The Journal of Open Source Software

pykoop: a Python Library for Koopman Operator Approximation

Steven Dahdah [1¶] **and James Richard Forbes** [1]

1 Department of Mechanical Engineering, McGill University, Montreal QC, Canada ¶ Corresponding author

DOI: 10.21105/joss.07947

Software

- Review
- Repository
- Archive

Editor: George K. Thiruvathukal

Reviewers:

- @kramea
- @matt-graham

Submitted: 04 October 2024
Published: 03 October 2025

Summary

pykoop is a Python package for learning differential equations in discretized form using the Koopman operator. Differential equations are an essential tool for modelling the physical world. Ordinary differential equations can be used to describe electric circuits, rigid-body dynamics, or chemical reaction rates, while the fundamental laws of electromagnetism, fluid dynamics, and heat transfer can be formulated as partial differential equations. The Koopman operator allows nonlinear differential equations to be rewritten as infinite-dimensional linear differential equations by viewing their time evolution in terms of an infinite number of nonlinear lifting functions. A finite-dimensional approximation of the Koopman operator can be identified from data given a user-selected set of lifting functions. Thanks to its linearity, the approximate Koopman model can be used for analysis, design, and optimal controller or observer synthesis for a wide range of systems using well-established linear tools. pykoop's documentation, along with examples in script and notebook form, can be found at at pykoop.readthedocs.io/en/stable. Its releases are also archived on Zenodo (Dahdah & Forbes, 2024b).

Statement of need

Designing Koopman lifting functions for a particular system is largely a trial-and-error process. As such, pykoop is designed to facilitate experimentation with Koopman lifting functions, with the ultimate goal of automating the lifting function optimization process. The pykoop library addresses three limitations in current Koopman operator approximation software packages. First, pykoop allows lifting functions to be constructed in a modular fashion, specifically through the composition of a series of lifting functions. Second, the library allows regressor hyperparameters and lifting functions to be selected automatically through full compatibility with `scikit-learn`'s (Pedregosa et al., 2011) existing cross-validation infrastructure. Third, pykoop handles datasets with control inputs and multiple training episodes at all stages of the pipeline. Furthermore, pykoop implements state-of-the-art Koopman operator approximation methods that enforce stability in the identified system, either through direct constraints on its eigenvalues or by regularizing the regression problem using its $\mathcal{H}_\infty$ norm (Dahdah & Forbes, 2022).

Open-source Python libraries for Koopman operator approximation include `PyKoopman` (Pan et al., 2024), `DLKoopman` (Dey & Davis, 2023), and `kooplearn` (Novelli & Kostic, 2023). While `PyKoopman` provides a similar selection of lifting functions to pykoop, it does not allow lifting functions to be composed. Furthermore, the library does not include built-in functionality to handle multiple training episodes. Unlike pykoop, which focuses on discrete-time, `PyKoopman` is able to identify continuous-time Koopman operators. As such, the library includes several built-in numerical differentiation methods. Neural network lifting functions are also supported by `PyKoopman`. The `DLKoopman` library focuses on learning Koopman models using only neural network lifting functions, which are not implemented in pykoop. The library supports multiple training episodes but does not support exogenous inputs and does not follow the `scikit-learn` interface. The `kooplearn` library includes kernel and neural network approaches to learning Koopman models, but does not include other types of lifting functions. While multiple training episodes are handled by `kooplearn`, exogenous inputs are not. Note that `pykoop` allows users to import Koopman matrices identified using other libraries for comparison and evaluation, provided that the lifting functions used are supported.

The main use cases for each Koopman operator approximation library are summarized below. Multiple training episodes are supported by `pykoop`, `DLKoopman`, and `kooplearn`. For use cases requiring neural network lifting functions, `PyKoopman`, `DLKoopman`, and `kooplearn` are all excellent options. The only package that currently supports continuous-time Koopman modelling is `PyKoopman`. For use cases that require composable lifting functions that can be optimized using standard hyperparameter selection tools, or for use cases that require control inputs, `pykoop` is the standout choice.

Scholarly publications using pykoop

The Koopman operator regression methods proposed in Dahdah & Forbes (2022) have been implemented within `pykoop`, while the methods proposed in Dahdah & Forbes (2024a), Lortie et al. (2024), Lortie & Forbes (2025), and Dahdah & Forbes (2024c) are all based on `pykoop`, but are implemented in their own repositories.

Acknowledgements

This work was supported financially by the NSERC Discovery Grants program, the FRQNT, IVADO, CIFAR, the CRM, and by Mecademic through the Mitacs Accelerate program.

References

Dahdah, S., & Forbes, J. R. (2022). System norm regularization methods for Koopman operator approximation. *Proceedings of the Royal Society A*, *478*(2265). https://doi.org/10.1098/rspa.2022.0162

Dahdah, S., & Forbes, J. R. (2024a). Closed-loop Koopman operator approximation. *Machine Learning: Science and Technology*, *5*(2), 025038. https://doi.org/10.1088/2632-2153/ad45b0

Dahdah, S., & Forbes, J. R. (2024b). *decargroup/pykoop*. Zenodo. https://doi.org/10.5281/zenodo.5576490

Dahdah, S., & Forbes, J. R. (2024c). Uncertainty modelling and robust observer synthesis using the Koopman operator. *`arXiv:2410.01057v1[eess.SY]`*. https://doi.org/10.48550/arXiv.2410.01057

Dey, S., & Davis, E. W. (2023). DLKoopman: A deep learning software package for Koopman theory. *Proceedings of Machine Learning Research*, *211*, 1467–1479.

Lortie, L., Dahdah, S., & Forbes, J. R. (2024). Forward-backward extended DMD with an asymptotic stability constraint. *`arXiv:2403.10623v1[eess.SY]`*. https://doi.org/10.48550/arXiv.2403.10623

Lortie, L., & Forbes, J. R. (2025). Asymptotically stable data-driven Koopman operator approximation with inputs using total extended DMD. *Machine Learning: Science and Technology*, *6*(1), 015003. https://doi.org/10.1088/2632-2153/ada33b

Novelli, P., & Kostic, V. (2023). *Machine-Learning-Dynamical-Systems/kooplearn v1.1.0*. https://github.com/Machine-Learning-Dynamical-Systems/kooplearn

Pan, S., Kaiser, E., Silva, B. M. de, Kutz, J. N., & Brunton, S. L. (2024). PyKoopman: A Python package for data-driven approximation of the Koopman operator. *Journal of Open Source Software*, *9*(94), 5881. https://doi.org/10.21105/joss.05881

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html