PyKoopman: A Python Package for Data-Driven Approximation of the Koopman Operator

07 May 2023

Summary

`PyKoopman` is a Python package for the data-driven approximation of the Koopman operator associated with a dynamical systems. The Koopman operator has emerged as a principled linear embedding of nonlinear dynamics and facilitates the prediction, estimation, and control of strongly nonlinear dynamics using linear systems theory. In particular, `PyKoopman` provides tools for data-driven system identification for unforced and actuated systems that build on the equation-free dynamic mode decomposition (DMD) [1] and its variants [2]–[4]. In this work, we provide a brief description of the mathematical underpinnings of the Koopman operator, an overview and demonstration of the features implemented in `PyKoopman` (with code examples), practical advice for users, and a list of potential extensions to `PyKoopman`. Software is available at https://github.com/dynamicslab/pyKoopman.

Statement of need

Engineers have long relied on linearization to bridge the gap between simplified, descriptions where powerful analytical tools exist, and the intricate complexities of nonlinear reality where analytical solutions are elusive [5], [6]. Local linearization, implemented via first-order Taylor series approximation, has been widely used in system identification [5], optimization [6], and many other fields to make problems tractable. However, many real-world systems are fundamentally nonlinear and require solutions outside of the local neighborhood where linearization is valid. Rapid progress in machine learning and the rise of big data are driving advances in the data-driven modeling of such nonlinear systems in science and engineering growth in measurement data and advances in machine learning are [7]. Koopman operator theory in particular has emerged as a principled approach to embed nonlinear dynamics in a linear framework that goes beyond simple linearization [4].

In the diverse landscape of data-driven modeling approaches, Koopman operator theory has received considerable attention in recent years [8]–[13]. These strategies encompass not only linear methodologies [5], [14] and dynamic mode decomposition (DMD) [1], [2], [15], but also more advanced techniques such as nonlinear autoregressive algorithms [16], [17], neural networks [18]–[27], Gaussian process regression [28], operator inference, and reduced-order modeling [31], among others [32]–[38]. The Koopman operator perspective is unique within data-driven modeling techniques due to its distinct aim of learning a coordinate system in which the nonlinear dynamics become linear. This methodology enables the application of closed-form, convergence-guaranteed methods from linear system theory to general nonlinear dynamics. To fully leverage the potential of data-driven Koopman theory across a diverse range of scientific and engineering disciplines, it is critical to have a central toolkit to automate state-of-the-art Koopman operator algorithms.

As a result, the `PyKoopman` is developed as a Python package for approximating the Koopman operator associated with natural and actuated dynamical systems from measurement data. Specifically, `PyKoopman` offers tools for designing the observables (i.e., functions of the system state) and inferring a finite-dimensional linear operator that governs the dynamic evolution of these observables in time. These steps can either be performed sequentially [10], [39] or combined, as demonstrated in more recent neural network models [21], [40]–[42]. Once a linear embedding is discovered from the data, the linearity of the transformed dynamical system can be leveraged for enhanced interpretability [43] or for designing near-optimal observers [44] or controllers for the original nonlinear system [45]–[49].

New features

The core component of the `PyKoopman` package is the `Koopman` model class. To make this package accessible to a broader user base, this class is implemented as a `scikit-learn` estimator. The external package dependencies are illustrated in Fig. 2. Additionally, users can create sophisticated pipelines for hyperparameter tuning and model selection by integrating `pyKoopman` with `scikit-learn`.

As illustrated in Fig. 3, `PyKoopman` is designed to lift nonlinear dynamics into a linear system with linear actuation. Specifically, our `PyKoopman` implementation involves two major steps:

- `observables`: the nonlinear observables used to lift $\mathbf{x}$ to $\mathbf{z}$, and reconstruct $\mathbf{x}$ from $\mathbf{z}$;
- `regression`: the regression used to find the optimal $\mathbf{A}$.

Additionally, we have a `differentiation` module that evaluates the time derivative from a trajectory and the `analytics` module for sparsifying arbitrary approximations of the Koopman operator.

At the time of writing, we have the following features implemented:

- Observable library for lifting the state $\mathbf{x}$ into the observable space
  - Identity (for DMD/DMDc or in case users want to compute observables themselves): `Identity`
  - Multivariate polynomials: `Polynomial` [10]
  - Time delay coordinates: `TimeDelay` [13], [50]
  - Radial basis functions: `RadialBasisFunctions` [10]
  - Random Fourier features: `RandomFourierFeatures` [51]
  - Custom library (defined by user-supplied functions): `CustomObservables`
  - Concatenation of observables: `ConcatObservables`
- System identification method for performing regression
  - Dynamic mode decomposition [1], [15]: `PyDMDRegressor`
  - Dynamic mode decomposition with control [52]: `DMDc`
  - Extended dynamic mode decomposition [10]: `EDMD`
  - Extended dynamic mode decomposition with control [45]: `EDMDc`
  - Kernel dynamic mode decomposition [39]: `KDMD`
  - Hankel DMD [37]: `HDMD`
  - Hankel DMD with control: `HDMDc`
  - Neural Network DMD [40], [41], [53]: `NNDMD`
- Sparse construction of Koopman invariant subspace
  - Multi-task learning based on linearity consistency [43]: `ModesSelectionPAD21`
- Numerical differentiation for computing $\dot{\mathbf{X}}$ from $\mathbf{X}$
  - Finite difference: `FiniteDifference`
  - 4th order central finite difference: `Derivative(kind="finite_difference")`
  - Savitzky-Golay with cubic polynomials: `Derivative(kind="savitzky-golay")`
  - Spectral derivative: `Derivative(kind="spectral")`
  - Spline derivative: `Derivative(kind="spline")`
  - Regularized total variation derivative: `Derivative(kind="trend_filtered")`
- Common toy dynamics
  - Discrete-time random, stable, linear state-space model: `drss`
  - Van del Pol oscillator: `vdp_osc`
  - Lorenz system: `lorenz`
  - Two-dimensional linear dynamics: `Linear2Ddynamics`
  - Linear dynamics on a torus: `torus_dynamics`
  - Forced Duffing Oscillator: `forced_duffing`
  - Cubic-quintic Ginzburg-Landau equation: `cqgle`
  - Kuramoto-Sivashinsky equation:`ks`
  - Nonlinear Schrodinger equation: `nls`
  - Viscous Burgers equation: `vbe`
- Validation routines for consistency checks

Conclusion

Our goal of the `PyKoopman` package is to provide a central hub for education, application and research development of learning algorithms for Koopman operator. The `PyKoopman` package is aimed at researchers and practitioners alike, enabling anyone with access to discover linear embeddings of nonlinear systems from data. Following `PySINDy` [54] and `Deeptime` [55], `PyKoopman` is designed to be accessible to users with basic knowledge of linear systems, adhering to `scikit-learn` standards, while also being modular for more advanced users. We hope that researchers and practioners will use `PyKoopman` as a platform for algorithms developement and applications of linear embedding.

Acknowledgments

The authors would like to acknowledge support from the National Science Foundation AI Institute in Dynamic Systems (Grant No. 2112085) and the Army Research Office ({W911NF-17-1-0306} and W911NF-19-1-0045).

References

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

[2] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: Data-driven modeling of complex systems*. SIAM, 2016.

[3] P. J. Schmid, "Dynamic mode decomposition and its variants," *Annual Review of Fluid Mechanics*, vol. 54, pp. 225–254, 2022.

[4] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern Koopman theory for dynamical systems," *SIAM Review*, vol. 64, no. 2, pp. 229–340, 2022, doi: 10.1137/21M1401243.

[5] L. Ljung, "Perspectives on system identification," *Annual Reviews in Control*, vol. 34, no. 1, pp. 1–12, 2010.

[6] S. Wright, J. Nocedal, *et al.*, "Numerical optimization," *Springer Science*, vol. 35, no. 67–68, p. 7, 1999.

[7] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control.* Cambridge University Press, 2022.

[8] M. Budišić, R. Mohr, and I. Mezić, "Applied Koopmanism," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 4, p. 047510, 2012.

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

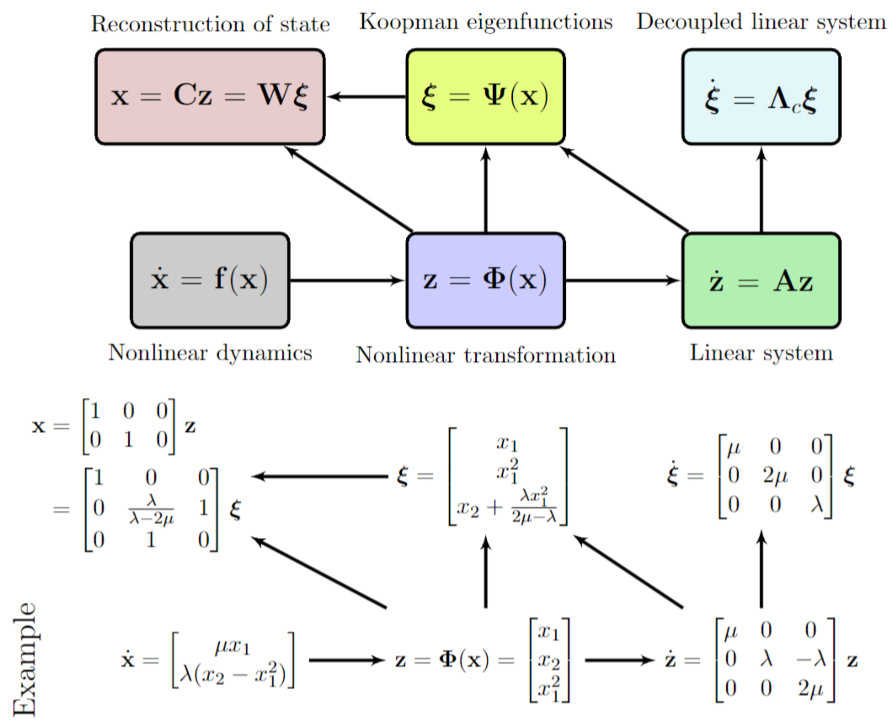

Figure 1: Lifting of the state $\mathbf{x}$ of the continuous autonomous dynamical system into a new coordinate system, in which the original nonlinear dynamics become linear and are easier to handle. One can also linearly reconstruct the state $\mathbf{x}$ from the new coordinate system. This is facilitated with `PyKoopman` in a data-driven manner.

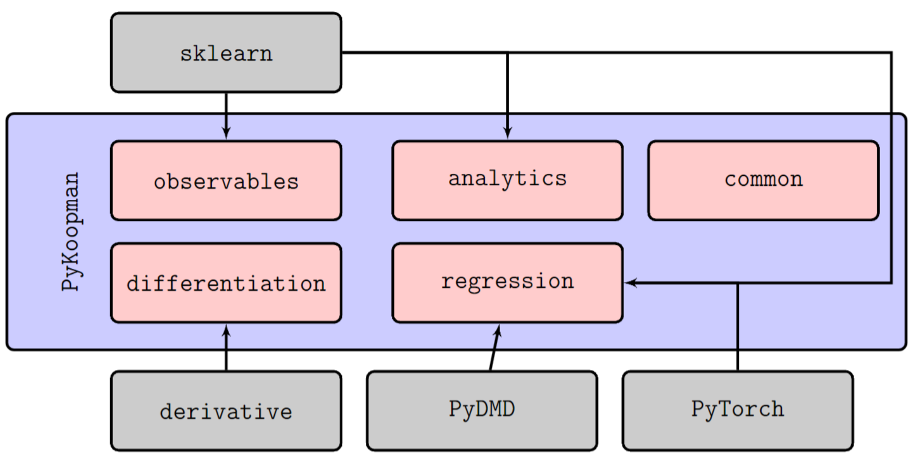

Figure 2: External package dependencies of PyKoopman.

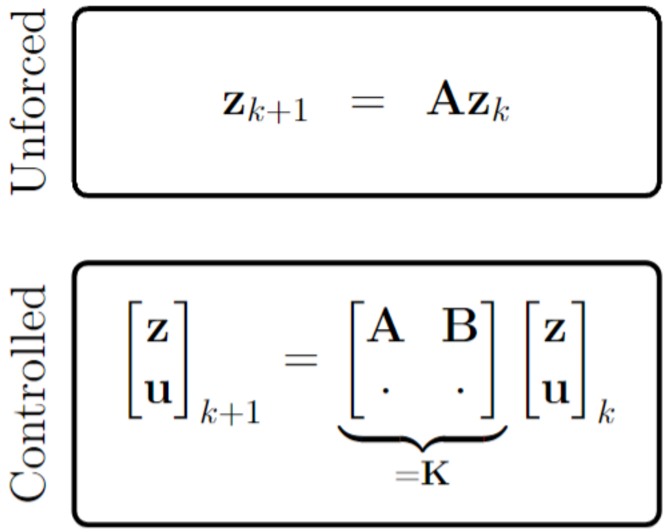

Figure 3: {Broad categorization of model types that can be identified with current `PyKoopman`. While the dotted parts (marked with "·") can be simultaneously discovered within the framework, they are typically ignored for control purposes.

[10] 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.

[11] S. Klus *et al.*, "Data-driven model reduction and transfer operator approximation," *Journal of Nonlinear Science*, vol. 28, no. 3, pp. 985–1010, 2018.

[12] 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.

[13] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, "Chaos as an intermittently forced linear system," *Nature communications*, vol. 8, no. 1, pp. 1–9, 2017.

[14] O. Nelles, "Nonlinear dynamic system identification," in *Nonlinear system identification*, Springer, 2020, pp. 831–891.

[15] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of fluid mechanics*, vol. 641, pp. 115–127, 2009.

[16] H. Akaike, "Fitting autoregressive models for prediction," *Annals of the institute of Statistical Mathematics*, vol. 21, no. 1, pp. 243–247, 1969.

[17] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.

[18] Z. Long, Y. Lu, X. Ma, and B. Dong, "Pde-net: Learning pdes from data," in *International conference on machine learning*, PMLR, 2018, pp. 3208–3216.

[19] L. Yang, D. Zhang, and G. E. Karniadakis, "Physics-informed generative adversarial networks for stochastic differential equations," *SIAM Journal on Scientific Computing*, vol. 42, no. 1, pp. A292–A317, 2020.

[20] C. Wehmeyer and F. Noé, "Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics," *The Journal of chemical physics*, vol. 148, no. 24, p. 241703, 2018.

[21] A. Mardt, L. Pasquali, H. Wu, and F. Noé, "VAMPnets for deep learning of molecular kinetics," *Nature communications*, vol. 9, no. 1, pp. 1–11, 2018.

[22] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 474, no. 2213, p. 20170844, 2018.

[23] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," *Physical review letters*, vol. 120, no. 2, p. 024102, 2018.

[24] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "DeepXDE: A deep learning library for solving differential equations," *SIAM Review*, vol. 63, no. 1, pp. 208–228, 2021.

[25] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019, doi: https://doi.org/10.1016/j.jcp.2018.10.045.

[26] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," *Proceedings of the National Academy of Sciences*, vol. 116, no. 45, pp. 22445–22451, 2019.

[27] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," *Science*, vol. 367, no. 6481, pp. 1026–1030, 2020.

[28] M. Raissi, H. Babaee, and G. E. Karniadakis, "Parametric gaussian process regression for big data," *Computational Mechanics*, vol. 64, pp. 409–416, 2019.

[29] P. Benner, S. Gugercin, and K. Willcox, "A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM review*, vol. 57, no. 4, pp. 483–531, 2015.

[30] B. Peherstorfer and K. Willcox, "Data-driven operator inference for nonintrusive projection-based model reduction," *Computer Methods in Applied Mechanics and Engineering*, vol. 306, pp. 196–215, 2016.

[31] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, "Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems," *Physica D: Nonlinear Phenomena*, vol. 406, p. 132401, 2020.

[32] D. Giannakis and A. J. Majda, "Nonlinear laplacian spectral analysis for time series with intermittency and low-frequency variability," *Proceedings of the National Academy of Sciences*, vol. 109, no. 7, pp. 2222–2227, 2012.

[33] O. Yair, R. Talmon, R. R. Coifman, and I. G. Kevrekidis, "Reconstruction of normal forms by learning informed observation geometries from data," *Proceedings of the National Academy of Sciences*, vol. 114, no. 38, pp. E7865–E7874, 2017.

[34] J. Bongard and H. Lipson, "Automated reverse engineering of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 104, no. 24, pp. 9943–9948, 2007.

[35] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *science*, vol. 324, no. 5923, pp. 81–85, 2009.

[36] B. C. Daniels and I. Nemenman, "Automated adaptive inference of phenomenological dynamical models," *Nature communications*, vol. 6, no. 1, pp. 1–8, 2015.

[37] 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, doi: 10.1073/pnas.1517384113.

[38] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Science Advances*, vol. 3, no. e1602614, 2017, doi: 10.1126/sciadv.1602614.

[39] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, "A kernel approach to data-driven Koopman spectral analysis," *Journal of Computational Dynamics*, vol. 2, pp. 247–265, 2015.

[40] 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.

[41] 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.

[42] N. Takeishi, Y. Kawahara, and T. Yairi, "Learning koopman invariant subspaces for dynamic mode decomposition," in *Advances in neural information processing systems*, 2017, pp. 1130–1140.

[43] 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.

[44] A. Surana and A. Banaszuk, "Linear observer synthesis for nonlinear systems using Koopman operator framework," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 716–723, 2016.

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

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

[47] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of Koopman eigenfunctions for control," *Machine Learning: Science and Technology*, vol. 2, no. 3, p. 035023, 2021.

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

[49] S. Peitz, S. E. Otto, and C. W. Rowley, "Data-driven model predictive control using interpolated koopman generators," *SIAM Journal on Applied Dynamical Systems*, vol. 19, no. 3, pp. 2162–2193, 2020.

[50] I. Mezić and A. Banaszuk, "Comparison of systems with complex behavior," *Physica D: Nonlinear Phenomena*, vol. 197, no. 1–2, pp. 101–133, 2004.

[51] A. M. DeGennaro and N. M. Urban, "Scalable extended dynamic mode decomposition using random kernel approximation," *SIAM Journal on Scientific Computing*, vol. 41, no. 3, pp. A1482–A1499, 2019.

[52] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.

[53] S. Pan and K. Duraisamy, "Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability," *SIAM Journal on Applied Dynamical Systems*, vol. 19, no. 1, pp. 480–509, 2020.

[54] B. M. de Silva, K. Champion, M. Quade, J.-C. Loiseau, J. N. Kutz, and S. L. Brunton, "Pysindy: A python package for the sparse identification of nonlinear dynamics from data," *arXiv preprint arXiv:2004.08424*, 2020.

[55] M. Hoffmann *et al.*, "Deeptime: A python library for machine learning dynamical models from time series data," *Machine Learning: Science and Technology*, vol. 3, no. 1, p. 015009, 2021.