$$\beta(t) = \sum_{i=1}^N \|\alpha_i\| |\phi'(p_0^i(t))| < \sum_{i=1}^N \|\alpha_i\| |\phi'(x_m)| = \beta_0 \quad \forall t \in \Gamma_0$$

by (28) and the bell function's property (27). As β is continuous we get from (39) that $\Delta_{C_0}^{V,W}(t) < C_0(t) \quad \forall t \in \Gamma_0$. For the induction step we first denote by $\beta_n(t)$ the function (41) when using $C_n(t)$ as bound. Now, for a $t \in \Gamma_0$ there exists without loss of generality an $i \in \{1 \dots N\}$ with $p_n^i(t) = d_i(t) - C_n(t)$. Now as $C_n(t) = \Delta_{C_{n-1}}^{V,W}(t) < C_{n-1}$ by induction assumption we have $p_n^i(t) = d_i(t) - C_n(t) > d_i(t) - C_{n-1}(t) = p_{n-1}^i(t)$ and hence $|\phi'(p_n^i(t))| < |\phi'(p_{n-1}^i(t))|$ by (27). But this immediately yields $\beta_n(t) < \beta_{n-1}(t)$, giving the desired result $\Delta_{C_n}^{V,W}(t) < \Delta_{C_{n-1}}^{V,W}(t) = C_n(t)$. The case $\Gamma_0 = \emptyset$ is easily obtained as each iteration simply reproduces C_0 .

So the key assumption here is that the coarsest estimation C_0 is small enough that for some time t and index i the point x_m is not contained in the ball $B_{C_0(t)}(\|Vz(t) - x_i\|)$, allowing for an improved estimation. This is a reasonable assumption as the expansion centers are drawn from sampled trajectories and are scattered over the high dimensional state space. As a worst case scenario, if the initial error E_{x_0} is big enough, i.e. $x_m \in B_{E_{x_0}}(\|Vz(t) - x_i\|) \quad \forall t \in [0, T], i \in \{1 \dots N\}$, then the estimation process cannot do better than $\|e(t)\| \leq C_0(t)$ as already $C_0(0) = E_{x_0}$ and C_0 is monotonously increasing.

The same theorem applies when using the LSL estimator and

$$\beta_0 := \sum_{i=1}^N \|\alpha_i\| |S_y(\varphi_\phi(y))|$$

for C_0 , albeit we have to do more analysis regarding the absolute secant gradient. One nice feature of this iterative scheme is that it allows to balance the error bound's sharpness against the computational costs.

3.5 Computation of error estimators

In order to compute the error estimators from Theorem 2 we propose a formulation which allows rapid computation of the error estimators parallel to the reduced simulation *independent* from the full system's high dimension. This is possible due to the full offline/online separability of the error estimator and an effective scheme how to solve the problem (31). Before we introduce our approaches we state a reformulation of the error estimator into a system of three additional ODEs that can be solved parallel to the reduced simulation. The central idea is to introduce two auxiliary functions u, v which yield the estimator via $\Delta^{V,W}(t) = \alpha(t) + e^{u(t)}v(t)$.

Lemma 5 (ODE formulation of improved error estimator) *Let the conditions of Theorem 2 hold. Then we can compute $\Delta^{V,W}(t)$ by solving the following system of ODE's:*

$$\alpha'(t) = \|(I - VW^t)(f(Vz(t)) + Bu(t))\| \quad \alpha(0) = E_{x_0} \quad (50)$$

$$u'(t) = \beta(t) \quad u(0) = 0 \quad (51)$$

$$v'(t) = \alpha(t)\beta(t)e^{-u(t)} \quad v(0) = 0. \quad (52)$$

3.5.1 Offline/online decomposition

The rapid computation of the error estimator (39) is possible by decomposing the ODE part (50) in an offline/online fashion similar to [2]. Here, we restrict ourselves to discuss the case for the local Lipschitz constant estimators presented in Theorem 2, as the global Lipschitz estimator from Theorem 1 is decomposed similarly but yet more simple.

Theorem 4 (Offline/online decomposition for local Lipschitz constant error estimators) *Let the error system (20)-(21) be given and the conditions from Theorem 2/Lemma 5 hold. Further let*

$$M_\alpha := (\alpha_1 \dots \alpha_N) \in \mathbb{R}^{d \times N}$$

$$\varphi_i(t) := (\Phi(Vz(t), x_i))_{i=1}^N \in \mathbb{R}^N, \quad t \in [0, T]$$

and recall as mentioned at the beginning of Section 2.3 that G is the matrix inducing the used norm $\|\cdot\|_G$. Then the offline computations are

$$\tilde{M} := V(W^t M_\alpha), \quad \tilde{B} := V(W^t B), \quad G_1 := M_\alpha^t G, \quad G_2 := B^t G \quad (53)$$

$$M_1 := G_1 M_\alpha - 2G_1 \tilde{M} + \tilde{M}^t G \tilde{M} \in \mathbb{R}^{N \times N} \quad (54)$$

$$M_2 := G_1 B - \tilde{M}^t G_2 - G_1 \tilde{B} + \tilde{M}^t G \tilde{B} \in \mathbb{R}^{N \times m} \quad (55)$$

$$M_3 := G_2 B - 2G_2 \tilde{B} + \tilde{B}^t G \tilde{B} \in \mathbb{R}^{m \times m}. \quad (56)$$

The online part consists of solving

$$\alpha'(t) = \sqrt{\varphi(t)^t M_1 \varphi(t) + 2\varphi(t)^t M_2 u(t) + u(t)^t M_3 u(t)} \quad \alpha(0) = E_{x_0} \quad (57)$$

$$u'(t) = \beta(t) \quad u(0) = 0 \quad (58)$$

$$v'(t) = \alpha_2(t)\beta(t)e^{-u(t)} \quad v(0) = 0. \quad (59)$$

Notice that all the auxiliary matrices (53)-(56) can be computed without having to create a full $d \times d$ matrix (assuming the norm-inducing matrix G is sparse) at any time, avoiding problems due to memory restrictions. Furthermore, if there are no inputs $u(t)$ for a given system only the matrix M_1 (54) and the first summand in (57) have to be computed since (55)-(56) vanish. One can see that the offline matrices M_1, M_2, M_3 are all small matrices independent of the original systems dimension and hence the online error part ODE system (57-59) allows for fast evaluation, can be attached to the reduced simulation ODE and can be computed simultaneously.

3.5.2 Computation of the secant gradient problem (31)

So far we have not given information about how to solve the maximization problem (31) efficiently. Before we get into details we create a link to the context of our kernel expansion setting. So let the conditions of Theorem 2 hold and recall the system function (4) with centers $x_i \in \mathbb{R}^d$ and the distances (42). Now for each i we must compute (36), which formulates our context as

$$p_{C(t)}(d_i(t)) = \begin{cases} d_i(t) - C(t) & r_i < d_i(t) - C(t) \\ r_i & r_i \in B_{C(t)}(d_i(t)), \quad i = 1 \dots N \\ d_i(t) + C(t) & r_i > d_i(t) + C(t). \end{cases} \quad (60)$$

In order to obtain the r_i we need to solve N small optimization problems

$$r_i = \varphi_\phi(d_i(t)) = \arg \max_{r \in \mathbb{R}_0^+} |S_{d_i(t)}(\phi)(r)|. \quad (61)$$

As already seen in Lemma 3 the problem (61) has a unique solution for any $t \in [0, T]$ and $i \in \{1 \dots N\}$. To provide an efficient way of solving it we will use the conditions (33)-(34) regarding the position of those r_i together with the following lemma which further characterizes (32):

Lemma 6 (Properties of φ_ϕ) *For any $\phi \in \mathcal{B}$ the mapping φ_ϕ from Lemma 3 is strictly monotonously decreasing. Further we have*

$$0 \leq \varphi_\phi(y) < x_R \quad \forall y \in \mathbb{R}^+$$

for

$$x_R := \frac{\phi(0)x_m}{\phi(0) - \phi(x_m)}$$

With these results the problem (61) can be efficiently solved using a penalized Newton algorithm and we can summarize our results in the following theorem:

Theorem 5 (Local secant Lipschitz constant estimation computation) *Let the conditions of Theorem 2 hold. For some penalty factor $c > 0$ and d_i as defined in (42) we can compute (60) via the following procedure:*

1. If $d_i(t) - C(t) > x_m$, set

$$p_{C(t)}(d_i(t)) = d_i(t) - C(t) \quad (62)$$

2. If $d_i(t) + C(t) < x_m$, set

$$p_{C(t)}(d_i(t)) = d_i(t) + C(t) \quad (63)$$

3. Else: Compute r_i via Newton iteration as root of $n_p(r)$:

- $d_i(t) < x_m$

$$n_p(r) = \begin{cases} n(x_m) - c(r - x_m)^2, & r \leq x_m \\ n(r), & x_m < r < x_R \\ n(x_R) + c(r - x_R)^2, & r \geq x_R \end{cases} \quad (64)$$

- $x_m < d_i(t)$

$$n_p(r) = \begin{cases} n(0) + cr^2, & r \leq 0 \\ n(r), & 0 < r < x_m \\ n(x_m) - c(r - x_m)^2, & r \geq x_m \end{cases} \quad (65)$$

Proof Recall that each function $\phi \in \mathcal{B}$ has a point $x_m > 0$ so that the conditions (33)-(34) apply. Without loss of generality we consider the case $d_i(t) > x_m$. This means if $d_i(t) - C(t) > x_m$ we automatically have $d_i(t) - C(t) > r_i$ since $d_i(t) > x_m \geq r_i$ by (34), meaning that we do not have to compute r_i as $p_{C(t)}(d_i(t))$ takes the value $d_i(t) - C(t)$ by (60) anyway. This yields both the cases (62) and (63). Otherwise we have the necessary condition $\phi'(r_i) = S_{d_i(t)}(\phi)(r_i)$ for a maximum. So in order to find candidates for r_i we need to find the roots of $n(r) := \phi'(r) - S_{d_i(t)}(\phi)(r)$. As $n(r)$ obviously also has a root at $r = d_i(t)$, a modification of our Newton function n is necessary in order to find the correct root:

1. $d_i(t) < x_m$

Condition (33) and Lemma 6 yields $x_m \leq r_i \leq x_R$. By elementary calculus we can show that there is exactly one root of $n(r)$ for $r > x_m$. We also know by $d_i(t) \neq x_m$ that $n(x_m) < 0$, hence $n(r) < 0$ for $x_m \leq r < r_i$ and $n(r) > 0$ for $r > r_i \geq x_R$. This gives us the required directions for the Newton function extensions and finally (64).

2. $x_m < d_i(t)$

By condition (34) and Lemma 6 we see that $0 < r_i \leq x_m$. Using $n(x_m) < 0$ and a similar argumentation as in the first case gives the target function (65).

Especially for a smaller $C(t)$ conditions (62) and (63) can reduce the number of optimization problems to be solved. As the state variable $Vz(t)$ changes continuously and (32) is continuous the number of Newton iterations for subsequent (next-timestep) computations of r_i are kept small when using the old values as starting point for the new iterations. The case $d_i(t) = x_m$ can either be checked for directly or one can just relax one of the conditions in (64) or (65) appropriately.

4 Numerical experiments

In this part we present numerical experiments for synthetic dynamical systems using kernel expansions as nonlinearity. Before we state the system settings, we introduce an heuristic variant of our local secant Lipschitz estimator motivated by the convergence result from Theorem 3. This is due to the fact that the limit function $\Delta_\infty^{V,W}$ reproduces itself when used as upper bound within the iterations. Thus numerically, when solving the integration of the error estimator in discrete time steps, using the error estimation $\Delta^{V,W}(t_{n-1})$ from a time step t_{n-1} as error bound $C(t_{n-1})$ in the computation of the next error estimation $\Delta^{V,W}(t_n)$ reproduces this behaviour up to the integration error. The experiments below show that this time-discrete method indeed seems to be a lower bound to the estimator variants from Theorem 2 when using the iterative scheme from Section 3.4.

To increase readability we shall refer to the global Lipschitz constant estimator from Theorem 1 by ‘‘GLE’’, and to the local Lipschitz constant estimators introduced in Theorem 2 using the results from Lemma 2 and 4 by ‘‘LGL’’ and ‘‘LSL’’, respectively. Additionally, we will perform 1 – 8 iterations of the estimators and the time-discrete method introduced at the beginning of this section shall be denoted by an appended ‘‘TD’’.

Our test setting is given as follows: The test system’s full dimension is $d = 120000$, the simulation end time is $T = 3$ and the system is solved using an explicit Euler scheme with time-step $\Delta t = 0.025$. The kernel used in the expansion is a Gaussian with $\beta = 288728$, which is chosen in a way that the kernel value is lower than $1 \cdot 10^{-5}$ for a distance greater or equal to the second next center. This way each kernel support effectively reaches only over the next two neighbouring expansion centers, ensuring locality of the kernel expansion. Further we choose $G = I_d$ and the Newton penalty factor in Theorem 5 as $c = 2$. For $v = (1, \dots, 1)^t \in \mathbb{R}^d$ and $\mathbf{U} := \text{span}\{v\}$ we set $N = 20$ and the expansion centers to be equally distributed via

$$x_i := v \frac{50}{N-1} (i-1) \in \mathbf{U}, \quad i = 1 \dots N.$$

The expansion coefficient vectors are given via

$$\alpha_i = \sin\left(\frac{x_i + \pi}{4\pi}\right) \in \mathbb{R}^d, \quad i = 1 \dots N,$$

which means $f(x) \in \mathbf{U}$. Further we set $x_0 = \frac{1}{2}v \in \mathbf{U}$ and choose $u(t) = 25 \sin(\pi t)$ which is mapped to state space via a random matrix $B \in [0, 1]^d \subset \mathbb{R}^{d \times 1}$. The subspace projection matrices $V = W$ are computed via a POD of the full trajectory state vectors and a subspace of dimension $r = 1$ is extracted. So we introduce an error by choosing an input $Bu(t)$ which does not lie in \mathbf{U} . Tests have also been performed introducing an initial condition error instead of input error and the simulations showed the same behaviour. The results for different error estimator configurations can be seen in Figures 1 and 2. The first observation is that the LSL estimators have a significantly lower exponential increase rate which is similar to the true error’s rate. One can see that all iterations and even the TD variants of the estimators fall back to their specific increase rate after certain times. Those times are determined by the moment all bounds are too big to have an positive effect regarding the choices (28)

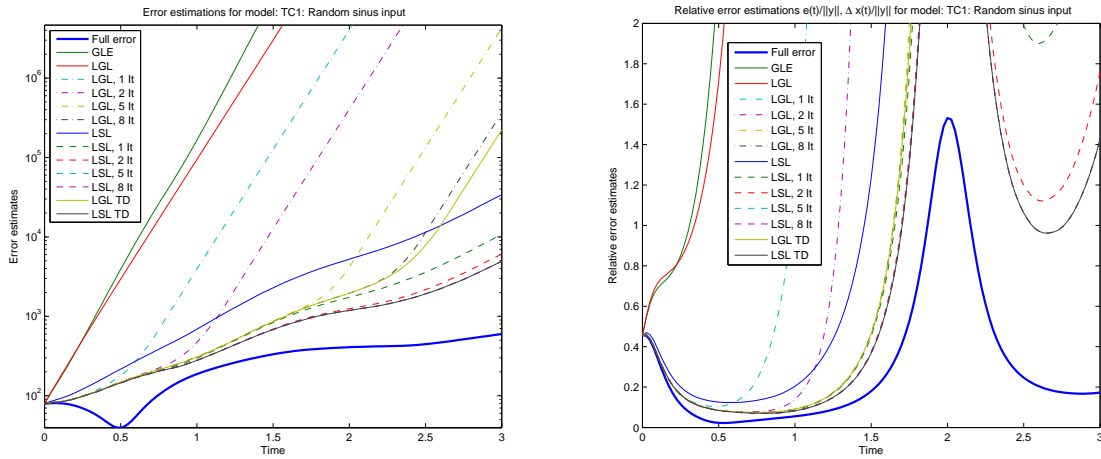


Fig. 1 Absolute (left) and relative (right) error estimates

or (36). Moreover, even the LSL estimator without any iterations performs better than the LGL TD version for $t > 2.6$ which clearly shows the superiority of the LSL estimation scheme. For the LGL estimator every iteration yields a visible improvement towards the TD version, and the LSL iterations give indistinguishable results compared to the TD variant already for iterations greater than three. The fact that both TD variants seem to be a lower bound for their corresponding iterative schemes agrees with our convergence results and strongly supports our heuristic approach. In Figure 2 we display the computation times versus the error estimations and plot the state variable norms of the full and reduced simulations along with the error bounds computed by the LSL TD estimator. We can see that

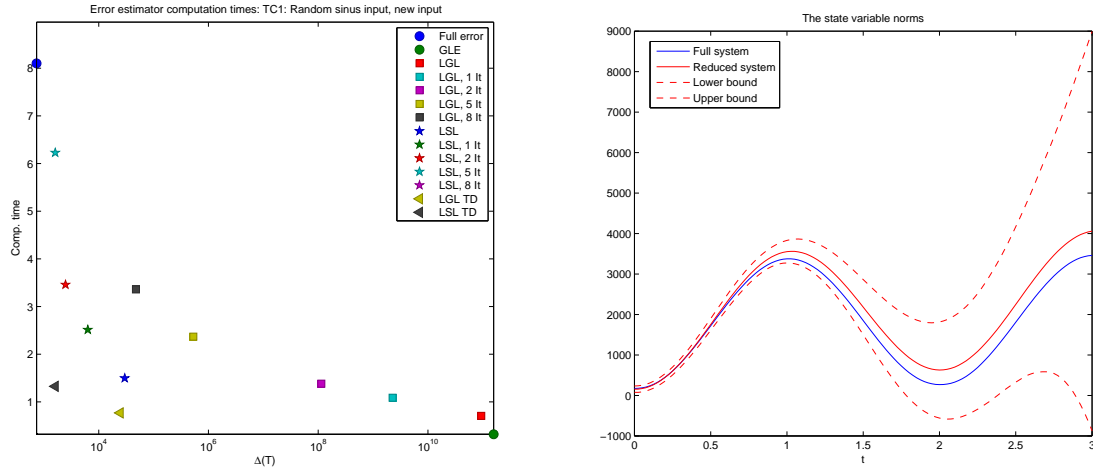


Fig. 2 Computation time/estimate relations (left) and full and reduced state variable norms with error bounds (right)

applying the LSL estimator without estimations requires roughly the same time as an LGL with 2 iterations but estimates about four orders of magnitude better. Of course, the more an estimator is placed in the lower-left corner the better the estimation/cost ratio. From this we can clearly see the effectiveness of the TD variants whose precision is only reached by greater numbers of iterations of their respective schemes. Even though the cost/estimation tightness increase rate for the LSL estimator is somewhat steeper than the LGL rate we are rewarded by an estimation sharpness starting orders of

magnitude lower than the LGL ones for $t = T$. Regarding the state variable norm plot we observe that $\|x(t)\|$ lies nicely in the error estimation tube $\|x^r(t)\| \pm \Delta^{V,W}(t)$, where $\Delta^{V,W}(t)$ is the estimation from the LSL TD estimator.

Now running the reduced model obtained in this setting again but using a different input $u(t) = 20 \sin(t)$ gives some information about how well the reduced system can cope with a new configuration since the subspace was not trained with the new configuration. The results are displayed in Figure 3 and we omitted the computation times as they are similar to the previous results. Here we see that

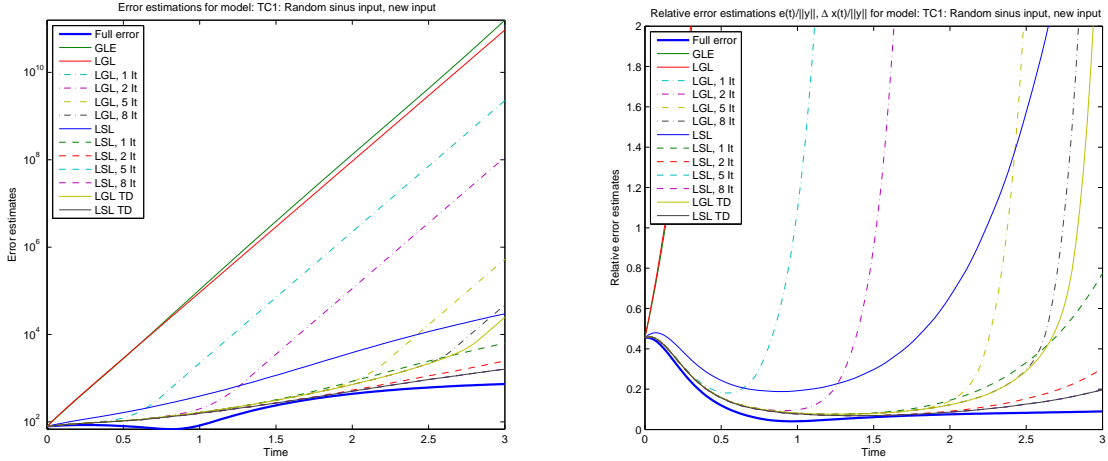


Fig. 3 Results for the test case 1-model with new input. Error (left), relative error (center) estimates and state variable norms (right)

our reduction method and the error estimators perform well in the new setting. Most importantly, all LSL estimators with at least one iteration and the TD variant produce a tight error bound at $t = T$ that only overestimate the full error by a factor of two for the case of the LSL TD estimator. Also the relative error is nicely bounded around 0.2 for those estimators. Table 1 contains an overview regarding the different estimators, computation times and overestimation factors for both the original and new input settings. There we see that 1-2 iterations always pay back in terms of estimation sharpness

Name	$\Delta(3)$	Time	Overestimation	Name	$\Delta(3)$	Time	Overestimation
Full error	$5.984 \cdot 10^2$	7.78s	$1.000 \cdot 10^0$	Full error	$7.348 \cdot 10^2$	8.10s	$1.000 \cdot 10^0$
GLE	$5.474 \cdot 10^{11}$	0.32s	$9.147 \cdot 10^8$	GLE	$1.574 \cdot 10^{11}$	0.32s	$2.142 \cdot 10^8$
LGL	$9.689 \cdot 10^{10}$	0.73s	$1.619 \cdot 10^8$	LGL	$9.369 \cdot 10^{10}$	0.71s	$1.275 \cdot 10^8$
LGL, 1 It	$4.072 \cdot 10^9$	1.08s	$6.804 \cdot 10^6$	LGL, 1 It	$2.272 \cdot 10^9$	1.09s	$3.092 \cdot 10^6$
LGL, 2 It	$4.105 \cdot 10^8$	1.39s	$6.860 \cdot 10^5$	LGL, 2 It	$1.140 \cdot 10^8$	1.38s	$1.551 \cdot 10^5$
LGL, 5 It	$4.242 \cdot 10^6$	2.38s	$7.088 \cdot 10^3$	LGL, 5 It	$5.300 \cdot 10^5$	2.37s	$7.212 \cdot 10^2$
LGL, 8 It	$3.649 \cdot 10^5$	3.39s	$6.098 \cdot 10^2$	LGL, 8 It	$4.768 \cdot 10^4$	3.36s	$6.488 \cdot 10^1$
LSL	$3.405 \cdot 10^4$	1.48s	$5.690 \cdot 10^1$	LSL	$2.986 \cdot 10^4$	1.50s	$4.063 \cdot 10^1$
LSL, 1 It	$1.074 \cdot 10^4$	2.54s	$1.795 \cdot 10^1$	LSL, 1 It	$6.335 \cdot 10^3$	2.51s	$8.622 \cdot 10^0$
LSL, 2 It	$6.092 \cdot 10^3$	3.51s	$1.018 \cdot 10^1$	LSL, 2 It	$2.481 \cdot 10^3$	3.46s	$3.376 \cdot 10^0$
LSL, 5 It	$4.937 \cdot 10^3$	6.51s	$8.249 \cdot 10^0$	LSL, 5 It	$1.615 \cdot 10^3$	6.23s	$2.197 \cdot 10^0$
LSL, 8 It	$4.933 \cdot 10^3$	9.47s	$8.242 \cdot 10^0$	LSL, 8 It	$1.613 \cdot 10^3$	8.97s	$2.195 \cdot 10^0$
LGL TD	$2.238 \cdot 10^5$	0.76s	$3.739 \cdot 10^2$	LGL TD	$2.469 \cdot 10^4$	0.77s	$3.361 \cdot 10^1$
LSL TD	$4.933 \cdot 10^3$	1.42s	$8.242 \cdot 10^0$	LSL TD	$1.613 \cdot 10^3$	1.33s	$2.195 \cdot 10^0$

Table 1 Error estimator overview for original (left) and new input simulation (right)

gain for both the LGL and LSL estimators. These results emphasize that by iterations one can trade-off between computation costs and estimation sharpness. To give an extreme case, performing eight iterations of the LSL estimator takes more time than the full simulation and so increasing the number of iterations beyond some limit is not always beneficial. For both scenarios the LSL TD estimator gives the best results. Also we can see the convergence behaviour of the iteration scheme against the TD

variants as e.g. the LSL 5It already matches 3 figures of the LSL TD version at $t = T$. Finally, we applied a mean value output mapping $C = \frac{1}{d}v^t$ and the outputs along with error estimation bounds for the LSL TD estimator are shown in Figure 4. We can observe from Figure 4 that the error estimates

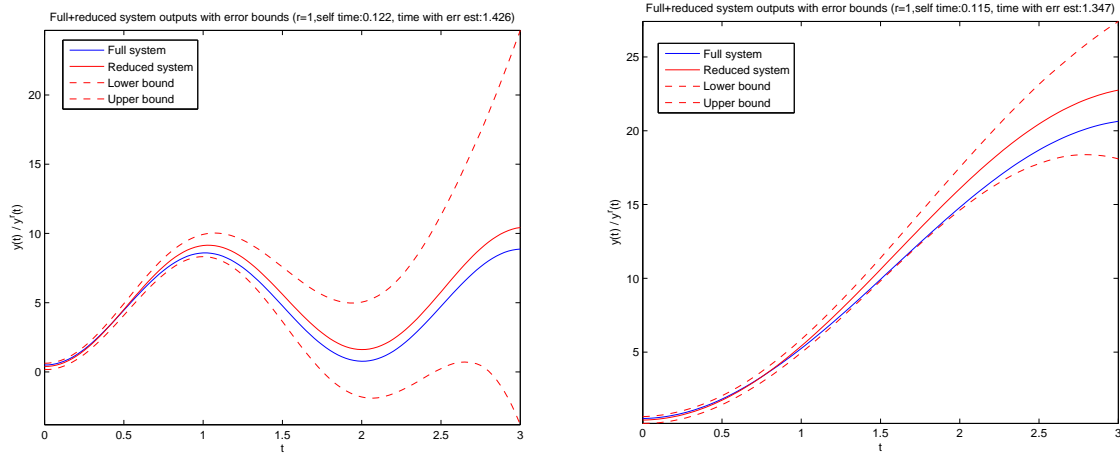


Fig. 4 Output plots with error bounds for original (left) and new input simulation (right)

bound especially the output of the new input simulation very sharply up to $t \leq 2$.

In order to check for integration error issues, all tests have also been run with $\Delta t = 0.0125$ and yielded indistinguishable results. This showed also that all computation times scale nicely with different Δt so the error estimators show no time-step dependent effects. Absolutely seen, the speedup factor of 5-6 regarding computation times for the full system compared to the best error estimator might not look too good but the important fact is here that the computation times of the error estimators are completely *independent* of the full system's dimension. Furthermore, no optimization regarding speed has been performed yet. Possible improvement will be a lookup tree for different maximum secant gradient positions that can be computed offline by solving a series of maximization problems (61).

5 Conclusion and Outlook

In this article we presented reduction techniques for kernel-based nonlinear dynamical systems along with a series of improving efficient a-posteriori error estimators. Special classes of kernels allow for the creation of a small reduced system (16)-(18) by lossless low-dimensional argument evaluations (11) or (14), ultimately enabling a rapid simulation of the reduced system. The error estimators which can be obtained simultaneously during the reduced simulation by solving a small auxiliary ODE system can be completely decomposed in an offline/online fashion as introduced in Section 3.5.1 which provides efficient computation.

Numerical experiments showed the superiority of the local Lipschitz estimator using maximum secant gradients obtained by a small Newton problem. Furthermore, the iterative scheme from Section 3.4 allows to balance the computation costs against estimation sharpness. More future work will target at the improvement of the error estimators regarding the coefficient vector norms $\|\alpha_i\|$ in (41) as locality can also be used there to sharpen the error estimations. The structure of the error estimators is also applicable to systems based on a piecewise-linear or -polynomial approximation using bell function induced kernels as weights and will be investigated more closely. Ultimately, incorporation of approximation techniques to obtain kernel expansions (4) from arbitrary nonlinear dynamical systems and extension to parameterized systems will be in the focus of our research.

References

1. K. Zhou, J. C. Doyle, K. Glover, *Robust and Optimal Control*, Prentice Hall PTR, Upper Saddle River, New Jersey 07458, 1996.
URL <http://www.gbv.de/dms/goettingen/18606599X.pdf>
2. B. Haasdonk, M. Ohlberger, Efficient reduced models and a-posteriori error estimation for parametrized dynamical systems by offline/online decomposition, *MCMDS, Mathematical and Computer Modelling of Dynamical Systems* Accepted.
3. P. Benner, Numerical linear algebra for model reduction in control and simulation, *GAMM-Mitt.* 29 (2) (2006) 275–296.
4. L. Knockaert, T. Dhaene, F. Ferranti, D. D. Zutter, Model order reduction with preservation of passivity, non-expansivity and markov moments, *Systems and Control Letters* In Press, Corrected Proof (2010) –. doi:DOI: 10.1016/j.sysconle.2010.10.006.
URL <http://www.sciencedirect.com/science/article/B6V4X-51F1SCJ-1/2/a9b3fd3deaa5eefa42bfcee874b98>
5. H. Sandberg, A. Rantzer, Balanced truncation of linear time-varying systems, *Automatic Control, IEEE Transactions on* 49 (2) (2004) 217 – 229. doi:10.1109/TAC.2003.822862.
6. N. Hara, H. Kokame, K. Konishi, Singular value decomposition for a class of linear time-varying systems with application to switched linear systems, *Systems and Control Letters* 59 (12) (2010) 792 – 798. doi:DOI: 10.1016/j.sysconle.2010.09.006.
URL <http://www.sciencedirect.com/science/article/B6V4X-518VXJ2-2/2/d1827e8d6b678a8555ad96146baf8>
7. T. Stykel, Balanced truncation model reduction of second-order systems, in: I. Troch, F. Breitenecker (Eds.), *Proceedings of the 5th Vienna Symposium on Mathematical Modelling*, Vol. 2, ARGESIM-Verlag, 2006.
8. B. Moore, Principal component analysis in linear systems: Controllability, observability, and model reduction, *IEEE Trans. Automat. Control* AC-26 (1) (1981) 17–32.
9. K. Willcox, J. Peraire, Balanced model reduction via the proper orthogonal decomposition, in: *15th AIAA Computational Fluid Dynamics Conference*, 2001.
10. B. Haasdonk, M. Ohlberger, Reduced basis method for finite volume approximations of parametrized linear evolution equations, *ESAIM: M2AN* 42 (2) (2008) 277–302. doi:10.1051/m2an:2008001.
URL <http://dx.doi.org/10.1051/m2an:2008001>
11. E. J. Grimme, Krylov projection methods for model reduction, Ph.D. thesis, University of Illinois (1997). doi:10.1.1.19.9254.
12. A. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM Publications, Philadelphia, PA, 2005.
13. M. Rewienski, A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems, Ph.D. thesis, MIT (2003).
14. B. Bond, L. Daniel, Parameterized model order reduction of nonlinear dynamical systems, in: *Computer-Aided Design, 2005. ICCAD-2005. IEEE/ACM International Conference on*, 2005, pp. 487 – 494. doi:10.1109/ICCAD.2005.1560117.
15. N. Dong, J. Roychowdhury, Piecewise polynomial nonlinear model reduction, in: *Design Automation Conference, 2003. Proceedings, 2003*, pp. 484 – 489. doi:10.1109/DAC.2003.1219054.
16. S. Lall, J. Marsden, S. Glavaski, A subspace approach to balanced truncation for model reduction of nonlinear control systems, *Int. J. on robust and nonlinear control* 12 (5) (2002) 519–535. doi:10.1002/rnc.657.
17. P. Tabuada, A. D. Ames, A. Julius, G. J. Pappas, Approximate reduction of dynamic systems, *Systems and Control Letters* 57 (7) (2008) 538 – 545. doi:DOI: 10.1016/j.sysconle.2007.12.005.
URL <http://www.sciencedirect.com/science/article/B6V4X-4RTCPKT-1/2/4a9075185a21c7fbc311bd717e530>
18. J. Phillips, Projection frameworks for model reduction of weakly nonlinear systems., in: *DAC, 2000*, pp. 184–189.
URL <http://dblp.uni-trier.de/db/conf/dac/dac2000.html>
19. P. Benner, T. Breiten, Krylov-subspace based model reduction of nonlinear circuit models using bilinear and quadratic-linear approximations, *max-Planck-Institute for Dynamics of Complex Technical Systems*, Magdeburg, Germany. Submitted (2010).
20. M. Barrault, Y. Maday, N. Nguyen, A. Patera, An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations, *C. R. Math. Acad. Sci. Paris*

- Series I 339 (2004) 667–672.
21. B. Haasdonk, M. Ohlberger, G. Rozza, A reduced basis method for evolution schemes with parameter-dependent explicit operators, *ETNA, Electronic Transactions on Numerical Analysis* 32 (2008) 145–161.
 22. S. Chaturantabut, D. Sorensen, Discrete empirical interpolation for nonlinear model reduction, in: *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference, 2009*, pp. 4316 –4321. doi:10.1109/CDC.2009.5400045.
 23. G. Berkooz, P. Holmes, J. L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, *Annual Review of Fluid Mechanics* 25 (1993) 539–575.
 24. K. Kunisch, V. Volkwein, Proper orthogonal decomposition for optimality systems, *M2AN Math. Model. Numer. Anal.* 42 (2008) 1–23.
 25. M. Meyer, H. Matthies, Efficient model reduction in non-linear dynamics using the Karhunen-Loève expansion and dual-weighted-residual methods, *Computational Mechanics* 31 (2003) 179–191.
 26. J. Phillips, J. Afonso, A. Oliveira, L. Silveira, Analog macromodeling using kernel methods, in: *Proc. of ICCAD-2003, International Conference on Computer Aided Design, 2003*, pp. 446–453. doi:http://dx.doi.org/10.1109/ICCAD.2003.1257815.
 27. B. Schölkopf, J. A. Smola, *Learning with Kernels*, Adaptive Computation and Machine Learning, The MIT Press, 2002.
 28. H. Wendland, *Scattered data approximation*, Vol. 17 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2005.
 29. L. Ma, Z. Wu, Kernel based approximation in sobolev spaces with radial basis functions, *Applied Mathematics and Computation* 215 (6) (2009) 2229 – 2237. doi:DOI: 10.1016/j.amc.2009.08.012. URL <http://www.sciencedirect.com/science/article/B6TY8-4XOPBWH-2/2/ab5128fe50ba3e7e9012a12efadb8>.
 30. H. Wendland, Computational aspects of radial basis function approximation, in: W. H. R. S. K. Jetter, M. D. Buhmann, J. Stöckler (Eds.), *Topics in Multivariate Approximation and Interpolation*, Vol. 12 of Studies in Computational Mathematics, Elsevier, 2006, pp. 231 – 256. doi:DOI: 10.1016/S1570-579X(06)80010-8. URL <http://www.sciencedirect.com/science/article/B8GXW-4NWOSWN-B/2/27818c185f2964fb4a4141a0984b1>.
 31. N. Aronszajn, Theory of reproducing kernels, *Transactions of the American Mathematical Society* 68 (3) (1950) 337–404. URL <http://dx.doi.org/10.2307/1990404>
 32. C. Berg, J. Christensen, P. Ressel, *Harmonic Analysis on Semigroups*, Springer, 1984.
 33. M. Hinze, S. Volkwein, Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control in dimension reduction of large-scale systems, in: *Dimension Reduction of Large-Scale Systems*, Lecture Notes in Computational and Applied Mathematics, Vol. 45, Springer Berlin Heidelberg, 2005, pp. 261–306. doi:10.1007/3-540-27909-1.
 34. N. Jung, B. Haasdonk, D. Kroner, Reduced basis method for quadratically nonlinear transport equations, *International Journal of Computing Science and Mathematics* 2 (4) (2009) 334 – 353. URL <http://inderscience.metapress.com/app/home/contribution.asp?referrer=parent&backto=issue,3,4>
 35. S. Volkwein, Model reduction using proper orthogonal decomposition, lecture notes, University of Graz (2008). URL <http://www.uni-graz.at/imawww/volkwein/publist.html>
 36. M. Drohmann, B. Haasdonk, M. Ohlberger, Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation, Preprint *Angewandte Mathematik und Informatik 02/10-N*, University of Münster, submitted to SISC (2010).
 37. T. H. Gronwall, Note on the derivatives with respect to a parameter of the solutions of a system of differential equations, *The Annals of Mathematics* 20 (4) (1919) 292–296. URL <http://www.jstor.org/stable/1967124>