Koopman Operator for Nonlinear Flight Dynamics

Andrew Lock, Viv Bone, Ingo Jahn

October 14, 2022

Version 1.2

Institute for Advanced Engineering and Space Sciences
University of Southern Queensland
Springfield, Australia

Contents

1 Introduction **1**

2 Background **1**
2.1 The Koopman operator 1
2.2 Dynamic mode decomposition (DMD) 3
2.3 The Galerkin method 3

3 Applications **4**
3.1 Past applications 4
3.2 Future applications 5

4 Implementation and development of the Galerkin method **6**
4.1 Basis functions 6
4.2 Domain selection and state-space scaling 7
4.3 System parameters and inputs 7
4.3.1 Parameters 7
4.3.2 Inputs 7
4.4 Efficient analytic computation of the Koopman matrix 8

5 Demonstration and discussion **10**
5.1 Test case: Duffing oscillator 10
5.1.1 Single attraction basin 11
5.1.2 Multiple attraction basins 13
5.2 Glider trajectory 15
5.2.1 Model with parameters 15
5.2.2 Model with autonomous control 18
5.3 Discussion 20
5.3.1 State-space domain 20
5.3.2 Linear system accuracy 20

6 Conclusion and recommendations **21**
6.1 Conclusion 21
6.2 Future work 22

1 Introduction

Dynamical systems representing vehicle flight are inherently nonlinear. When the underlying continuous-time dynamics are known, solutions are typically obtained by integrating the system equations over appropriately small time-steps Methods such as multiple-shooting and pseudo-spectral analysis are used to understand the behaviour of such systems. However, there is currently no generalised framework for explicit characterisation and solution of nonlinear systems.

Conversely, linear systems are well understood, and many efficient algorithms are available for explicit characterisation and prediction. Linear approximations of nonlinear systems is an active area of research, with the objective of applying spectral analysis and linear theory to nonlinear dynamics. One such method is the Koopman operator theory, which was first proposed for continuous dynamical systems in 1932 [1, 2]. The Koopman operator *unfolds* a nonlinear dynamical system into an infinite-dimension linear system [3]. When these lifted dimensions are truncated to a finite space, existing linear system analysis tools can be applied.

Most applications of Koopman operator theory to dynamical systems utilise the methods based on dynamic mode decomposition (DMD), in particular extended dynamic mode decomposition (EDMD). EDMD methods uses time series data to approximate the linear system which advances the lifted states, using best-fit methods. The primary focus of recent research is the application of EDMD to create data-driven models, often when the underlying system is unknown. However, a solution to the lifted state linear system can also be obtained using analytic techniques using the underlying continuous non-linear equations. Analytic techniques provide an exact solution to the system projection onto a set of basis functions, such as would be found using EDMD with infinite data points [4]. For the purpose of predicting and analysing flight dynamics, the governing continuous dynamics are typically understood, such as aerodynamics and propulsion. Analytic derivations of the linear Koopman system therefore offers benefits compared to the EDMD alternative, including a guaranteed order of captured dynamics.

This paper demonstrates the application of an analytically-derived Koopman linearisation of flight dynamical systems, and is structured as follows: Sec. 2 summarises the history fundamental theory of the Koopman operator, DMD, and the Galerkin method. Sec. 3 describes past and future uses of Koopman linear systems for aerospace dynamics. Sec. 4 describes the implementation of the Galerkin method in this work, including the incorporation of system parameters and inputs, and new efficient calculation algorithms. Sec. 5 demonstrates the performance of Koopman lifted systems for a Duffing oscillator and autonomous glider. The paper concludes with suggestions for future work in Sec. 6.

2 Background

2.1 The Koopman operator

This section describes the theory which underpins the Koopman operator. We start with a continuous dynamical system of the form

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)) \tag{1}$$

where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ is a set of n system states, and $\mathbf{f}(\mathbf{x}) : \mathcal{X} \to \mathcal{X}$ represents the system dynamics. For an observable function of the space $g(\mathbf{x})$, the continuous-time Koopman operator $\mathcal{K}$ is defined as

$$\mathcal{K} \cdot g(\mathbf{x}) := g(\mathbf{f}(\mathbf{x})) = \frac{d}{dt} g(\mathbf{x}) \ . \tag{2}$$

This operation is linear, and so

$$\mathcal{K} \cdot (a_1 g_1(\mathbf{x}) + a_2 g_2(\mathbf{x})) = a_1 \mathcal{K} \cdot g_1(\mathbf{x}) + a_2 \mathcal{K} \cdot g_2(\mathbf{x}) \ , \tag{3}$$

where a_1 and a_2 are constants. Although $\mathcal{K}$ is linear it, is not guaranteed to have a finite number of terms. In fact, $\mathcal{K}$ is very likely to have infinite terms for any real system and arbitrary choice of basis space $\mathcal{G}(\mathcal{X})$. A solution to this problem is to consider an observable function which is an eigenfunction of $\mathcal{K}$, which by definition evolves linearly in time. For an eigenfunction $\phi(\mathbf{x})$ of $\mathcal{K}$, we express this identity as

$$\frac{d}{dt}\phi(\mathbf{x}) = \mathcal{K} \cdot \phi(\mathbf{x}) = \lambda\phi(\mathbf{x}) \ . \tag{4}$$

where λ is the corresponding eigenvalue. We can extend (4) to observables composed of a linear combination of eigenfunctions

$$g(\mathbf{x}) = \sum_{i=1}^{\infty} v_i\phi_i(\mathbf{x}) \ , \tag{5}$$

with a corresponding linear time evolution,

$$\frac{d}{dt}g(\mathbf{x}) = \sum_{i=1}^{\infty} \lambda_i v_i\phi_i(\mathbf{x}) \ . \tag{6}$$

Eq. (5) maps $\mathcal{X}$ to a new Hilbert space $\mathcal{G}(\mathcal{X})$, which has an eigenfunction coordinate basis. However, instead of a single scalar observable, we are normally interested in multiple system observables

$$\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_m(\mathbf{x})]^{\mathrm{T}} \ . \tag{7}$$

which can similarly be expressed as

$$\mathbf{g}(\mathbf{x}) = \sum_{i=1}^{\infty} \mathbf{v}_i\phi_i(\mathbf{x}) \ . \tag{8}$$

where $\mathbf{v}_i$ is a vector known as the *Koopman mode* corresponding to eigenfunction ϕ_i. With the exception of special cases such as Hamiltonian dynamical systems, there are generally infinite eigenfunctions in the Hilbert space from (8), which cannot be captured in a finite linear system. Instead, most applied Koopman methods choose or identify a finite set of observable functions $\mathbf{z}(\mathbf{x})$ which approximate a set of eigenfunctions, called a *Koopman-invariant subspace*, so that

$$\mathbf{z}(\mathbf{x}) \approx \sum_{i=1}^{m} \mathbf{v}_i z_i(\mathbf{x}) \tag{9}$$

and

$$\frac{d}{dt}\mathbf{z}(\mathbf{x}) \approx \sum_{i=1}^{m} \lambda_i \mathbf{v}_i z_i(\mathbf{x}) \ . \tag{10}$$

Combining these finite Koopman modes and eigenvalues into a matrix $\mathbf{K} = [\lambda_1\mathbf{v}_1, \lambda_2\mathbf{v}_2, \ldots, \lambda_m\mathbf{v}_m]$ yields the linear system

$$\frac{d}{dt}\mathbf{z}(\mathbf{x}) \approx \mathbf{K}\mathbf{z}(\mathbf{x}) \ . \tag{11}$$

Defining the basis functions $\mathbf{z}(\mathbf{x})$ and Koopman matrix $\mathbf{K}$ is the focus of various applied Koopman techniques, and the number and type of basis functions have large influence of the accuracy of this approximation. In some cases, such as DMD fluid dynamics analysis, the objective is to identify a small dominant set of $\mathbf{z}(\mathbf{x})$ which give a reasonable approximation in (9), in order to understand the dominant spectral properties and modes from data. Alternatively, such as in the Galerkin method described Sec 2.3 a large set of orthogonal $\mathbf{z}(\mathbf{x})$ are chosen in order to give very close approximation to (9) for accurate system prediction.

2.2 Dynamic mode decomposition (DMD)

The dynamic mode decomposition (DMD) method was initially proposed by Schmid [5] in 2010 without inferring a direct link to the Koopman operator. Later works by Rowley, Mezic, and collaborators, demonstrated that DMD is a finite approximation to the Koopman operator [6, 4, 7], and initial applications of DMD were limited to analysis of fluid dynamics data [8, 9]. Reviews by Tu [10] and Brunton et al. [3] provide further background to the history, variants, and applications, of DMD.

DMD algorithms describe the standard methods to approximate a Koopman operator from measured or simulated data. For this purpose, it deals with the discrete-time Koopman operator $\mathcal{K}_t$ and its approximate finite linear equivalent $\mathbf{K}_t$, which is related to the continuous-time Koopman matrix $\mathbf{K}$ by

$$\mathbf{K}_t = e^{(\mathbf{K}\cdot\Delta t)} \tag{12}$$

where Δt is the discrete time-step. DMD methods are fundamentally based on a least-square procedure to find the best fit matrix $\mathbf{K}_t$ which advances a set of states $\mathbf{x} \in \mathbb{R}^n$ forward in time according to

$$\mathbf{x}_{k+1} = \mathbf{K}_t \mathbf{x}_k \ , \tag{13}$$

where $\mathbf{x}_{k+1} = \mathbf{x}(t_k + \Delta t)$. Tu [10] showed that with the *exact* DMD algorithm, all this is needed is a set of data incorporating m number of consecutive state pairs $\{(\mathbf{x}_k, \mathbf{x}_{k+\Delta t})\}_{k=1}^{m}$. Separating the vectors of $\mathbf{x}_k$ and $\mathbf{x}_{k+\Delta t}$ into two matrices $\mathbf{X}$ and $\mathbf{Y}$ respectively, computation of $\mathbf{K}_t$ is described by

$$\mathbf{K}_t = \min_{\mathbf{K}_t} \|\mathbf{Y} - \mathbf{K}_t \mathbf{X}\|_{\mathrm{F}} \tag{14}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. The data need not be ordered or regularly spaced in time, other than the constant Δt between the states of each pair. DMD is therefore a powerful tool for data-driven Koopman operator approximation when the underlying dynamical equations are unknown, and algorithms exist for efficient computation with high-dimensional systems, such as spatially-sampled fluid velocity fields. Particular focus is typically given to the eigendecomposition of $\mathbf{K}$, to identify dominant modes. Extended Dynamic Mode Decomposition (EDMD) furthers this approach by replacing the data set in (13) and (14) with an extended set of observables calculated from the data, $\mathbf{z}(\mathbf{X})$ and $\mathbf{z}(\mathbf{Y})$, which allows closer approximation of the true Koopman operator.

2.3 The Galerkin method

Instead of the data-driven approach in Sec 2.2, a finite approximation to the Koopman operator can be calculated from known underlying dynamic equations. Using the Galerkin method, the original dynamic equations $\mathbf{f}(\mathbf{x})$ are projected onto a set of basis functions $\mathbf{z}(\mathbf{x})$. This projection uses the inner product, which for a dynamic function $f(\mathbf{x})$ and basis function $z(\mathbf{x})$ is defined as

$$\langle f(\mathbf{x}), z(\mathbf{x}) \rangle = \int_{\mathcal{B}} f(\mathbf{x}) z(\mathbf{x}) w(\mathbf{x}) d\mathbf{x} \tag{15}$$

where $\mathcal{B}$ is the domain of the projection, and $w(\mathbf{x})$ is a weighting function. For a finite set of observables $\mathbf{z}(\mathbf{x})$, the error of the approximation in (9) is

$$\mathbf{e}(\mathbf{x}) = \frac{d}{dt}\mathbf{z}(\mathbf{x}) - \mathbf{K}\mathbf{z}(\mathbf{x}) \tag{16}$$

$$= (\nabla_{\mathbf{x}} \mathbf{z}(\mathbf{x}))\, \mathbf{f}(\mathbf{x}) - \mathbf{K}\mathbf{z}(\mathbf{x}) \ , \tag{17}$$

where $\nabla_{\mathbf{x}} = [\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_n}]$. From the Galerkin method, we can state that $\mathbf{e}(\mathbf{x})$ is orthogonal to all basis function $\mathbf{z}(\mathbf{x})$, expressed as

$$\langle \mathbf{z}(\mathbf{x}), \mathbf{e}(\mathbf{x}) \rangle = 0 \ . \tag{18}$$

Expanding (18) and simplifying notation leads to

$$\langle \mathbf{z}, (\nabla_{\mathbf{x}} \mathbf{z})\mathbf{f} - \mathbf{K}\mathbf{z} \rangle = 0 \tag{19}$$

$$\langle \mathbf{z}, \mathbf{K}\mathbf{z} \rangle = \langle \mathbf{z}, (\nabla_{\mathbf{x}} \mathbf{z})\mathbf{f} \rangle \ . \tag{20}$$

We select a set of orthogonal and scaled basis functions such that

$$\langle z_i, z_j \rangle = \begin{cases} 1 & \text{for } \mathrm{i} = \mathrm{j} \\ 0 & \text{for } \mathrm{i} \neq \mathrm{j} \end{cases} \ , \tag{21}$$

which reduces the LHS of (20) so that each element of $\mathbf{K}$ is computed by

$$\mathrm{K}_{ij} = \langle z_i, (\nabla_{\mathbf{x}} z_j)\mathbf{f} \rangle \ . \tag{22}$$

To relate the lifted states to our original states, we use the projection matrix $\mathbf{G}$, where

$$\mathrm{G}_{ij} = \langle x_i, z_j \rangle \ , \tag{23}$$

resulting in

$$\mathbf{x} = \mathbf{G}\mathbf{z} \ . \tag{24}$$

Mezi´c et al. [4] showed that for a similar set of basis functions, the Koopman approximation calculated with DMD approaches the Galerkin solution. Sec. 4.4 introduces methods for efficient computation of (22), which can otherwise quickly become intensive as the size of $\mathbf{x}$ and $\mathbf{z}$ increase.

3 Applications

3.1 Past applications

Past use of the Koopman operator has primarily focussed on spectral analysis of measured or simulated data using DMD, particularly for computational fluid dynamics (CFD) simulations and fluid dynamics experiments. In this application, accurate future-state prediction is of lesser importance, and the Koopman matrix is deliberately reduced to a small set of dominant modes. Conversely, using a Koopman linear system for accurate future state prediction of known systems is a relatively new field and less studied application, with most public literature published in the last three years. Some examples of Koopman lifted linear system of known nonlinear dynamics include:

- Near-space hypersonic vehicle control by Mi et al. [11] (2019).
- Optimisation and uncertainty propagation of an airdrop mission by Leonard et al. [12] (2019).
- A DMD-based linear model of optimal vehicle control by Cibulka et al. [13] (2020).
- Spacecraft trajectory prediction and control by Chen and Shan [14] (2020).
- Solution to orbital zonal harmonits by Arnas and Linares [15] (2021).

Of these past examples which make use of the Galerkin method, specific dynamic systems have been selected to simplify the problem, such as restricting the dynamics to stable systems with smooth functions with closed-form integrals. However, many practical applications of Koopman lifted systems will need to be compatible with any arbitrary dynamics function and non-smooth functions (such as interpolated tabulated data or piecewise functions). This paper describes the extension of the Galerkin method to efficiently capture arbitrary and non-smooth dynamics.

3.2 Future applications

The benefits of approximating nonlinear dynamics with a Koopman linear system are broadly applicable to a vast array of problems. However, the advantages of linearity are particularly suited for aerospace vehicle dynamics, which require fast real-time analysis and high performance control. Some key examples of these benfits are listed below, although many further applications exist with respect to vehicle sub-systems, multi-agent systems, and vehicle tracking and interception.

- Fast and real-time vehicle trajectory prediction, including determining possible future-state envelopes of a trajectory with known constraints. For example, determining the future position envelope of a vehicle with knowledge of maximum thrust, dynamic pressure, and acceleration g-force.
- Using a linear system representation with *parameters* (see Sec. 4.3.1) for:
  - Fast solutions to trajectory initial-value problems, such as vehicle launch conditions, and flight settings (see example in Sec. 5.2.1).
  - Preliminary parametric vehicle design, considering vehicle properties such as mass, aerodynamics, and propulsion.
  - Modelling the response of autonomous control systems (see example in Sec. 5.2.2).
  - Spectral analysis, particularly for autonomous control parameters (such as controller gain). Eigendecomposition of the lifted linear system provides useful information on dynamic modes, frequencies, and stability.
- Using a system representation with bilinear *inputs* (see Sec. 4.3.2) for:
  - Optimal control, making use of bilinear optimisers and sequential quadratic programming to find globally optimum time-varying control inputs. Optimal control inputs may include thrust control and control surface positions. Optimal control with a bilinear system is much faster, more robust, and more likely to find globally optimal values than a nonlinear optimal control problem.
  - Model predictive control (MPC). Real-time MPC for fast-reacting systems requires a linear system model. A lifted-state linear model provides much greater accuracy further from the current operating point than local linearisation, which allows longer MPC time-horizons and better control performance.
- Nonlinear uncertainty propagation. Linear systems are capable of explicit uncertainty propagation using Gaussian operations (such as the working principle of Kalman filters). Nonlinear system uncertainty propagation typically require expensive Monte-Carlo simulations, or local linearisation which reduces accuracy. For example, a lifted linear system can be used to get explicit future-position uncertainty of an object trajectory with nonlinear dynamics.

In general, the computational cost of linear systems is much less than nonlinear systems, and increases favourably for larger systems. As well as permitting explicit calculations without time-integration, linear systems are highly compatible with vectorised and parallel computing, which can be utilised to a much greater extent than with nonlinear systems. Therefore, lifted linear systems offer the possibility to model and optimise larger and more complex systems than possible with nonlinear equivalents.

4 Implementation and development of the Galerkin method

4.1 Basis functions

In this work, each multivariate basis function $z_i(\mathbf{x})$ is the product of univariate Legendre polynomials,

$$z_i(\mathbf{x}) = c_i \prod_{j=1}^{n} \mathcal{P}_{k_j}(x_j), \tag{25}$$

where $\mathcal{P}_k$ is the Legendre polynomial of order k, and c_i is a normalisation parameter to satisfy (21). Legendre polynomials have the benefits of a bounded domain between [-1,1], a constant weighting value $w(\mathbf{x}) = 1$ in (15), and efficient numerical quadrature integration. We construct a set of basis functions $\mathbf{z}(\mathbf{x}) = [z_1(\mathbf{x}), z_2(\mathbf{x}), \ldots, z_m(\mathbf{x})]^{\mathrm{T}}$ to include all functions up to a combined polynomial order $k_{\max}$, expressed as

$$\sum_{j=1}^{n} k_j \leq k_{\max}\,. \tag{26}$$

The number of basis functions is typically much larger than the number of original state variables. Alternative sets of orthogonal functions such as Hermite, Laguerre, and trigonometric polynomials are equally valid. For example, considering a two-state system

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\,, \tag{27}$$

and basis functions up to $k_{\max} = 3$, the univariate Legendre polynomials are

$$\mathcal{P}_0(x) = 1 \tag{28a}$$
$$\mathcal{P}_1(x) = x \tag{28b}$$
$$\mathcal{P}_2(x) = 0.5(3x^2 - 1) \tag{28c}$$
$$\mathcal{P}_3(x) = 0.5(5x^3 - 3x)\,, \tag{28d}$$

and the corresponding 10 basis functions are

$$\mathbf{z}(\mathbf{x}) = \begin{bmatrix} z_0(\mathbf{x}) \\ z_1(\mathbf{x}) \\ z_2(\mathbf{x}) \\ z_3(\mathbf{x}) \\ z_4(\mathbf{x}) \\ z_5(\mathbf{x}) \\ z_6(\mathbf{x}) \\ z_7(\mathbf{x}) \\ z_8(\mathbf{x}) \\ z_9(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} c_0\mathcal{P}_0(x_1)\mathcal{P}_0(x_2) \\ c_1\mathcal{P}_0(x_1)\mathcal{P}_1(x_2) \\ c_2\mathcal{P}_0(x_1)\mathcal{P}_2(x_2) \\ c_3\mathcal{P}_0(x_1)\mathcal{P}_3(x_2) \\ c_1\mathcal{P}_1(x_1)\mathcal{P}_0(x_2) \\ c_4\mathcal{P}_1(x_1)\mathcal{P}_1(x_2) \\ c_5\mathcal{P}_1(x_1)\mathcal{P}_2(x_2) \\ c_2\mathcal{P}_2(x_1)\mathcal{P}_0(x_2) \\ c_5\mathcal{P}_2(x_1)\mathcal{P}_1(x_2) \\ c_3\mathcal{P}_3(x_1)\mathcal{P}_0(x_2) \end{bmatrix} \tag{29}$$

4.2 Domain selection and state-space scaling

Although the nonlinear dynamics exist in the space $\mathbf{x} \in \mathcal{X}$, the lifted states $\mathbf{z}(\mathbf{x})$ are only valid for $\mathbf{x} \in \mathcal{B} \subseteq \mathcal{X}$, where $\mathcal{B}$ is the domain of the basis functions, which for the case of multivariate Legendre polynomials $\mathcal{B} \in [-1, 1]^n$. Typical formulations of flight dynamics equations in terms of physical states (such as position, velocity etc.) are not contained within $\mathcal{B}$, or may not be valid at all in this domain. Therefore, the original unscaled dynamics equations must be scaled with a function $s(\mathbf{x'}) : \mathcal{X} \to \mathcal{B}$, so that $\mathbf{f}(s(\mathbf{x'})) : \mathcal{B} \to \mathcal{B}$, where $\mathbf{x}'$ are the original unscaled states, $\mathcal{X}'$ is the original unscaled domain, and $\mathbf{x} \in \mathcal{X} \subseteq \mathcal{X}'$ is the valid domain of the Koopman approximation. The lifted states are then calculated from the scaled states, $\mathbf{z}(s(\mathbf{x}'))$. In this work we use linear scaling $\mathbf{x} = s(\mathbf{x}') = \mathbf{a}\mathbf{x}' + \mathbf{b}$, where $\mathbf{a}$ and $\mathbf{b}$ are constants, however alternative may include trigonometric or exponential scaling.

4.3 System parameters and inputs

The approach to approximate the Koopman operator described thus far is sufficient to model fixed autonomous systems. However, the utility of a lifted linear system is greatly expanded if we can use a single linear system to model:

1. Parameters $\mathbf{p} = [p_1, p_2, \ldots, p_{n_p}]^{\mathrm{T}}$ that can vary at the system initial state, but do not vary in time. Examples include vehicle mass during a parametric design study, or the gain of an autonomous thrust controller.

2. Inputs $\mathbf{u}(t) = [u_1(t), u_2(t), \ldots, u_{n_u}(t)]^{\mathrm{T}}$ that vary in time. Examples include propulsion throttle, control-surface actuator position, or a time-varying controller setpoint.

This section describes how parameters and inputs can be incorporated into Koopman linear models.

4.3.1 Parameters

Incorporating parameters into the Koopman linear system is relatively simple and without limitation. Each parameter is treated as a system state variable with nill dynamics, and the nonlinear system is augmented to incorporate these extra equations, represented as

$$\frac{d}{dt}\begin{bmatrix}\mathbf{x}_{\mathrm{f}} \\ \mathbf{p}\end{bmatrix} = \begin{bmatrix}\mathbf{f}(\mathbf{x}_{\mathrm{f}}) \\ \mathbf{0}\end{bmatrix} . \tag{30}$$

The process of constructing the Koopman linear system is then otherwise identical, where the basis functions are now $\mathbf{z}(\mathbf{x}, \mathbf{p})$ and $\mathbf{p}$ are scaled to a range of interest similar to state variables (see Sec. 4.2). The lifted state vector $\mathbf{z}$ can be initialised for different values of $\mathbf{p}$, and then modelled in time using the same linear system

$$\frac{d}{dt}\mathbf{z}(\mathbf{x}, \mathbf{p}) = \mathbf{K}\mathbf{z}(\mathbf{x}, \mathbf{p}) \tag{31}$$

4.3.2 Inputs

Time-varying inputs can be incorporated into the linear model with some limitations. The trivial solution is to describe the nonlinear system as an augmented system with inputs (similar to parameters), and re-initialise the lifted state vector $\mathbf{z}(\mathbf{x}, \mathbf{u})$ at each time-step. However, this would negate most advantages of a linear system, and likely provide little speed benefit over integrating the original nonlinear equations.

Alternatively, it has been shown by Bruder et al. that a control-affine nonlinear system can be described accurately with a bilinear Koopman linear representation. A control-affine nonlinear system can be expressed as

$$\frac{d}{dt}\mathbf{x(t)} = \mathbf{f}(\mathbf{x}(t)) + \mathbf{B}_x\mathbf{u}(t) \ , \tag{32}$$

where $\mathbf{B}_x$ is a linear matrix. We can construct a set of basis functions suitable for a bilinear lifted system as

$$\mathbf{z}_u(\mathbf{x},\mathbf{u}) = \Big[[\mathbf{z}(\mathbf{x})],\ [cu_1\mathbf{z}(\mathbf{x})],\ [cu_2\mathbf{z}(\mathbf{x})],\ldots,\ [cu_{n_u}\mathbf{z}(\mathbf{x})]\Big]^{\mathrm{T}} \tag{33}$$

where c is a normalisation parameter to satisfy (5) for any two functions in $\mathbf{z}_u$. The lifted linear system is then calculated as

$$\frac{d}{dt}\begin{bmatrix} \mathbf{z}(\mathbf{x}) \\ cu_1\mathbf{z}(\mathbf{x}) \\ cu_2\mathbf{z}(\mathbf{x}) \\ \vdots \\ cu_{n_u}\mathbf{z}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} [\mathbf{K}_z] & [\mathbf{B}_{u_1}] & [\mathbf{B}_{u_2}] & \cdots & [\mathbf{B}_{u_{n_u}}] \\ & & - & & \\ & & - & & \\ & & \vdots & & \\ & & - & & \end{bmatrix} \begin{bmatrix} \mathbf{z}(\mathbf{x}) \\ cu_1\mathbf{z}(\mathbf{x}) \\ cu_2\mathbf{z}(\mathbf{x}) \\ \vdots \\ cu_{n_u}\mathbf{z}(\mathbf{x}) \end{bmatrix} \ , \tag{34}$$

which can be re-written in terms of $\mathbf{z}(\mathbf{x})$ as the bilinear system

$$\frac{d}{dt}\mathbf{z}(\mathbf{x}) = \mathbf{K}_z\mathbf{z}(\mathbf{x}) + c\sum_{i=1}^{n_u}(\mathbf{B}_{u_i}u_i)\ \mathbf{z}(\mathbf{x}) \ . \tag{35}$$

Although (35) is still nonlinear, a significant body of past literature has focussed on efficient algorithms to optimise bilinear systems, including sequential quadratic programming, and sequential linear-quadratic programming. Moreover, Bruder et al. showed that efficient linear model predictive control (MPC) can function accurately using the assumption that $\mathbf{z}(\mathbf{x})$ in the bilinear term is constant over the predictive time horizon.

4.4 Efficient analytic computation of the Koopman matrix

Using the Galerkin method, each cell of the Koopman matrix $\mathbf{K}$ must be computed individually by

$$\mathrm{K}_{ij} = \langle z_i, (\nabla_{\mathbf{x}} z_j)\mathbf{f}\rangle \ , \tag{36}$$

which expands to

$$\mathrm{K}_{ij} = \int_{\mathcal{B}} z_i \sum_{k=1}^{n} \frac{\partial z_j}{\partial x_k} f_k \ d\mathbf{x} \ . \tag{37}$$

In this paper, we numerically evaluate this integral using Gauss-Legendre quadrature, which does not restrict the dynamics functions to closed-form solutions of (37), and allows the process to be automated. However, considering the high number of quadrature points necessary to evaluate multivariate integrals and the large dimensions of $\mathbf{K}$, we make use of various techniques to avoid an otherwise burdensome computational time. Two of the most advantageous methods to improve computational speed are described below.

First, we split the integral of (37) into

$$\mathrm{K}_{ij} = \sum_{k=1}^{n} \int_{\mathcal{B}} z_i \frac{\partial z_j}{\partial x_k} f_k \ d\mathbf{x} \ . \tag{38}$$

Knowing that each basis function is composed of univariate orthogonal polynomials

$$z_i = c_i \prod_{a=0}^{n} \mathcal{P}_{i_a}(x_a) \ , \tag{39}$$

and the arguments of each dynamic function are a subset of the state vector,

$$f_k(\mathbf{x}_{fk}), \ \mathbf{x}_{fk} \subseteq \mathbf{x}, \tag{40}$$

we can identify combinations of orthogonal functions in the integral, as well as terms which are zero, using

$$\int_{\mathcal{B}} z_i \frac{\partial z_j}{\partial x_k} f_k \ d\mathbf{x} \ = 0 \text{ for } \begin{cases} \text{any } i_a \neq j_a \ \wedge \ a \neq k \ \wedge \ x_a \notin \mathbf{x}_{fk}, & \text{or} \\ j_k = 0 \ \wedge \ x_k \notin \mathbf{x}_{xk} \ , \end{cases} \tag{41}$$

where a are the polynomial orders from (39). Using this method we can quickly identify that most of the integrals in (38) are zero.

Second, we observe that dynamic functions and basis functions need to be evaluated at the quadrature points once only to construct the $\mathbf{K}$ matrix. Moreover, for basis functions constructed using the product of univariate functions (such as the Legendre polynomials in this work), each univariate function need only be evaluated once. After initial evaluation, each non-zero integral term in (41) can be computed by the tensor dot products of quadrature values. For example, consider a three-state nonlinear system where $\mathbf{x} = [x_1, x_2, x_3]^{\mathrm{T}}$. The first term ($k = 1$) in the sum from (38) is

$$\int_{\mathcal{B}} z_i \frac{\partial z_j}{\partial x_1} f_1 \ d\mathbf{x} \ . \tag{42}$$

Assume the dynamics function $f_1(x_1, x_2)$ is only a function of x_1 and x_2, and the two basis functions z_i and z_j are described by

$$z_i = c_i \ \mathcal{P}_1(x_1) \ \mathcal{P}_1(x_2) \ \mathcal{P}_1(x_3) \ , \tag{43}$$

$$z_j = c_j \ \mathcal{P}_2(x_2) \ \mathcal{P}_2(x_2) \ \mathcal{P}_1(x_3) \ . \tag{44}$$

We can expand the integral to

$$\int_{\mathcal{B}} z_i \frac{\partial z_j}{\partial x_1} f_k \ d\mathbf{x} = c_i c_j \int_{\mathcal{B}} \mathcal{P}_1(x_1) \frac{d\mathcal{P}_2(x_1)}{dx_1} \ \mathcal{P}_1(x_2)\mathcal{P}_2(x_2) \ \mathcal{P}_1(x_3)\mathcal{P}_1(x_3) \ f_1(x_1, x_2) \ d\mathbf{x} \ . \tag{45}$$

We know from (21) that

$$\int_{\mathcal{B}} \mathcal{P}_1(x_3)\mathcal{P}_1(x_3) \ dx_3 = 1 \ . \tag{46}$$

To compute the remainder of the integral, we use tensors of the dynamics function and Legendre polynomials evaluated as the quadrature points $\mathbf{Q} = [q_1, q_2, \ldots, q_p]$. The tensor for f_1 is

$$f_1 \to \mathbf{F}_1 = \begin{bmatrix} f_1(q_1, q_1) & \ldots & f_1(q_p, q_1) \\ \vdots & \ddots & \vdots \\ f_1(q_1, q_p) & \ldots & f_1(q_p, q_p) \end{bmatrix} \ , \tag{47}$$

and the similar-dimensioned Legendre polynomial tensors are

$$\mathcal{P}_1 \to \mathbf{P}_1 = \begin{bmatrix} \mathcal{P}_1(q_1) & \dots & \mathcal{P}_1(q_p) \\ \vdots & \ddots & \vdots \\ \mathcal{P}_1(q_1) & \dots & \mathcal{P}_1(q_p) \end{bmatrix} , \tag{48}$$

$$\frac{d\mathcal{P}_1}{dt} \to \mathbf{P'}_1 = \begin{bmatrix} \mathcal{P}'_1(q_1) & \dots & \mathcal{P}'_1(q_p) \\ \vdots & \ddots & \vdots \\ \mathcal{P}'_1(q_1) & \dots & \mathcal{P}'_1(q_p) \end{bmatrix} , \tag{49}$$

$$\mathcal{P}_2 \to \mathbf{P}_2 = \begin{bmatrix} \mathcal{P}_2(q_1) & \dots & \mathcal{P}_2(q_p) \\ \vdots & \ddots & \vdots \\ \mathcal{P}_2(q_1) & \dots & \mathcal{P}_2(q_p) \end{bmatrix} . \tag{50}$$

The integral can then be computed using the tensor dot product

$$\int_{\mathcal{B}} z_i \frac{\partial z_j}{\partial x_1} f_k \, d\mathbf{x} = c_i c_j \sum \left(\mathbf{P}_1 \cdot \mathbf{P}'_1 \cdot \mathbf{P}_1^{\mathrm{T}} \cdot \mathbf{P}_2^{\mathrm{T}} \cdot \mathbf{F}_1 \cdot \mathbf{W} \right) , \tag{51}$$

where $\mathbf{W}$ is a tensor of corresponding quadrature weights. This method can be expanded to any number of variables and corresponding tensor dimensions. The Legendre polynomials, their derivatives, and the system dynamic functions are evaluated once only, and for each integration the Legendre polynomial tensors are rotated to match the variable axes of the dynamics function, and extended uniformly in additional tensor dimensions. Using this integration method provided significant speed increases in the current work, but also importantly demonstrates the suitability for future speed optimisation using graphical processing units (GPUs) which are highly suited to tensor operations.

5 Demonstration and discussion

5.1 Test case: Duffing oscillator

We first demonstrate the application of the Galerkin method for computing the Koopman matrix on a damped duffing oscillator, governed by

$$\frac{d^2}{dt^2}x + \delta \frac{d}{dt}x + \alpha x + \beta x^3 = 0 , \tag{52}$$

where the damping coefficient $\delta = 0.5$, and the stationary points are determined by $\alpha x + \beta x^3 = 0$. Eq. 52 is formulated as a nonlinear system of two state variables $x_1 = x$, and $x_2 = \frac{d}{dt}x$,

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\delta x_2 - \alpha x_1 - \beta x_1^3 \end{bmatrix} . \tag{53}$$

We consider the domain $x_1 \in [-2, 2]$ and $x_2 \in [-2, 2]$. This simple test case is useful to demonstrate the performance and limitations of the Galerkin method. We separate the case of a single attraction basin (Sec. 5.1.1), to the case of multiple stationary points and attraction basins (Sec. 5.1.2). A stationary point is defined as

$$\frac{d}{dt}\mathbf{x} = 0 , \tag{54}$$

and an attraction basin describes the state-spate region for which any the system either tends towards a single stationary point as $t \to \infty$, or infinitely oscillates within the basin.

5.1.1 Single attraction basin

First, we let α , $\beta = 1$ resulting in a global attracting stationary point at $x = 0$, as shown in Figure 1. Figure 2 shows the Koopman linear approximation for basis functions with maximum

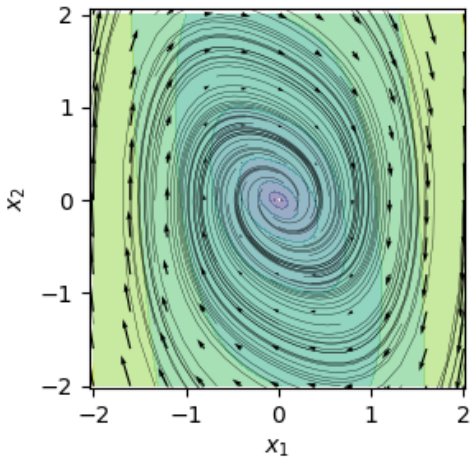


Figure 1: Damped duffing oscillator gradient field with a single attractor. $x_1 = x$, $x_2 = \frac{dx}{dt}$.

polynomial orders of 3, 5, and 7, which correspond to 10, 21 and 36 lifted states respectively. Figure 3 shows the corresponding eigenvalues for the discrete-time flow matrix, with $\Delta t = 1$ s. The linear approximation approaches the true solution with an increasing number of lifted states. All three linear models are stable with real eigenvalues within the unit circle, and all converge to the stationary point of $x = 0$.

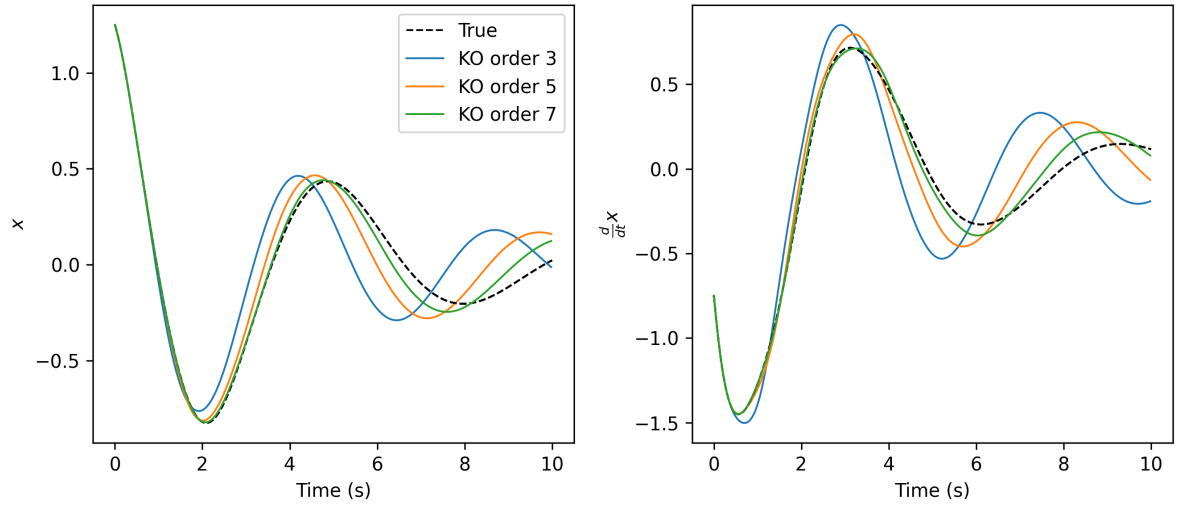
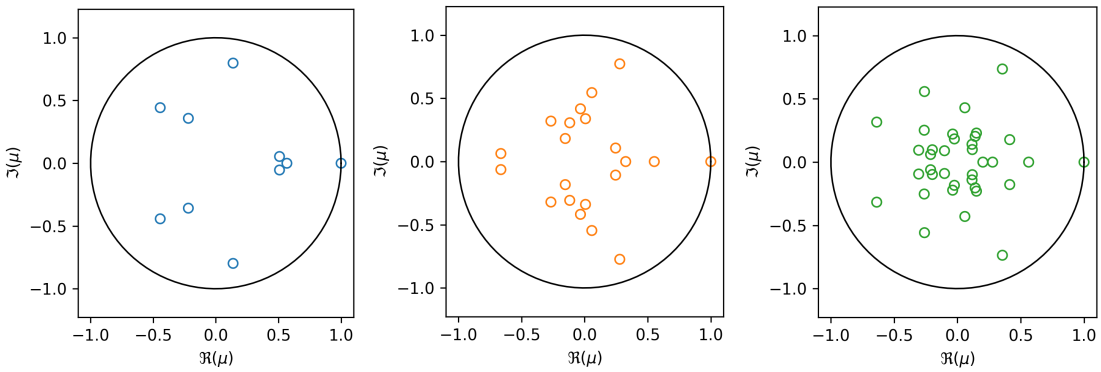


Figure 2: Comparison of linear approximation to a Duffing oscillator for three orders of basis functions. The dynamics function has a global attractor at $x = 0$.

Mezi´c et al. [4] proved that with increasing sample points, the linear approximation derived through EDMD converges to the Galerkin Koopman approximation described in Sec. 2.3. We demonstrate that again here for the damped duffing oscillator with $\alpha = 1$, $\beta = 1$. Training points for the EDMD model are selected using a pseudo-random hypercube method. Figure 4 shows the progression of the function for increasing numbers of EDMD training points, which are seen to converge to the Galerkin Koopman solution.



(a) Order 3 (b) Order 5 (c) Order 7

Figure 3: Eigenvalues of the discrete-time Koopman matrix representing a Duffing oscillator with a single attracting basin, for three basis function orders.

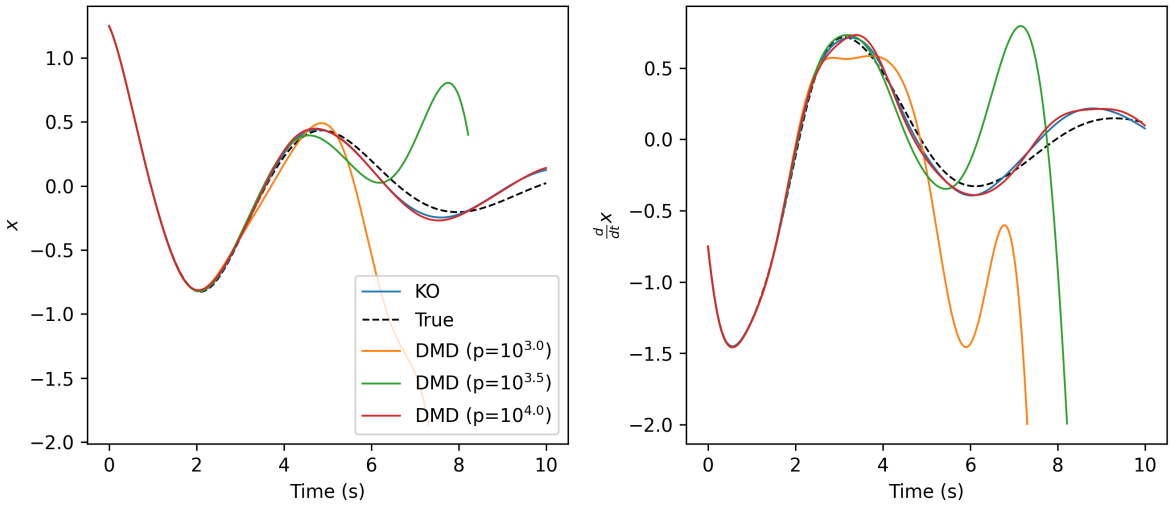


Figure 4: Comparison of Galerkin and EDMD linear approximation to a Duffing oscillator with basis functions up to order 7, and varying quantity of of EDMD sample points.

5.1.2 Multiple attraction basins

A finite linear system can never perfectly represent a nonlinear system with multiple attraction basins, and approximating the Koopman operator over multiple basins significantly reduces its accuracy. To demonstrate this, we consider the Duffing oscillator, (52), with values $\alpha = -1$ and $\beta = 1$, so that there are two attracting stationary points of $x = -1$ and $x = 1$, and a non-attracting stationary point at the origin, as shown in Figure 5. Figure 6 shows the Koopman

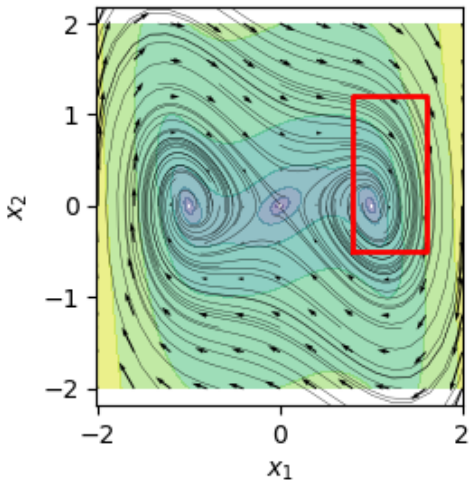


Figure 5: Subspace of a duffing oscillator with multiple attractor basins. The red box indicates a subspace within a single basin.

linear approximations for three orders of basis functions, which are all significantly worse than the case of a single stationary point in Figure 2. All linear model tracks the true function until approximately $t = 1$, where divergence becomes apparent. Figure 7 shows the eigenvalues for the approximation using seventh order basis functions, where it is evident the linear approximation is unstable, and will eventually diverge.

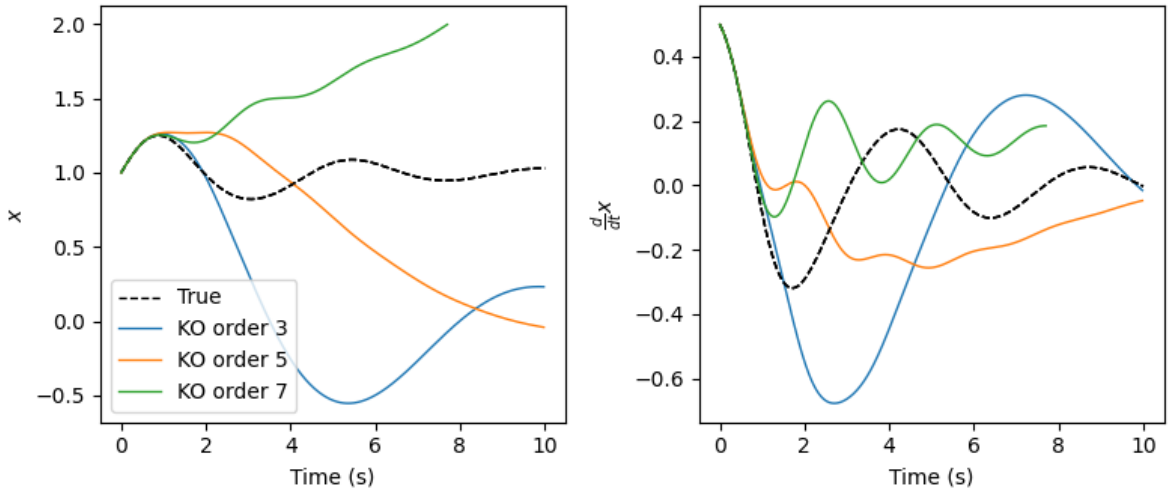


Figure 6: Comparison of linear approximation to a Duffing oscillator for three orders of basis functions. The Duffing oscillator has two attracting stationary points at at $x = 1$ and $x = -1$.

The easiest way to resolve this issue and improve accuracy is to restrict the domain to a subspace which contains a single attraction region, shown by the red rectangle in Figure 5.

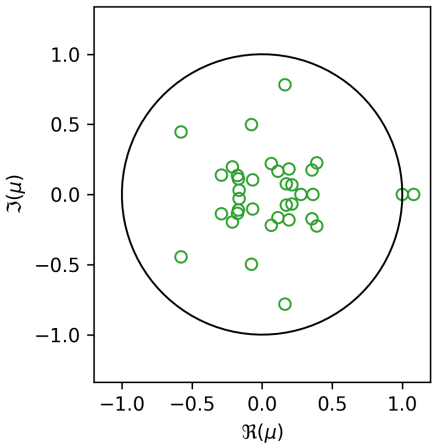


Figure 7: Eigenvalues of the linear approximation to the Duffing oscillator with basis functions up to order 7. The Duffing oscillator has two attracting stationary points at $x = 1$ and $x = -1$.

Computing the linear system for this subspace results in a far more accurate representation of the nonlinear system, as shown in Figure 8. However, this requires knowledge of the nonlinear attraction basin boundaries within the function, which may not be obvious for many systems. An alternative method is to utilise multiple short-horizon predictions using the linear model

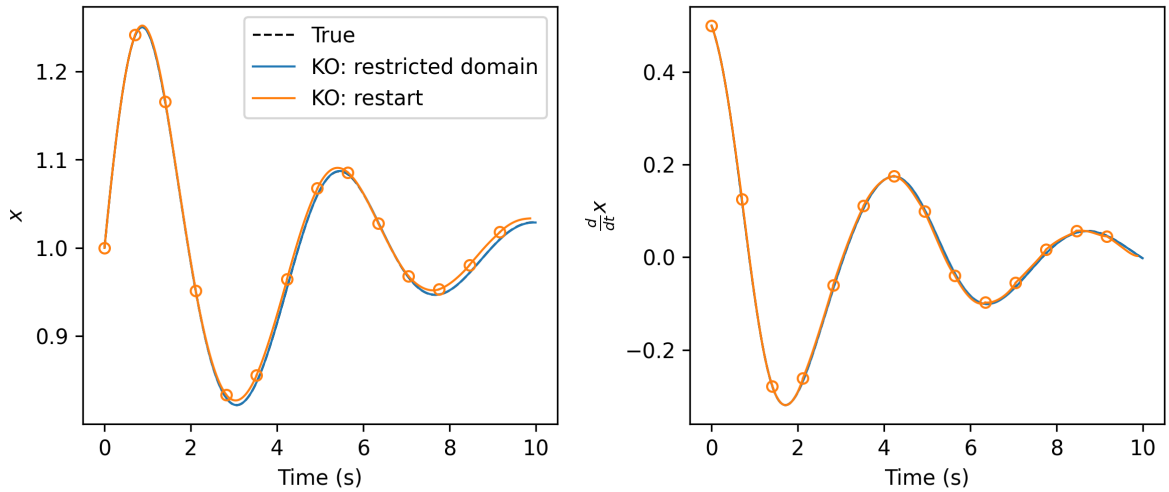


Figure 8: Methods to handle dynamic functions with multiple stationary points: (1) restrict the domain to a subspace with a single attractor; or (b) limit the linear model prediction horizon to a small value and re-initialise the lifted states, marked with circles.

which encompasses the entire domain. After each linear prediction, the lifted states are re-initialised from the basis functions as $\mathbf{z}(\mathbf{G} \cdot \mathbf{z})$. This is demonstrated in Figure 8, where circles represent the points of re-initialization, and shows much improved accuracy. However, this reduces the utility of the lifted system, in particular due to the nonlinear nature of the basis functions. Additional proposed methods to improve the ability of Koopman linear systems are described in Sec. 5.3.1.

5.2 Glider trajectory

5.2.1 Model with parameters

The Koopman linear approximations is applied to the case of an trimmed three degrees of freedom glider, which has 6 fundamental state variables: position coordinates in the inertial frame x, y, z, velocity magnitude v, flight path angle χ, and heading angle γ. A trimmed flight model assumes angle of attack, α, and bank (roll) angle, ϕ, are system inputs, and are autonomously achieved by adjustments to the vehicle's control surfaces. Additionally, it is assumed that zero net moment forces are acting on the vehicle for a constant α and ϕ. Thus, Euler rotation dynamics are not modelled, which simplifies the system for the current demonstration purpose; this example could equally be applied to a full six degree of freedom system (with increased computation time). We incorporate angle of attack and bank angle as parameters $\mathbf{p} = [\alpha, \phi]^{\mathrm{T}}$, so the augmented system dynamics is governed by

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \\ v \\ \chi \\ \gamma \\ \alpha \\ \phi \end{bmatrix} = \begin{bmatrix} \cos(\gamma)\cos(\chi)v \\ \cos(\gamma)\sin(\chi)v \\ -\sin(\gamma)v \\ f_1/m - sin(\gamma)g_z \\ (1/\cos(\gamma)V)(f_2/m + \cos(\chi)g_z) \\ (1/V)(f_3/m + \sin(\gamma)\sin(\chi)g_z) \\ 0 \\ 0 \end{bmatrix}, \tag{55}$$

where the body forces in vehicle velocity coordinates are

$$f_1 = 0.5\rho V^2 C_{\mathrm{d}}(\alpha) A \tag{56}$$

$$f_2 = 0.5\rho V^2 C_{\mathrm{l}}(\alpha)\sin(\phi) A \tag{57}$$

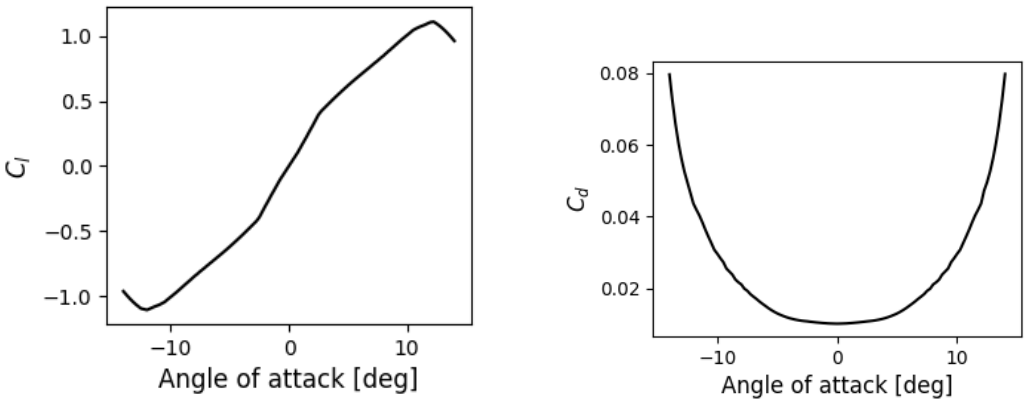
$$f_3 = 0.5\rho V^2 C_{\mathrm{l}}(\alpha)\cos(\phi) A, \tag{58}$$

and A is representative area. Defining vehicle motion in polar coordinates allows the domain to be restricted to $v > 0$, avoiding the stationary singularity otherwise encountered with Cartesian velocity coordinates. Lift and drag coefficients, C_{l} and C_{d}, are based of the NACA 0012 airfoil at a Reynolds number of 2×10^5, calculated by Xfoil [17] and are a function of α, shown in Figure 9. Atmospheric density ρ is determined using the piecewise-linear 1976 U.S. Standard Atmosphere model [18] as a function of elevation.

$$\rho = f(-z) \tag{59}$$

$$\tag{60}$$

For altitudes up to 11 km, this is a single linear relationship. Figure 10 shows the trajectory of the glider when released with the initial conditions presented in Table 1. Four different angles of attack are modelled with the same linear system, demonstrating the utility of the augmented system with parameters. Basis functions up to order 5 are used for the linear system, resulting in 792 lifted states (although some are unused). The nonlinear system does not have a global stationary point due to the varying atmospheric density. However, the linear model still closely follows the true solution for about 100 seconds, before some divergence between the two is apparent.



(a) Lift (b) Drag

Figure 9: Aerodynamic lift and drag coefficients.

Table 1: Initial conditions for uncontrolled glider trajectory in Figure 10.

Parameter	**Initial value**
Altitude	11 km
Velocity	160 m/s
Heading angle (χ)	0 °
Flight path angle (γ)	0 °
Angle of attack (α)	5 - 12.5 °
Bank angle (ϕ)	2.85 °

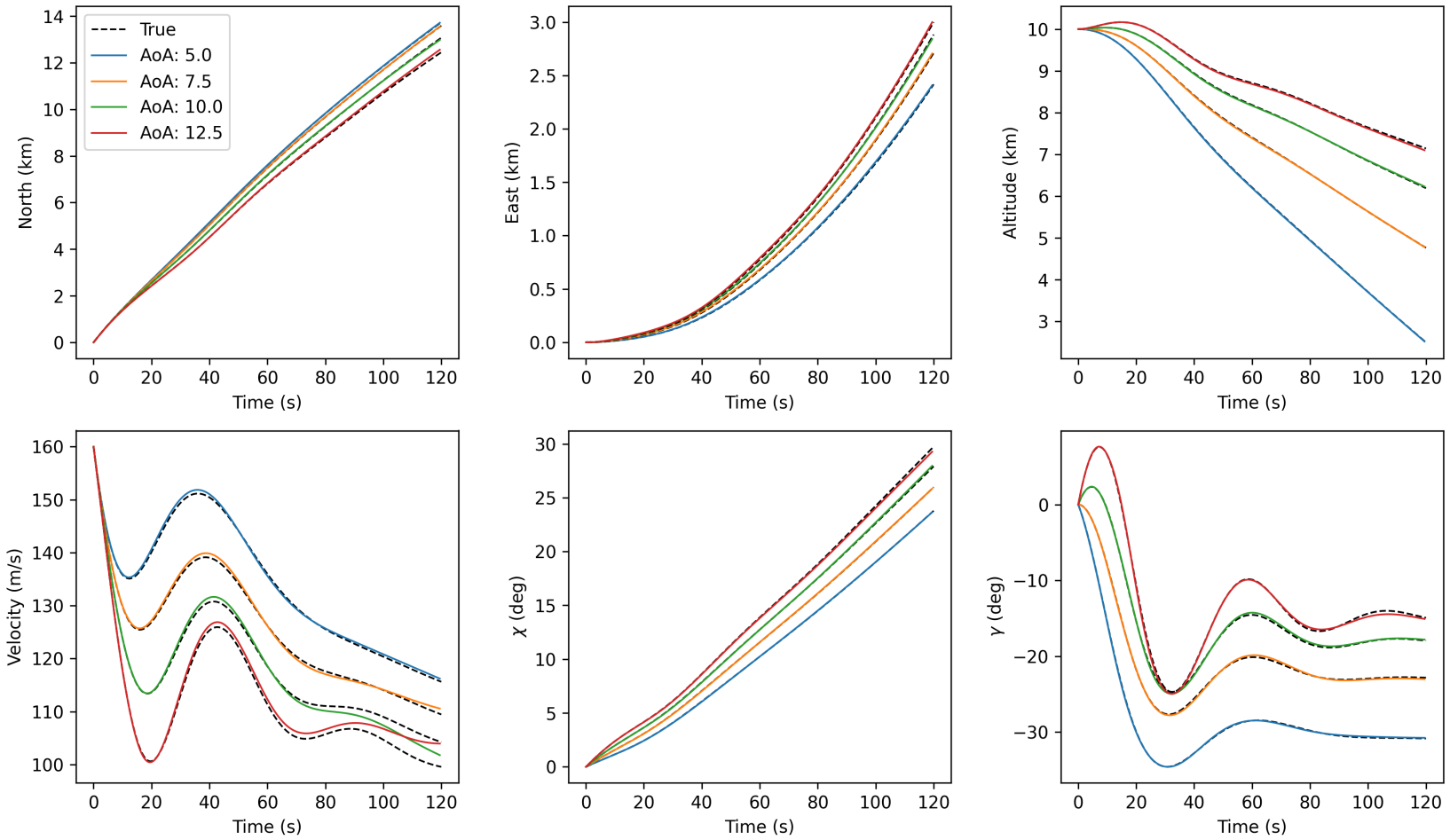


Figure 10: Linear approximations to a glider trajectory with varying angles of attack. All angles of attack are modelled with the same augmented linear system.

5.2.2 Model with autonomous control

To demonstrate the ability to add controllers to the linearised system, we now replace the angle of attack with a proportional-integral controller of the form

$$\alpha = G_{\mathrm{P}}(v_{\mathrm{sp}} - v) + G_{\mathrm{I}} \int_0^t (v_{\mathrm{sp}} - v)\ dt = G_{\mathrm{P}}(v_{\mathrm{sp}} - v) + G_{\mathrm{I}} I_c\ , \tag{61}$$

with the dynamics system

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \\ v \\ \chi \\ \gamma \\ I_c \\ \phi \end{bmatrix} = \begin{bmatrix} \cos(\gamma)\cos(\chi)v \\ \cos(\gamma)\sin(\chi)v \\ -\sin(\gamma)v \\ f_1/m - sin(\gamma)g_z \\ (1/\cos(\gamma)V)(f_2/m + \cos(\chi)g_z) \\ (1/V)(f_3/m + \sin(\gamma)\sin(\chi)g_z) \\ v_{\mathrm{sp}} - v \\ 0 \end{bmatrix}\ , \tag{62}$$

where v_{sp} is the velocity setpoint, and G_{P} and G_{I} are proportional and integral gains respectively. Figure 11 shows the glider trajectory with a deliberately overactive controller, in order demonstrate the ability of the linear system to capture its effect. Initial conditions of the simulation are presented in Table 2.

Table 2: Initial conditions for glider with autonomous velocity control in Figure 11.

Parameter	**Initial value**
Altitude	10 km
Velocity	140 m/s
Heading angle (χ)	0 rad
Flight path angle (γ)	0 rad
Bank angle (ϕ)	2.85 °
Control integral term	0

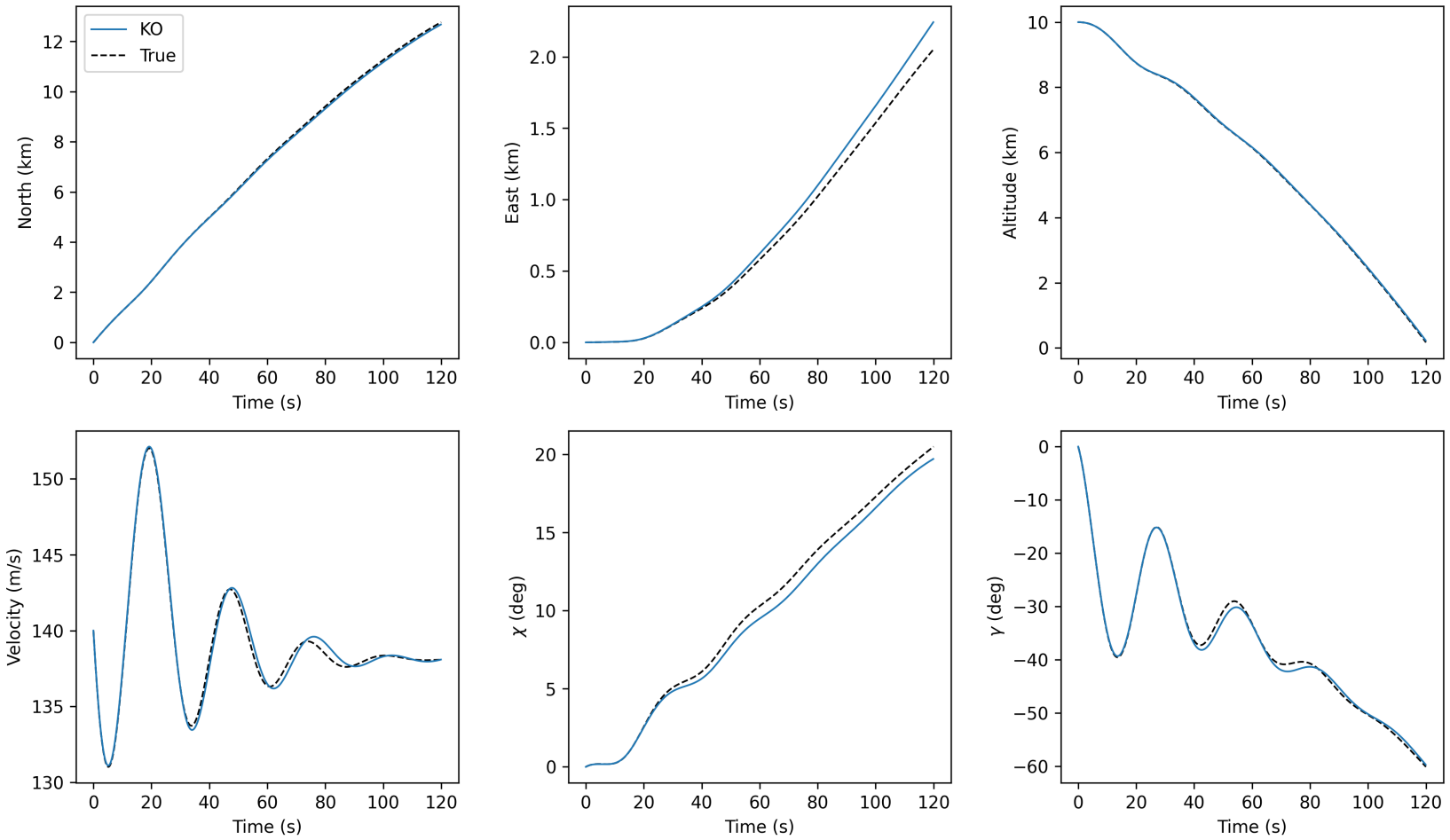


Figure 11: Linear approximations to a glider trajectory with an angle of attack controller.

5.3 Discussion

5.3.1 State-space domain

In contrast to local linearisation which is only accurate around a small region, a Koopman linear system may have an exact finite solution for an entire basin of attraction around a stationary point. Nevertheless, Koopman linear systems are still limited by the fundamental linear system characteristic of either a single stationary point at the origin, or a subspace of infinite stationary points. The nonlinear state-space domain $\mathcal{X} \in \mathcal{X}'$ chosen for linearisation (see Sec. 4.2) can therefore have a strong impact on the accuracy of the Koopman linear system approximation, particularly with respect to whether $\mathcal{X}$ is contained within one or more attraction basins. Although a Koopman linear system is not capable of providing an exact solution over multiple basins, it may still provide an approximate solution valid for short time horizon predictions, although spectral properties of such a linear system will be inaccurate, as demonstrated in Sec. 5.1.2. One approach to extend the size of attraction basins is to reformulate the dynamics. For example, systems with conserved energy can be described by their Hamiltonian, which provides a global attraction basin, often at the expense of some information of the system. Alternatively, the Riccati equation may be used to extend the attraction basin for models incorporating control.

Even when $\mathcal{X}$ is contained within a single attracting basin, a trade-off is encountered between the size of $\mathcal{X}$, the number of lifted states, and Koopman approximation accuracy. Some regions of $\mathcal{X}'$ may contain highly nonlinear dynamics requiring high order basis functions, whereas other regions can be modelled accurately with lower order functions. Multiple different linearised domains and systems may be combined to represent different state-space domains. For example we may define two subspaces $\mathcal{X}_1,\ \mathcal{X}_2 \in \mathcal{X}'$, where $\mathcal{X}_1$ may span vehicle low-acceleration cruise dynamics with higher accuracy (and longer time-horizon accuracy), while $\mathcal{X}_2$ may span a larger space encompassing high-acceleration states, but with poorer long-term accuracy.

A limitation of the current Galerkin method state-space domain selection is that state-space domains $\mathcal{X}$ must be n-dimensional rectangular. That is, the domain is scaled by restricting the range of each original state x_i between some lower and upper bounds, requiring that the dynamics associated with all 'corners' of the state-space must be captured by the linear system, even if they are irrelevant. For example, we may be interested in vehicle velocity between 100-600 m/s, and an altitude range between 0-20 kms. The vehicle state at the extremes of 100 m/s at 20 kms altitude, or 600 m/s at ground level cannot be achieved in reality result in high-order gradients which must be captured by the linear system at the expense of accuracy at more typical operating states (e.g. 300 m/s at 10 kms). Importantly, these 'corners' of the state-space domain often extend outside an attraction basin, and significantly affect linear system accuracy. This effect can be minimised by reformulating the dynamics into more meaningful state variables; instead of velocity, the state vector $\mathbf{x}$ can include Mach number which is more meaningful at various altitudes. However, this does not fully resolve the issue, and some states variables such as angle of attack or heading angle have no obvious substitution. At a theoretical level, there is no impediment to restrict the subspace $\mathcal{X}$ to any arbitrary and non-rectangular domain of $\mathcal{X}'$ when implementing the Galerkin method. Indeed, inherent choice of state-space domain is one of the benefits of DMD, for which the training data set can only incorporate data points from the relevant region of $\mathcal{X}'$. Incorporating non-rectangular domains in the Galerkin method is a strongly suggested direction for future work, and should yield large improvements in the applicability and performance of derived linear models.

5.3.2 Linear system accuracy

So far we have discussed the accuracy of Koopman linear systems qualitatively. We may often need a quantitative understanding of upper bounds of this error, to inform the level of confidence

placed on linear model results. The error of a lifted linear system across a discrete time-step $e_{\mathbf{z},\Delta t}$ can be estimated manually by mass sampling states across the domain and calculating as

$$e_{\mathbf{z},\Delta t} = \frac{1}{p}\sum_{i=1}^{p}\left[\mathbf{K}_t\mathbf{z}(\mathbf{x}_k) - \mathbf{z}(\mathbf{x}_{k+1})\right] \tag{63}$$

where p is a large number of distributed sample points. In addition, there already exists a body of literature for a priori estimates of Galerkin projections, which can be utilised before computing the linear matrix $\mathbf{K}$ to inform the choice of the corresponding basis functions and state-space domain. However, we are primarily interest in the error of the original state vector $\mathbf{x}$ rather than $\mathbf{z}(\mathbf{x})$, and the relationship between the two for varying discrete time-steps is not apparent. As small errors $e_{\mathbf{z},\Delta t}$ are introduced, the lifted state falls off the manifold $\mathbf{z}(\mathbf{x})$ with unpredictable consequences for the error in $\mathbf{x}$. During future development of the Galerkin method, attempts should be made to estimate the upper bounds error in $\mathbf{x}$, which may provide useful insights on such things as choice of basis function type and order, state-space domain, and prediction time-horizon in order to achieve acceptable accuracy.

6 Conclusion and recommendations

6.1 Conclusion

Koopman operator theory presents a framework for constructing finite linear approximations to nonlinear systems, by projecting the nonlinear dynamics onto a Hilbert space constructed of Koopman operator eigenfunctions. Koopman linear approximations have historically been used for spectral analysis of unknown dynamics, such as fluid turbulence, using dynamic mode decomposition (DMD). However, recent work suggests Koopman linear systems can also provide accurate lifted linear approximations to known dynamical systems constructed using Galerkin projections. Lifted linear systems offer many potential benefits including but not limited to explicit future state prediction, efficient parametric studies, spectral and stability analysis, and control optimisation.

This work summarises the theory underpinning the Galerkin method of constructing Koopman linear approximations of known dynamics using polynomial basis functions. Details of a system-agnostic program for computing the Koopman linear matrix is presented. New methods to significantly optimise the Galerkin method calculation using numerical techniques are described, which offer a pathway to efficiently construct linear systems of any arbitrary dynamical system. Linear approximations are constructed for two nonlinear systems: a simple Duffing oscillator, and a three degree-of-freedom glider trajectory. The ability of linear systems to closely follow the time-integrated nonlinear solution is presented, and an example demonstrates a linear model capable of incorporating variable system parameters. The paper discusses methods to incorporate bilinear time-varying inputs, allowing for optimal control calculation, as well as methods to avoid the inherent linear system limitation of a singular discrete stationary point.

Lifted linear systems are particularly suited to vehicle dynamics and trajectory problems, where real-time future state prediction and high-performance control are required. Applications include vehicle future-state envelope calculations, parametric initial-value solutions, and optimal time-varying control inputs. Linear computations are highly compatibility with vectorised and parallel computing, resulting in favourable time scaling for complex systems compared to nonlinear equivalents. Therefore, further development of this technique may offer designers of aerospace systems a distinct advantage to construct fast and optimal systems for system design and control.

6.2 Future work

Throughout this paper, many directions to further develop the theory, capability, and application of Koopman linear systems are proposed. In no particular order, these include:

1. Optimal control of lifted linear systems with bilinear time-varying inputs using model predictive control (MPC) and sequential quadratic programming (SQP);
2. Explicit uncertainty propagation of lifted linear systems using linear Gaussian operations;
3. Spectral decomposition of lifted linear systems to study stability of autonomous vehicle controllers;
4. Methods to construct linear systems of any arbitrary state-space domain;
5. Investigation of alternative basis function types, including trigonometric basis functions;
6. A priori estimates of Galerkin system projection errors;
7. Methods to reformulating nonlinear vehicle dynamics to a system with a single stationary point; and
8. Extending lifted linear systems to multi-agent systems.

Acknowledgements

The researchers would like to acknowledge the funding and support received from Northrop Grumman Corporation and Northrop Grumman Australia to complete this work.

References

[1] B O Koopman and V J VNeumann. Dynamical Systems of Continuous Spectra. *Proc. Natl. Acad. Sci.*, 18(3):255–263, mar 1932. doi: 10.1073/pnas.18.3.255. URL `https://doi.org/10.1073/pnas.18.3.255`.

[2] J v Neumann. Zur Operatorenmethode in der klassischen Mechanik. *Ann. Math.*, pages 587–642, 1932. ISSN 0003-486X.

[3] Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. Modern Koopman theory for dynamical systems. *arXiv Prepr. arXiv2102.12086*, 2021.

[4] Igor Mezi´c Mezi´c, Milan Korda, Igor Mezi´c Mezi´c, and Milan Korda. On Convergence of Extended Dynamic Mode Decomposition to the Koopman Operator. Technical report, 2017.

[5] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[6] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009. ISSN 1469-7645.

[7] Igor Mezić. On Numerical Approximations of the Koopman Operator. *Mathematics*, 10(7): 1180, apr 2022. ISSN 2227-7390. doi: 10.3390/math10071180. URL `https://www.mdpi.com/2227-7390/10/7/1180`.

[8] Peter J Schmid, Larry Li, Matthew P Juniper, and O Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1):249–259, 2011. ISSN 1432-2250.

[9] Peter J Schmid. Application of the dynamic mode decomposition to experimental data. *Experiments in fluids*, 50(4):1123–1130, 2011. ISSN 1432-1114.

[10] Jonathan H Tu. *Dynamic Mode Decomposition: Theory and Applications*. PhD thesis, Princeton University, 2013.

[11] Peichao Mi, Qingxian Wu, and Yuhui Wang. Suboptimal control law for a near-space hypersonic vehicle based on Koopman operator and algebraic Riccati equation. 0(0):1–17, 2019. doi: 10.1177/09544100211045594.

[12] Andrew Leonard, Jonathan Rogers, and Adam Gerlach. Koopman Operator Approach to Airdrop Mission Planning Under Uncertainty. 2019. doi: 10.2514/1.G004277. URL `http://arc.aiaa.org`.

[13] Vít Cibulka, Tomáš Haniš, Milan Korda, and Martin Hromčík. Model Predictive Control of a Vehicle using Koopman Operator. *IFAC PapersOnLine*, 53(2):4228–4233, 2020. doi: 10.1016/j.ifacol.2020.12.2469. URL `www.sciencedirect.com`.

[14] Ti Chen and Jinjun Shan. Koopman-operator-based attitude dynamics and control on SO3. *Journal of Guidance, Control, and Dynamics*, 43(11):2112–2126, 2020. ISSN 15333884. doi: 10.2514/1.G005006.

[15] David Arnas and Richard Linares. Approximate Analytical Solution to the Zonal Harmonics Problem Using Koopman Operator Theory. *J. Guid. Control. Dyn.*, 44(11):1909–1923, nov 2021. ISSN 1533-3884. doi: 10.2514/1.G005864. URL `https://arc.aiaa.org/doi/10.2514/1.G005864`.

[16] Daniel Bruder, Xun Fu, and Ram Vasudevan. Advantages of Bilinear Koopman Realizations for the Modeling and Control of Systems with Unknown Dynamics. Technical report.

[17] Mark Drela. XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils BT - Low Reynolds Number Aerodynamics. pages 1–12, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg. ISBN 978-3-642-84010-4.

[18] National Oceanic and Atmospheric Administration, United States Air Force, and National Aeronautics and Space Administration. U.S. Standard Atmosphere 1976. Technical report, 1976.