








# Reachability of Koopman Linearized Systems Using Random Fourier Feature Observables and Polynomial Zonotope Refinement

Stanley Bak<sup>1</sup> , Sergiy Bogomolov<sup>2</sup> , Brandon Hency<sup>3</sup> ,  
Niklas Kochdumper<sup>1</sup> , Ethan Lew<sup>4</sup>,  
and Kostiantyn Potomkin<sup>2</sup> 



<sup>1</sup> Department of Computer Science, Stony Brook University, Stony Brook, NY, USA

{stanley.bak,niklas.kochdumper}@stonybrook.edu

<sup>2</sup> School of Computing, Newcastle University, Newcastle Upon Tyne, UK

{sergiy.bogomolov,k.potomkin2}@newcastle.ac.uk

<sup>3</sup> Air Force Research Laboratory, Wright-Patterson Air Force Base,  
Dayton, OH, USA

brandon.hency@us.af.mil

<sup>4</sup> Galois Inc., Portland, OR, USA

elew@galois.com

**Abstract.** Koopman operator linearization approximates nonlinear systems of differential equations with higher-dimensional linear systems. For formal verification using reachability analysis, this is an attractive conversion, as highly scalable methods exist to compute reachable sets for linear systems. However, two main challenges are present with this approach, both of which are addressed in this work. First, the approximation must be sufficiently accurate for the result to be meaningful, which is controlled by the choice of *observable functions* during Koopman operator linearization. By using random Fourier features as observable functions, the process becomes more systematic than earlier work, while providing a higher-accuracy approximation. Second, although the higher-dimensional system is linear, simple convex initial sets in the original space can become complex non-convex initial sets in the linear system. We overcome this using a combination of Taylor model arithmetic and polynomial zonotope refinement. Compared with prior work, the result is more efficient, more systematic and more accurate.

**Keywords:** Koopman operator · Reachability analysis · Polynomial zonotopes · Random Fourier features · Formal verification

## 1 Introduction

Despite recent advances, systems described by nonlinear ordinary differential equations are still hard to analyze, control, and verify. On the other hand, a

© The Author(s) 2022

S. Shoham and Y. Vizel (Eds.): CAV 2022, LNCS 13371, pp. 490–510, 2022.

[https://doi.org/10.1007/978-3-031-13185-1\\_24](https://doi.org/10.1007/978-3-031-13185-1_24)

powerful body of methods and theories exists for linear systems making analysis, control, and verification much easier, even for high-dimensional systems. The efficiency of techniques related to reachability analysis for linear systems [4, 6, 15] motivates the use of Koopman operator linearization, where a higher-dimensional linear system approximates the dynamic behavior of a nonlinear system. Koopman operator techniques are also well-suited for data-driven approaches since the Koopman linearized system can be directly created from measurements, bypassing a potentially complex modeling step. The Koopman framework has been successfully applied to many applications, including control [26, 28], state estimation [31] and recently, formal verification [5].

The main contribution of this paper is to advance the state-of-the-art in formal verification using reachability analysis on Koopman operator linearized systems. First, we improve the accuracy of the finite Koopman linearization by employing random Fourier features [29]. In contrast with an *ad hoc*, finite-dimensional feature space, random Fourier features leverage the powerful *kernel trick* from machine learning [36, 38] to generate a computationally tractable mapping over an infinite-dimensional feature space. Second, we improve speed. Instead of using an SMT solver to reason over non-convex initial sets, we propose combining Taylor models with polynomial zonotope refinement. A comparison on the same nonlinear system benchmarks used in the earlier Koopman verification work [5] demonstrates both the improved accuracy and the improved verification speed.

## 1.1 Related Work

The concept of Koopman operator linearization was originally introduced in 1931 [22]. Instead of investigating the dynamic evolution of the original system state, the Koopman approach considers the evolution of so-called *observable functions* or *observables* defined by nonlinear transformations of the original system state. Since the set of all possible observables defines a vector space, it then holds that the dynamic behavior of every nonlinear system can be equivalently represented by an infinite dimensional linear system. Because it is obviously infeasible to handle infinite dimensions, a finite set of observables is used in practice. Given such a set, the system matrix resulting in the most accurate linear approximation of the original system behavior can be determined using extended dynamic mode decomposition [41].

Many different methods for determining good observables have been proposed: Carleman linearization [7] equivalently represents the dynamic behavior of polynomial systems with an infinite dimensional linear system. The corresponding observables are multi-variate monomials, which are determined by repeatedly computing the time-derivative of the current observables. Terminating this iteration after a certain number of steps yields a finite set of observables. Carleman linearization can be extended to general nonlinear systems by using a Taylor series expansion. A finite set of observables defines an exact linear representation of the original system if the vector space spanned by the observables is closed under the operation of Lie-derivatives [34]. Consequently, a natural approach is

to refine an initial set of observables by removing observables that violate the condition [34]. This concept can be extended to obtain polynomial instead of linear representations for the original nonlinear system [35]. Another class of approaches uses neural networks as observables [16, 43], where the weights of the network are trained on traces of the real system. Since these approaches usually train the system matrix together with neural networks, they circumvent the subsequent application of dynamic mode decomposition. If one aims to reason about the original system based on the Koopman linearization, some quantification of the approximation error is required. Several approaches derive error bounds for truncated Carleman linearization [3, 12, 24] considering quadratic systems [24], polynomial systems [12], as well as general nonlinear systems [3].

The main motivation for using the Koopman framework for reachability analysis is that reachable sets for linear systems can be computed efficiently [11, 15, 23] even for high-dimensional systems [2, 4, 6], while reachability analysis for nonlinear systems [1, 8, 27] is often computationally demanding and potentially results in large over-approximations. Another advantage is that the Koopman approach can also be applied to data-driven systems where no model is available. Due to the nonlinear transformation of the initial state defined by the observables, reachability analysis for Koopman operator linearized system represents a special type of reachability problem. To the best of our knowledge only two approaches exist for far: The first approach [13] utilizes the error bounds for quadratic systems [24] to compute an enclosure of the reachable set for weakly nonlinear systems based on a finite Carleman linearization, where interval arithmetic [17] is applied to enclose the image of the initial set through the observables. The second approach [5], which represents the work closest to our method, presents two different verification strategies: 1) Direct encoding of the nonlinear transformation defined by the observables using a SMT solver, and 2) zonotope domain splitting, where the initial set is recursively split into smaller sets until the specification can be verified or falsified.

## 1.2 Overview

In this work we address the two main bottlenecks of formal verification for Koopman operator linearized systems, which are the selection of observables and the computation of the image of the initial set through the nonlinear transformation defined by the observables. In particular, while currently observables often have to be selected manually by the user, we generate observables in a systematic fashion using random Fourier features. As we demonstrate with numerical experiments, these observables yield high-accuracy approximations of the real system behavior. Moreover, while previous approaches either compute very conservative convex enclosures of the image through the observables [13] or have to split the initial set in order to achieve a desired precision [5], we calculate tight non-convex enclosures of the image by combining Taylor model arithmetic with polynomial zonotopes. To conduct collision checks between the resulting non-convex reachable set enclosures and unsafe regions we then use a novel polynomial zonotope

refinement strategy, which is significantly faster than the previous SMT solver and zonotope domain splitting approaches [5].

The remainder of the paper is structured as follows: We first recapitulate some preliminary results that are required throughout the paper in Sect. 2. In the main part we then describe the systematic generation of observables using random Fourier features in Sect. 3, before we present our proposed verification algorithm in Sect. 4. Finally, we demonstrate the superior performance of random Fourier feature observables and our verification algorithm in comparison with existing techniques on various benchmark systems in Sect. 5.

### 1.3 Notation

In the remainder of this paper, we will use the following notations: Sets are denoted by calligraphic letters, matrices by uppercase letters, vectors by lowercase letters, and lists by bold uppercase letters. Given a vector  $b \in \mathbb{R}^n$ ,  $b_{(i)}$  refers to the  $i$ -th entry. Given a matrix  $A \in \mathbb{R}^{n \times m}$ ,  $A_{(i,\cdot)}$  represents the  $i$ -th matrix row,  $A_{(\cdot,j)}$  the  $j$ -th column, and  $A_{(i,j)}$  the  $j$ -th entry of matrix row  $i$ . Given a discrete set of positive integer indices  $\mathcal{H} = \{h_1, \dots, h_w\}$  with  $1 \leq h_i \leq m \forall i \in \{1, \dots, w\}$ ,  $A_{(\cdot,\mathcal{H})}$  is used for  $[A_{(\cdot,h_1)} \dots A_{(\cdot,h_w)}]$ , where  $[C \ D]$  denotes the concatenation of two matrices  $C$  and  $D$ . The symbols  $\mathbf{0}$  and  $\mathbf{1}$  represent matrices of zeros and ones of proper dimension, the empty matrix is denoted by  $[\ ]$ , and  $I_n \in \mathbb{R}^{n \times n}$  is the identity matrix. Given an ordered list  $\mathbf{L} = (l_1, \dots, l_n)$ ,  $\mathbf{L}_{(i)}$  refers to the  $i$ -th entry and  $|\mathbf{L}| = n$  denotes the number of elements in the list. Moreover, the concatenation of two lists  $\mathbf{L}_1$  and  $\mathbf{L}_2$  is denoted by  $(\mathbf{L}_1, \mathbf{L}_2)$ . The left multiplication of a matrix  $M \in \mathbb{R}^{m \times n}$  with a set  $\mathcal{S} \subset \mathbb{R}^n$  is defined as  $M\mathcal{S} = \{Ms \mid s \in \mathcal{S}\}$ , and the Cartesian product of two sets is denoted by the  $\times$  operator. We further introduce an  $n$ -dimensional interval as  $\mathcal{I} = [l, u]$ ,  $\forall i \ l_{(i)} \leq u_{(i)}$ ,  $l, u \in \mathbb{R}^n$ .

## 2 Preliminaries

Our approach utilizes several existing techniques and concepts, which we shortly recapitulate here. We use the nonlinear system

$$\begin{aligned} \dot{x}_1 &= x_1 \\ \dot{x}_2 &= x_2 - x_1^4 \end{aligned} \tag{1}$$

in combination with the initial set  $\mathcal{X}_0 = [-2, 2] \times [0, 4]$  as a running example throughout this section.

### 2.1 Koopman Operator Linearization

First, we describe the general concept of Koopman operator linearization [22]. Given a nonlinear system

$$\frac{\partial x}{\partial t} = f(x) \quad \text{with } x \in \mathbb{R}^n, f: \mathbb{R}^n \rightarrow \mathbb{R}^n, \tag{2}$$

our goal is to find observables  $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$  such that the dynamics of the resulting new variables  $g_i(x)$  is linear:

$$\frac{\partial g(x)}{\partial t} = A g(x) \quad \text{with } A \in \mathbb{R}^{m \times m}, \tag{3}$$

where  $g(x) = [g_1(x) \dots g_m(x)]^T$  is the observable function. Since the new variables  $g_i(x)$  are functions of the original system state  $x$ , the linear system (3) defines an equivalent representation of the dynamic behavior of the original system (2). Usually, the number of observables  $m$  is significantly larger than the dimension  $n$  of the original system.

Let us demonstrate Koopman linearization for our exemplary system in (1). By choosing the observables  $g_1(x) = x_1$ ,  $g_2(x) = x_2$ , and  $g_3(x) = x_1^4$  we obtain the linear system

$$\frac{\partial}{\partial t} \begin{bmatrix} g_1(x) \\ g_2(x) \\ g_3(x) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} g_1(x) \\ g_2(x) \\ g_3(x) \end{bmatrix}$$

since  $\partial g_1(x)/\partial t = \dot{x}_1 = x_1$ ,  $\partial g_2(x)/\partial t = \dot{x}_2 = x_2 - x_1^4 = g_2(x) - g_3(x)$ , and  $\partial g_3(x)/\partial t = 4x_1^3 \dot{x}_1 = 4x_1^4 = 4g_3(x)$ .

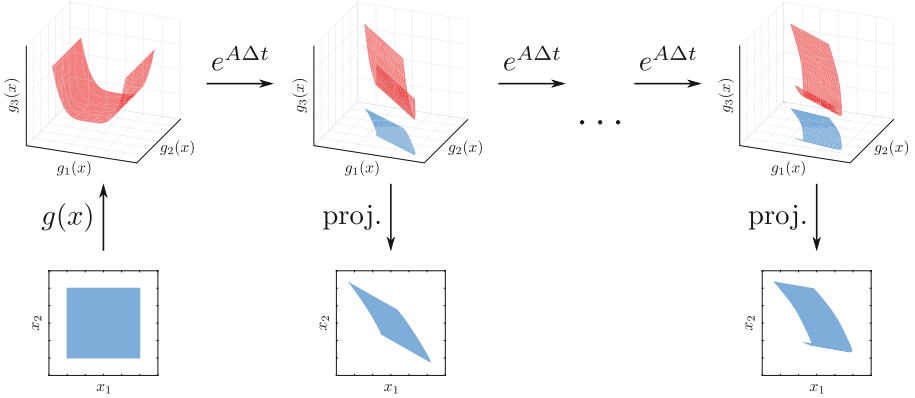
The exact linearization using a finite number of observables demonstrated by the example above is unfortunately only possible for a small number of special systems. In practice one therefore usually aims to instead determine a linear system (3) that approximates the dynamic behavior of the nonlinear system (2) well enough. Given observables  $g_i(x)$ , the system matrix  $A$  resulting in the best approximation can be determined by applying extended dynamic mode decomposition [41] to traces of the original system. Since those traces can also be generated by simulating black-box systems or by measuring the real system behavior, we do not necessarily require a model (2) of the original system. This is one of the biggest advantages of the Koopman framework making it well suited for data-driven approaches. The approach we present in this work verifies Koopman linearized systems using reachability analysis:

**Definition 1.** (*Reachable set*) Given an initial set  $\mathcal{X}_0 \subset \mathbb{R}^n$ , the reachable set for a Koopman linearized system is

$$\mathcal{R}(t) := \{ \xi(t, g(x_0)) \mid x_0 \in \mathcal{X}_0 \},$$

where  $\xi(t, g(x_0))$  is the solution to (3) at time  $t \in \mathbb{R}_{\geq 0}$  for the initial state  $g(x_0)$ .

Consequently, to compute the reachable set for a Koopman linearized system one first needs to propagate the initial set through the nonlinear transformation defined by the observables, followed by the calculation of the reachable set for the linear system in (3) using a reachability algorithm. This procedure is visualized in Fig. 1. Definition 1 defines the reachable set for the observables  $g_i(x)$ . However, since safety specifications are typically defined on the original system state  $x$  rather than on  $g(x)$ , we usually require the reachable set for the original state  $\mathcal{R}_x(t)$  for verification. This issue can easily be resolved by using the original system state  $x$  for the first  $n$  observables  $g_i(x) = x_{(i)}$ ,  $i = 1, \dots, n$ , in which case  $\mathcal{R}_x(t)$  can be obtained via projection:  $\mathcal{R}_x(t) = [I_n \ \mathbf{0}] \mathcal{R}(t)$ .



**Fig. 1.** Schematic visualization of reachability analysis for Koopman linearized systems: We first transform the initial set to the higher-dimensional observable space using  $g(x)$ , then compute the reachable set of the linear system using the matrix exponential  $e^{A\Delta t}$  with time-step size  $\Delta t$ , and finally obtain the reachable set in the original state space via projection.

## 2.2 Taylor Model Arithmetic

Taylor model arithmetic [25] can be utilized to compute tight non-convex enclosures for the image through a nonlinear function. It is based on a set representation called Taylor models:

**Definition 2.** (Taylor model) Given a polynomial function  $p : \mathbb{R}^s \rightarrow \mathbb{R}^n$ , an interval domain  $\mathcal{D} \subset \mathbb{R}^s$ , and an interval remainder  $\mathcal{Y} \subset \mathbb{R}^n$ , a Taylor model  $\mathcal{T}(x)$  is defined as

$$\forall x \in \mathcal{D} : \mathcal{T}(x) := \{p(x) + y \mid y \in \mathcal{Y}\}.$$

The Taylor order  $\kappa \in \mathbb{N}$  defines an upper bound for the polynomial degree of the polynomial  $p(x)$ . The set defined by a Taylor model is

$$\{\mathcal{T}(x) \mid x \in \mathcal{D}\} = \{p(x) + y \mid x \in \mathcal{D}, y \in \mathcal{Y}\}.$$

For a concise notation we use the shorthand  $\mathcal{T}(x) = \langle p(x), \mathcal{Y}, \mathcal{D} \rangle_T$ .

The general concept of Taylor model arithmetic is to define rules on how to perform the arithmetic operations  $+$ ,  $-$ ,  $\cdot$ , and  $/$  as well as elementary functions such as  $\sin(x)$  or  $\sqrt{x}$  on Taylor models [25, Sec. 2]. Since every nonlinear function represents a composition of arithmetic operations and elementary functions, the image through the function can then be computed by successively evaluating those rules. Given two one-dimensional Taylor models  $\mathcal{T}_1(x) = \langle p_1(x), \mathcal{Y}_1, \mathcal{D} \rangle_T$  and  $\mathcal{T}_2(x) = \langle p_2(x), \mathcal{Y}_2, \mathcal{D} \rangle_T$  the rules for addition and multiplication are for example given as

$$\begin{aligned} \mathcal{T}_1(x) + \mathcal{T}_2(x) &:= \langle p_1(x) + p_2(x), \mathcal{Y}_1 + \mathcal{Y}_2, \mathcal{D} \rangle_T \\ \mathcal{T}_1(x) \cdot \mathcal{T}_2(x) &:= \langle p_1(x) \cdot p_2(x), \mathcal{Y}_1 \cdot \mathcal{Y}_2 + \mathcal{I}_1 \cdot \mathcal{Y}_2 + \mathcal{Y}_1 \cdot \mathcal{I}_2, \mathcal{D} \rangle_T, \end{aligned}$$

where  $\mathcal{I}_1 = \{p_1(x) \mid x \in \mathcal{D}\}$  and  $\mathcal{I}_2 = \{p_2(x) \mid x \in \mathcal{D}\}$ . The rules for elementary functions are obtained using a finite Taylor series expansion, where the order of the Taylor series is equal to the Taylor order  $\kappa$ . For  $\sin(x)$  we for example obtain with  $\kappa = 2$  the rule

$$\sin(\mathcal{T}_1(x)) := \langle \sin(c) + \cos(c)(p_1(x) - c) - 0.5 \sin(c)(p_1(x) - c)^2, \mathcal{Y}, \mathcal{D} \rangle_T,$$

where the expansion point  $c$  is chosen as  $c = p_1(c_d)$  with  $c_d$  being the center of the domain  $\mathcal{D}$ , and the interval  $\mathcal{Y}$  computed according to [25, Sec. 2] encloses the remainder of the Taylor series. Due to the finite Taylor series approximation, Taylor model arithmetic yields a tight enclosure rather than the exact image. The accuracy of the enclosure can be improved by choosing a larger Taylor order.

For our verification approach we apply Taylor model arithmetic to compute the image of the initial set through the observable function. The initial set  $\mathcal{X}_0 = [-2, 2] \times [0, 4]$  for the exemplary system in (1) can be represented by the Taylor model  $\mathcal{T}(x) = \langle x, \emptyset, \mathcal{X}_0 \rangle_T$ . Applying Taylor model arithmetic to the observable function  $g(x)$  defined by the observables  $g_1(x) = x_1$ ,  $g_2(x) = x_2$ , and  $g_3(x) = x_1^4$  then yields the Taylor model

$$\{g(x) \mid x \in \mathcal{X}_0\} \subseteq \left\langle \begin{bmatrix} x_1 \\ x_2 \\ x_1^4 \end{bmatrix}, \emptyset, [-2, 2] \times [0, 4] \right\rangle_T, \tag{4}$$

which represents the exact image in this case since the observables contain polynomial functions only.

### 2.3 Set Representations

In this work we use polynomial zonotopes to represent reachable sets, polytopes to represent unsafe sets, and zonotopes for efficient collision checking. Let us first introduce polytopes, for which we consider the halfspace representation:

**Definition 3.** (Polytope) Given a matrix  $H \in \mathbb{R}^{s \times n}$  and vector  $d \in \mathbb{R}^s$ , the halfspace representation of a polytope  $\mathcal{P} \subset \mathbb{R}^n$  is defined as

$$\mathcal{P} := \{x \in \mathbb{R}^n \mid Hx \leq d\}.$$

We use the shorthand  $\mathcal{P} = \langle H, d \rangle_{\mathcal{P}}$ .

A halfspace  $\mathcal{H} \subset \mathbb{R}^n$  is a special case of a polytope consisting of a single inequality constraint  $h^T x \leq d$  with  $h \in \mathbb{R}^n$ ,  $d \in \mathbb{R}$ . We use the shorthand  $\mathcal{H} = \langle h, d \rangle_H$ . Another special type of polytopes are zonotopes, which can be stored efficiently using so-called generators:

**Definition 4.** (Zonotope) Given a center vector  $c \in \mathbb{R}^n$  and a generator matrix  $G \in \mathbb{R}^{n \times p}$ , a zonotope  $\mathcal{Z} \subset \mathbb{R}^n$  is defined as

$$\mathcal{Z} := \left\{ c + \sum_{i=1}^p \alpha_i G_{(\cdot,i)} \mid \alpha_i \in [-1, 1] \right\},$$

where the scalars  $\alpha_i$  are called factors. We use the shorthand  $\mathcal{Z} = \langle c, G \rangle_{\mathcal{Z}}$ .

Polynomial zonotopes are a novel non-convex set representation that has been originally introduced for reachability analysis of nonlinear systems [1]. We use the sparse representation of polynomial zonotopes [20]<sup>1</sup>:

**Definition 5.** (*Polynomial zonotope*) Given a constant offset  $c \in \mathbb{R}^n$ , a generator matrix of dependent generators  $G \in \mathbb{R}^{n \times h}$ , a generator matrix of independent generators  $G_I \in \mathbb{R}^{n \times q}$ , and an exponent matrix  $E \in \mathbb{N}_0^{p \times h}$ , a polynomial zonotope  $\mathcal{PZ} \subset \mathbb{R}^n$  is defined as

$$\mathcal{PZ} := \left\{ c + \sum_{i=1}^h \left( \prod_{k=1}^p \alpha_k^{E_{(k,i)}} \right) G_{(\cdot,i)} + \sum_{j=1}^q \beta_j G_{I(\cdot,j)} \mid \alpha_k, \beta_j \in [-1, 1] \right\}.$$

The scalars  $\alpha_k$  are called *dependent factors* since a change in their value affects multiplication with multiple generators. Consequently, the scalars  $\beta_j$  are called *independent factors* because they only affect multiplication with one generator. We use the shorthand  $\mathcal{PZ} = \langle c, G, G_I, E \rangle_{\mathcal{PZ}}$ .

Using polynomial zonotopes for verification has two main advantages:

1. Due to the similarity with Taylor models the set defined by a Taylor model can be equivalently represented as a polynomial zonotope [20, Prop. 4].
2. Due to the similarity with zonotopes tight enclosing zonotopes can be computed efficiently for polynomial zonotopes [20, Prop. 5].

For verification we therefore convert the Taylor model representing the image of the initial set through the observable function to a polynomial zonotope, for which collision checks with the unsafe sets can be efficiently realized using zonotope enclosures that are iteratively refined by splitting the polynomial zonotope.

The conversion of the Taylor model in (4) corresponding to our running example in (1) yields the following polynomial zonotope

$$\begin{aligned} \left\langle \begin{bmatrix} x_1 \\ x_2 \\ x_1^4 \end{bmatrix}, \emptyset, [-2, 2] \times [0, 4] \right\rangle_T &= \left\langle \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 16 \end{bmatrix}, [\cdot], \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 0 \end{bmatrix} \right\rangle_{\mathcal{PZ}} \\ &= \left\{ \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \alpha_1 + \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} \alpha_2 + \begin{bmatrix} 0 \\ 0 \\ 16 \end{bmatrix} \alpha_1^4 \mid \alpha_1, \alpha_2 \in [-1, 1] \right\}, \end{aligned}$$

where the high-level idea of the conversion is to represent the interval domain  $\mathcal{D}$  with dependent zonotope factors  $\alpha_i \in [-1, 1]$ .

### 3 Linearization via Fourier Features

We now present the automated generation of observables using random Fourier features [10]. Let us first motivate why Fourier features are a good choice for

<sup>1</sup> In contrast to [20, Def. 1], we explicitly do not integrate the constant offset  $c$  in  $G$ . Moreover, we omit the identifier vector used in the original work [20] for simplicity.



observables. For Koopman linearization, the observables  $g(x)$  define a transformation to a high-dimensional space. One commonly used approach to handle such high-dimensional spaces efficiently is the *kernel trick*: In many algorithms the data points  $x, y \in \mathbb{R}^n$  only appear in the form of inner products  $g(x)^T g(y)$ . In this case it suffices to define a kernel function  $k(x, y)$  that represents the similarity measure  $g(x)^T g(y)$  between data points in the high-dimensional feature space, rather than explicitly defining a transformation  $g(x)$  to this space. Kernel functions can also represent more general features that are not vectors and even infinite dimensional features, which motivates their application in the Koopman framework. The kernel trick is mainly applied for machine learning techniques [36], such as regression [38], clustering [18], and classification [39]. However, also the extended dynamic mode decomposition algorithm [41] can be formulated in terms of inner-products [42], so that the kernel trick can be applied for Koopman linearization. Rather than explicitly choosing observables  $g(x)$  we can therefore select a kernel function instead, which implicitly defines the observable function  $g(x)$  through the kernel's relation to an inner product space. Commonly used kernels are radial basis function kernels, polynomial kernels, and spline kernels.

The kernel trick cannot be applied directly to our reachability technique since we require an explicit formulation of the observables  $g(x)$ . We therefore first select a kernel function  $k(x, y)$ , and then determine observables  $g(x)$  that yield a good approximation of the kernel function  $k(x, y) \approx g(x)^T g(y)$ . Random Fourier features are a common technique to approximate kernel functions [10, 29]. They are based on Bochner's theorem [33, Sec. 1.4.3], which links a weakly stationary kernel function to a Fourier transform:

$$k(x, y) = \int_{\mathbb{R}^n} e^{j\omega^T(x-y)} d\mu(\omega) = \mathbb{E}_\omega \left( e^{j\omega^T x} \overline{e^{j\omega^T y}} \right), \quad (5)$$

where the function  $\mu : \mathbb{R}^n \rightarrow [0, 1]$  defines a probability distribution,  $\mathbb{E}_\omega(\cdot)$  denotes the expected value with respect to  $\omega$ ,  $j$  is the imaginary unit, and  $\bar{a}$  denotes the complex conjugate for a complex number  $a \in \mathbb{C}$ . The distribution  $\mu(\omega)$  associated with a specific kernel can be obtained by taking the inverse Fourier transform of  $k(x, y)$  [29]. We can collect  $m$  samples from the distribution  $\mu(\omega)$  to approximate the expected value in (5), which finally yields

$$k(x, y) = \mathbb{E}_\omega \left( e^{j\omega^T x} \overline{e^{j\omega^T y}} \right) \approx \frac{1}{m} \sum_{i=1}^m \underbrace{e^{j\omega_i^T x}}_{g_i(x)} \underbrace{e^{j\omega_i^T y}}_{g_i(y)}.$$

The random Fourier features are the resulting observables  $g_i(x)$  that approximate the kernel function. Note that we can omit the constant factor  $\frac{1}{m}$  since extended dynamic mode decomposition will automatically scale the observables accordingly. We consider real-valued kernels only, so we use Euler's formula  $e^{jx} = \cos(x) + j \sin(x)$  to simplify the random Fourier features to

$$g_i(x) = \sqrt{2} \cos(\omega_i^T x + b_i), \quad i = 1, \dots, m, \quad (6)$$

where the shift  $b_i$  is selected uniformly from the interval  $[0, 2\pi]$  and  $\omega_i$  is drawn randomly from the probability distribution  $\mu(\omega)$  corresponding to the kernel

that is used. While this random selection might appear to be a disadvantage at first sight, it is guaranteed that the random Fourier feature approximation converges to the exact kernel function when increasing the number of observables [29]. Moreover, we observed from our numerical experiments that changes in the values for  $b_i$  and  $\omega_i$  do not significantly influence the accuracy of the resulting linear approximation.

In summary, the random Fourier features presented above represent a systematic method for selecting a finite set of accurate observables, which requires only few hyperparameters. These hyperparameters include the type of kernel that is used, the kernel parameters, and the number of observables. For the numerical experiments in this paper we use a radial basis function kernel

$$k(x, y) = e^{-\frac{\|x-y\|_2^2}{2\ell^2}},$$

which contains the lengthscale  $\ell$  as the only parameter. The probability distribution  $\mu(\omega)$  for this kernel is the multivariate normal distribution with covariance matrix  $\ell^2 \cdot I_n$  centered at the origin [29, Fig. 1].

## 4 Verification Using Reachability Analysis

We now present our novel verification algorithm for Koopman linearized systems, which is summarized in Algorithm 1. For simplicity we assume that the specification we aim to verify is described by a single unsafe set  $\mathcal{U}$ , but the extension to multiple unsafe sets is straightforward. We first apply Taylor model arithmetic (see Sect. 2.2) to compute a tight non-convex enclosure for the image of the initial set  $\mathcal{X}_0$  through the observable function  $g(x)$  in Line 3. Since it simplifies the computation of the zonotope enclosures required later on, we then convert the resulting Taylor model to a polynomial zonotope in Line 4. This polynomial zonotope is used as the initial set for the computation of the reachable set for the Koopman linearized system as performed in Line 5, for which we can use any reachability algorithm for linear systems. For simplicity we assume here that the obtained reachable sets are exact. In the general case where the exact reachable set cannot be computed one can for example incorporate the error measures from [14] and [40] into the verification algorithm.

The problem we are facing now is that the reachable sets  $\mathcal{R}_0, \dots, \mathcal{R}_{t_F/\Delta t}$  are represented by polynomial zonotopes, a set representation for which exact collision checks with the unsafe set  $\mathcal{U}$  are computationally demanding. We resolve this issue by applying a novel polynomial zonotope refinement procedure in lines 6–19, where we recursively split the polynomial zonotopes until we can either verify or falsify the specification using zonotope enclosures of the split sets. In particular, we first enclose each polynomial zonotope in the queue  $\mathbf{L}$  with a zonotope in Line 9. For a zonotope  $\mathcal{Z} = \langle c, G \rangle_{\mathcal{Z}}$  collision checks with an unsafe set as performed in Line 10 are very efficient: If the unsafe set is a halfspace  $\mathcal{U} = \langle h, d \rangle_H$ , we have according to [15, Sec. 5.1]

$$(\mathcal{Z} \cap \mathcal{U} \neq \emptyset) \Leftrightarrow \left( h^T c - \sum_{i=1}^p |h^T G_{(\cdot, i)}| \leq d \right) \quad (7)$$

**Algorithm 1.** Verification of Koopman linearized systems

---

**Require:** Koopman linearized system  $\dot{g}(x) = Ag(x)$ , initial set  $\mathcal{X}_0$ , final time  $t_F$ , specification given as an unsafe set  $\mathcal{U}$ , time step size  $\Delta t$ , initial Taylor order  $\kappa_0$ .

**Ensure:** System is safe (res =  $\top$ ) or unsafe (res =  $\perp$ ).

```

1: res  $\leftarrow \perp$ ,  $\kappa \leftarrow \kappa_0$  (initialization)
2: repeat
3:    $\mathcal{T}(x) \leftarrow \{g(x) \mid x \in \mathcal{X}_0\}$  (comp. using Taylor model arithmetic with order  $\kappa$ )
4:    $\mathcal{PZ} \leftarrow \mathcal{T}(x)$  (convert Taylor model to polynomial zonotope, see [20, Prop. 4])
5:    $\mathcal{R}_0, \dots, \mathcal{R}_{t_F/\Delta t} \leftarrow$  reachability analysis of  $\dot{g}(x) = Ag(x)$  for initial set  $\mathcal{PZ}$ 
6:    $\mathbf{L} \leftarrow (\mathcal{R}_0, \dots, \mathcal{R}_{t_F/\Delta t})$  (initialize queue of not yet verified sets)
7:   repeat
8:      $\mathcal{PZ} \leftarrow \mathbf{L}_{(1)}$ ,  $\mathbf{L} \leftarrow (\mathbf{L}_{(2)}, \dots, \mathbf{L}_{(|\mathbf{L}|)})$  (pop first element from queue)
9:      $\mathcal{Z} \leftarrow$  zonotope enclosure of  $\mathcal{PZ}$  (see [20, Prop. 5])
10:    if  $\mathcal{Z} \cap \mathcal{U} \neq \emptyset$  then (check if specification is satisfied, see (7) and (8))
11:       $x_0, t \leftarrow$  most critical initial state and corresponding time
12:      if  $[I_n \ \mathbf{0}] e^{At} g(x_0) \in \mathcal{U}$  then
13:        return (specification falsified  $\Rightarrow$  system is unsafe)
14:      else
15:         $\mathcal{PZ}_1, \mathcal{PZ}_2 \leftarrow$  split  $\mathcal{PZ}$  (see Prop. 1 and (11))
16:         $\mathbf{L} \leftarrow (\mathbf{L}, \mathcal{PZ}_1, \mathcal{PZ}_2)$  (add new sets to queue)
17:      end if
18:    end if
19:    until  $\mathbf{L} = ()$  or splitting does not yield any further improvement
20:     $\kappa \leftarrow \kappa + 1$  (increase Taylor order)
21:  until  $\mathbf{L} = ()$  (queue empty  $\Rightarrow$  no intersection with  $\mathcal{U}$ )
22: res  $\leftarrow \top$  (if this line is reached no reach. set intersects  $\mathcal{U} \Rightarrow$  system is safe)

```

---

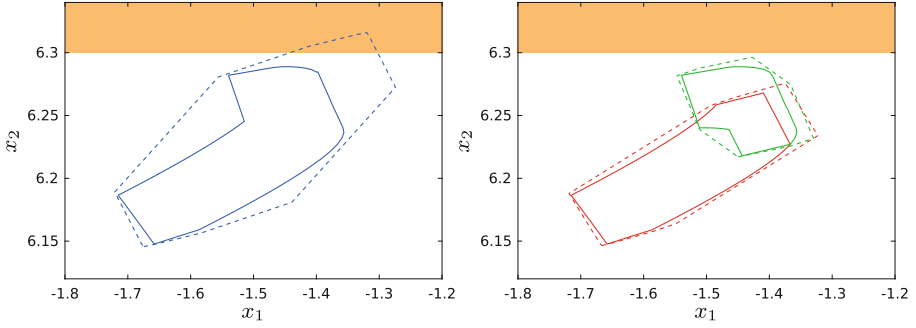
For general polytopes  $\mathcal{U} = \langle H, d \rangle_P$  collision checks can be realized using linear programming:

$$(\mathcal{Z} \cap \mathcal{U} \neq \emptyset) \Leftrightarrow (\delta = 0), \quad (8)$$

where

$$\delta = \min_{\alpha, x} \|c + G\alpha - x\|_1 \quad \text{s.t.} \quad \alpha \in [-\mathbf{1}, \mathbf{1}], \quad Hx \leq d. \quad (9)$$

If the specification cannot be verified, we next try to falsify it in lines 11–13 by extracting the initial point  $x_0$  that is expected to violate the specification the most from  $\mathcal{Z}$ . For a halfspace  $\mathcal{U} = \langle h, d \rangle_H$  the vector of zonotope factors  $\alpha = [\alpha_1 \dots \alpha_p]^T$  resulting in the largest violation is given as  $\alpha = -\text{sign}(h^T G)$ , where the signum function is interpreted elementwise. Since the factors  $\alpha$  of the zonotope enclosure are related to the dependent factors of the original polynomial zonotope and since polynomial zonotopes preserve dependencies during reachability analysis [21], we can then directly extract the initial point  $x_0$  corresponding to  $\alpha$  from the polynomial zonotope. For general polytopes we can use the optimal  $\alpha$  from the linear program in (9) to estimate the most critical initial point. If we can neither verify nor falsify the specification we have a so called spurious counterexample that arises due to the over-approximation introduced by the zonotope enclosure. We therefore split the polynomial zonotope in



**Fig. 2.** Reachable set for the Roessler system (see Sect. 5.1) at time  $t = 2.95$ , where polynomial zonotopes are depicted by solid lines, the corresponding zonotope enclosures are depicted by dashed lines, and the unsafe set is shown in orange. While the zonotope enclosure of the original polynomial zonotope is too conservative to verify the specification (left), splitting the polynomial zonotope once reduces the over-approximation enough for verification to succeed (right).

this case in Line 15 since splitting reduces the over-approximation in the zonotope enclosure (see Fig. 2). The split sets are then added to the queue in Line 16, where we use a first-in, first-out scheme for the queue to detect easy falsifications fast before excessively splitting the sets.

One remaining issue we are facing is that Taylor model arithmetic is not exact. Due to the over-approximation in the initial set it can therefore happen that we can neither verify nor falsify the specification by splitting the polynomial zonotope. To solve this issue we embed our whole algorithm into a repeat-until-loop that iteratively increases the order  $\kappa$  used for Taylor model arithmetic (see Line 20). Since Taylor model arithmetic converges to the exact result if the order goes to infinity, we obtain a complete algorithm that is guaranteed to terminate. In practice we can often prevent computationally expensive iterations of the outer loop by choosing the initial order  $\kappa_0$  large enough. It remains to decide when to stop splitting the polynomial zonotopes and increase the Taylor order instead (see Line 19). The simplest method is to just use an upper bound for the number of recursive splits that are performed. A more sophisticated approach is to abort splitting if the distance between the most critical point  $[I_n \ \mathbf{0}] e^{At} g(x_0)$  and the unsafe set  $\mathcal{U}$  is smaller than the over-approximation in the polynomial zonotope  $\mathcal{PZ}$ , which is given by the independent generators.

Finally, we provide a closed-form expression for splitting a polynomial zonotope since this operation is not specified in the original work [20]:

**Proposition 1.** (*Split*) Given a polynomial zonotope  $\mathcal{PZ} = \langle c, G, G_I, E \rangle_{\mathcal{PZ}} \subset \mathbb{R}^n$  and the index  $r \in \{1, \dots, p\}$  of one dependent factor, the operation  $\text{split}(\mathcal{PZ}, r)$  returns two polynomial zonotopes  $\mathcal{PZ}_1, \mathcal{PZ}_2$  satisfying  $\mathcal{PZ}_1 \cup \mathcal{PZ}_2 = \mathcal{PZ}$ :

$$\begin{aligned} \mathcal{PZ}_1 &= \left\langle c, [\widehat{G}_1^{(1)} \dots \widehat{G}_h^{(1)}], G_I, [\widehat{E}_1 \dots \widehat{E}_h] \right\rangle_{PZ} \\ \mathcal{PZ}_2 &= \left\langle c, [\widehat{G}_1^{(2)} \dots \widehat{G}_h^{(2)}], G_I, [\widehat{E}_1 \dots \widehat{E}_h] \right\rangle_{PZ} \end{aligned}$$

with

$$\begin{aligned} \widehat{E}_i &= \begin{bmatrix} E_{(\{1, \dots, r-1\}, i)} & E_{(\{1, \dots, r-1\}, i)} \cdots E_{(\{1, \dots, r-1\}, i)} & E_{(\{1, \dots, r-1\}, i)} \\ 0 & 1 & \dots & E_{(r, i)} - 1 & E_{(r, i)} \\ E_{(\{r+1, \dots, p\}, i)} & E_{(\{r+1, \dots, p\}, i)} \cdots E_{(\{r+1, \dots, p\}, i)} & E_{(\{r+1, \dots, p\}, i)} & E_{(\{r+1, \dots, p\}, i)} \end{bmatrix}, \\ \widehat{G}_i^{(k)} &= \left[ b_{i,0}^{(k)} \cdot G_{(\cdot, i)} \cdots b_{i, E_{(r, i)}}^{(k)} \cdot G_{(\cdot, i)} \right], \\ b_{i,j}^{(1)} &= 0.5^{E_{(r, i)}} \binom{E_{(r, i)}}{j}, \quad b_{i,j}^{(2)} = -0.5^{E_{(r, i)}} (2(E_{(r, i)} \bmod 2) - 1) \binom{E_{(r, i)}}{j}, \end{aligned}$$

where  $x \bmod y$ ,  $x, y \in \mathbb{N}_0$  is the modulo operation and  $\binom{w}{z}$ ,  $w, z \in \mathbb{N}_0$  denotes the binomial coefficient. To remove redundancies we subsequently apply the compact operation as defined in [20, Prop. 2] to  $\mathcal{PZ}_1$  and  $\mathcal{PZ}_2$ .

*Proof.* The **split** operation is based on the substitution of the selected dependent factor  $\alpha_r$  with two new dependent factors  $\alpha_{r,1}$  and  $\alpha_{r,2}$ :

$$\begin{aligned} \{ \alpha_r \mid \alpha_r \in [-1, 1] \} &= \{ 0.5(1 + \alpha_{r,1}) - 0.5(1 + \alpha_{r,2}) \mid \alpha_{r,1}, \alpha_{r,2} \in [-1, 1] \} \\ &\cup \{ 0.5(1 + \alpha_{r,1}) \mid \alpha_{r,1} \in [-1, 1] \} \cup \{ -0.5(1 + \alpha_{r,2}) \mid \alpha_{r,2} \in [-1, 1] \}. \end{aligned} \quad (10)$$

Inserting this substitution into the definition of polynomial zonotopes in Definition 5 yields

$$\begin{aligned} \mathcal{PZ} &= \left\{ c + \sum_{i=1}^h \left( \prod_{k=1}^p \alpha_k^{E_{(k, i)}} \right) G_{(\cdot, i)} + \sum_{j=1}^q \beta_j G_{I(\cdot, j)} \mid \alpha_k, \beta_j \in [-1, 1] \right\} \stackrel{(10)}{=} \\ &\underbrace{\left\{ c + \sum_{i=1}^h \left( \prod_{\substack{k=1 \\ k \neq r}}^p \alpha_k^{E_{(k, i)}} \right) \left( \frac{1 + \alpha_{r,1}}{2} \right)^{E_{(r, i)}} G_{(\cdot, i)} + \sum_{j=1}^q \beta_j G_{I(\cdot, j)} \mid \alpha_k, \beta_j, \alpha_{r,1} \in [-1, 1] \right\}}_{=\mathcal{PZ}_1} \\ &\cup \underbrace{\left\{ c + \sum_{i=1}^h \left( \prod_{\substack{k=1 \\ k \neq r}}^p \alpha_k^{E_{(k, i)}} \right) \left( \frac{1 + \alpha_{r,2}}{-2} \right)^{E_{(r, i)}} G_{(\cdot, i)} + \sum_{j=1}^q \beta_j G_{I(\cdot, j)} \mid \alpha_k, \beta_j, \alpha_{r,2} \in [-1, 1] \right\}}_{=\mathcal{PZ}_2}. \end{aligned}$$

Finally, with

$$\begin{aligned} \left( \frac{1 + \alpha_{r,1}}{2} \right)^{E_{(r, i)}} &= b_{i,0}^{(1)} + b_{i,1}^{(1)} \alpha_{r,1} + b_{i,2}^{(1)} \alpha_{r,1}^2 + \cdots + b_{i, E_{(r, i)}}^{(1)} \alpha_{r,1}^{E_{(r, i)}} \\ \left( \frac{1 + \alpha_{r,2}}{-2} \right)^{E_{(r, i)}} &= b_{i,0}^{(2)} + b_{i,1}^{(2)} \alpha_{r,2} + b_{i,2}^{(2)} \alpha_{r,2}^2 + \cdots + b_{i, E_{(r, i)}}^{(2)} \alpha_{r,2}^{E_{(r, i)}} \end{aligned}$$

we obtain the equations above.

The split operation for polynomial zonotopes is not exact, meaning that the resulting sets usually overlap (see Fig. 2). To minimize the size of the overlapping region we split the dependent factor with index  $r$  that maximizes the following heuristic:

$$\max_{r \in \{1, \dots, p\}} \sum_{\substack{i=1 \\ E_{(r,i)} > 1}}^h (1 - 0.5^{E_{(r,i)}}) \|G_{(\cdot, i)}\|_2, \quad (11)$$

where  $G \in \mathbb{R}^{n \times h}$  and  $E \in \mathbb{N}_0^{p \times h}$  are the generator and exponent matrix of the polynomial zonotope. Moreover, since the goal of splitting in Algorithm 1 is to verify a certain specification, it is advisable to first project the polynomial zonotope onto the halfspace normal directions of the unsafe set  $\mathcal{U}$  before evaluating the heuristic (11) in order to direct the splitting process towards directions that are beneficial for verification.

Note that the polynomial zonotope refinement technique presented in this section is not restricted to verification of Koopman linearized systems, but can equally be applied for collision checks of polynomial zonotopes or Taylor models with halfspaces and polytopes in general. Moreover, by inverting the inequality constraints polynomial zonotope refinement can also be applied to check if a Taylor model or polynomial zonotope is contained in a halfspace or polytope.

## 5 Experimental Results

We now evaluate the performance of random Fourier feature observables and our novel reachability algorithm on various benchmark systems. For this, we compare our approach with the closest method from the literature [5]. Since the algorithms presented there are implemented in Julia, we also implemented our approach in Julia to obtain a fair comparison of the computation time. In our implementation we use the package `TaylorModels.jl`<sup>2</sup> for Taylor model arithmetic and the package `DataDrivenDiffEq.jl`<sup>3</sup> for extended dynamic mode decomposition. All computations are carried out on a 3.2 GHz 8-core AMD Ryzen 7 5800H processor with 16 GB memory. We published our implementation together with a repeatability package that reproduces the results shown in this paper as a CodeOcean compute capsule<sup>4</sup>.

### 5.1 Benchmarks

Let us first define all benchmarks that we use for the evaluation. Again, we consider the same systems and specifications as in [5] for a fair comparison:

<sup>2</sup> <https://github.com/JuliaIntervals/TaylorModels.jl>.

<sup>3</sup> <https://datadriven.sciml.ai/>.

<sup>4</sup> <https://codeocean.com/capsule/8730054/tree/v1>.

**Roessler Attractor:** The dynamic equations for the Roessler attractor [32] are

$$\begin{aligned}\dot{x}_1 &= -x_2 - x_3 \\ \dot{x}_2 &= x_1 + 0.2 x_2 \\ \dot{x}_3 &= 0.2 + x_3 (x_1 - 5.7),\end{aligned}$$

and we consider the initial set  $\mathcal{X}_0 = [-0.05, 0.05] \times [-8.45, -8.35] \times [-0.05, 0.05]$ , the final time  $t_F = 6$ , and the unsafe region  $x_2 \geq 6.375 - 0.025 \cdot i$  parameterized by  $i \in [0, 20]$ .

**Steam Governor:** The dynamic equations for the steam governor [37] are

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3^2 \sin(x_1) \cos(x_1) - \sin(x_1) - 3 x_2 \\ \dot{x}_3 &= \cos(x_1) - 1,\end{aligned}$$

and we consider the initial set  $\mathcal{X}_0 = [0.95, 1.05] \times [-0.05, 0.05] \times [0.95, 1.05]$ , the final time  $t_F = 3$ , and the unsafe set  $x_2 \leq -0.25 + 0.01 \cdot i$  parameterized by  $i \in [0, 10]$ .

**Coupled Van-der-Pol Oscillator:** The dynamic equations for the coupled Van-der-Pol oscillator [30] are

$$\begin{aligned}\dot{x}_1 &= x_2 & \dot{x}_3 &= x_4 \\ \dot{x}_2 &= (1 - x_1^2) x_2 - x_1 + (x_3 - x_1) & \dot{x}_4 &= (1 - x_3^2) x_4 - x_3 + (x_1 - x_3),\end{aligned}$$

and we consider the initial set  $\mathcal{X}_0 = [-0.025, 0.025] \times [0.475, 0.525] \times [-0.025, 0.025] \times [0.475, 0.525]$ , the final time  $t_F = 2$ , and the unsafe set  $x_1 \geq 1.25 - 0.05 \cdot i$  parameterized by  $i \in [1, 16]$ .

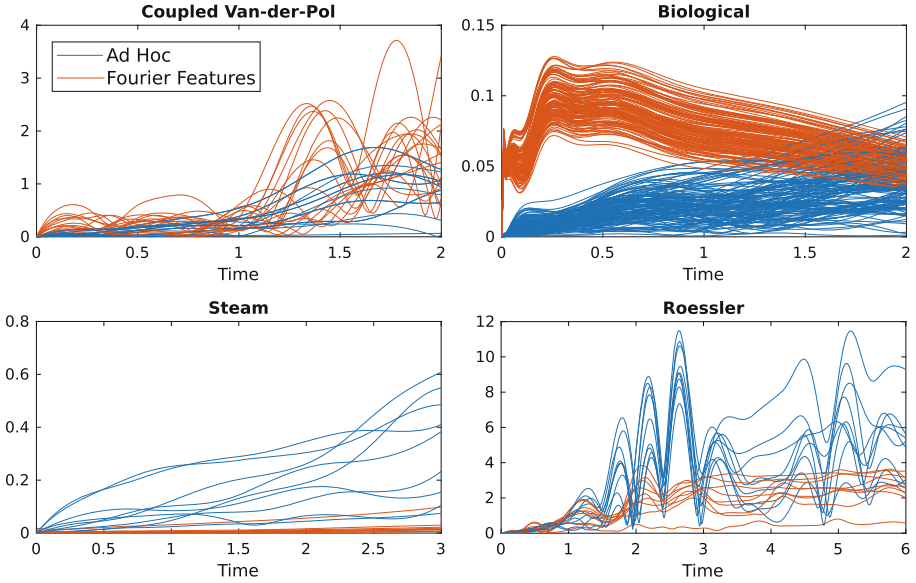
**Biological System:** The dynamic equations for the biological system [19] are

$$\begin{aligned}\dot{x}_1 &= -0.4 x_1 + 5 x_3 x_4 & \dot{x}_5 &= -5 x_5 x_6 + 5 x_3 x_4 \\ \dot{x}_2 &= 0.4 x_1 - x_2 & \dot{x}_6 &= 0.5 x_7 - 5 x_5 x_6 \\ \dot{x}_3 &= x_2 - 5 x_3 x_4 & \dot{x}_7 &= -0.5 x_7 + 5 x_5 x_6, \\ \dot{x}_4 &= 5 x_5 x_6 - 5 x_3 x_4\end{aligned}$$

and we consider the initial set  $\mathcal{X}_0 = [0.99, 1.01] \times \dots \times [0.99, 1.01]$ , the final time  $t_F = 2$ , and the unsafe set  $x_4 \leq 0.883 + 0.002 \cdot i$  parameterized by  $i \in [0, 20]$ .

## 5.2 Approximation Error

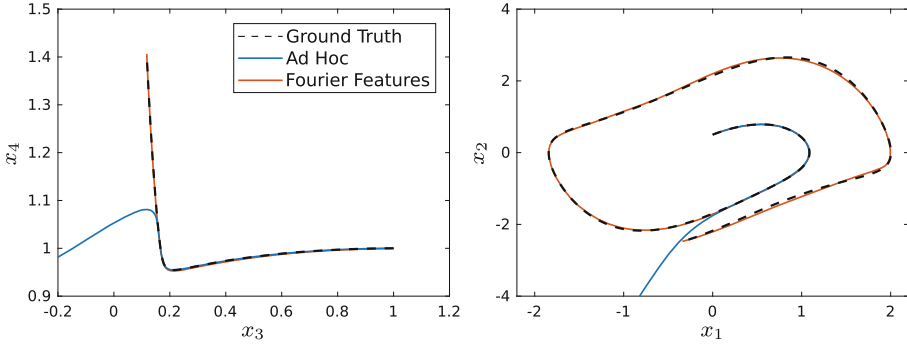
We first investigate the accuracy of the Koopman linearized system with respect to the original nonlinear dynamics, where we compare our random Fourier feature observables with the ad hoc observables from [5]. These ad hoc observables consist of multi-variate polynomials of the system state  $x$  up to a fixed order, trigonometric functions of the time  $t$ , and combinations of these



**Fig. 3.** Relative simulation error between Koopman linearized systems and the original nonlinear system in percent.

(e.g.,  $x_1 x_2 \sin^2(t) \cos(t)$ ). To obtain the data traces required for extended dynamic mode decomposition we simulate the original nonlinear systems for 500 points sampled from the corresponding initial set, where a Sobol sequence is used for sampling. For the generation of the random Fourier feature observables according to (6) we use the parameter  $\ell = 0.3$  and  $m = 71$  for the Roessler attractor,  $\ell = 1.62$  and  $m = 72$  for the steam governor,  $\ell = 1.24$  and  $m = 132$  for the coupled Van-der-Pol oscillator, and  $\ell = 1.81$  and  $m = 105$  for the biological system, where  $\ell$  is the lengthscale parameter of the kernel and the number of observables  $m$  is chosen identical to the one used for the ad hoc observables [5]. As a measure for the accuracy we use the Euclidean distance between simulated trajectories for the original nonlinear system and the Koopman linearized system. The initial points for these trajectories are the center and the vertices of the initial set. According to Fig. 3 random Fourier feature observables are for the steam governor and the Roessler attractor more accurate than than the ad hoc observables used in earlier work [5]. Moreover, while for the short time horizons considered in Fig. 3 it seems that the ad hoc observables are more precise for the coupled Van-der-Pol oscillator and the biological system, over longer time horizons the error of the ad hoc observables is exploding. This is visualized in Fig. 4, where the trajectory corresponding to the ad hoc observables progresses into a completely different direction than the original system, while random Fourier features stay accurate. In this way, random Fourier features are not only a more systematic approach for choosing observables, but also improve the precision of the resulting Koopman linearized system.





**Fig. 4.** Comparison of simulations for Koopman linearized systems with the ground truth from the original nonlinear system for a time horizon of  $t_F = 10$ , where the biological system is shown on the left and the coupled Van-der-Pol oscillator is shown on the right.

### 5.3 Verification Using Reachability Analysis

We now compare our novel verification algorithm for Koopman linearized systems with the verification strategies presented in [5]. In particular, we compare to verification of the original nonlinear system using Flow\* [9], direct encoding of nonlinear constraints using a SMT solver [5, Sec. 4.1], and zonotope domain splitting [5, Sec. 4.4]. Both approaches from [5] consider discrete-time safety, where the system is considered to be safe if the specification is satisfied at time points  $0, \Delta t, 2\Delta t, \dots, t_F$  with  $\Delta t = 0.05$ . While our verification algorithm also supports continuous-time safety, we consider discrete-time safety here to obtain a fair comparison. Note that for discrete-time safety the reachable set computation in Line 5 of Algorithm 1 simplifies to  $\mathcal{R}_i = [I_n \ \mathbf{0}] e^{A_i \Delta t} \mathcal{X}_0$ ,  $i = 0, \dots, t_F/\Delta t$ . For the comparison we consider both, the ad hoc observables used in [5] as well as the random Fourier feature observables presented here.

The resulting computation times for verification are summarized in Table 1. For all benchmark instances our novel verification algorithm has the lowest computation time, and is often even magnitudes faster than the other verification approaches. The main reason for this is that with our polynomial refinement strategy we can completely avoid the computationally expensive calls to SMT solvers used by the other methods. Moreover, while the computation time for the other approaches often depends on how difficult it is to verify or falsify the specification, our algorithm exhibits roughly equal runtimes for all specifications. The explanation for this is that the polynomial zonotope refinement approach that we use for the collision checks with unsafe sets is very efficient, so that the majority of the runtime is spent on the computation of the image through the observable function using Taylor model arithmetic, a task which is independent from the specification. Interestingly, using random Fourier features instead of ad hoc observables can either prolong or accelerate the verification process, depending on the benchmark instance and verification approach used. However, even if

**Table 1.** Computation time in seconds for verification or falsification of the benchmark systems from Sect. 5.1 using different approaches, where the symbol – indicates that the computation timed-out after 2 h. The parameter  $i$  specified in the second column changes the specification, and the third column shows weather the specification can be verified or falsified.

	$i$	Safe?	Flow*	Direct Enc.		Zono. Split.		Our App.	
				ad hoc	fourier	ad hoc	fourier	ad hoc	fourier
Coupled VP	1	✓	251	788	398	0.57	171	<b>0.20</b>	<b>3.00</b>
	8	×	497	680	120	53	232	<b>0.79</b>	<b>3.77</b>
	16	×	1665	557	373	18	38	<b>0.20</b>	<b>2.99</b>
Biological	1	✓	260	470	–	0.59	–	<b>0.44</b>	<b>1.95</b>
	5	✓	250	426	–	49	–	<b>0.44</b>	<b>1.73</b>
	10	✓	238	427	–	179	–	<b>0.46</b>	<b>1.76</b>
Steam	0	✓	61	197	149	182	42	<b>0.12</b>	<b>0.25</b>
	5	×	285	59	40	37	38	<b>0.38</b>	<b>0.56</b>
	10	×	77	29	20	18	27	<b>0.12</b>	<b>0.26</b>
Roessler	0	✓	55	181	291	9.53	117	<b>0.55</b>	<b>0.35</b>
	10	×	78	177	385	5.01	241	<b>0.22</b>	<b>0.75</b>
	20	×	55	174	158	3.5	86	<b>0.21</b>	<b>0.34</b>

they prolong the time required for verification in some cases, the usage of random Fourier feature observables can be justified by their superior accuracy demonstrated in Sect. 5.2. Yet another observation is that direct encoding and zonotope domain splitting are not able to verify or falsify the high-dimensional biological model at all if random Fourier feature observables are used. The reason for this is that both of these approaches apply an SMT solver for verification, which do not scale to high-dimensions and are not well-suited for handling the trigonometric functions as well as the high coupling between variables used for random Fourier feature observables. So in summary our proposed verification algorithm outperforms all exiting verification techniques for Koopman linearized systems in terms of runtime. In addition, it handles different types of observables well and scales to high-dimensional systems.

## 6 Conclusion

We presented two major improvements for reachability analysis of Koopman operator linearized systems: First, we use random Fourier features as observable functions, which yields a systematic approach requiring much less user insight than previous methods. Second, we handle the nonlinear transformation of the initial state by combining Taylor model arithmetic with polynomial zonotope refinement. As demonstrated on several nonlinear system benchmarks, the combination of these two techniques is both extremely accurate and extremely fast.

The main trade-off with Koopman linearized systems is that the guarantees are on the system approximation, not the original system. Despite this, we believe

the method could still be useful for verification in systems engineering, where the goal is to produce evidence that the system meets its requirements. It could also be effective for finding unsafe counterexamples—falsification—or to analyze systems where only simulation code is provided, or even real-world systems where sensor measurements could be used to create a Koopman linearized model for analysis. As such systems do not have models given with symbolic differential equations, most traditional reachability methods cannot be applied.

**Acknowledgements.** This material is based upon work supported by the Air Force Office of Scientific Research, the DARPA Assured Autonomy program under the United States Air Force, and the Office of Naval Research under award numbers FA9550-19-1-0288, FA9550-21-1-0121, FA9550-22-1-0450, FA2386-17-1-4065, FA8750-19-C-0092, and N00014-22-1-2156. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force, DARPA, or the United States Navy. Distribution Statement A: Approved for Public Release; Distribution is Unlimited. PA: AFRL-2022-1356.

## References

1. Althoff, M.: Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In: Proceedings of the International Conference on Hybrid Systems: Computation and Control, pp. 173–182 (2013)
2. Althoff, M.: Reachability analysis of large linear systems with uncertain inputs in the Krylov subspace. *Trans. Autom. Control* **65**(2), 477–492 (2019)
3. Amini, A., et al.: Error bounds for Carleman linearization of general nonlinear systems. In: Proceedings of the International Conference on Control and its Applications, pp. 1–8 (2021)
4. Bak, S., et al.: Numerical verification of affine systems with up to a billion dimensions. In: Proceedings of the International Conference on Hybrid Systems: Computation and Control, pp. 23–32 (2019)
5. Bak, S., et al.: Reachability of black-box nonlinear systems after Koopman operator linearization. In: Proceedings of the International Conference on Analysis and Design of Hybrid Systems, pp. 253–258 (2021)
6. Bogomolov, S., et al.: Reach set approximation through decomposition with low-dimensional sets and high-dimensional matrices. In: Proceedings of the International Conference on Hybrid Systems: Computation and Control, pp. 41–50 (2018)
7. Carleman, T.: Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires. *Acta Math.* **59**, 63–87 (1932)
8. Chen, X., et al.: Taylor model flowpipe construction for non-linear hybrid systems. In: Proceedings of the Real-Time Systems Symposium, pp. 183–192 (2012)
9. Chen, X., Abraham, E., Sankaranarayanan, S.: Flow\*: an analyzer for non-linear hybrid systems. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 258–263. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-39799-8\\_18](https://doi.org/10.1007/978-3-642-39799-8_18)
10. DeGennaro, A.M., Urban, N.M.: Scalable extended dynamic mode decomposition using random kernel approximation. *SIAM J. Sci. Comput.* **41**(3), 1482–1499 (2019)

11. Duggirala, P.S., Viswanathan, M.: Parsimonious, simulation based verification of linear systems. In: Proceedings of the International Conference on Computer Aided Verification, pp. 477–494 (2016)
12. Forets, M., Pouly, A.: Explicit error bounds for Carleman linearization. arXiv preprint [arXiv:1711.02552](https://arxiv.org/abs/1711.02552) (2017)
13. Forets, M., Schilling, C.: Reachability of weakly nonlinear systems using Carleman linearization. In: Bell, P.C., Totzke, P., Potapov, I. (eds.) RP 2021. LNCS, vol. 13035, pp. 85–99. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-89716-1\\_6](https://doi.org/10.1007/978-3-030-89716-1_6)
14. Frehse, G., et al.: SpaceEx: scalable verification of hybrid systems. In: Proceedings of the International Conference on Computer Aided Verification, pp. 379–395 (2011)
15. Girard, A.: Reachability of uncertain linear systems using zonotopes. In: Proceedings of the International Conference on Hybrid Systems: Computation and Control, pp. 291–305 (2005)
16. Han, Y., et al.: Deep learning of Koopman representation for control. In: Proceedings of the International Conference on Decision and Control, pp. 1890–1895 (2020)
17. Jaulin, L., Kieffer, M., Didrit, O., Walter, E.: Interval analysis. In: Applied Interval Analysis, pp. 11–43. Springer (2001)
18. Kim, D.W., et al.: Evaluation of the performance of clustering algorithms in kernel-induced feature space. *Pattern Recogn.* **38**(4), 607–611 (2005)
19. Klipp, E., et al.: *Systems Biology in Practice: Concepts, Implementation and Application*. Wiley, Hoboken (2005)
20. Kochdumper, N., Althoff, M.: Sparse polynomial zonotopes: a novel set representation for reachability analysis. *Trans. Autom. Control* **66**(9), 4043–4058 (2021)
21. Kochdumper, N., et al.: Utilizing dependencies to obtain subsets of reachable sets. In: Proceedings of the International Conference on Hybrid Systems: Computation and Control (2020)
22. Koopman, B.O.: Hamiltonian systems and transformation in Hilbert space. *Proc. Natl. Acad. Sci. U.S.A.* **17**(5), 315–318 (1931)
23. Kurzhanski, A.B., Varaiya, P.: Ellipsoidal techniques for reachability analysis. In: Proceedings of the International Conference on Hybrid Systems: Computation and Control, pp. 202–214 (2000)
24. Liu, J.P., et al.: Efficient quantum algorithm for dissipative nonlinear differential equations. *Proc. Natl. Acad. Sci. U.S.A.* **118**(35), e2026805118 (2021)
25. Makino, K., Berz, M.: Taylor models and other validated functional inclusion methods. *Int. J. Pure Appl. Math.* **4**(4), 379–456 (2003)
26. Mauroy, A., Mezić, I., Susuki, Y. (eds.): *The Koopman Operator in Systems and Control*. LNCIS, vol. 484. Springer, Cham (2020). <https://doi.org/10.1007/978-3-030-35713-9>
27. Mitchell, I.M., et al.: A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *Trans. Autom. Control* **50**(7), 947–957 (2005)
28. Otto, S.E., Rowley, C.W.: Koopman operators for estimation and control of dynamical systems. *Annu. Rev. Control Robot. Auton. Syst.* **4**, 59–87 (2021)
29. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: Proceedings of the International Conference on Neural Information Processing Systems, pp. 1177–1184 (2007)
30. Rand, R., Holmes, P.: Bifurcation of periodic motions in two weakly coupled Van der Pol oscillators. *Int. J. Non-Linear Mech.* **15**(4–5), 387–399 (1980)

31. Rauh, A., et al.: Carleman linearization for control and for state and disturbance estimation of nonlinear dynamical processes. In: Proceedings of the International Conference on Methods and Models in Automation and Robotics, pp. 455–460 (2009)
32. Rössler, O.E.: An equation for continuous chaos. *Phys. Lett. A* **57**(5), 397–398 (1976)
33. Rudin, W.: *Fourier Analysis on Groups*. Courier Dover Publications, Mineola (2017)
34. Sankaranarayanan, S.: Automatic abstraction of non-linear systems using change of bases transformations. In: Proceedings of the International Conference on Hybrid Systems: Computation and Control, pp. 143–152 (2011)
35. Sankaranarayanan, S.: Change-of-bases abstractions for non-linear hybrid systems. *Nonlinear Anal. Hybrid Syst.* **19**, 107–133 (2016)
36. Schölkopf, B., Smola, A.J.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge (2018)
37. Sotomayor, J., et al.: Bifurcation analysis of the Watt governor system. *Comput. Appl. Math.* **26**(1), 19–44 (2007)
38. Takeda, H., et al.: Kernel regression for image processing and reconstruction. *Trans. Image Process.* **16**(2), 349–366 (2007)
39. Tuia, D., et al.: Learning relevant image features with multiple-kernel classification. *Trans. Geosci. Remote Sens.* **48**(10), 3780–3791 (2010)
40. Wetzlinger, M., et al.: Adaptive parameter tuning for reachability analysis of linear systems. In: Proceedings of the International Conference on Decision and Control, pp. 5145–5152 (2020)
41. Williams, M.O., et al.: A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. *J. Nonlinear Sci.* **25**(6), 1307–1346 (2015)
42. Williams, M.O., et al.: A kernel-based method for data-driven Koopman spectral analysis. *J. Comput. Dyn.* **2**(2), 247–265 (2015)
43. Yeung, E., et al.: Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. In: Proceedings of the American Control Conference, pp. 4832–4839 (2019)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

