Koopman Operators in Robot Learning

Lu Shi,[1,7] Masih Haseli,[2] Giorgos Mamakoukas,[3] Daniel Bruder,[4] Ian Abraham,[5]
Todd Murphey,[6] Jorge Cortés,[2] and Konstantinos Karydis[1]

Abstract—**Koopman operator theory offers a rigorous treatment of dynamics and has been emerging as a powerful modeling and learning-based control method enabling significant advancements across various domains of robotics. Due to its ability to represent nonlinear dynamics as a linear operator, Koopman theory offers a fresh lens through which to understand and tackle the modeling and control of complex robotic systems. Moreover, it enables incremental updates and is computationally inexpensive making it particularly appealing for real-time applications and online active learning. This review comprehensively presents recent research results on advancing Koopman operator theory across diverse domains of robotics, encompassing aerial, legged, wheeled, underwater, soft, and manipulator robotics. Furthermore, it offers practical tutorials to help new users get started as well as a treatise of more advanced topics leading to an outlook on future directions and open research questions. Taken together, these provide insights into the potential evolution of Koopman theory as applied to the field of robotics.**

Index Terms—**Koopman operator theory, robotics, modeling, control, learning.**

I. INTRODUCTION

RUNTIME learning is an open challenge in robotics. The wealth of robot learning methods being actively investigated (for example, neural ODEs [1], [deep] reinforcement learning [2], and generative AI [3]) all rely on large amounts of data collected in an offline manner, mostly via high-fidelity simulators. However, real-world deployment requires the robots to operate in environments that are often *novel*, characterized by phenomena not anticipated by *a priori* datasets, and incorporating interactions that are *unsimulable*. These three foundational characteristics transcend robotics applications. For example, human-robot interaction (HRI) [4] may take place in an engineered environment, but it may be impossible to model people based on first-principle statements or average characteristics such as those obtained from large datasets. Similarly, wholly physical systems—e.g., soft matter [5], systems in turbulent flow [6], [7], mechanoreception

[1]Dept. of Electrical and Computer Engineering, University of California, Riverside, [2]Dept. of Mechanical and Aerospace Engineering, University of California San Diego, [3]Dept. of Planning and Control, Zoox Inc., [4]Dept. of Mechanical Engineering, University of Michigan, [5]Dept. of Mechanical Engineering, Yale University, [6]Dept. of Mechanical Engineering, Northwestern University, [7]Institute of AI Industry Research (AIR), Tsinghua University. Emails: shilu@air.tsinghua.edu.cn, kkarydis@ucr.edu, {mhaseli, cortes}@ucsd.edu, t-murphey@northwestern.edu, ian.abraham@yale.edu, bruderd@umich.edu, giorgosmamakoukas@u.northwestern.edu.
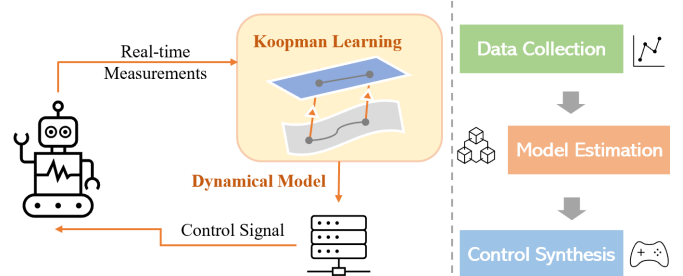
Fig. 1. Left: Overview of the Koopman-based **runtime learning**, modeling, and control of robots. Right: Key components of the implementation.

of touch [8]—may not be meaningfully "simulatable" based on first principles with current technology, either because the complexity of such a simulation is impractical (e.g., necessitating to solve complex partial differential equations [e.g., Navier-Stokes]) or because the system has both unknown and unknowable parameters and boundary conditions (e.g., predictive mechanics of mechanoreception in haptics); in some cases, first principles may not even be readily available (e.g., robots interacting with people with unknown intent). Other cases include wearable technologies [9]–[11], and robotics in unstructured and dynamic environments [12], [13].

The emphasis in the last decade on deep learning has contributed to reliance on 'big data' in robotics [14], [15], driven by slow algorithms operating on offline data that have been labeled at scale (e.g., the massive availability of labeled images, contributing to the prioritization of vision as a sensing modality in robotics). In contrast, our effort herein centers on the following motivating question. *If robots—with their physical bodies and embodied learning constraints—are going to function in novel settings without relying on offline data, what tools are available that are appropriate for runtime learning using only 'small data?'*

Koopman operators provide a partial answer to this question. The main way they are integrated within core robotics foundational research (learning, modeling, and control) is summarized in Fig. 1. The essential ingredients in Koopman operators are an infinite basis of observables, a linear operator predicting the evolution of those observables, and some method for approximating the operator using finite calculations. These ingredients are sufficient to represent any nonlinear system, including nonsmooth and discontinuous ones [16]. These operators have only recently made their way into the field of robotics, both theoretically and experimentally. Machine learning (data-driven) techniques using Koopman operators are suited to robotics needs—both in terms of **empirical properties** and **formal properties**. Empirically, Koopman operators only require sparse data sets, making them

amenable to runtime computation using only small datasets. Moreover, this enables complex adaptive behavior (e.g., spontaneous control response to unmodeled turbulent flow). Formal properties include applying physically relevant properties to otherwise unknown models (e.g., stability, invariance, or symmetry), the ability to directly calculate information measures for continuous active learning, the benefit of applying linear control techniques (e.g., LQR) to nonlinear systems and the ease of certifying stability (e.g., constructive Control Lyapunov Functions [17]), and guarantees on sample efficiency. These benefits collectively contribute data-driven techniques that can be implemented synchronously with robotic execution, rather than in an offline batch processing manner.

The emergence of Koopman-based models presents a potential paradigm shift, offering a unified framework for modeling nonlinear dynamics by providing more efficient and versatile representations of robotic behaviors and dynamics that have the following distinct advantages.

1) **Interpretability**: Unlike many machine learning approaches that represent the input-output relationship as a black box neural network (NN), the Koopman operator theory provides dynamical model descriptions of the underlying system rooted at principled geometric and algebraic properties that can be leveraged to explain the performance of data-driven approximations.

2) **Data-efficiency**: The Koopman operator theory demands only a limited number of measurements when compared to most of the NN-based methods, making it suitable for real-time implementation.

3) **Linear Representation**: The Koopman operator leads to a linear representation of the dynamics, enabling the use of linear systems tools and methods for analysis, estimation, and control synthesis.

As Koopman operator theory continues to rapidly evolve, several studies have delved into algorithmic enhancements and validation techniques for modeling and control purposes. While the main body of relevant research results started to appear only within the past five years, it is rapidly growing. A literature search with the specific keywords "Koopman" and "Robot*" yields over 17,800 results on Google Scholar.[1] The publications with the most relevance and their number of citations are shown in Fig. 2. The graphs clearly illustrate the increasing trend in this area. While existing review articles have extensively discussed data-driven methodologies [18] and operator-based algorithms [19], recent review articles on Koopman operator theory have predominantly focused on theoretical analyses [20]–[22], encompassing topics such as operator estimation, spectral properties analysis, and controller design [23]. More recently, there has been a surge in reviews specific to soft robotics [24]. Here, our goal is to offer a comprehensive introduction to the Koopman operator, complete with a tutorial featuring executable code for implementing different procedures, along with recent advancements in its application within the field of robotics. We aim to elucidate the benefits of Koopman-based approaches for tasks such as modeling, state estimation, and control. Additionally, we seek
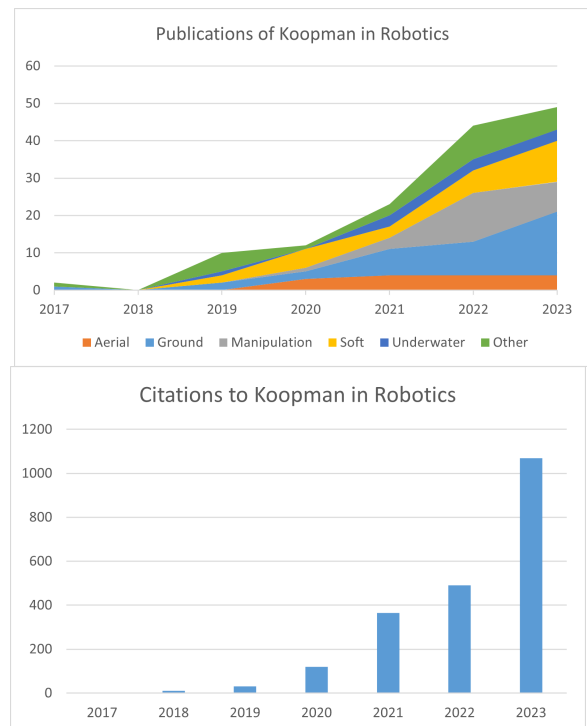

Fig. 2. Literature review results of the Koopman operator theory in robotics with application instances.

to consolidate and highlight recent trends in research efforts related to the practical utilization of Koopman operator theory across various robotic systems.

The review is structured to be accessible to both newcomers and those with prior experience in Koopman and robotics. The recommended reading roadmap is outlined in Fig. 3. Specifically, in Sections II and III, we provide an overview of the Koopman operator along with a practical tutorial, augmented by code available online.[2] This serves as a quick reference for those new to the field, allowing them to grasp the basic concepts of how Koopman operator theory functions and how to implement it for robot modeling and control. Readers can choose to stop here and experiment with the provided code with their systems. Then, the interested reader may proceed to Section VI to explore state-of-the-art applications across various robotic platforms, such as aerial robots, ground robots, robot arms, underwater robots, soft robots, and others, including examples in multi-agent systems. For readers seeking more in-depth understanding, Section IV goes into theoretical details of the Koopman operator, covering aspects like Koopman operators in discrete-time and continuous-time systems, Koopman eigenfunctions as well as Koopman invariant spaces, data-driven approximation approaches, and strategies for handling systems with inputs. Continuing to Section V, we present the implementation details of the Koopman operator in robotic systems. This encompasses guidance on data collection, selection of lifting functions, and controller design. In the concluding Section VII of the review, we provide a summary of the covered materials and discuss several open challenges and potential future directions in the field.

---

[1] Data are current up to April 24th, 2024.
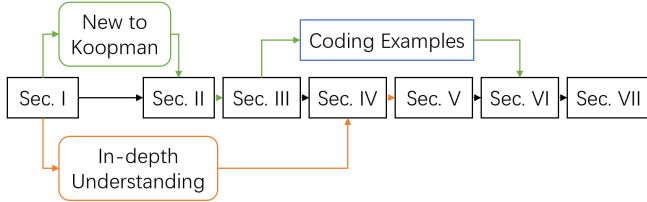
[2] The code is available on Github.

Fig. 3. The illustration of **recommended reading sequence**. For readers who are new to the Koopman operator in robotics, we recommend following the green line to read this review paper. In addition, interested readers who seek to learn more about this area are suggested to follow the orange line.

II. THE KOOPMAN OPERATOR AT A GLANCE

A. The Koopman Operator

Let $x \in \mathcal{X} \subseteq \mathbb{R}^{N_x}$ be the state vector for a discrete-time dynamical system. The system's propagation rule is represented by the nonlinear function T (Fig. 4). Through the process of lifting the original states x using the observable functions $x \mapsto g(x)$, the system's dynamics can be redefined in a new space where a linear operator, denoted as the Koopman operator $\mathcal{K}$, describes the dynamics. In other words, although T and $\mathcal{K}$ act on different spaces, they encapsulate the same dynamics. For example, given the current state x, one can propagate it to the subsequent time step and observe it through two pathways: either by employing T to compute $T(x)$ and observe the resultant state (the bottom route) or by utilizing an observable function, applying $\mathcal{K}$ and evaluating it at x (the top route). The concept of the "equivalence" or "substitution" offers several advantages. (1) It enables a global linear representation of the nonlinear dynamics T, thus enabling the application of techniques designed for linear systems. (2) It facilitates the estimation of the underlying dynamics by learning the linear operator in real time, thus eliminating the need for least-square regression of the nonlinear function, which in general requires a large amount of data.

B. Data-driven Prediction via Koopman Theory

Consider a discrete-time system[3] expressed as

$$x_{t+1} = T(x_t) \ . \tag{1}$$

Consider a vector space $\mathcal{F}$ of complex-valued functions (known as observables) with the system's state space as their domain. One can think of the observables $g \in \mathcal{F}$ as *mapped* or *lifted* states. The evolution of the observables g using the Koopman operator $\mathcal{K}: \mathcal{F} \to \mathcal{F}$ associated with system (1) is

$$\mathcal{K}g = g \circ T, \quad \forall g \in \mathcal{F} \ ,$$

where $\circ$ denotes function composition. To ensure $\mathcal{K}$ is well defined, $\mathcal{F}$ must be closed under composition with T. This condition might force $\mathcal{F}$ to be infinite-dimensional if one insists that $\mathcal{F}$ contains specific predefined functions (such as functions that return the full state values).

The Koopman operator propagates the system one step forward in time according to

$$\mathcal{K}g(x_t) = g \circ T(x_t) = g(T(x_t)) = g(x_{t+1}) \ . \tag{2}$$

---

[3]One can also describe similar constructions for continuous-time dynamical systems, but we delay its presentation to Section IV.
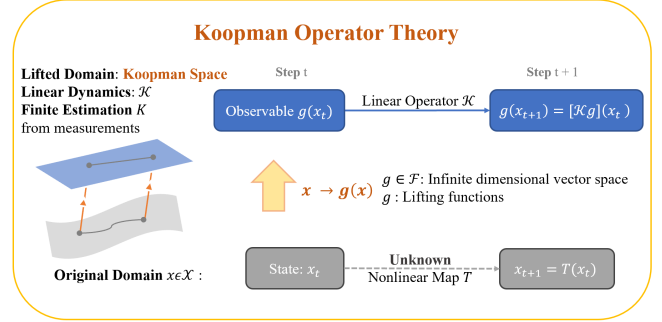

Fig. 4. A glance at the Koopman Operator Theory.

The lifted states $g(x)$ essentially transform the original (nonlinear) system into a linear one evolving at the Koopman space which can afford the design and implementation of a variety of linear controllers (such as LQR).

To avoid dealing with infinite-dimensional spaces, a finite-dimensional representation of the Koopman operator can be employed instead. Thus, finite-dimensional subspaces of $\mathcal{F}$ that are invariant under the Koopman operator play a crucial role. Formally, let $\mathcal{S} \subset \mathcal{F}$ be a finite-dimensional Koopman-invariant subspace and restrict the action of the Koopman operator to $\mathcal{S}$ as $\mathcal{K}{\restriction}_{\mathcal{S}}: \mathcal{S} \to \mathcal{S}$. Since $\mathcal{S}$ is finite-dimensional, given any basis for it, one can represent the operator $\mathcal{K}{\restriction}_{\mathcal{S}}$ by a matrix. Formally, let $\boldsymbol{\Psi}$ be a vector-valued function whose elements form a basis for $\mathcal{S}$, then there exists a matrix $K \in \mathbb{C}^{\dim(\mathcal{S}) \times \dim(\mathcal{S})}$ such that

$$\mathcal{K}\boldsymbol{\Psi} = \boldsymbol{\Psi} \circ T = K\boldsymbol{\Psi} \ , \tag{3}$$

where in the first term, the Koopman operator acts elements-wise on $\boldsymbol{\Psi}$. Combing (3) and (2) yields a linear evolution of system trajectories, thus enabling the use of methods from linear systems theory for (1), that is

$$\boldsymbol{\Psi}(x_{t+1}) = K\boldsymbol{\Psi}(x_t) \ . \tag{4}$$

By defining $z_t := \boldsymbol{\Psi}(x_t)$, (4) can be written as $z_{t+1} = Kz_t$ which is in linear form. If the space $\mathcal{S}$ contains the state observables $g_i(x) = x^i$, where x^i is the ith element of state x, then one can choose the basis $\boldsymbol{\Psi}$ in a way that contains g_i's as its elements.[4] In this case, the lifted linear system (4) captures the complete information of the original nonlinear system (1). Moreover, the eigendecomposition of matrix K allows one to identify the Koopman eigenfunctions and eigenvalues. We refer the reader to Section IV for more details.

It is generally challenging (sometimes impossible) to find exact finite-dimensional invariant subspaces that contain all the state observables. A popular way to tackle this issue is via approximations. Suppose the space $\mathcal{F}$ is equipped with an inner product, given any arbitrary space $\mathcal{S}$ (not necessarily invariant under the Koopman operator). In that case, one can consider the operator $\mathcal{P}_{\mathcal{S}}\mathcal{K} : \mathcal{F} \to \mathcal{F}$, where $\mathcal{P}_{\mathcal{S}}$ is the orthogonal projection operator on $\mathcal{S}$. For this operator, the space $\mathcal{S}$ is invariant by construction. Therefore, we can restrict

---

[4] This is referred to as the full-state observability assumption [25]. More generally, state observables can be written as a linear combination of elements of $\boldsymbol{\Psi}$, if they are contained in $\mathcal{S}$.

the action of this operator to $\mathcal{S}$ as $\mathcal{P}_\mathcal{S}\mathcal{K}\!\restriction_\mathcal{S}\colon \mathcal{S} \to \mathcal{S}$ and given a basis $\boldsymbol{\Psi}$ for $\mathcal{S}$, the operator admits a matrix representation

$$\mathcal{P}_\mathcal{S}\mathcal{K}\boldsymbol{\Psi} = \hat{K}\boldsymbol{\Psi}\,,$$

where $\hat{K} \in \mathbb{C}^{\dim(\mathcal{S}) \times \dim(\mathcal{S})}$. In this case, one can write an approximate version of (4) as follows

$$\boldsymbol{\Psi}(x_{t+1}) = [\mathcal{K}\boldsymbol{\Psi}](x_t) \approx [\mathcal{P}_\mathcal{S}\mathcal{K}\boldsymbol{\Psi}](x_t) = \hat{K}\boldsymbol{\Psi}(x_t)\,. \quad (5)$$

By defining $z_t := \boldsymbol{\Psi}(x_t)$, the previous equation can be written in an approximate linear form $z_{t+1} \approx \hat{K}z_t$, enabling the use of efficient linear methods to approximate the system's behavior.

In practice, we typically have access to discretely sampled measurement data of the system. This can then be used to obtain a finite-dimensional approximated matrix of the infinite-dimensional Koopman operator and these Koopman-based components. Of all the approximation approaches, one of the most popular algorithms is the Extended Dynamic Mode Decomposition (EDMD) approach [25]. In particular, if $\mathcal{F}$ is the L_2 space based on the empirical measure on the data set, the procedure above corresponds to EDMD, which allows one to compute the matrices K and $\hat{K}$ solely based on data by solving least-squares problems (see Section IV-B for details).

The Koopman operator theory, along with its data-driven estimation approach EDMD, was initially formulated without accounting for inputs. Several ways have been proposed to include control inputs. For example, control inputs can be integrated into the definition of $\boldsymbol{\Psi}$ by treating them as an augmented state [26]. Then, the process described above can be employed similarly for estimating the system $x_{t+1} = T(x_t, u_t)$. A detailed discussion on how to deal with this case is provided in Section IV-C.

III. IMPLEMENTATION AT A GLANCE

This section summarizes how to train a Koopman-based model and use it for control synthesis. There are mainly three steps: data collection (from physical experiments or simulations), model training, and control synthesis as shown in Algorithm 1. Next, we present a high-level overview of each step; more details are also provided in Section V.

The quality of the training data greatly impacts the ability to learn an accurate representation of the underlying system and, subsequently, apply effective control. The aim is to collect sequences of state and control measurements that allow one to learn how an initial state and control input map to the next state: $(x_t, u_t) \mapsto x_{t+1}$. To learn a globally accurate representation, training data must sample the entire state and control spaces $\mathcal{X}$ and $\mathcal{U}$, respectively. While this is practically infeasible, one should strive to capture training data that cover as much of the operating space of the robot as possible. A non-uniform data distribution will bias the learned model toward the more frequently observed states in the training dataset.

Given a training dataset, one can learn a Koopman model after selecting the basis functions as well as the optimization criteria and parameters. The basis functions determine the structure of the Koopman representation (i.e. linear, bilinear, nonlinear) [27] and can be either composed based on a pre-determined dictionary of functions or learned (e.g., via a neural
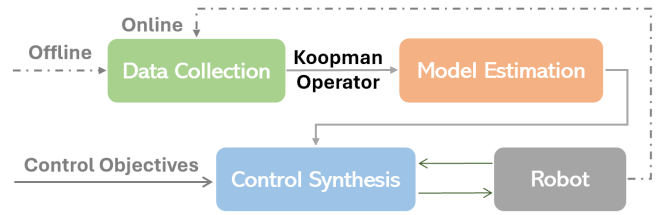

Fig. 5. Overall structure of the modeling and control of robotic systems with the Koopman operator theory.

Algorithm 1 Implementation Algorithm

Data Collection:

- *States and Inputs Determination*: Based on the control objective and sensors available on the system, determine the state and input.
- *Data Collection*: Collect data with approaches like those discussed in Section V-A.
- *Data Preprocessing (as required)*: Perform data curation: normalize when the magnitude among different states varies too much; filter and/or average to remove noise; interpolate between measurements to ensure consistent temporal spacing across all training samples, etc.

Model training (Online or Offline):

- *Lifting Function Selection*: select lifting functions, e.g., polynomials or NN-based approaches, etc. Refer to Sec. V-B for ways to define the dictionary of observables.
- *Koopman operator estimation*: Utilize X, U to estimate the Koopman operator K.
- *Koopman eigenfunctions calculation (Optional)*: obtain the Koopman eigenfunction of the operator for analysis if needed.
- *Compute Koopman Mode, eigenvalue and eigenvector expressions (Optional)*

Control Synthesis

- *Design the Controller*: design control structure, e.g., LQR, based on the estimated Koopman model.
- *Feedback Signal Collection (if applied)*: obtain real-time observations from the robot.
- *Obtain Control Signal*: calculate the control input to drive the robotic system to achieve the control objective.

---

network). The optimization, too, can differ and, besides optimizing for modeling error, it can include regularization terms, constraints that enforce, or cost terms that promote model properties (i.e. stability). Modeling errors can be assessed by different metrics, and this choice affects the answer one obtains, as well as the computational complexity of solving the corresponding optimization. The most direct implementation involves a least-squares regression solution that can be computed efficiently in real-time. After training a Koopman model, and depending on the structure of the dynamics, one can deploy appropriate feedback control methods. For example, linear Koopman models are used with LQR or linear MPC, and bilinear Koopman models can pair up with nonlinear MPC.

An implementation overview of Koopman operator theory in robotic systems is included in the supplementary tutorial

(https://shorturl.at/ouE59). We also recommend investigating the impact of different dictionary selections and experimenting with various approaches to estimate the Koopman operator using the code provided by [28]. Additionally, designing and implementing different controllers can be explored using examples provided in [29], while the scripts used in [30], [31] offer insights into active data collection.

IV. THE KOOPMAN OPERATOR AND ITS APPROXIMATION

We present the Koopman operator for both discrete- and continuous-time dynamical systems and present data-driven approaches to approximate it with finite-dimensional representations. Key notation is summarized in Table I.

TABLE I
KEY NOTATION USED IN THE REVIEW.

Notation	Description
x	State
u	Input
N_x	Dimension of state x
N_u	Dimension of input u
t	Index for online system propagation
g	Observable function
$\mathcal{K}$	Koopman operator of discrete-time system
$\{\mathcal{K}^t\}$	Koopman operator group of continuous-time system
K_{EDMD}	Approximated Koopman operator via EDMD
$\mathcal{L}_G$	Koopman generator
$\mathfrak{P}_{\mathcal{K}f}^{\text{EDMD}}$	EDMD Predictor
ϕ	Koopman eigenfunction
λ_ϕ	Koopman eigenvalue w.r.t ϕ
v_ϕ	Left eigenvector of K_{EDMD}
N	Dimension of observables' dictionary; Size of estimated Koopman operator
ψ_n	The n-th lifting function
$\mathbf{\Psi}$	Vector-valued observation

A. Koopman Operator

We briefly review the Koopman operator for discrete- and continuous-time systems following [32], [33].

1) Discrete-Time Case: Consider the discrete-time system

$$x_{t+1} = T(x_t), \quad x \in \mathcal{X} \subseteq \mathbb{R}^{N_x} \ . \tag{6}$$

Let $\mathcal{F}$ be a vector space defined over the field $\mathbb{C}$, comprised of complex-valued functions with domain $\mathcal{X}$. Assume $\mathcal{F}$ is closed under the composition with T, i.e. $g \circ T \in \mathcal{F}$ for all $g \in \mathcal{F}$. Then, we define the Koopman operator $\mathcal{K} : \mathcal{F} \to \mathcal{F}$ associated with system (6) as

$$\mathcal{K}g = g \circ T, \quad \forall g \in \mathcal{F} \ . \tag{7}$$

This definition readily implies that $\mathcal{K}$ is a linear operator, i.e. $\mathcal{K}(c_1 g_1 + c_2 g_2) = c_1 \mathcal{K} g_1 + c_2 \mathcal{K} g_2$, for all $c_1, c_2 \in \mathbb{C}$ and $g_1, g_2 \in \mathcal{F}$. The linearity of the operator naturally leads to the concept of spectrum. Specifically, the point spectrum of the Koopman operator and its eigenfunctions play an important role in system analysis. The nonzero function $\phi \in \mathcal{F}$ is an eigenfunction of $\mathcal{K}$ with eigenvalue $\lambda_\phi \in \mathbb{C}$ if

$$\mathcal{K}\phi = \lambda_\phi \phi \ . \tag{8}$$

It is crucial to observe that Koopman eigenfunctions evolve according to linear difference equations on the system trajectories, i.e. $\phi(x_{t+1}) = \phi \circ T(x_t) = \mathcal{K}\phi(x_t) = \lambda_\phi \phi(x_t)$.

Since the space $\mathcal{F}$ is generally infinite dimensional, implementing the Koopman operator's action on digital hardware is not feasible. To tackle this issue, finding finite-dimensional representations for the operator is of utmost importance. This leads to the concept of finite-dimensional invariant subspaces under the operator. Formally, $\mathcal{L} \subset \mathcal{F}$ is an invariant subspace under the Koopman operator if $\mathcal{K}f \in \mathcal{L}$ for all $f \in \mathcal{L}$. If $\mathcal{L}$ is finite-dimensional, one can take a basis $\mathbf{\Psi}$ for it and describe the operator's action (and consequently the system's dynamics) by a matrix K as

$$\mathbf{\Psi}(x^+) = \mathbf{\Psi} \circ T(x) = K\mathbf{\Psi}(x) \ . \tag{9}$$

Note the parallelism between the composition with T in this equation and the definition of the operator in (7).

2) Continuous-Time Case: Let us now consider the system

$$\dot{x} = G(x), \quad x \in \mathcal{X} \subseteq \mathbb{R}^{N_x} \ , \tag{10}$$

where the map G is continuously differentiable. For $t \in \mathbb{R}_{\geq 0}$, let $\mathcal{G}^t : \mathcal{X} \to \mathcal{X}$ be the associated flow map defined as

$$\mathcal{G}^t(x(0)) := x(0) + \int_{\tau=0}^t G(x(\tau))d\tau \ , \tag{11}$$

for all initial conditions $x(0) \in \mathcal{X}$. We assume the flow map (11) is well-defined for all $t \in \mathbb{R}_{\geq 0}$, i.e. it is a complete flow.

Similarly to the discrete-time case, consider the vector space $\mathcal{F}$ defined over the field $\mathbb{C}$, comprised of complex-valued functions defined on the domain $\mathcal{X} \subseteq \mathbb{R}^{N_x}$. Assume $\mathcal{F}$ is closed under composition with the flow map $\mathcal{G}^t$, for all $t \in \mathbb{R}_{\geq 0}$. Then, for each $t \in \mathbb{R}_{\geq 0}$, we can define a Koopman operator $\mathcal{K}^t : \mathcal{F} \to \mathcal{F}$, similarly to (7), as

$$\mathcal{K}^t f = f \circ \mathcal{G}^t, \quad \forall f \in \mathcal{F} \ . \tag{12}$$

If $\mathcal{F}$ is a Banach space with norm $\| \cdot \|$ and the family of operators $\{\mathcal{K}^t\}_{t \in \mathbb{R}_{\geq 0}}$ is a strongly continuous semi-group,[5] i.e. it satisfies

1) $\mathcal{K}^0 = \text{id}$,
2) $\mathcal{K}^{t_1+t_2} = \mathcal{K}^{t_1}\mathcal{K}^{t_2}$, for all $t_1, t_2 \in \mathbb{R}_{\geq 0}$,
3) $\lim_{t \searrow 0} \|\mathcal{K}^t f - f\| = 0$, for all $f \in \mathcal{F}$,

where id is the identity operator on $\mathcal{F}$, then we can define the infinitesimal generator $\mathcal{L}_G : \mathcal{F} \to \mathcal{F}$ of the semi-group $\{\mathcal{K}^t\}_{t \in \mathbb{R}_{\geq 0}}$ as

$$\mathcal{L}_G f := \lim_{t \searrow 0} \frac{\mathcal{K}^t f - f}{t} = G \cdot \nabla f, \quad \forall f \in \mathcal{F} \ , \tag{13}$$

where $\cdot$ and ∇ represent the dot product and the gradient operator respectively. $\mathcal{L}_G$ is often referred to as the *Koopman generator*.[6] By convention, we define the eigenfunctions of the Koopman semi-group slightly differently from the typical definition. A nonzero function $\phi \in \mathcal{F}$ is an eigenfunction of the semi-group $\{\mathcal{K}^t\}_{t \in \mathbb{R}_{\geq 0}}$ with eigenvalue $\lambda_\phi \in \mathbb{C}$ if

$$\mathcal{K}^t \phi = e^{\lambda_\phi t}\phi, \quad \forall t \in \mathbb{R}_{\geq 0} \ . \tag{14}$$

---

[5] See for example [33, Chapter 1] for specific choices of G and $\mathcal{F}$ that turn $\{\mathcal{K}^t\}_{t \in \mathbb{R}_{\geq 0}}$ into a strongly continuous semi-group.

[6] The definition of Koopman generator can be slightly relaxed to hold on a dense subset of $\mathcal{F}$ [33, Chapter 1].

For strongly continuous semi-group of operators, given the Koopman generator, (14) leads to the equivalent definition

$$\mathcal{L}_G \phi = \lambda_\phi \phi \ . \tag{15}$$

It is important to observe the similarity between (15) and (8). The concept of invariant subspaces for the continuous case is then analogous to the discrete-time case.

B. Extended Dynamic Mode Decomposition (EDMD)

In practice, we have access to discrete measurement data of the system to obtain a finite-dimensional approximated matrix of the infinite-dimensional Koopman operator. Extended Dynamic Mode Decomposition (EDMD) approach [25] is among the most widely-used approximation methods.

Consider data matrices taken from system (6) as

$$X = [x_1, x_2, \ldots, x_M] \ ,$$
$$Y = [y_1, y_2, \ldots, y_M] \ ,$$

where $y_i = T(x_i)$ for all $i \in \{1, \ldots, M\}$; we can also use the state history $Y = [x_2, \ldots, x_M, x_{M+1}]$. We need a dictionary of functions that lift state variables to the higher-dimensional space where the observable dynamics is approximately linear. Define the dictionary $\boldsymbol{\Psi}(x_m) = [\psi_1(x_m), \ldots, \psi_N(x_m)]^T$. Then, the Koopman operator can be approximated by minimizing the total residual between snapshots,[7]

$$J = \frac{1}{2} \sum_{m=1}^{M} \| (\boldsymbol{\Psi}(x_{m+1}) - K\boldsymbol{\Psi}(x_m) \|^2 \ . \tag{16}$$

This can be solved as a least-square problem, yielding

$$K_{\text{EDMD}} = \boldsymbol{A}\boldsymbol{G}^\dagger, \quad \text{where} \begin{cases} \boldsymbol{G} = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\Psi}(x_m)\boldsymbol{\Psi}^*(x_m), \\ \boldsymbol{A} = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\Psi}(x_{m+1})\boldsymbol{\Psi}^*(x_m), \end{cases} \tag{17}$$

with $\dagger$ being the pseudoinverse and $*$ the conjugate transpose.

Define the action of the dictionary on a data matrix X by $\boldsymbol{\Psi}(X) = [\boldsymbol{\Psi}(x_1), \boldsymbol{\Psi}(x_2), \ldots, \boldsymbol{\Psi}(x_M)]$. Then, EDMD can be equivalently formulated as a least Frobenius-norm problem

$$\underset{K}{\text{minimize}} \|\boldsymbol{\Psi}(Y) - K\boldsymbol{\Psi}(X)\|_F \ , \tag{18}$$

with the closed-form solution

$$K_{\text{EDMD}} = \boldsymbol{\Psi}(Y)\boldsymbol{\Psi}(X)^\dagger \ . \tag{19}$$

The matrix K_{EDMD} captures important information about the Koopman operator. Formally, given a function $f \in \text{span}(\boldsymbol{\Psi})$ denoted as $f(\cdot) = v_f^T \boldsymbol{\Psi}(\cdot) = \sum_{i=1}^{N} (v_f)_i \psi_i(\cdot)$, the EDMD predictor for the function $\mathcal{K}f$ is defined as

$$\mathfrak{P}_{\mathcal{K}f}^{\text{EDMD}} := v_f^T K_{\text{EDMD}} \boldsymbol{\Psi} \ . \tag{20}$$

Note that $\mathfrak{P}_{\mathcal{K}f} \in \text{span}(\boldsymbol{\Psi})$ even if $\mathcal{K}f \notin \text{span}(\boldsymbol{\Psi})$. An important special case of (20) is for functions of the form $\phi(\cdot) = v_\phi^T \boldsymbol{\Psi}(\cdot)$, where v_ϕ is a left eigenvector of K_{EDMD} (that is, $v_\phi^T K_{\text{EDMD}} = \lambda_\phi v_\phi^T$), which leads to the approximated Koopman eigenfunction

$$\mathfrak{P}_{\mathcal{K}\phi}^{\text{EDMD}} := v_\phi^T K_{\text{EDMD}} \boldsymbol{\Psi} = \lambda_\phi v_\phi^T \boldsymbol{\Psi} = \lambda_\phi \phi \ . \tag{21}$$

---

[7] For ease of exposition our notation is transpose of the notation in [25].

The approximation error of the predictors in (20)-(21) depends on the selected dictionary of functions $\boldsymbol{\Psi}$. More precisely, the error depends on how close $\text{span}(\boldsymbol{\Psi})$ is to be invariant under the Koopman operator. To understand this, it is important to observe that K_{EDMD} does not capture the Koopman operator itself but it encodes the projection of the operator's action on $\text{span}(\boldsymbol{\Psi})$, i.e. EDMD approximates the following linear operator

$$\mathcal{P}_{\text{span}(\boldsymbol{\Psi})}\mathcal{K} : \mathcal{F} \to \mathcal{F} \ , \tag{22}$$

where $\mathcal{P}_{\text{span}(\boldsymbol{\Psi})} : \mathcal{F} \to \mathcal{F}$ is the $L_2(\mu_X)$-orthogonal projection operator on $\text{span}(\boldsymbol{\Psi})$ and the L_2 inner product is calculated based on the empirical measure

$$\mu_X = \frac{1}{M} \sum_{i=1}^{M} \delta_{x_i} \ , \tag{23}$$

with δ_{x_i} the Dirac measure at point x_i.

Since $\text{span}(\boldsymbol{\Psi})$ is always invariant under the operator in (22), one can restrict the action of (22) to $\text{span}(\boldsymbol{\Psi})$ and represent the restricted operator by a matrix (which, for exact data, coincides with K_{EDMD}) [34], [35]. This connection between EDMD and the Koopman operator leads to many important properties, including the convergence – as the dictionary of basis functions grows – of the operator defined by EDMD matrix to the Koopman operator in operator topology, capturing the Koopman operator's eigenvalues, as well as weak convergence of eigenfunctions. We refer the reader to [35] for the detailed analysis and statements.

It is important to realize that none of the aforementioned convergence results imply that a larger finite-dimensional space is necessarily better for prediction. For example consider the linear system $x^+ = 0.5x$ with two dictionaries $\boldsymbol{\Psi}_1(x) = x$ and $\boldsymbol{\Psi}_2(x) = [x, \sin(x)]$. Despite $\text{span}(\boldsymbol{\Psi}_1) \subsetneq \text{span}(\boldsymbol{\Psi}_2)$, prediction on $\text{span}(\boldsymbol{\Psi}_1)$ is exact (since it is invariant under the Koopman operator associated with the system). In contrast, the prediction on $\text{span}(\boldsymbol{\Psi}_2)$ has large errors for some functions (see e.g., [36]–[38] for methods that remove functions to prune subspaces and improve the prediction accuracy).

In general, to observe the asymptotic convergence phenomena of EDMD to the Koopman operator described in [35], the dimension of the dictionary and the number of data points must be *sufficiently large*. Moreover, without a system's model, it is not possible to estimate a lower bound on the dictionary's dimension to achieve a predetermined level of accuracy. In addition, if the dictionary sequence is chosen based on a given basis for the space $\mathcal{F}$ (that is not chosen based on the system's knowledge), the dimension of the dictionary required to achieve a high accuracy level might be extremely large. Therefore, for practical applications, it is imperative to design or learn dictionaries based on information available from the system and/or data to achieve a reasonable accuracy on relatively low-dimensional subspaces. We refer the reader to Section V-B for more information on such methods. One last observation is that these convergence results only hold for systems without control inputs. Further discussion on these can be found in Section VII.

Remark 4.1: (Dynamic Mode Decomposition): Dynamic Mode Decomposition (DMD) is a data-driven method originally proposed to extract coherent features of fluid flows [39]. Although developed before EDMD, the exact variant of DMD [40] can be seen as a special case of EDMD where there is no lifting on the data. Alternatively, the EDMD dictionary can be set as the identity map to recover the exact DMD.

C. Koopman Approximation for System with Inputs

The Koopman operator theory and EDMD were originally proposed without the consideration of inputs. To generalize the Koopman theory to allow for control inputs, the corresponding dynamical systems can be extended from (6) and (10) as

$$x_{t+1} = T_u(x_t, u_t) \text{ and } \dot{x} = G_u(x, u) , \qquad (24)$$

with $u \in \mathbb{R}^{N_u}$. For convenience, we only discuss the case of discrete-time systems in the following (the treatment for continuous-time systems is similar). Define a set of observables $g(x, u)$ that are functions of both the state and the input. The Koopman operator acting on the space of those observables is expressed as

$$\mathcal{K}g(x_t, u_t) = g \circ T_u = g(T_u(x_t, u_t), u_{t+1}) . \qquad (25)$$

Because (24) only prescribes how the state evolves, this formulation is, in general, ill-posed as it treats the state and the input as being the same when they are not. However, this formulation can work in specific cases [41]:

1) If the input is a time-invariant constant forcing signal, i.e. $u_{t+1} = u_t = c$, (25) can be rewritten as $\mathcal{K}g(x_t, c) = g(T_u(x_t, c), c)$.
2) If the input is generated as a function of states, i.e. via closed-loop control so that $u_t = f_x(x_t)$, (25) can be simplified to contain only the state as $\mathcal{K}\hat{g}(x_t) = \hat{g}(\hat{T}_u(x_t))$, where $\hat{T}_u$ is a function of T_u and f_x, and $\hat{g}$ is a function of g and f_x. Thus, a known closed-loop control law can reduce the complexity of the Koopman expression.
3) If the input follows its dynamics, i.e. $u_{t+1} = f_u(u_t)$, (25) can also be simplified as $\mathcal{K}g(\hat{x}_t) = g(\hat{T}_u(\hat{x}_t))$ by treating the inputs as states, i.e. $\hat{x} = [x; u]$.

When the input is generated from exogenous forcing, which is common in robotics, then (25) keeps its general form. Approaches to consider in this case include approximating the action of the inputs on the evolution of observables linearly (i.e. $g \circ T_u = \mathcal{K}g(x_t) + Bu_t$) or bilinearly (i.e. $g \circ T_u = \mathcal{K}g(x_t) + Bu_t + g(x_t)^T Fu_t$). Through the Koopman canonical transform [29], [42], control-affine systems $x_{t+1} = T(x_t) + \sum_i h_i(x_t)u_t^i$ can be transformed into a bilinear form.

Dealing with control system (24) is significantly more difficult than system (6). The difficulty is rooted in the difference between the input and the state. The state is a fundamental property of the system and evolves according to its dynamics, while the input in an open loop system is not known a-priori, and its choice can significantly alter the system's behavior. Below, we discuss two different ways to deal with this fundamental issue.

Considering All Infinite Input Sequences: The work in [43] tackles the aforementioned issue by considering the system's behavior over all possible input sequences. Formally, consider the space $l(\mathcal{U})$ comprised of *all possible infinite input sequences* $\{u_i\}_{i=0}^\infty$ with $u_i \in \mathcal{U}$ and let $S : l(\mathcal{U}) \to l(\mathcal{U})$ denote the left-shift operator on $l(\mathcal{U})$, mapping the sequence $\{u_i\}_{i=0}^\infty$ to $\{u_i\}_{i=1}^\infty$. Now, consider the extended state $x_s := (x, u_s) \in \mathcal{X} \times l(\mathcal{U})$ and define the dynamical system

$$x_s^+ = (x, u_s)^+ = (T_u(x, u_s(0)), Su_s) =: L(x_s) , \qquad (26)$$

where $u_s(0)$ is the first element of u_s. As in (7), we can define a Koopman operator $\mathcal{K}_L : \mathcal{H} \to \mathcal{H}$ for system L in (26) as

$$\mathcal{K}_L f = f \circ L, \ \forall f \in \mathcal{H} , \qquad (27)$$

where the vector space $\mathcal{H}$ is defined over the field $\mathbb{C}$, consists of functions with domain $\mathcal{X} \times l(\mathcal{U})$ and codomain $\mathbb{C}$, and is closed under composition with L, i.e., $g \circ L \in \mathcal{H}$ for all $g \in \mathcal{H}$.

Note that due to the dependency on infinite input sequences, working with operator (27) is significantly more difficult than in the case for (7), as it does not afford a direct way to find general finite-dimensional models for systems with access to finite input sequences. However, since [43] aims to use model predictive control, which requires *relatively accurate short-term prediction*, it assumes the dynamics can be approximated by a lifted linear model (termed linear predictor) as

$$z^+ = Az + Bu , \qquad (28)$$

where z is the lifted state starting from the initial condition $z_0 = \mathbf{\Psi}(x_0)$ (here $\mathbf{\Psi} : \mathcal{X} \to \mathbb{R}^{N_\Psi}$ is the lifting map). One can estimate the matrices A and B in (28) with an EDMD-like method. However, it is crucial to keep in mind that since the model does not consider infinite-input sequences, it does not generally capture all the information of the operator $\mathcal{K}_L$. Even if the dimension of the lifted state goes to infinity, one cannot generally conclude convergence of the lifted linear model trajectories to the trajectories of the nonlinear system (24); see also [43, *Discussion after Corollary 1*].

Koopman Control Family: An alternative Koopman-based approach to model the system (24) does not rely on infinite input sequences and it is easier to use for finite-dimensional approximations [44]. The central idea is that one can represent the behavior of (24) by a family of systems in the form of (6) generated by setting the input in (24) to be a constant signal. Consider the family of constant input dynamics $\{T_{\hat{u}}\}_{\hat{u} \in \mathcal{U}}$ defined by

$$x^+ = T_{\hat{u}}(x) := T_u(x, u \equiv \hat{u}), \ \hat{u} \in \mathcal{U} . \qquad (29)$$

For any trajectory $\{x_i\}_{i=1}^{m+1}$ of (24) generated by input sequence $\{u_i\}_{i=1}^m$ and initial condition x_0, one can write

$$x_{m+1} = T_{u_m} \circ T_{u_{m-1}} \circ \cdots \circ T_{u_0}(x_0) , \qquad (30)$$

Hence, the subsystems of the family $\{T_{\hat{u}}\}_{\hat{u} \in \mathcal{U}}$ completely capture the behavior of (24). Moreover, we can use (7) to define Koopman operators for each subsystem. Consider the vector space $\mathcal{F}$ over field $\mathbb{C}$ comprised of complex-valued functions with domain $\mathcal{X}$ that is closed under composition

with members of the family $\{T_{\hat{u}}\}_{\hat{u}\in\mathcal{U}}$. The *Koopman Control Family* $\{\mathcal{K}_{\hat{u}}\}_{\hat{u}\in\mathcal{U}}$ is defined such that for all $\hat{u} \in \mathcal{U}$ we have

$$\mathcal{K}_{\hat{u}}g = g \circ T_{\hat{u}}, \ \forall g \in \mathcal{F} \ .$$

The Koopman Control Family can completely capture the evolution of functions in $\mathcal{F}$ over the trajectories of (24). Given a trajectory $\{x_i\}_{i=1}^{m+1}$ of (24) generated by input sequence $\{u_i\}_{i=1}^{m}$ and initial condition x_0, one can write

$$g(x_{m+1}) = [\mathcal{K}_{u_0}\mathcal{K}_{u_1}\ldots\mathcal{K}_{u_m}g](x_0), \ \forall g \in \mathcal{F} \ .$$

Similarly to the case without control inputs where Koopman-invariant subspaces lead to a finite-dimensional linear form (cf. (9)), the models on *common* invariant subspaces of the Koopman control family are *all* in the following form (termed *"input-state separable model"*, cf. [44, Theorem 4.3])

$$\mathbf{\Phi}(x^+) = \mathbf{\Phi} \circ \mathcal{T}(x, u) = \mathcal{A}(u)\mathbf{\Phi}(x) \ ,$$

where $\mathbf{\Phi} : \mathcal{X} \to \mathbb{C}^{N_\Phi}$ is the lifting function and $\mathcal{A} : \mathcal{U} \to \mathbb{C}^{N_\Phi \times N_\Phi}$ is a matrix-valued function. Note that the input-state separable model is *linear* in the lifted state but *nonlinear* in the input. This is because, in an open-loop system, the input does not abide by predefined dynamics. Therefore, one cannot use the structure of the Koopman operator to represent the nonlinearity in the input as a linear operator. Interestingly, the popular lifted linear, bilinear, and linear switched models (e.g., [45]) are all special cases of the input-state separable form (cf. [44, Lemmas 4.4-4.5]). In case there is no suitable common invariant subspace, one can approximate the action of the Koopman Control Family on any given subspace via orthogonal projections. See [44] for theoretical analysis, data-driven learning methods, and accuracy bounds.

V. IMPLEMENTATION DETAILS

Having discussed the overall implementation as well as the fundamental theoretical results underlying the Koopman operator in robotics, we now take a deeper look into important implementation steps covering data collection, model estimation (with a specific focus on the selection of lifting functions), and controller design. Here, we present a comprehensive array of specific approaches and engage in detailed discussions regarding various options within each step.

A. Data Collection

As in all data-driven algorithms, the type and amount of (training) data to collect play an important role in model accuracy and control efficacy. Optimizing the data collection process in Koopman-based methods is an active research direction [46], [47]. Methods focus on the collection of both offline and online measurements.

One of the most common methods to collect data is random selection [48], i.e. generating measurements by propagating the system with random initial conditions and input signals. To enrich the diversity of the data, one should select initial conditions and inputs over the entire operating range of the robot [27], [46]. An enclosed area might be required for safety [49]. The random collection approaches are more widely

used for soft robots [50]–[52] as soft parts do less harm to their surroundings if an aggressive command is selected by chance.

However, generating data with random inputs can compromise safety, especially for rigid robots. One attempt to solve this challenge is to design a baseline controller, which can be open-loop and naive but is improved over time. For example, [31] utilizes the controller obtained from the previous episode to generate training data used for the current episode. By doing that, we can obtain observations in a safer way and with more valuable information. Another way that emphasizes the effectiveness of data is to account for the value of information, i.e. add constraints on information richness when solving the optimization problem. These ideas are closely connected to active learning [30], which we will discuss in Section V-C2.

For methods that rely on extracting dynamics from data, it is important to consider the stochastic case, when the measurements used for estimation are noisy, as the efficiency and quality of data have a significant impact on the performance of the estimated model and hence the controller design. To quantify and emphasize the effect of noise, [53] presents a procedure for deriving loose and tight bounds of the prediction error using Dynamic Mode Decomposition (DMD). The general version for EDMD is explored in [54]. Once the explicit prediction error due to noisy measurements is derived, this work describes how the proposed algorithm and the resulting prediction error can be utilized in controller design, providing robustness guarantees for the perturbed systems' performance. Another approach [48] to minimize the effect of modeling uncertainties and external disturbance is to include the usage of a Kalman filter. First, an augmented model is derived to include the disturbance model in the ordinary Koopman structure, and then a Kalman filter is adapted to estimate the observables and disturbances simultaneously. We delve into a comprehensive discussion about understanding and dealing with the impact of measurement noise when estimating the Koopman operator through EDMD and DMD.

1) EDMD and DMD with Measurement Noise: Here we discuss methods that deal with the effect of measurement noise in data following [55]. To better understand the effect of *measurement* noise in EDMD, we reformulate (18) as

$$\begin{aligned} &\underset{K,\Delta}{\text{minimize}} && \|\Delta\|_F^2 \\ &\text{subject to} && \mathbf{\Psi}(Y)^T + \Delta = \mathbf{\Psi}(X)^T K^T \ . \end{aligned} \quad (31)$$

This reformulation [8] immediately reveals two issues when data matrices are noisy.

- *Lack of correction in $\mathbf{\Psi}(X)$:* Based on (31), EDMD finds a correction for $\mathbf{\Psi}(Y)$ with minimum Frobenius norm subject to the optimization constraints. However, EDMD does not provide any correction for $\mathbf{\Psi}(X)$. If $\mathbf{\Psi}(X)$ is noisy, the EDMD solution will be incorrect. This is a limitation inherent to least-squares methods where noise in both dictionary matrices $\mathbf{\Psi}(Y)$ and $\mathbf{\Psi}(X)$ can lead to inconsistent solutions [56].
- *Noise distortion after applying the nonlinear map $\mathbf{\Psi}$:* Since we measure the data X, Y and pass them

---

[8] The notation in [55] is the transpose of the notation employed here.

through $\mathbf{\Psi}$, noise in X, Y gets distorted. (Even if the noise in X, Y is zero-mean and i.i.d, distorted noise in $\mathbf{\Psi}(Y), \mathbf{\Psi}(X)$ is in general neither.) Corrections to $\mathbf{\Psi}(Y), \mathbf{\Psi}(X)$ are needed to counter the noise distortion.

To address these issues, one should add a correction term for $\mathbf{\Psi}(X)$ and introduce additional terms canceling the noise's distortion. This leads to

$$\underset{K, \Delta_1, \Delta_2}{\text{minimize}} \quad \sum_{i=1}^{M} \left\| [\Delta_1, \Delta_2]_i \, C_i^{-1/2} \right\|_2^2$$

$$\text{subject to} \quad \mathbf{\Psi}(Y)^T - B_{\mathbf{\Psi}(Y)^T} + \Delta_2$$
$$= (\mathbf{\Psi}(X)^T - B_{\mathbf{\Psi}(X)^T} + \Delta_1) K^T , \quad (32)$$

where $[\Delta_1, \Delta_2]_i$ denotes the ith row of the matrix created by concatenating Δ_1, Δ_2 side by side. Here, $B_{\mathbf{\Psi}(X)^T}, B_{\mathbf{\Psi}(Y)^T}$ account for the bias created by noise distortion in $\mathbf{\Psi}(X)^T, \mathbf{\Psi}(Y)^T$, and $C_i^{-1/2}$'s account for the covariance distortion (for simplicity, we do not provide the precise expressions, but explicit formulas of the bias and covariance corrections can be found in [55]). The problem (32) is an element-wise weighted total least squares optimization [57], [58]. Under appropriate conditions on data-sampling and dictionary functions, (32) applied on noisy data provides a weakly consistent estimator of the EDMD's solution applied on noise-free data (see e.g., [55], [57]).

The measurement noise problem in DMD can be addressed more directly since there is no lifting via a nonlinear dictionary, so noise distortion does not occur. Hence, instead of solving (32) one only needs to form a total least squares problem [59], [60] as

$$\underset{K, \Delta_1, \Delta_2}{\text{minimize}} \quad \left\| [\Delta_1, \Delta_2] \right\|_F^2$$

$$\text{subject to} \quad Y^T + \Delta_2 = (X^T + \Delta_1) K^T , \quad (33)$$

Under some conditions, (33) provides a strongly consistent estimator for the DMD's solution on noise-free data [61].

An alternative approach to deal with noise in DMD is to consider both forward and backward dynamics. Given that for a nonsingular linear system, the forward and backward dynamics are inverse of each other, the work in [59] proposes the use of forward and backward DMD on the data to account for the effect of noise that is assumed to be additive to data, normally distributed, and with zero mean. In this case, one can approximate the correct DMD matrix as

$$\tilde{K}_{\text{DMD}} \approx \left(A_{\text{DMD}}^f A_{\text{DMD}}^b \right)^{-\frac{1}{2}} ,$$

where A_{DMD}^f and A_{DMD}^b are the debiased solutions of DMD on the noisy data forward and backward in time. Alternatively, [62] proposes a different method to consider the forward and backward dynamics by solving

$$\underset{K}{\text{minimize}} \quad \frac{1}{2} \| Y - KX \|_F^2 + \frac{1}{2} \| X - K^{-1}Y \|_F^2 .$$

This optimization problem aims to find the matrix K to account for forward and backward dynamics. The problem is highly non-convex and hence difficult to solve: the work in [62] provides several algorithms to solve it efficiently.

B. Model Estimation — Lifting Functions Design

The lifting functions are crucial because they serve as the basis for constructing a linear approximation of the (nonlinear) system's state evolution. A poor choice of the lifting functions can significantly impact the estimation accuracy of the Koopman operator and the higher-dimensional linear dynamics.

Existing examples regarding the design of the lifting functions fall under several main directions. The first one is empirical selection. For example, discontinuous spectral elements can make the observation matrix block diagonal [25], Hermite polynomials are best suited to problems where data are normally distributed, and radial basis functions are effective for systems with complex geometry. However, empirical approaches can be time- and effort-demanding and cannot guarantee generalization to and efficiency in new cases. Another direction is mechanics-inspired selections since robotic systems have certain characteristic properties, e.g., degrees of freedom, configuration spaces, and workspaces, that can be acquired without knowledge of their exact dynamical models. These properties reveal fundamental information about system states and dynamics and can provide intuition on how to select lifting functions required for Koopman operator approximation [63]. A third approach is to synthesize basis functions out of higher-order derivatives of the states (in the continuous time) [64] or past measurements (commonly referred to as Hankel Koopman models) [65], which can be used for completely unknown dynamics. Lastly, one can compare and optimize over multiple sets of lifting functions to find a proper basis set, but the question of how to select these sets efficiently remains open.

Next, we describe in detail data-driven methods for the identification of dictionaries/subspaces that lead to accurate Koopman approximations. These methods have in common their reliance on the notion of Koopman-invariant subspace and the fact that the restriction of the Koopman operator to such subspaces is exact. They are highly connected to the convergence analysis introduced earlier in Section IV-B.

1) Optimization-based Methods: **Learning in original coordinates:** Available data can be used to learn an appropriate dictionary. The work [66] takes advantage of the eigenfunction evolution on non-recurrent sets and provides an optimization-based approach to directly learn Koopman eigenfunctions which naturally span invariant subspaces, leading to accurate long-term prediction. The work [67] uses a sparsity-promoting approach to derive the learned subspace to being close to invariant under the Koopman operator. Another popular approach is the use of neural networks or other parametric families (such as polynomials, Fourier, or radial-basis functions) to learn appropriate dictionaries (while fixing the state variables as a part of the dictionary) through the optimization

$$\underset{K \in \mathbb{R}^{M \times M}, \mathbf{\Psi} \in \text{PF}}{\text{minimize}} \| \mathbf{\Psi}(Y) - K\mathbf{\Psi}(X) \|_F , \quad (34)$$

where PF is the parametric family of choice. This optimization can be solved simultaneously in both variables K and $\mathbf{\Psi}$ [68], in a sequential manner in each variable [69], or one can simply use the closed-form solution (19) of EDMD [70], to write

$$\underset{\mathbf{\Psi} \in \text{PF}}{\text{minimize}} \| \mathbf{\Psi}(Y) - \mathbf{\Psi}(Y)\mathbf{\Psi}(X)^{\dagger} \, \mathbf{\Psi}(X) \|_F . \quad (35)$$

Note that the cost function in (35) is the residual error of EDMD at the optimal solution. It is important to note that (34)-(35) are prone to over-fitting, therefore regularization may be necessary. More importantly, optimization problems (34)-(35) are solved over a parametric family of functions (not just the EDMD matrix); therefore, they are generally non-convex and finding globally optimal solutions is not guaranteed.

Enforcing subspace invariance under the Koopman operator: Minimizing the EDMD's residual error in (34)-(35) does not necessarily lead to a close-to-invariant subspace. Figure 6 illustrates this point with an example of a non-invariant subspace whose residual error can be made arbitrarily small depending on the selected basis of functions. Therefore, optimization problems (34)-(35) might lead to models not suitable for long-term prediction.

The work in [71] addresses this problem by providing an alternative loss function that captures the quality of the subspace. The notion of *temporal forward-backward consistency index* (or *consistency index* for brevity) is defined as [9]

$$\mathcal{I}_C(\mathbf{\Psi}, X, Y) := \lambda_{\max}\big(I - K_F K_B\big) . \tag{36}$$

$\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of its argument, and

$$K_F = \mathbf{\Psi}(Y)\mathbf{\Psi}(X)^\dagger ,$$
$$K_B = \mathbf{\Psi}(X)\mathbf{\Psi}(Y)^\dagger , \tag{37}$$

are the EDMD matrices applied forward and backward in time on the data set. The intuition behind this definition is that when the subspace is Koopman invariant, the forward and backward EDMD matrices are inverse of each other and therefore $\mathcal{I}_C = 0$; otherwise, $\mathcal{I}_C \neq 0$ and the larger $\mathcal{I}_C$, the larger the deviation between forward and backward EDMD matrices from being inverse of each other.

The consistency index has several important properties. (1) $\mathcal{I}_C \in [0, 1]$; (2) Unlike the cost functions in (34)-(35), $\mathcal{I}_C$ only depends on the subspace $\mathrm{span}(\mathbf{\Psi})$ and not any specific basis (Fig. 7 illustrates this); (3) Under specific change of basis, $\mathcal{I}_C$ can be viewed as the maximum eigenvalue of a positive semidefinite matrix ($I - K_F K_B$ is not generally symmetric) enabling the use of common optimization solvers; (4) Importantly, it provides a tight upper bound on the relative prediction error of EDMD on the entire subspace. Formally,

$$\sqrt{\mathcal{I}_C(\mathbf{\Psi}, X, Y)} = \max_{f \in \mathrm{span}(\mathbf{\Psi}), \|\mathcal{K}f\|_{L_2(\mu_X)} \neq 0} \frac{\|\mathcal{K}f - \mathfrak{P}_{\mathcal{K}f}\|_{L_2(\mu_X)}}{\|\mathcal{K}f\|_{L_2(\mu_X)}} ,$$

where $\mathfrak{P}_{\mathcal{K}f}$ is the EDMD's predictor for $\mathcal{K}f$ in (20) and the L_2 norm is calculated based on the empirical measure (23).

Minimizing the consistency index is equivalent to the following robust minimax problem (where we have a closed-form expression for the max part)

$$\underset{\mathbf{\Psi} \in \mathrm{PF}}{\mathrm{minimize}} \; \max_{f \in \mathrm{span}(\mathbf{\Psi}), \|\mathcal{K}f\|_{L_2(\mu_X)} \neq 0} \frac{\|\mathcal{K}f - \mathfrak{P}_{\mathcal{K}f}\|_{L_2(\mu_X)}}{\|\mathcal{K}f\|_{L_2(\mu_X)}} , \tag{38}$$

which minimizes the maximum EDMD function prediction error over the subspace. Note that, by minimizing the consistency index, one accounts for accurate prediction of all uncountably many members of the function space, as opposed

[9] This definition is equivalent but different from the definition in [71].
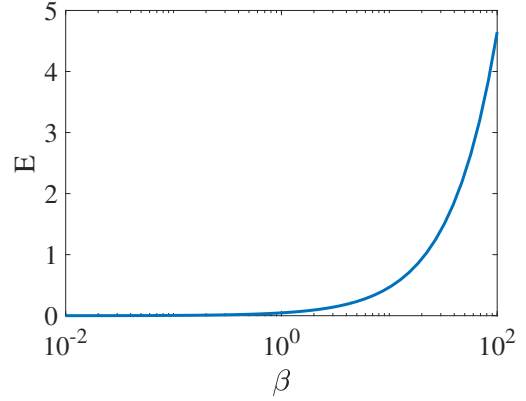

Fig. 6. EDMD's residual error for linear system $x^+ = 0.6x$ and the family of dictionaries $D_\beta(x) = [x, x + \beta \sin(x)]$ with $\beta \in [0.01, 100]$. Note that for all $\beta \in \mathbb{R} \setminus \{0\}$, all dictionaries span the same subspace $\mathcal{S} = \mathrm{span}\{x, \sin(x)\}$. The residual error depends on the choice of basis for subspace $\mathcal{S}$. More importantly, $\mathcal{S}$ is not Koopman-invariant but the residual error can be arbitrarily close to zero depending on the basis.
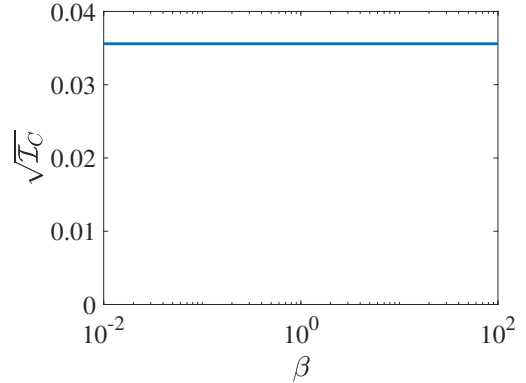

Fig. 7. The square root of consistency index for the system and family of dictionaries of Fig. 6. Unlike EDMD's residual error depicted in Fig. 6, the consistency index does not depend on the choice of basis and accurately measures the approximation quality of the subspace.

to only finitely many functions in the optimizations (34)-(35). Therefore, by minimizing the consistency index, one can expect a more accurate prediction, especially in the long term. Figure 8 illustrates this point in a nonlinear pendulum example.

Learning a Change of Coordinates: The previous optimization methods aim to learn the dynamics in the original coordinates. However, the most intuitive coordinate system is generally not the best. A better approach would be to choose Koopman eigenfunctions as the basis for our coordinates along which dynamical evolution is linear, i.e. we can choose the injective (so it can be inverted) vector-valued function $\mathbf{\Phi}(x) : \mathbb{R}^{N_x} \to \mathbb{R}^{N_\Phi}$ that spans an (approximately) invariant subspace as our new coordinates and build an inverse map $\mathbf{\Theta} : \mathcal{R}(\mathbf{\Phi}) \to \mathbb{R}^{N_x}$, where $\mathcal{R}(\mathbf{\Phi})$ is the range of $\mathbf{\Phi}$ and ideally we have $\mathbf{\Theta}\big(\mathbf{\Phi}(x)\big) = x$ for all $x \in \mathbb{R}^{N_x}$. Therefore, the dynamics in the new coordinates can be written as

$$\mathbf{\Phi}(x^+) = A\mathbf{\Phi}(x) ,$$
$$x^+ = \mathbf{\Theta}\big(\mathbf{\Phi}(x^+)\big) ,$$

where $A \in \mathbb{R}^{N_\Phi \times N_\Phi}$. Note that if $\mathbf{\Phi}$ is injective, we can work in the $\mathbf{\Phi}$ coordinates and do not necessarily need to use the inverse map $\mathbf{\Theta}$. However, analytically checking whether a map is injective or not is computationally intensive or even
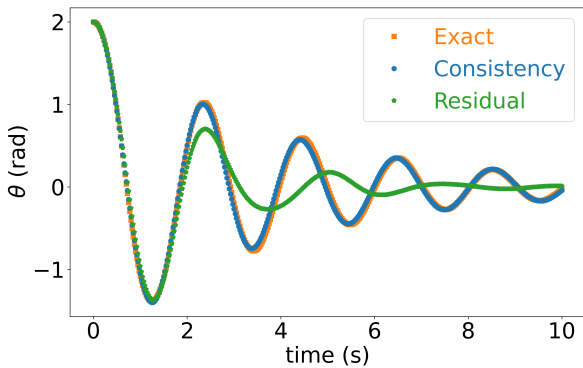
Fig. 8. We consider here the pendulum $[\dot{\theta}, \dot{\omega}] = [\omega, -9.81\sin(\theta) - 0.1\omega]$ and a parametric family of dictionaries comprised of 5 functions in the form $\Psi(\theta, \omega) = [\theta, \omega, \mathrm{NN}_1, \mathrm{NN}_2, \mathrm{NN}_3]$, where each NN is a feedforward neural network. The plot compares the prediction of the pendulum's angle evolution given linear predictors on the subspaces learned by minimizing the consistency index (equivalent to robust optimization (38)) and minimizing the residual error of EDMD (cf. (35)). The subspace learned by minimizing the consistency index is superior in long-term prediction. This is due to the fact that it accounts for all (uncountably many) members of the function space rather than only finitely many members considered in the residual error of EDMD.

intractable (see e.g., [72]); therefore, the maps $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$ are often learned simultaneously from data as a part of an auto-encoder neural network structure [73]–[75]. The work in [76] formulates a similar problem using rank-constrained semi-definite programs and provides convex relaxations to identify the linear embedding and the mapping that leads to the system's states.

Stability of attractors and dictionary learning: An important system property is the stability of attractors. This phenomenon is directly related to the existence of certain Koopman eigenfunctions [77]. Even though the Koopman operator is infinite-dimensional, the stability of the attractors is generally decidable by *finitely many* eigenfunctions. Moreover, these eigenfunctions lend themselves to a Lyapunov formulation, thus enabling the use of already existing rich literature of Lyapunov-based tools. The work [78] studies the stability problem from the angle of contractions and shows the equivalence of contraction and Koopman-based results regarding stability. Using the connection of Koopman eigenfunctions to attractors' stability, [79] provides a neural network-based method to learn Lyapunov functions from data. Moreover, [80] provides a numerical method to build Lyapunov functions using Koopman eigenfunctions leading to estimates for the regions of attractions and their geometric features. In the context of dictionary/subspace learning, by enforcing the stability in the learned model (given the knowledge about the attractor's stability), one can ensure a more realistic and accurate model [81]–[83].

2) Algebraic Search for Koopman-Invariant Subspaces: The optimization methods described above do not generally come with guarantees on the quality of the identified subspaces unless a global optimizer is found. However, this can be difficult since they often rely on objective functions that are non-convex and employ parametric families that lack desirable algebraic structures (e.g., neural networks). Interestingly, imposing the structure of a vector space on the parametric family

allows one to effectively utilize the linearity of the Koopman operator for the identification of both exact and approximate eigenfunctions and invariant subspaces.

The work [36] provides a data-driven necessary and almost surely sufficient condition to identify all Koopman eigenfunctions in an arbitrary finite-dimensional function space. This result directly relies on the eigendecomposition of forward and backward EDMD matrices (cf. (37)). Moreover, [36] provides an efficient and provably correct algebraic algorithm termed *Symmetric Subspace Decomposition (SSD)* that finds the maximal Koopman-invariant subspace in arbitrary finite-dimensional function spaces (termed "search space"). In addition, [37] provides a parallel version of the SSD algorithm suitable for searching through high-dimensional spaces. Given as search space the finite-dimensional function space spanned by the dictionary $\boldsymbol{\Psi}_s$, any basis $\boldsymbol{\Psi}$ with elements in $\mathrm{span}(\boldsymbol{\Psi}_s)$ can be described using a matrix C with full column rank as $\boldsymbol{\Psi}^T(\cdot) = \boldsymbol{\Psi}_s^T(\cdot)C$. If the subspace $\mathrm{span}(\boldsymbol{\Psi})$ is invariant under the Koopman operator, this gets reflected in the data, $\mathcal{R}(\boldsymbol{\Psi}(X)^T) = \mathcal{R}(\boldsymbol{\Psi}(Y)^T)$ (where $\mathcal{R}(\cdot)$ stands for the range space of its argument), which can also be written as

$$\mathcal{R}(\boldsymbol{\Psi}_s(X)^T C) = \mathcal{R}(\boldsymbol{\Psi}_s(Y)^T C) . \qquad (39)$$

Therefore, finding the largest invariant subspace within the finite-dimensional function space spanned by the dictionary $\boldsymbol{\Psi}_s$ can be equivalently posed as finding the matrix C with the maximum number of columns that satisfies (39). The SSD algorithm and its parallel implementation build on this observation to devise an algebraic procedure to identify the largest invariant subspace contained in $\mathrm{span}(\boldsymbol{\Psi}_s)$.

Exact Koopman-invariant subspaces capturing complete information about the dynamics are rare. A typical and useful approach is to allow for some error in the model to capture more (but inexact) information. However, it is crucial to characterize and tune the approximation accuracy. The *Tunable Symmetric Subspace Decomposition (T-SSD)* [38] is an algebraic search algorithm able to find a suitable subspace with any predetermined error level (specified by a tuning variable) on the function approximations. Instead of requiring the equality in (39), the T-SSD algorithm enforces the subspaces $\mathcal{R}(\boldsymbol{\Psi}_s(X)^T C)$ and $\mathcal{R}(\boldsymbol{\Psi}_s(Y)^T C)$ to be *close*. Formally, this is captured by an accuracy parameter $\epsilon \in [0, 1]$ that specifies the distance between the two subspaces. The T-SSD algorithm can be viewed as the generalization of the EDMD and SSD algorithms, cf. Fig. 9. If we do not impose any accuracy level (allowing for up to a maximum 100% prediction error by setting $\epsilon = 1$), T-SSD is equivalent to applying EDMD on the search space. On the other hand, if we specify zero-prediction error by setting $\epsilon = 0$, then T-SSD is equivalent to SSD and finds the maximal Koopman-invariant subspace of the search space with (almost surely) exact prediction on the entire subspace. In this sense, T-SSD balances the accuracy and the expressiveness (the dimension of the identified subspace) of the model based on the parameter ϵ. Figure 10 shows the relative prediction error of the Duffing system over the search space of all polynomials up to degree 10 and the identified subspace by T-SSD given $\epsilon = 0.02$.
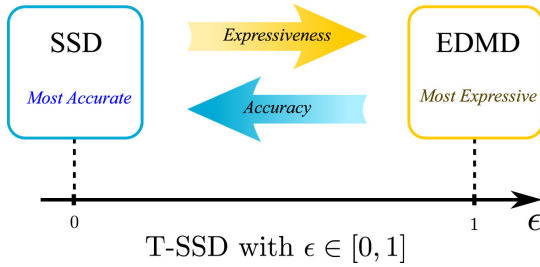
Fig. 9. The variable $\epsilon \in [0,1]$ in T-SSD sets the balance between the accuracy and expressiveness of the model. Both SSD and EDMD algorithms are special cases of T-SSD with $\epsilon = 0$ and $\epsilon = 1$, respectively. (Image taken from [38] and is available under license (CC BY 4.0): https://creativecommons.org/licenses/by/4.0/.)
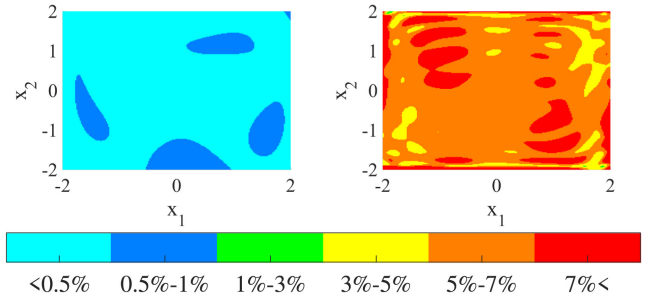

Fig. 10. Consider the Duffing System $[\dot{x}_1, \dot{x}_2] = [x_2, -0.5x_2 + x_1(1-x_1)^2]$ over the state space $[-2, 2]^2$ with the search space comprised of all polynomials up to degree 10. The right plot shows the relative EDMD dictionary prediction error for a normalized basis of the search space and the left plot shows the same error for the dictionary identified by T-SSD with $\epsilon = 0.02$. (Image taken from [38] and is available under license (CC BY 4.0): https://creativecommons.org/licenses/by/4.0/.)

C. Controller Design

1) Model-based Controller: A system model plays a pivotal role in facilitating the design of model-based controllers, enabling us to exploit model predictions for the selection of appropriate control inputs aligned with a given task. As elucidated in Section V, we can derive both linear and nonlinear models from measurements using Koopman operator theory, seamlessly integrating them into the architecture of model-based controllers. The centerpiece and critical determinant impacting the efficacy of these controllers is the inherent "model." The Koopman operator, adept at accommodating new real-time measurements and/or pre-existing offline measurements, can serve as a defining model constraint within the formulation of the controller structure. The incorporation of the Koopman operator into model-based control was initially introduced in [84]. Subsequently, the concepts of Koopman Model Predictive Control (Koopman MPC) [43] and Koopman Nonlinear Model Predictive Control (Koopman NMPC) [29] ensued. Through the process of estimating and refining models, utilizing either online measurements or offline learning gleaned from data acquired within physical environments, a Koopman-based model controller can effectively accomplish control objectives even for systems with incomplete knowledge or that are entirely unknown.

In a Koopman-based optimal controller, control inputs are determined by iteratively solving

$$\underset{\{\boldsymbol{z}_i\}_{i=0}^{N_h}, \{\boldsymbol{u}_i\}_{i=0}^{N_h}}{\text{minimize}} \quad J\left(\{\boldsymbol{z}_i\}_{i=0}^{N_h}, \{\boldsymbol{u}_i\}_{i=0}^{N_h}\right)$$
$$\text{subject to} \quad \boldsymbol{z}_{i+1} = F_{\mathcal{K}}\left(\boldsymbol{z}_i, \boldsymbol{u}_i\right)$$
$$\boldsymbol{z}_0 = \boldsymbol{\Psi}(\boldsymbol{x}_t) , \quad (40)$$

where N_h is the number of time steps in the controller horizon, J is a quadratic cost function, $\boldsymbol{\Psi}$ is the lifting dictionary, $F_{\mathcal{K}}$ is the Koopman-based system model, and $\boldsymbol{z}_i$ and $\boldsymbol{u}_i$ are the lifted system state and input at the ith time-step in the horizon, respectively. The problem's objective function penalizes deviations from a desired trajectory, and its constraints enforce consistency with a Koopman system model. With a linear Koopman model realization, the optimization (40) becomes

a quadratic program (QP)

$$\underset{\{\boldsymbol{z}_i\}_{i=0}^{N_h}, \{\boldsymbol{u}_i\}_{i=0}^{N_h}}{\text{minimize}} \quad \sum_{i=0}^{N_h} \boldsymbol{z}_i^\top G_i \boldsymbol{z}_i + \boldsymbol{u}_i^\top H_i \boldsymbol{u}_i + \boldsymbol{g}_i^\top \boldsymbol{z}_i + \boldsymbol{h}_i^\top \boldsymbol{u}_i$$
$$\text{subject to} \quad \boldsymbol{z}_{i+1} = K_A \boldsymbol{z}_i + B \boldsymbol{u}_i$$
$$\boldsymbol{z}_0 = \boldsymbol{\Psi}(\boldsymbol{x}_t) . \quad (41)$$

As discussed in Section IV-A, the Koopman operator can be used to describe a system's dynamics in either linear or nonlinear form. A major advantage of a linear Koopman model realization is that it makes the formulation to be convex. This implies it has a unique globally optimal solution that can be efficiently computed without initialization even for high-dimensional models [85]–[87], making Koopman MPC well-suited for real-time feedback control applications. Nonlinear/bilinear Koopman model realizations render (40) non-convex, which are less efficient to solve and may only yield locally optimal solutions [88]. Sometimes a nonlinear Koopman realization generates more accurate predictions, in which case such a trade-off is warranted. Bilinear Koopman model realizations have been explored recently to combine some of the advantages of linear and nonlinear models [27].

2) Active Learning: The linear structure of the Koopman operator yields several advantages for learning the dynamics of a robotic system. The closed-form solution of the linear least-squares model fitting problem (16), (17), and (19) to approximate the Koopman operator can be leveraged to formulate active-learning controllers [30]. Specifically, assume that the Koopman dynamics from the dictionary of observables $\boldsymbol{\Psi}(x_{t+1}) = K_A \boldsymbol{\Psi}(x_t) + B\mathbf{u}_k$ form the mean of a normally distributed state in the lifted space $p(\mathbf{z}_{t+1}|K, \mathbf{z}_k)$. Then, one can form an approximation of the *Fisher-Information Matrix*

$$\mathbf{I} = \mathbb{E}\left[\frac{\partial}{\partial K} \log p(\mathbf{z}_{t+1}|K, \mathbf{z}_k) \frac{\partial}{\partial K} \log p(\mathbf{z}_{t+1}|K, \mathbf{z}_k)^\top\right]. \quad (42)$$

For normally distributed systems with zero-mean and variance Σ, this attains the closed-form expression

$$\mathbf{I} = \frac{\partial \mathbf{z}_{t+1}}{\partial K}^\top \Sigma^{-1} \frac{\partial \mathbf{z}_{t+1}}{\partial K} \leq \text{Var}[K^\star]^{-1} , \quad (43)$$

where $\mathbf{z}_{t+1} = K_A\mathbf{z}_t + B\mathbf{u}_t$, and $\mathrm{Var}[K^\star]^{-1}$ is the posterior variance of the approximate Koopman operator (as calculated by the Grammian matrix in (17)). The Fisher-information is a matrix that lower-bounds the posterior uncertainty in estimation, defined as the Cramér-Rao bound [89], [90]. The Fisher-information Matrix is differentiable and actionable, i.e. one can optimize controllers that directly maximize the Fisher information, thus improving the best-case posterior variance on the Koopman operator, given the current operator.

Such controllers can be obtained via optimization using the optimality conditions on the Fisher-information [90], i.e.

$$\begin{aligned}
\operatorname*{minimize}_{\{\mathbf{u}_i\}_{i=0}^{N_h}} \quad & \sum_{i=0}^{N_h-1} \mathfrak{I}(\mathbf{z}_i, {}^tK) + \mathbf{u}_i^\top R\mathbf{u}_t \\
\text{subject to} \quad & \mathbf{z}_{i+1} = {}^tK_A\mathbf{z}_i + {}^tB\mathbf{u}_i \\
& \mathbf{z}_0 = \mathbf{\Psi}(x_t) \quad\quad\quad\quad\quad (44)
\end{aligned}$$

where $R \succ 0$ is a positive definite matrix, $\mathfrak{I}(\mathbf{z}_i, {}^tK)$ is an optimality condition (e.g., D-, or T- Optimality) that reduces the matrix into a scalar value, N_h is the time-horizon, and the superscript ^{t} indicates the current estimate of the Koopman operator given state-control data collected in the past. Note that the above optimization problem is done in a receding-horizon manner to account for changes in the Koopman operator. Controllers that are formulated from the optimization (44) effectively lower the overall posterior variance of the model yielding an effective Koopman operator model with a few data points. The resulting model can be used across different tasks such as for an aerial robot to quickly recover from an unstable tumble or a legged robot to learn complex interaction models with granular media (see [30]).

The simple structure of the Koopman operator has advanced learning in other directions. In particular, the use of deep models to approximate the function observables has made significant strides in expanding the use-case of methods based on the Koopman operator [30], [68], [91]. Deep function observable can further be integrated into an active-learning problem with some success [30]. While the added complexity in the observables provides more flexibility in the modeling range of the Koopman operators, it does reduce the effectiveness of active learning. This is a result of more data being required to effectively learn the nonlinear observables. When compared to deep neural network models, the Koopman-based linear model still has a significant advantage in data-efficiency and control through active learning [30].

D. Robustness and Stability

Efforts have also been made to ensure robustness and stability in the face of modeling errors. In [92], the concept of a deep stochastic Koopman operator is introduced to establish stability in nonlinear stochastic systems. This approach involves designing a probabilistic neural network to estimate the distribution of observables, which is then propagated through the Koopman matrices. The training process simultaneously optimizes the parameters of both the probabilistic neural network and the Koopman operator. Further, in [83], researchers compute the nearest stable matrix solution to minimize the least-squares reconstruction error. Rather than addressing uncertainty solely at the modeling stage, alternative approaches explore strategies for considering stability and model mismatch directly during the controller design phase. For instance, modeling errors can be effectively managed by employing a constraint-tightening approach within a proposed tracking MPC strategy [93]. This approach ensures a recursively feasible and input-to-state stable system even in the presence of bounded modeling errors. Another approach to tackle this issue involves constructing a conformant model that combines trace conformance and reach-set conformance [94]. To establish a robust MPC structure, [95] leverages an approximation of the Lipschitz constant for both state- and control-dependent model errors obtained during training. This approach can bound the prediction error across the planning horizon and enable the formulation of an online robust model-based controller.

VI. APPLICATIONS IN ROBOTIC PLATFORMS

In this section, we introduce the challenges involved in modeling different robotic platforms and how Koopman-based implementations can help address them.

A. Aerial Robots

The inherent nonlinearity of aerial robots presents significant challenges for achieving precise control. Key challenges arise when there is interaction with the environment that is governed by hard-to-model aerodynamic effects [96], such as variable wind gusts and ground effects when landing or flying at a low altitude [97], which can corrupt the nominal model and lead to unexpected performance.

Data-driven Koopman-based approaches have attempted to mitigate the effect of such uncertain aerodynamic interactions by extracting and adapting online system models used to linearly control the aerial robot. This overall idea can be achieved in different ways. For example, [31] uses episodic learning to estimate the Koopman eigenfunction pairs and obtain the resulting control inputs in real time to handle the ground effect when a multirotor is landing. A sequence of Koopman eigenfunctions is iteratively learned from data. Accordingly, nonlinear control signals are improved from the nominal control law. Another implementation of the Koopman operator theory in the aerial robots is to jointly learn a function dictionary and the lifted Koopman bilinear model to achieve quadrotor trajectory tracking at a low altitude [98]. A neural network is combined with the Koopman operator to update both the lifted states and inputs of the robot with online measurements. In a different effort, a hierarchical structure is proposed that refines the reference signal sent to the high-rate, pre-tuned low-level controller of the aerial robot to deal with uncertainty [53]. A model of the reference and the actual output of the disturbed robot is learned via Koopman operator theory and is utilized in the outer controller to decrease the effect of environmental disturbance in real time.

B. Ground Robots

Wheeled robots are subject to nonlinear dynamic effects governed by continuous contact with the ground that are hard

to model. In many applications, wheeled robots are called to navigate smoothly and efficiently over varying types of terrain, including rugged and deformable terrain [99]. This is usually formulated as an optimization problem over a time horizon, where a model of the underlying system in the changing environment is required. Koopman operator-based methods can serve as the model extraction method without imposing stringent requirements on information to be known (or assumed) a priori. Then, a model-based controller, e.g., Model Predictive Controller (MPC), can be designed based on this data-driven model to achieve a trajectory tracking objective [93]. Other considerations including rotational motions of the mobile wheeled robots are emphasized via coordinate transformation by introducing virtual control inputs [100].

Legged robots can traverse different and more complicated terrains than wheeled robots. Legged robots usually have highly nonlinear and sophisticated dynamics, while locomotion is governed by discrete impacts, thus making modeling and control a hard task [101] An initial attempt at modeling legged robots with the Koopman operator is done with the quadruped leg dynamics on deformable terrains in [102]. An experimental framework has been proposed to obtain a data-driven Koopman-based model of a quadruped's leg dynamics as a switched dynamical system. Experimental results have shown that the learned switched system model can be used to predict gait trajectories when operating over unknown terrains. Though current developments in this area are still in their infancy, more research activity is expected, e.g., controller design with Koopman-based models, which may become a potentially powerful tool in legged locomotion.

Distinctively yet crucially, the Koopman operator has also found its way into the realm of autonomous driving [103]. A key challenge is how to identify a global vehicle dynamics model considering the inherent complexity of the different subsystems and the induced nonlinearities and uncertainties. Considerable efforts have been directed toward utilizing the Koopman operator theory for estimating vehicle models [104]. The resulting approximated models can then be integrated into controller design, such as Model Predictive Control (MPC) [105]–[107]. In some cases, like in operation over deformable terrains with significant height variations and the presence of bumps [108], it has been shown possible to obtain a linear representation of both vehicle and terrain interaction dynamics using Koopman estimation. Further advancements have considered bilinear model formations for autonomous vehicles [109] and their combination with deep neural networks [91]. The integration of driver-in-the-loop dynamics has been explored in [110], reflecting a holistic approach to autonomous driving. Additionally, [111] introduces the embedding of the stochastic Koopman operator structure within an attention-based framework. This innovative approach is leveraged for abnormal behavior detection in autonomous driving systems, thereby enhancing the safety of vehicle controllers. The widespread application of the Koopman operator demonstrates its versatility and efficacy in advancing autonomous driving technologies.

C. Robot Manipulators

(Rigid) robot manipulators are another popular robotic platform because of their stability, feasibility, and safety. Modeling and control for most robot arms have been well established, and these robots are robust to environmental uncertainties. However, recent advances in embodied intelligence consider to a large extent robot manipulators, and even for these systems the simulation-to-real gap remains a challenge. Besides works on model approximation approaches for robot arms [27], [46], [112], Koopman operator theory is usually employed as assistance for other learning methods such as reinforcement learning. For instance, to guide the learning process, Koopman-based methods have been used to build a human intent model from human-demonstrated trajectories [113]. Another example is embedding the Koopman operator into LQR formulation for efficient and flexible policy learning [114].

D. Underwater Robots

Beyond several works using the Koopman operator to analyze fluid flows directly [115], Koopman-based methods have been also implemented for modeling and control of underwater robotic systems that interact with those changing flows. Underwater robots are typically underactuated and nonlinear because of propulsion designs aimed at reducing weight and cost. On the other hand, such robots often are subject to uncertain disturbance forces and moments that can be generated by fluid fields or other objects [116]. The use of the Koopman operator can introduce linearity in the model expression, which can be further utilized as model constraints in the controller design [117], [118]. Further, feedback policies that can learn or adapt online to unmodeled changes provide directions to solve the aforementioned challenges. In [64], a Koopman-LQR approach is proposed and experimentally tested in the dynamic estimation and control of a tail-actuated robotic fish. Results show that the online-updated data-driven model constructed using the Koopman operator significantly improves control performance in trajectory-tracking tasks compared to a tuned PID controller. Even though Koopman-based methods display better performance than other state-of-the-art data-driven modeling methods, like NARX (neural network approach [119]) and SINDy [120] in this illustration, the control of underwater robots with the Koopman operator is still an open direction in more complicated environments with harder tasks.

E. Soft Robots

Soft robots have received significant attention in recent years, and have grown into an important research topic. In contrast to their rigid counterparts, soft robots have bodies made out of intrinsically soft or extensible materials, which exhibit unprecedented adaptation to complex environments and can absorb impact energy for safe interactions with the environment, other robots, and even humans. Soft robots are particularly well-suited for data-driven modeling approaches because in most cases they can be operated even under randomized control inputs without posing much of a physical risk to themselves or their surroundings. Data can thus be

collected from them much more safely than from traditional rigid-body robots.

Koopman operator theory provides a data-driven alternative to traditional modeling and control methods for soft robots. It is attractive for soft robot applications because it generates global linear representations of their dynamics (which may be otherwise intractable) that are compatible with traditional control techniques such as MPC. Koopman theory has been applied to the modeling of various soft robotic systems, enabling real-time control [48], [92], [121]–[127].

Major challenges remain in applying Koopman methods to the modeling and control of soft robots. One disadvantage of Koopman-based linearization is that it involves increasing the dimension of the state space to achieve linearity. Soft robots and other continuum systems have inherently high-dimensional dynamics, therefore it is not always possible to linearize their dynamics through lifting without exceeding reasonable computational complexity. Instead, low-dimensional parameterizations of soft robots must be utilized to make them compatible with Koopman modeling techniques, which inadvertently rely on assumptions. In turn, the resulting models may not generalize well, especially in contact-rich applications. How to choose the most appropriate description of a soft robot for compatibility with Koopman methods remains an open area of research.

F. Other Types of Robot Platforms

Beyond the categories described above, Koopman operator theory can also contribute to advancing the modeling and control of other robot platforms. One example is the case of autonomous robotic excavators, where the excavator works and interacts with the surrounding soil in a highly nonlinear and complex manner [128], [129]. These nonlinear bucket-soil dynamics are learned as a set of linear equations with the Koopman operator. System identification and control applied to rehabilitation robots or limb assistive devices [130], [131] is another nascent area of application. Legged robots and manipulator robots involved in rehabilitation often undergo discrete transitions between contact and non-contact states, leading to a switch in governing equations. The application of Koopman operator theory to these hybrid systems offers the advantage of directly converting a diverse array of heterogeneous systems into a unified global linear model [132]. This characteristic proves particularly beneficial for control purposes in managing the intricacies of these dynamic systems. More instances include but are not limited to model extraction of snake-like robots [133], smarticle ensembles [134], and surgery robots [135].

G. Multi-agent Systems

In addition to its applications in single-agent systems, Koopman operator theory has proven valuable in the domain of multi-agent systems. In a multi-agent robotic system, multiple robots or agents collaborate to execute tasks within a shared environment. A comprehensive introduction and tutorial on the application of Koopman system estimation and control in multi-agent systems is provided in [136], [137].

A critical aspect of multi-agent systems is formation control, where a group of robots must coordinate to maintain a specific spatial arrangement while in motion. However, real-world applications may introduce external environmental disturbances, posing challenges to the multi-agent system's robustness. The Koopman operator can help address these challenges. It has been employed to estimate the disturbed model of agents within a group, either through online adaptation [138] or offline training [139] methods. This estimation aids in managing environmental uncertainties, allowing the team to maintain a desired formation. Koopman operator theory has recently been used to address scenarios involving disconnection and signal recovery within the system [140], by essentially aiding in recovering missed signals of the leader by capturing inherent features through linear motion evolution.It has also been applied for modeling high-dimensional biological or engineering swarms [141], thereby facilitating the learning of local interactions within homogeneous swarms based on observational data and enabling the generation of similar swarming behavior using the acquired model.

VII. DISCUSSION AND CONCLUSION

This review paper provided a comprehensive examination of the Koopman operator theory, a mathematical tool that can enable global linearization by elevating an original nonlinear system to a higher-dimensional linear space. We reviewed recent breakthroughs across a spectrum of robotic platforms, demonstrating how Koopman-based implementations are instrumental in addressing challenges related to robot runtime learning, modeling, and control. The paper meticulously described several foundational and more advanced yet key components regarding its deployment onto robots, including aspects related to data collection, the selection of lifting functions, and controller synthesis.

Despite the expanding use of the Koopman operator in modern robotics, there are still considerable challenges. The points that follow contain a curated summary of various discussions and insights derived from the authors' collective past and ongoing research. Certainly, the list is not exhaustive. However, we believe that it offers a solid starting point for future research directions catering to both practical implementations as well as further theoretical advancements.

1) **Incorporating Constraints into Koopman Space:** As in any optimization problem, incorporating the various (often conflicting) constraints appropriately is crucial. While we have outlined some methods to do so, how to lift different types of constraints from the original space into the Koopman space requires more research. We believe that exploiting the algebraic and geometric structure of the Koopman operator can play a key role.

2) **Formal Accuracy Measures in General Hilbert and Banach Spaces:** A significant limitation in Koopman-based modeling is the lack of formal accuracy measures applicable to general Hilbert and Banach spaces. Developing such measures is essential for optimization-based dictionary learning and ensuring the safety and reliability of applications in critical domains. These measures

should be independent of the choice of basis for finite-dimensional subspaces, providing a standardized way to assess model accuracy and performance.

3) **Identification of Classes of Lifting Functions:** To ensure the effectiveness of Koopman-based control, it is crucial to identify classes of appropriate observable functions for lifting. While we have outlined different ways that have been proposed already, methods for higher-dimensional dynamical systems, or for cases where certain properties (like invariance) must be maintained, remain underdeveloped.

4) **Stochastic Simulation and Belief-Space Planning:** Real-world cases involve uncertainty, leading to the need for stochastic simulation and belief-space planning. Handling multi-modal distributions and generating robust plans under uncertainty are vital research areas. Advancements in techniques like the stochastic Koopman operator [92], [142], [143] can pave the way to incorporate stochasticity into Koopman-based control methods. However, more work related to efficient implementation and the study of reliability and adaptability is needed.

5) **Sampling Rate Selection:** One of the ongoing inquiries surrounding the application of Koopman-based approaches in robotic systems pertains to the selection of an optimal sampling rate. Particularly crucial for online learning, the sampling rate directly influences the quality of the derived Koopman operator. Additionally, it governs the error bounds associated with predictions. Selecting the rate for data collection and control signal execution will be vital to the performance of the robot.

6) **Systems with Control Inputs**: In forced systems, the linearity of observables does not automatically imply the linearity of the control inputs. For this reason, selected observables in existing works are often constrained to be linear functions of the input. However, this constraint can limit the quality of the resulting Koopman-based approximation since it may be too restrictive and fail to comprehensively capture the intricate interplay between input and state variables. A direct implication is that any convergence properties of the original control system may be altered, or even lost. In this regard, we believe there is much to be gained by exploiting the concept of the input-state separable form described earlier.

7) **Extension to Hybrid Systems**: We have shown how Koopman operator theory can handle both discrete-time and continuous-time dynamical systems independently. Yet, the case of hybrid dynamical systems, which encapsulates many mechatronic and robotic systems, is more subtle and has received less attention to date. A very recent effort has shown that, under certain conditions, unforced heterogeneous systems (i.e. a class of hybrid systems) can still be lifted into a higher-dimensional linear system via Koopman-based method [16]. However, extending to systems with inputs and systems whose evolution is governed by broader hybrid dynamics remains open.

8) **Uncertainty in Lifted Features:** Lastly yet importantly, the assumption made in prior work regarding the zero-mean, Gaussian uncertainty model of the approximate Koopman operator should be revisited as well. Recent advancements have demonstrated that one can still operate and plan controllers that are robust to this specific choice of uncertainty model, as done in [95], [144]. Indeed, it is not clear if the nature of uncertainty in observables, especially with the addition of control input, can be accurately modeled as a normal Gaussian distribution. As an example, one might have a system where the states x_t are well modeled as Gaussian (in this case the underlying dynamics might also be linear). It is not clear whether, under which conditions, Gaussian distributions pushed into the observable lifted space should form Gaussians. Thus, finding cases where lifting observables also preserves some statistical structure remains open.

In conclusion, the application of Koopman operator theory in robotics represents a transformative shift in how we approach the modeling, control, and optimization of complex robotic systems in support of runtime learning. By leveraging the power of linear representation for inherently nonlinear dynamics, Koopman-based methods provide a robust and computationally efficient framework that addresses many of the limitations of traditional approaches. The theoretical advancements and practical implementations discussed in this review demonstrate the versatility and efficacy of Koopman operators across a wide array of robotic domains, including aerial, legged, wheeled, underwater, soft, and manipulator robots. As we continue to refine these techniques and explore new applications, the potential for Koopman operator theory to drive innovation in robotics is immense. Future research will undoubtedly focus on overcoming current challenges, such as incorporating constraints, improving real-time performance, and enhancing robustness to uncertainty. By doing so, we can fully harness the capabilities of Koopman theory to enable smarter, more adaptive, and more efficient robotic systems that can operate autonomously in complex, dynamic environments.

REFERENCES

[1] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[2] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research (IJRR)*, vol. 32, no. 11, pp. 1238–1274, 2013.

[3] M. Mandapuram, S. S. Gutlapalli, A. Bodepudi, and M. Reddy, "Investigating the prospects of generative artificial intelligence," *Asian Journal of Humanity, Art and Literature*, vol. 5, no. 2, pp. 167–174, 2018.

[4] T. B. Sheridan, "Human–robot interaction: Status and challenges," *Human Factors*, vol. 58, no. 4, pp. 525–532, 2016.

[5] C. Lee, M. Kim, Y. J. Kim, N. Hong, S. Ryu, H. J. Kim, and S. Kim, "Soft robot review," *International Journal of Control, Automation and Systems*, vol. 15, pp. 3–15, 2017.

[6] L. Christensen, J. de Gea Fernández, M. Hildebrandt, C. E. S. Koch, and B. Wehbe, "Recent advances in ai for navigation and control of underwater robots," *Current Robotics Reports*, vol. 3, no. 4, pp. 165–175, 2022.

[7] K. Karydis, I. Poulakakis, J. Sun, and H. G. Tanner, "Probabilistically valid stochastic extensions of deterministic models for systems with uncertainty," *The International Journal of Robotics Research (IJRR)*, vol. 34, no. 10, pp. 1278–1295, 2015.

[8] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Transactions on Robotics (T-RO)*, vol. 36, no. 3, pp. 582–596, 2020.

[9] A. Godfrey, V. Hetherington, H. Shum, P. Bonato, N. Lovell, and S. Stuart, "From a to z: Wearable technology explained," *Maturitas*, vol. 113, pp. 40–47, 2018.

[10] E. Kokkoni, Z. Liu, and K. Karydis, "Development of a soft robotic wearable device to assist infant reaching," *Journal of Engineering and Science in Medical Diagnostics and Therapy*, vol. 3, pp. 021 109–1, 2020.

[11] C. Mucchiani, Z. Liu, I. Sahin, J. Dube, L. Vu, E. Kokkoni, and K. Karydis, "Closed-loop position control of a pediatric soft robotic wearable device for upper extremity assistance," in *31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2022, pp. 1514–1519.

[12] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Autonomous Robots*, vol. 35, pp. 51–76, 2013.

[13] X. Kan, H. Teng, and K. Karydis, "Online exploration and coverage planning in unknown obstacle-cluttered environments," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 4, pp. 5969–5976, 2020.

[14] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research (IJRR)*, vol. 37, no. 4-5, pp. 421–436, 2018.

[15] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.

[16] H. H. Asada, "Global, unified representation of heterogeneous robot dynamics using composition operators: A Koopman direct encoding method," *IEEE/ASME Transactions on Mechatronics*, 2023.

[17] S. A. Deka, A. M. Valle, and C. J. Tomlin, "Koopman-based neural lyapunov functions for general attractors," in *IEEE Conference on Decision and Control (CDC)*, 2022, pp. 5123–5128.

[18] S. Brunton and J. Kutz, *Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[19] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven approximations of dynamical systems operators for control," *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*, pp. 197–234, 2020.

[20] S. E. Otto and C. W. Rowley, "Koopman operators for estimation and control of dynamical systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 59–87, 2021.

[21] P. Bevanda, S. Sosnowski, and S. Hirche, "Koopman operator dynamical models: Learning, analysis and control," *Annual Reviews in Control*, vol. 52, pp. 197–212, 2021.

[22] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern Koopman theory for dynamical systems," *SIAM Review*, vol. 64, no. 2, 2022.

[23] A. T. Taylor, T. A. Berrueta, and T. D. Murphey, "Active learning in robotics: A review of control principles," *Mechatronics*, vol. 77, p. 102576, 2021.

[24] L. Shi, Z. Liu, and K. Karydis, "Koopman operators for modeling and control of soft robotics," *Current Robotics Reports*, pp. 1–9, 2023.

[25] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.

[26] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing Koopman theory to allow for inputs and control," *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 909–930, 2018.

[27] D. Bruder, X. Fu, and R. Vasudevan, "Advantages of bilinear Koopman realizations for the modeling and control of systems with unknown dynamics," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 3, pp. 4369–4376, 2021.

[28] S. Pan, E. Kaiser, B. M. de Silva, J. N. Kutz, and S. L. Brunton, "Pykoopman: A python package for data-driven approximation of the koopman operator," *Journal of Open Source Software*, vol. 9, no. 94, p. 5881, 2024.

[29] C. Folkestad and J. W. Burdick, "Koopman NMPC: Koopman-based learning and nonlinear model predictive control of control-affine systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 7350–7356.

[30] I. Abraham and T. D. Murphey, "Active learning of dynamics for data-driven control using Koopman operators," *IEEE Transactions on Robotics (T-RO)*, vol. 35, no. 5, pp. 1071–1083, 2019.

[31] C. Folkestad, D. Pastor, and J. W. Burdick, "Episodic Koopman learning of nonlinear robot dynamics with application to fast multirotor landing," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9216–9222.

[32] M. Budišić, R. Mohr, and I. Mezić, "Applied Koopmanism," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 4, 2012.

[33] A. Mauroy, Y. Susuki, and I. Mezić, *Koopman operator in systems and control*. Springer, 2020.

[34] S. Klus, P. Koltai, and C. Schütte, "On the numerical approximation of the Perron-Frobenius and Koopman operator," *Journal of Computational Dynamics*, vol. 3, no. 1, pp. 51–79, 2016.

[35] M. Korda and I. Mezić, "On convergence of extended dynamic mode decomposition to the Koopman operator," *Journal of Nonlinear Science*, vol. 28, no. 2, pp. 687–710, 2018.

[36] M. Haseli and J. Cortés, "Learning Koopman eigenfunctions and invariant subspaces from data: Symmetric Subspace Decomposition," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3442–3457, 2022.

[37] ——, "Parallel learning of Koopman eigenfunctions and invariant subspaces for accurate long-term prediction," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 4, pp. 1833–1845, 2021.

[38] ——, "Generalizing dynamic mode decomposition: balancing accuracy and expressiveness in Koopman approximations," *Automatica*, vol. 153, p. 111001, 2023.

[39] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.

[40] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: theory and applications," *"Journal of Computational Dynamics"*, 2014.

[41] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing Koopman theory to allow for inputs and control," *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 909–930, 2018.

[42] A. Surana, "Koopman operator based observer synthesis for control-affine nonlinear systems," in *IEEE Conference on Decision and Control (CDC)*, 2016, pp. 6492–6499.

[43] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[44] M. Haseli and J. Cortés, "Modeling nonlinear control systems via Koopman control family: universal forms and subspace invariance proximity," *https://arxiv.org/abs/2307.15368*, 2024, submitted.

[45] S. Peitz and S. Klus, "Koopman operator-based model reduction for switched-system control of PDEs," *Automatica*, vol. 106, pp. 184–191, 2019.

[46] M. Bakhtiaridoust, M. Yadegar, and N. Meskin, "Data-driven fault detection and isolation of nonlinear systems using deep learning for Koopman operator," *ISA transactions*, vol. 134, pp. 200–211, 2023.

[47] R. Mahmood, J. Lucas, J. M. Alvarez, S. Fidler, and M. Law, "Optimizing data collection for machine learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 29 915–29 928, 2022.

[48] J. Chen, Y. Dang, and J. Han, "Offset-free model predictive control of a soft manipulator using the Koopman operator," *Mechatronics*, vol. 86, p. 102871, 2022.

[49] Z. C. Guo, V. Korotkine, J. R. Forbes, and T. D. Barfoot, "Koopman linearization for data-driven batch state estimation of control-affine systems," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 866–873, 2021.

[50] M. L. Castaño, A. Hess, G. Mamakoukas, T. Gao, T. Murphey, and X. Tan, "Control-oriented modeling of soft robotic swimmer with Koopman operators," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2020, pp. 1679–1685.

[51] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Data-driven control of soft robots using Koopman operator theory," *IEEE Transactions on Robotics (T-RO)*, vol. 37, no. 3, pp. 948–961, 2020.

[52] J. Wang, B. Xu, J. Lai, Y. Wang, C. Hu, H. Li, and A. Song, "An improved Koopman-MPC framework for data-driven modeling and control of soft actuators," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 2, pp. 616–623, 2022.

[53] L. Shi, H. Teng, X. Kan, and K. Karydis, "A data-driven hierarchical control structure for systems with uncertainty," in *IEEE Conference on Control Technology and Applications (CCTA)*, 2020, pp. 57–63.

[54] L. Shi and K. Karydis, "Enhancement for robustness of Koopman operator-based data-driven mobile robotic systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 2503–2510.

[55] M. Haseli and J. Cortés, "Approximating the Koopman operator using noisy data: noise-resilient extended dynamic mode decomposition," in *American Control Conference (ACC)*, Philadelphia, PA, Jul. 2019, pp. 5499–5504.

[56] L. J. Gleser, "Estimation in a multivariate errors in variables regression model: large sample results," *The Annals of Statistics*, pp. 24–44, 1981.

[57] A. Kukush and S. V. Huffel, "Consistency of elementwise-weighted total least squares estimator in a multivariate errors-in-variables model AX= B," *Metrika*, vol. 59, no. 1, pp. 75–97, 2004.

[58] I. Markovsky, M. L. Rastello, A. Premoli, A. Kukush, and S. V. Huffel, "The element-wise weighted total least-squares problem," *Computational Statistics & Data Analysis*, vol. 50, no. 1, pp. 181–209, 2006.

[59] S. T. M. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley, "Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition," *Experiments in Fluids*, vol. 57, no. 3, p. 42, 2016.

[60] M. S. Hemati, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, "De-biasing the dynamic mode decomposition for applied Koopman spectral analysis of noisy datasets," *Theoretical and Computational Fluid Dynamics*, vol. 31, no. 4, pp. 349–368, 2017.

[61] K. Aishima, "Strong consistency of the projected total least squares dynamic mode decomposition for datasets with random noise," *Japan Journal of Industrial and Applied Mathematics*, vol. 40, no. 1, pp. 691–707, 2023.

[62] O. Azencot, W. Yin, and A. Bertozzi, "Consistent dynamic mode decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 3, pp. 1565–1585, 2019.

[63] L. Shi and K. Karydis, "ACD-EDMD: Analytical construction for dictionaries of lifting functions in Koopman operator-based nonlinear robotic systems," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 906–913, 2021.

[64] G. Mamakoukas, M. L. Castano, X. Tan, and T. D. Murphey, "Derivative-based Koopman operators for real-time control of robotic systems," *IEEE Transactions on Robotics (T-RO)*, vol. 37, no. 6, pp. 2173–2192, 2021.

[65] M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz, "Time-delay observables for Koopman: Theory and applications," *SIAM Journal on Applied Dynamical Systems*, vol. 19, no. 2, pp. 886–917, 2020.

[66] M. Korda and I. Mezic, "Optimal construction of Koopman eigenfunctions for prediction and control," *IEEE Transactions on Automatic Control*, vol. 65, no. 12, pp. 5114–5129, 2020.

[67] S. Pan, N. Arnold-Medabalimi, and K. Duraisamy, "Sparsity-promoting algorithms for the discovery of informative Koopman-invariant subspaces," *Journal of Fluid Mechanics*, vol. 917, p. A18, 2021.

[68] E. Yeung, S. Kundu, and N. Hodas, "Learning deep neural network representations for Koopman operators of nonlinear dynamical systems," in *American Control Conference (ACC)*, 2019, pp. 4832–4839.

[69] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 10, p. 103111, 2017.

[70] N. Takeishi, Y. Kawahara, and T. Yairi, "Learning Koopman invariant subspaces for dynamic mode decomposition," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 1130–1140.

[71] M. Haseli and J. Cortés, "Temporal forward-backward consistency, not residual error, measures the prediction accuracy of extended dynamic mode decomposition," *IEEE Control Systems Letters*, vol. 7, pp. 649–654, 2023.

[72] E. C. Balreira, O. Kosheleva, and V. Kreinovich, "Algorithmics of checking whether a mapping is injective, surjective, and/or bijective," *Constraint programming and decision making*, pp. 1–7, 2014.

[73] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Communications*, vol. 9, no. 1, p. 4950, 2018.

[74] S. E. Otto and C. W. Rowley, "Linearly recurrent autoencoder networks for learning dynamics," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 1, pp. 558–593, 2019.

[75] O. Azencot, N. B. Erichson, V. Lin, and M. Mahoney, "Forecasting sequential data using consistent Koopman autoencoders," in *International Conference on Machine Learning*. PMLR, 2020, pp. 475–485.

[76] M. Sznaier, "A data driven, convex optimization approach to learning Koopman operators," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 436–446.

[77] A. Mauroy and I. Mezić, "Global stability analysis using the eigenfunctions of the Koopman operator," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3356–3369, 2016.

[78] B. Yi and I. R. Manchester, "On the equivalence of contraction and Koopman approaches for nonlinear stability and control," in *IEEE Conference on Decision and Control (CDC)*, 2021, pp. 4609–4614.

[79] S. A. Deka, A. M. Valle, and C. J. Tomlin, "Koopman-based neural Lyapunov functions for general attractors," in *IEEE Conference on Decision and Control (CDC)*, 2022, pp. 5123–5128.

[80] L. Zheng, X. Liu, Y. Xu, W. Hu, and C. Liu, "Data-driven estimation for region of attraction for transient stability using Koopman operator," *CSEE Journal of Power and Energy Systems*, 2022.

[81] P. Bevanda, M. Beier, S. Kerz, A. Lederer, S. Sosnowski, and S. Hirche, "Diffeomorphically learning stable Koopman operators," *IEEE Control Systems Letters*, vol. 6, pp. 3427–3432, 2022.

[82] F. Fan, B. Yi, D. Rye, G. Shi, and I. R. Manchester, "Learning stable Koopman embeddings," in *American Control Conference (ACC)*. IEEE, 2022, pp. 2742–2747.

[83] G. Mamakoukas, I. Abraham, and T. D. Murphey, "Learning stable models for prediction and control," 2020.

[84] I. Abraham, G. De La Torre, and T. D. Murphey, "Model-based control using Koopman operators," in *Robotics: Science and Systems, RSS*. MIT Press Journals, 2017.

[85] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[86] J. A. Paulson, A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz, "Fast stochastic model predictive control of high-dimensional systems," in *IEEE Conference on Decision and Control (CDC)*, 2014, pp. 2802–2809.

[87] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," in *UKACC 12th International Conference on Control (CONTROL)*. IEEE, 2018, pp. 339–339.

[88] E. Polak, *Optimization: algorithms and consistent approximations*. Springer Science & Business Media, 2012, vol. 124.

[89] H. Cramér, "A contribution to the theory of statistical estimation," *Scandinavian Actuarial Journal*, vol. 1946, no. 1, pp. 85–94, 1946.

[90] C. Radhakrishna Rao, "Information and accuracy attainable in the estimation of statistical parameters," *Bulletin of the Calcutta Mathematical Society*, vol. 37, no. 3, pp. 81–91, 1945.

[91] Y. Xiao, X. Zhang, X. Xu, X. Liu, and J. Liu, "Deep neural networks with Koopman operators for modeling and control of autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 135–146, 2022.

[92] M. Han, J. Euler-Rolle, and R. K. Katzschmann, "Desko: Stability-assured robust control with a deep stochastic Koopman operator," in *International Conference on Learning Representations (ICLR)*, 2021.

[93] Y. Wang, Y. Yang, Y. Pu, and C. Manzie, "Data-driven predictive tracking control based on Koopman operators," *arXiv preprint arXiv:2208.12000*, 2022.

[94] N. Kochdumper and S. Bak, "Conformant synthesis for Koopman operator linearized control systems," in *IEEE Conference on Decision and Control (CDC)*, 2022, pp. 7327–7332.

[95] G. Mamakoukas, S. Di Cairano, and A. P. Vinod, "Robust model predictive control with data-driven Koopman operators," in *American Control Conference (ACC)*. IEEE, 2022, pp. 3885–3892.

[96] K. Karydis and V. Kumar, "Energetics in robotic flight at small scales," *Interface focus*, vol. 7, no. 1, p. 20160088, 2017.

[97] X. Kan, J. Thomas, H. Teng, H. G. Tanner, V. Kumar, and K. Karydis, "Analysis of ground effect for small-scale uavs in forward flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3860–3867, 2019.

[98] C. Folkestad, S. X. Wei, and J. W. Burdick, "Koopnet: Joint learning of Koopman bilinear models and function dictionaries with application to quadrotor trajectory tracking," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1344–1350.

[99] P. V. Borges, T. Peynot, S. Liang, B. Arain, M. Wildie, M. Minareci, S. Lichman, G. Samvedi, I. Sa, N. Hudson *et al.*, "A survey on terrain traversability analysis for autonomous ground vehicles: Methods, sensors, and challenges." *Field Robotics*, vol. 2, no. 1, pp. 1567–1627, 2022.

[100] C. Ren, H. Jiang, C. Li, W. Sun, and S. Ma, "Koopman-operator-based robust data-driven control for wheeled mobile robots," *IEEE/ASME Transactions on Mechatronics*, 2022.

[101] J. Carpentier and P.-B. Wieber, "Recent progress in legged robots locomotion control," *Current Robotics Reports*, vol. 2, no. 3, pp. 231–238, 2021.

[102] A. Krolicki, D. Rufino, A. Zheng, S. S. Narayanan, J. Erb, and U. Vaidya, "Modeling quadruped leg dynamics on deformable terrains using data-driven Koopman operators," *IFAC-PapersOnLine*, vol. 55, no. 37, pp. 420–425, 2022.

[103] W. Manzoor, S. Rawashdeh, and A. Mohammadi, "Vehicular applications of Koopman operator theory–a survey," *IEEE Access*, 2023.

[104] V. Cibulka, T. Haniš, and M. Hromčík, "Data-driven identification of vehicle dynamics using Koopman operator," in *22nd International Conference on Process Control (PC19)*. IEEE, 2019, pp. 167–172.

[105] M. Švec, Š. Ileš, and J. Matuško, "Model predictive control of vehicle dynamics based on the Koopman operator with extended dynamic mode decomposition," in *22nd IEEE International Conference on Industrial Technology (ICIT)*, vol. 1, 2021, pp. 68–73.

[106] E. Sheng, S. Yu, H. Chang, Y. Zhang, Y. Li, and Y. Hao, "Automated driving control of vehicles with guidances," in *IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 1817–1822.

[107] S. Yu, E. Sheng, Y. Zhang, Y. Li, H. Chen, and Y. Hao, "Efficient nonlinear model predictive control of automated vehicles," *Mathematics*, vol. 10, no. 21, p. 4163, 2022.

[108] J. Buzhardt and P. Tallapragada, "A Koopman operator approach for the vertical stabilization of an off-road vehicle," *IFAC-PapersOnLine*, vol. 55, no. 37, pp. 675–680, 2022.

[109] S. Yu, C. Shen, and T. Ersal, "Autonomous driving using linear model predictive control with a Koopman operator based bilinear vehicle model," *IFAC-PapersOnLine*, vol. 55, no. 24, pp. 254–259, 2022.

[110] W. Guo, S. Zhao, H. Cao, B. Yi, and X. Song, "Koopman operator-based driver-vehicle dynamic model for shared control systems," *Applied Mathematical Modelling*, vol. 114, pp. 423–446, 2023.

[111] N. Chakraborty, A. Hasan, S. Liu, T. Ji, W. Liang, D. L. McPherson, and K. Driggs-Campbell, "Structural attention-based recurrent variational autoencoder for highway vehicle anomaly detection," in *International Conference on Autonomous Agents and Multiagent Systems*, 2023, pp. 1125–1134.

[112] G. Mamakoukas, O. Xherija, and T. Murphey, "Memory-efficient learning of stable linear dynamical systems for prediction and control," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 527–13 538, 2020.

[113] A. Sinha and Y. Wang, "Koopman operator-based knowledge-guided reinforcement learning for safe human-robot interaction," *Frontiers in Robotics and AI*, vol. 9, p. 779194, 2022.

[114] H. Yin, M. C. Welle, and D. Kragic, "Embedding Koopman optimal control in robot policy learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 392–13 399.

[115] I. Mezić, "Analysis of fluid flows via spectral properties of the Koopman operator," *Annual Review of Fluid Mechanics*, vol. 45, pp. 357–378, 2013.

[116] C. Rodwell and P. Tallapragada, "Embodied hydrodynamic sensing and estimation using Koopman modes in an underwater environment," in *American Control Conference (ACC)*. IEEE, 2022, pp. 1632–1637.

[117] S. Li, Z. Xu, J. Liu, and C. Xu, "Learning-based extended dynamic mode decomposition for addressing path-following problem of underactuated ships with unknown dynamics," *International Journal of Control, Automation and Systems*, vol. 20, no. 12, pp. 4076–4089, 2022.

[118] P. Bevanda, M. Beier, S. Heshmati-Alamdari, S. Sosnowski, and S. Hirche, "Towards data-driven LQR with Koopmanizing flows," *IFAC-PapersOnLine*, vol. 55, no. 15, pp. 13–18, 2022.

[119] O. Nelles, *Nonlinear System Identification*. Springer, 2001.

[120] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[121] D. Bruder, B. Gillespie, C. D. Remy, and R. Vasudevan, "Modeling and control of soft robots using the Koopman operator and model predictive control," in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.

[122] D. Bruder, C. D. Remy, and R. Vasudevan, "Nonlinear system identification of soft robot dynamics using Koopman operator theory," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6244–6250.

[123] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Koopman-based control of a soft continuum manipulator under variable loading conditions," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 4, pp. 6852–6859, 2021.

[124] D. A. Haggerty, M. J. Banks, P. C. Curtis, I. Mezić, and E. W. Hawkes, "Modeling, reduction, and control of a helically actuated inertial soft robotic arm via the Koopman operator," *arXiv preprint arXiv:2011.07939*, 2020.

[125] E. Kamenar, N. Ćrnjarić-Žic, D. Haggerty, S. Zelenika, E. W. Hawkes, and I. Mezić, "Prediction of the behavior of a pneumatic soft robot based on Koopman operator theory," in *43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*. IEEE, 2020, pp. 1169–1173.

[126] L. Shi, C. Mucchiani, and K. Karydis, "Online modeling and control of soft multi-fingered grippers via Koopman operator theory," in *IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2022, pp. 1946–1952.

[127] H. Wang, W. Liang, B. Liang, H. Ren, Z. Du, and Y. Wu, "Robust position control of a continuum manipulator based on selective approach and Koopman operator," *IEEE Transactions on Industrial Electronics*, 2023.

[128] N. S. Selby and H. H. Asada, "Learning of causal observable functions for Koopman-dfl lifting linearization of nonlinear controlled systems and its application to excavation automation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 4, pp. 6297–6304, 2021.

[129] F. E. Sotiropoulos and H. H. Asada, "Dynamic modeling of bucket-soil interactions using Koopman-DFL lifting linearization for model predictive contouring control of autonomous excavators," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 1, pp. 151–158, 2021.

[130] T. Goyal, S. Hussain, E. Martinez-Marroquin, N. A. Brown, and P. K. Jamwal, "Learning Koopman embedding subspaces for system identification and optimal control of a wrist rehabilitation robot," *Transactions on Industrial Electronics*, vol. 70, no. 7, pp. 7092–7101, 2022.

[131] A. Kalinowska, T. A. Berrueta, A. Zoss, and T. Murphey, "Data-driven gait segmentation for walking assistance in a lower-limb assistive device," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1390–1396.

[132] H. H. Asada, "Global, unified representation of heterogenous robot dynamics using composition operators: A Koopman direct encoding method," *IEEE/ASME Transactions on Mechatronics*, 2023.

[133] K. Zhu, L. Wu, C. Ren, X. Li, and S. Ma, "Path following controller for snake robots based on data-driven MPC and extended state observer," in *41st Chinese Control Conference (CCC)*. IEEE, 2022, pp. 5592–5597.

[134] W. Savoie, T. A. Berrueta, Z. Jackson, A. Pervan, R. Warkentin, S. Li, T. D. Murphey, K. Wiesenfeld, and D. I. Goldman, "A robot made of robots: Emergent transport and control of a smarticle ensemble," *Science Robotics*, vol. 4, no. 34, p. eaax4316, 2019.

[135] X. Kong, H. Mo, E. Dong, Y. Liu, and D. Sun, "Automatic tracking of surgical instruments with a continuum laparoscope using data-driven control in robotic surgery," *Advanced Intelligent Systems*, vol. 5, no. 2, p. 2200188, 2023.

[136] G. Tao and Q. Zhao, "Koopman system approximation based optimal control of multiple robots–part I: Concepts and formulations," *arXiv preprint arXiv:2305.03777*, 2023.

[137] Q. Zhao and G. Tao, "Koopman system approximation based optimal control of multiple robots–part II: Simulations and evaluations," *arXiv preprint arXiv:2305.03778*, 2023.

[138] Y. Wang, T. Baba, and T. Hikihara, "Realization of lattice formation in nonlinear two-dimensional potential by mobile robots," *CoRR*, 2022.

[139] Y. Wang, M. J. Banks, I. Mezic, and T. Hikihara, "Trajectory estimation in unknown nonlinear manifold using Koopman operator theory," *arXiv preprint arXiv:2312.05428*, 2023.

[140] W. Zhan, Z. Miao, Y. Chen, Y. Feng, and Y. Wang, "Koopman operation-based leader-following formation control for nonholonomic mobile robots under denial-of-service attacks," in *IEEE International Conference on Unmanned Systems (ICUS)*, 2023, pp. 110–115.

[141] E. Hansen, S. L. Brunton, and Z. Song, "Swarm modeling with dynamic mode decomposition," *IEEE Access*, vol. 10, pp. 59 508–59 521, 2022.

[142] M. Wanner and I. Mezic, "Robust approximation of the stochastic Koopman operator," *SIAM Journal on Applied Dynamical Systems*, vol. 21, no. 3, pp. 1930–1951, 2022.

[143] N. Takeishi, Y. Kawahara, and T. Yairi, "Subspace dynamic mode decomposition for stochastic Koopman analysis," *Physical Review E*, vol. 96, no. 3, p. 033310, 2017.

[144] S. Sinha, B. Huang, and U. Vaidya, "On robust computation of Koopman operator and prediction in random dynamical systems," *Journal of Nonlinear Science*, vol. 30, no. 5, pp. 2057–2090, 2020.