Koopman Operator for Control

Yuhan Zhao

NYU TANDON SCHOOL OF ENGINEERING

LARX Group Metting

Outline

Outline

What is Koopman Operator

Given an autonomous dynamical system

$$x_{t+1} = f(x_t), \tag{1}$$

where $x_t \in \mathcal{M} \subset \mathbb{R}^n$, $f : \mathcal{M} \to \mathcal{M}$.

Let $g : \mathcal{M} \to \mathbb{R}$ be a scalar valued function (called an **observable**) in the Hilbert space $\mathcal{H}$. We define the Koopman operator $\mathcal{K}$

$$\mathcal{K}g = g \circ f. \tag{2}$$

- $\mathcal{K}$ maps one function to another function.
- $\mathcal{K}$ is an infinite dimensional linear operator.

$$\mathcal{K}(\alpha_1 g_1 + \alpha_2 g_2) = \alpha_1 \mathcal{K} g_1 + \alpha_2 \mathcal{K} g_2, \quad \forall g_1, g_2 \in \mathcal{H}, \alpha_1, \alpha_2 \in \mathbb{R}.$$

Explanation

$\mathcal{K}$ governs the evolution of g along the trajectory of dynamical systems.

For discrete dynamical system (1):

- Trajectory is a sequence $x_t, x_{t+1}, x_{t+2}, \ldots$
- Koopman operator (2) becomes

$$\mathcal{K}g(x_t) = g(f(x_t)) = g(x_{t+1}), \tag{3}$$

which holds for any $g \in \mathcal{H}$ and any $x_t \in \mathcal{M}$.

A new linear dynamical system in an infinite dimensional space:

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

Why Infinite Dimension

- $\mathcal{K}$ acts on functions rather than finite dimensional x_t.[1]
- $\mathcal{K}$ has infinite dimensional bases. No matrix representation[2].

Koopman operator is linear, define eigenvalue and eigen-function:

$$\mathcal{K}\phi_i = \lambda_i \phi_i. \tag{4}$$

- $\mathcal{K}$ has infinite eigen-functions $\{\phi_1, \phi_2, \dots\}$, $\phi_i : \mathcal{M} \to \mathbb{R}$.
- Eigen-functions serve as bases of the Hilbert space $\mathcal{H}$.
- For any observable $g \in \mathcal{H}$,

$$g(x) = \sum_{i=1}^{\infty} c_i \phi_i(x),$$

where c_i is the coordinate associated with the basis ϕ_i.

[1] We can evaluate the value at x_t.

[2] Any finite dimensional linear operator has a matrix representation.

Why Infinite Dimension

For one-step evolution,

$$\sum_{i=1}^{\infty} \tilde{c}_i \phi_i(x_{t+1}) := g(x_{t+1})$$
$$= \mathcal{K} g(x_t) = \sum_{i=0}^{\infty} c_i \lambda_i \phi_i(x_t).$$

Observation

- The coordinates $\tilde{c}_i$ and c_i using eigen-function bases are infinite although the coordinate transformation is finite: $\tilde{c}_i = \lambda_i c_i$.
- The coordinate transformation can be infinite if we use other function bases.

Koopman Invariant Subspace

Definition

Koopman invariant space is defined as $\mathcal{G} \subset \mathcal{H}$ such that $\mathcal{K}g \in \mathcal{G}$ $\forall g \in \mathcal{G}$.

If $\mathcal{G}$ is spanned by finite functions $\{\psi_1, \dots, \psi_p\}$, then $\mathcal{K}$ becomes a finite dimensional linear operator. i.e., for any $g \in \mathcal{G}$, there exists $\{\alpha_i\}_{i=1}^p$ and $\{\beta_i\}_{i=1}^p$ such that

$$g = \alpha_1\psi_1 + \cdots + \alpha_p\psi_p, \quad \Rightarrow \quad \mathcal{K}g = \beta_1\psi_1 + \cdots + \beta_p\psi_p.$$

- Any combination of Koopman eigen-functions forms an invariant subspace.
- $\mathcal{K}$ has a finite matrix representation if we know $\{\psi_1, \dots, \psi_p\}$.
- There can be multiple or zero Koopman invariant subspaces, depending on f (Brunton et al., 2016a).

Multiple Observables

An observable g is a scalar valued function.

We can define multiple observables for more observations:

$$\mathbf{g} = \begin{bmatrix} g_1 & g_2 & \cdots & g_m \end{bmatrix}^\mathsf{T}.$$

Each g_j follows the same reasoning. For example, if $\mathbf{g} \in \text{span}\{\psi_1, \ldots, \psi_p\}$, there exists $[\alpha_{ij}] \in \mathbb{R}^{m \times p}, [\beta]_{ij} \in \mathbb{R}^{m \times p}$ such that $\mathcal{K}\mathbf{g} = [\beta]_{ij}\psi$.

Two Examples

Example 1

Consider the dynamical system with $x \in \mathbb{R}^2$:

$$x_{1,t+1} = \mu x_{1,t}, \quad x_{2,t+1} = \lambda(x_{2,t} - x_{1,t}^2).$$

We define three observables with $g_1(x) = x_1$, $g_2(x) = x_2$, $g_3(x) = x_1^2$. Then $\mathcal{K}\mathbf{g}(x_t)$ can be represented by

$$\begin{bmatrix} g_1(x_{t+1}) \\ g_2(x_{t+1}) \\ g_3(x_{t+1}) \end{bmatrix} = \begin{bmatrix} \mu & 0 & 0 \\ 0 & \lambda & -\lambda \\ 0 & 0 & \mu^2 \end{bmatrix} \begin{bmatrix} g_1(x_t) \\ g_2(x_t) \\ g_3(x_t) \end{bmatrix}.$$

Let $y_t = \mathbf{g}(x_t)$, we have $y_{t+1} = Ay_t$.

Two Examples

Example 1 cont.

- $\{g_1, g_2, g_3\}$ spans a Koopman invariant subspace.
- We can capture the dynamics of trajectory x_t by $\mathbf{g}$.
- To infer x_t from y_t, we need to learn $\mathbf{g}^{-1}$. Trick: augment $\tilde{\mathbf{g}}$ by $[x, \mathbf{g}]$.

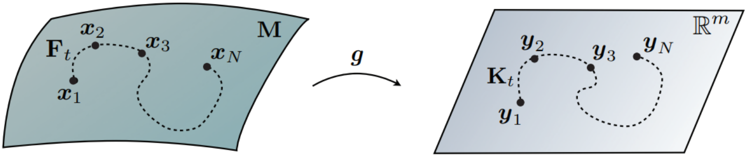

Figure: Illustration of x and y trajectories. From (Brunton and Kutz, 2019).

Two Example

Example 2

Consider the 1D logistic map

$$x_{t+1} = \beta x_t(1 - x_t).$$

If we select $y_1 := g_1(x) = x$, $y_2 := g_2(x) = x^2$, we have

$$\begin{bmatrix} y_{1,t+1} \\ y_{2,t+1} \end{bmatrix} = \begin{bmatrix} \beta & -\beta \\ ? & ? \end{bmatrix} \begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix}.$$

Note that

$$x_{t+1}^2 = \beta^2(x_t^2 - 2x_t^3 + x_t^4),$$

we need to add a new observable $y_3 := g_3(x) = x^3$ to capture the dynamics of x_t^2.

Two Examples

Example 2 cont.

$$
\begin{bmatrix} x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ \vdots \end{bmatrix}_{k+1} =
\begin{array}{c}
\begin{array}{cccccccccccc} x & x^2 & x^3 & x^4 & x^5 & x^6 & x^7 & x^8 & x^9 & x^{10} & & \end{array} \\
\begin{bmatrix}
\beta & -\beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & \beta^2 & -2\beta^2 & r^2 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & \beta^3 & -3\beta^3 & 3\beta^3 & \beta^3 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & \beta^4 & -4\beta^4 & 6\beta^4 & -4\beta^4 & \beta^4 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & \beta^5 & -5\beta^5 & 10\beta^5 & -10\beta^5 & 5\beta^5 & -\beta^5 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\end{array}
\begin{bmatrix} x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ \vdots \end{bmatrix}_{k}.
$$

- $\mathcal{K}$ has infinite dimensions under the chosen bases $\mathbf{g}$.
- Polynomial bases do not span a Koopman invariant space.
- Any truncation is a not good approximation for $\beta > 1$.

Challenges

Koopman operator theory

- "perceives" the state evolution through $\mathbf{g}$ and $\mathcal{K}$;
- provides an indirect approach for system identification;
- estimate $\mathbf{g}$ and $\mathcal{K}$ are sufficient; no need to estimate f.

Challenges to use Koopman operator:

- Identify the invariant space.
- Choose the right bases $\mathbf{g}$ (or eigen-functions) for the invariant space.
- Estimate Koopman operator $\mathcal{K}$ from observation data.
- Estimate states from observations if we want to know the real state.

Data-Driven Approaches

What we have is N snapshots of observation:

$$Y_1 = \begin{bmatrix} h(x_1) & h(x_2) & \cdots & h(x_N) \end{bmatrix}, \quad Y_2 = \begin{bmatrix} h(x_1') & h(x_2') & \cdots & h(x_N') \end{bmatrix},$$

where $h : \mathcal{M} \to \mathbb{R}^r$ is a state observation function[3].

- For full state observation: $h(x) = x$.
- $x_i' = f(x_i)$. For uniform time intervals, $x_i' = x_{i+1}$.

[3] The observable $\mathbf{g}$ is different from the observation function h. We can understand $\mathbf{g}$ as embedding functions of state x.

Data-Driven Approaches

$$\min_{\phi,\theta,K} \quad \mathcal{L}(\phi,\theta,K) = \|\mathbf{g}_\theta(z_\phi(Y_2)) - K\mathbf{g}_\theta(z_\phi(Y_1))\|_F^2 .$$

- z_ϕ: estimated state $\tilde{x}$ from observations.
- $\mathbf{g}_\theta$: basis of the Koopman invariant subspace.
- K: Koopman operator.

Finding Koopman invariant subspace:

- Learning invariant subspace bases (Takeishi et al., 2017).
- Learning eigen-functions (Lusch et al., 2018), K becomes diagnoal.

Finding Koopman operator:

- Dynamic mode decomposition, $K = \mathbf{g}_\theta(z_\phi(Y_2))\mathbf{g}_\theta(z_\phi(Y_1))^\dagger$.
- Direct learning (Yeung et al., 2019).

Related Literature

Koopman operator

- Begin with the seminal works (Koopman, 1931; Koopman and Neumann, 1932).
- First studied in physics and fluid mechanics (Mezić, 2005).
- Review on Koopman operator (Brunton et al., 2022; Bevanda et al., 2021).
- Survey on vehicular applications using Koopman operator (Manzoor et al., 2023).

Related Literature

Other approaches to system identification.

- Dynamic Mode Decoposition (DMD) (Tu et al., 2014)
  - find the transition matrix using full state observations.
- Extended DMD (eDMD) (Williams et al., 2015)
  - find the transition matrix using specified nonlinear bases;
  - closely related to Koopman operator.
- Sparse identification of nonlinear dynamics (SINDy) (Brunton et al., 2016b),
  - specify nonlinear bases of dynamical systems and identify the basis coefficients.
- Neural networks, RNN (Chen et al., 1990; Delgado et al., 1995)
  - find an input-output map directly. Black box model.
  - has a long history.

Outline

1. Koopman Operator
2. Koopman Operator for Control

Extending Koopman Operator for Control

Koopman operator

- Enables data-driven methods for indirect system identification.
- Works for autonomous dynamical systems.

We are interested in

- Data-driven methods for control.
- Applying Koopman operator to control.

Extending Koopman Operator for Control

Given a dynamical control system

$$x_{t+1} = f(x_t, u_t), \tag{5}$$

where $x_t \in \mathcal{M} \subset \mathbb{R}^n$, $u_t \in \mathcal{U} \subset \mathbb{R}^m$, $f : \mathcal{M} \times \mathcal{U} \to \mathcal{M}$.

Basic idea:

- Reflect the evolution of f and the impact of an arbitrary u.
- Extend the state space to $\mathcal{M} \times \mathcal{U}$.

We define the Koopman operator $\mathcal{K} : \mathcal{H} \to \mathcal{H}$

$$\mathcal{K}g(x_t, u_t) = g(f(x_t, u_t), u_{t+1}) = g(x_{t+1}, u_{t+1}), \tag{6}$$

where $g : \mathcal{M} \times \mathcal{U} \to \mathbb{R}$ is a observable in $\mathcal{H}$.

Control Variants

Different forms of control:

- Closed loop control: $u_t = h(x_t)$.

$$\mathcal{K}g(x_t, h(x_t)) = g(x_{t+1}, h(x_{t+1})).$$

  Reduce to Koopman operator for the associated autonomous system.

- Open loop control with internal control dynamics: $u_{t+1} = h(u_t)$.

$$\mathcal{K}g(x_t, u_t) = g(f(x_t, u_t), h(u_t)).$$

  Reduce to Koopman operator for the associated autonomous system where u is also a state.

- Open loop control with exogenous controls: unknown inputs.

Practical Constraints

Questions:

- Do we predict controls?

We can impose constraint on future input prediction by defining

$$\mathcal{K}g(x_t, u_t) = g(f(x_t, u_t), 0) = g(x_{t+1}, 0).$$

- Suitable for systems with Markov property.

In practice, the recovery of finite approximation of $\mathcal{K}$ for any arbitrary inputs requires a rich set of measurements, control profiles, and initial conditions.

Observable Bases

Next step: find Koopman invariant subspace and linearization.

We select $\mathbf{g} = [g_1, \ldots, g_p]$ such that $\mathcal{K}\mathbf{g} \in \text{span}\{g_1, \ldots, g_p\}$ such that

$$\mathbf{g}(x_{t+1}, u_{t+1}) \approx K\mathbf{g}(x_t, u_t).$$

- Eigen-functions are viable choices.

$$\mathcal{K}\phi_i(x, u) = \lambda_i \phi_i(x, u), \quad i = 1, 2, \ldots .$$

Attention

Choosing the correct observables is an art but critical.

Observable Bases — Special Structures

People assume special structures on the observable g for control.

- Partition g into two parts:

$$g(x,u) = g_x(x,u) + g_u(x,u).$$

- First part is only related to the state:

$$g_x(x,u) = g_x(x).$$

- Linear[4] of bilinear structure in the second part:

$$g_u(x,u) = a^\top u, \quad \text{or} \quad g_u(x,u) = \psi(x)(a^\top u).$$

[4] Linear structure is the most used case.

Observable Bases — Special Structures

Using linearity and causality (Korda and Mezić, 2018), we can write

$$\mathbf{g}(x, u) = \begin{bmatrix} \mathbf{g}_x(x) & u \end{bmatrix}^{\mathsf{T}}.$$

Then we have

$$\mathbf{g}(x_{t+1}, u_{t+1}) = \begin{bmatrix} \mathbf{g}_x(x_{t+1}) \\ u_{t+1} \end{bmatrix} = \begin{bmatrix} K_{xx} & K_{xu} \\ K_{ux} & K_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{g}_x(x_t) \\ u_t \end{bmatrix}. \tag{7}$$

We get rid of u_{k+1} since we do not predict controls, resulting in

$$\mathbf{g}_x(x_{t+1}) = K_{xx}\mathbf{g}_x(x_t) + K_{xu}u_t. \tag{8}$$

Transform Nonlinear Optimal Control Problem

Nonlinear optimal control problem (NOCP):

$$\begin{aligned} \min \quad & l_T(x_T) + \sum_{t=0}^{T-1} l_t(x_t) + u_t^\top R_t u_t + r_t^\top u_t \\ \text{s.t.} \quad & x_{t+1} = f(x_t, u_t), \quad t = 0, \dots, T-1, \\ & h_t(x_t) + c^\top u_t \leq 0, \quad t = 0, \dots, T-1, \\ & h_T(x_T) \leq 0. \end{aligned} \tag{9}$$

Tricks to select $\mathbf{g}_x$:

- Augment state itself: $\mathbf{g}_x = [x, \tilde{g}]$. ($C = [I \ 0]$, $x = Cg_x$).
- Augment nonlinear functions in the NOCP: $\mathbf{g}_x = [\tilde{g}, l_0, \dots, l_T, h_0, \dots, h_T]$.

Transform Nonlinear Optimal Control Problem

- Let $z_t = \mathbf{g}_x(x_t)$.
- Compute finite-dimensional Koopman operator K.
- Find A and B for dynamical systems.
- Convert nonlinear constraints.

Linearized optimal control problem:

$$
\begin{aligned}
\min \quad & y_T^\top Q_T y_T + \sum_{t=0}^{T-1} y_t^\top Q_t y_t + u_t^\top R_t u_t + r_t^\top u_t \\
\text{s.t.} \quad & y_{t+1} = A y_t + B u_t, \quad t = 0, \dots, T-1, \\
& E_t z_t + F_t u_t \le 0, \quad t = 0, \dots, T-1, \\
& z_0 = \mathbf{g}_x(x_0).
\end{aligned}
\tag{10}
$$

Example

Example 3

Consider the dynamical system with control:

$$x_{1,t+1} = \mu x_{1,t}, \quad x_{2,t+1} = \lambda(x_{2,t} - x_{1,t}^2) + \delta u_t.$$

We define $g_1(x,u) = x_1$, $g_2(x,u) = x_2$, $g_3(x,u) = x_1^2$, $g_4(x,u) = u$. Then $\mathbf{g}_x(x) = [g_1(x)\ g_2(x)\ g_3(x)]$. $\mathcal{K}\mathbf{g}(x_t, u_t)$ can be represented by

$$\mathbf{g}_x(x_{t+1}) = \begin{bmatrix} \mu & 0 & 0 \\ 0 & \lambda & -\lambda \\ 0 & 0 & \mu^2 \end{bmatrix} \mathbf{g}_x(x_t) + \begin{bmatrix} 0 \\ \delta \\ 0 \end{bmatrix} g_4(u_t).$$

Let $y_t = \mathbf{g}_x(x_t)$, we have $y_{t+1} = Ay_t + Bu_t$.

Discussions

Questions:

- Why do we partition $\mathbf{g}$ into two parts?
- Why do we assume linear or affine structure in u rather than in x?
- Why do we need linear-quadratic structure in u in NOCP (9)?

Discussions:

- Partition provides a notion of "control" in the lifted linear system. More convenient to process.
- x can be unknown but we must know u. Otherwise, we cannot control the original system.
- Linear or affine structure allows us access u directly. Otherwise, we need to learn the inverse function $\mathbf{g}_u^{-1}$ to perform control.
- Linear-quadratic structure in u is required by the linear structure in $\mathbf{g}$.

Approaches

We have N snaoshots of measurements:

$$Y_1 = [h(x_1) \ \cdots \ h(x_N)], Y_2 = [u_1 \ \cdots \ u_N], Y_3 = [h(x_1') \ \cdots, h(x_N')].$$

- Assume full state observation $h(x) = x$.
- $x_i' = f(x_i, u_i)$. For uniform time intervals, $x_i' = x_{i+1}$.

Approaches:

- Extended DMD, the bases $\mathbf{g}$ are given (Korda and Mezić, 2018).

$$\min_{A,B} \quad \|A\mathbf{g}_x(Y_1) + BY_2 - \mathbf{g}_x(Y_3)\|_F^2 .$$

- Deep learning, learning $\mathbf{g}$ and (or) K (Shi and Meng, 2022).
  - K step prediction loss.
  - Add regularization if necessary.

Related Literature

Koopman operator for control

- Starts from (Korda and Mezić, 2018; Proctor et al., 2018).
- Widely used in many fields, including robotics, aerospace, and traffic. See Manzoor et al. (2023).

Other approaches to system identification for control.

- Dynamic Mode Decomposition with control (DMDc) (Proctor et al., 2016).
- SINDy for model predictive control (Kaiser et al., 2018).
- Neural networks for model predictive control (Chen et al., 2018; Li et al., 2019; Drgona et al., 2020).

Related Literature

Koopman control in robotics[5]

- Soft robots (Bruder et al., 2019, 2020; Wang et al., 2022; Alora et al., 2023).
- Rehabilitation (Goyal et al., 2022).
- Human-robot interaction (Broad et al., 2020)
- UAV/UGV (Folkestad et al., 2020; Ren et al., 2022).
- Manipulator (Zhang and Wang, 2023).
- General learning for control systems and applications in robotics
  - Deep learning (Shi and Meng, 2022; Yin et al., 2022).
  - Bilinear Koopman operator (Bruder et al., 2021).
  - Stable koopman operator (Mamakoukas et al., 2023).
  - Control affine systems (Guo et al., 2021).
  - Derivative-based Koopman operator and error bound (Mamakoukas et al., 2021).
  - ...

[5] Soft robots are the most studied application area. General frameworks for Koopman learning are heavily discussed. Only find one paper in HRI.

Summary

We have briefly introduced

- Koopman operator for nonlinear system identification.
- Extending Koopman operator for control.

Recommended Reference

Koopman operator theory and DMD

- Brunton et al. (2022); Tu et al. (2014)

Learning Koopman operator

- Takeishi et al. (2017); Lusch et al. (2018)

Koopman operator for control

- Korda and Mezić (2018); Proctor et al. (2018); Shi and Meng (2022)

References I

John Irvin Alora, Mattia Cenedese, Edward Schmerling, George Haller, and Marco Pavone. Data-driven spectral submanifold reduction for nonlinear optimal control of high-dimensional robots. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2627–2633. IEEE, 2023.

Petar Bevanda, Stefan Sosnowski, and Sandra Hirche. Koopman operator dynamical models: Learning, analysis and control. *Annual Reviews in Control*, 52:197–212, 2021.

Alexander Broad, Ian Abraham, Todd Murphey, and Brenna Argall. Data-driven koopman operators for model-based shared control of human–machine systems. *The International Journal of Robotics Research*, 39(9):1178–1195, 2020.

Daniel Bruder, C David Remy, and Ram Vasudevan. Nonlinear system identification of soft robot dynamics using koopman operator theory. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6244–6250. IEEE, 2019.

Daniel Bruder, Xun Fu, R Brent Gillespie, C David Remy, and Ram Vasudevan. Data-driven control of soft robots using koopman operator theory. *IEEE Transactions on Robotics*, 37(3):948–961, 2020.

Daniel Bruder, Xun Fu, and Ram Vasudevan. Advantages of bilinear koopman realizations for the modeling and control of systems with unknown dynamics. *IEEE Robotics and Automation Letters*, 6(3):4369–4376, 2021.

Steven L. Brunton and J. Nathan Kutz. *Data-Driven Dynamical Systems*, page 229–275. Cambridge University Press, 2019. doi: 10.1017/9781108380690.008.

Steven L Brunton, Bingni W Brunton, Joshua L Proctor, and J Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2):e0150171, 2016a.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016b.

Steven L. Brunton, Marko Budišić, Eurika Kaiser, and J. Nathan Kutz. Modern koopman theory for dynamical systems. *SIAM Review*, 64(2):229–340, 2022. doi: 10.1137/21M1401243.

Sheng Chen, Stephen A Billings, and PM Grant. Non-linear system identification using neural networks. *International journal of control*, 51(6):1191–1214, 1990.

Yize Chen, Yuanyuan Shi, and Baosen Zhang. Optimal control via neural networks: A convex approach. *arXiv preprint arXiv:1805.11835*, 2018.

References II

A Delgado, C Kambhampati, and Kevin Warwick. Dynamic recurrent neural network for system identification and control. *IEE Proceedings-Control Theory and Applications*, 142(4):307–314, 1995.

Jan Drgona, Karol Kis, Aaron Tuor, Draguna Vrabie, and Martin Klauco. Differentiable predictive control: An mpc alternative for unknown nonlinear systems using constrained deep learning. *arXiv preprint arXiv:2011.03699*, 1, 2020.

Carl Folkestad, Daniel Pastor, and Joel W Burdick. Episodic koopman learning of nonlinear robot dynamics with application to fast multirotor landing. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9216–9222. IEEE, 2020.

Tanishka Goyal, Shahid Hussain, Elisa Martinez-Marroquin, Nicholas AT Brown, and Prashant K Jamwal. Learning koopman embedding subspaces for system identification and optimal control of a wrist rehabilitation robot. *IEEE Transactions on Industrial Electronics*, 70(7):7092–7101, 2022.

Zi Cong Guo, Vassili Korotkine, James R Forbes, and Timothy D Barfoot. Koopman linearization for data-driven batch state estimation of control-affine systems. *IEEE Robotics and Automation Letters*, 7(2):866–873, 2021.

Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A*, 474(2219):20180335, 2018.

Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

Bernard O Koopman and J v Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences*, 18(3):255–263, 1932.

Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

Yunzhu Li, Jiajun Wu, Jun-Yan Zhu, Joshua B Tenenbaum, Antonio Torralba, and Russ Tedrake. Propagation networks for model-based control under partial observation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1205–1211. IEEE, 2019.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

References III

Giorgos Mamakoukas, Maria L Castano, Xiaobo Tan, and Todd D Murphey. Derivative-based koopman operators for real-time control of robotic systems. *IEEE Transactions on Robotics*, 37(6):2173–2192, 2021.

Giorgos Mamakoukas, Ian Abraham, and Todd D Murphey. Learning stable models for prediction and control. *IEEE Transactions on Robotics*, 2023.

Waqas Manzoor, Samir Rawashdeh, and Alireza Mohammadi. Vehicular applications of koopman operator theory–a survey. *IEEE Access*, 2023.

Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005.

Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. Generalizing koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930, 2018. doi: 10.1137/16M1062296.

Chao Ren, Hongjian Jiang, Chunli Li, Weichao Sun, and Shugen Ma. Koopman-operator-based robust data-driven control for wheeled mobile robots. *IEEE/ASME Transactions on Mechatronics*, 28(1):461–472, 2022.

Haojie Shi and Max Q-H Meng. Deep koopman operator with control for nonlinear systems. *IEEE Robotics and Automation Letters*, 7(3):7700–7707, 2022.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. *Advances in neural information processing systems*, 30, 2017.

Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014. ISSN 2158-2491. doi: 10.3934/jcd.2014.1.391.

Jiajin Wang, Baoguo Xu, Jianwei Lai, Yifei Wang, Cong Hu, Huijun Li, and Aiguo Song. An improved koopman-mpc framework for data-driven modeling and control of soft actuators. *IEEE Robotics and Automation Letters*, 8(2):616–623, 2022.

References IV

Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25:1307–1346, 2015.

Enoch Yeung, Soumya Kundu, and Nathan Hodas. Learning deep neural network representations for koopman operators of nonlinear dynamical systems. In *2019 American Control Conference (ACC)*, pages 4832–4839. IEEE, 2019.

Hang Yin, Michael C Welle, and Danica Kragic. Embedding koopman optimal control in robot policy learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13392–13399. IEEE, 2022.

Jinxin Zhang and Hongze Wang. Online model predictive control of robot manipulator with structured deep koopman model. *IEEE Robotics and Automation Letters*, 2023.