A Unified Framework for Robust Discovery of Hybrid Physical Dynamics from Data

Eymeric Chauchat,[1*] A. Collaborator,[1,2] Lead Scientist[1]

[1]Department of Robotics, Graduate School of Engineering, Tohoku University, Sendai, Japan.

[2]Another Department, Different Institution, City & Postal Code, Country.

[*]Corresponding author. Email: [Your Email Here] [†]These authors contributed equally to this work.

Discovering governing equations from data is a fundamental challenge in science and engineering. Sparse Identification of Nonlinear Dynamics (SINDy) has emerged as a powerful technique for this task, but existing methods often treat different physical paradigms, such as Lagrangian and Newtonian mechanics, as separate problems and struggle with noisy data or implicit dynamical formulations. Here, we introduce PY-XL-SINDY, a unified computational framework that overcomes these limitations. PY-XL-SINDYfeatures a modular catalog system that seamlessly combines Lagrangian, classical (Newtonian), and external force models into a single, hybrid regression problem. It incorporates a robust implicit solver based on constrained optimization to identify systems with rational or implicit dynamics, a common feature in complex mechanical systems. Furthermore, the framework leverages JAX for high-performance, parallelizable computation, enabling large-scale analysis and integration with data-intensive applications. We demonstrate the framework's superior performance and robustness on a series of challenging mechanical systems, validating the discovered models against the high-fidelity MuJoCo physics simulator. This work provides a powerful, extensible tool for the automated discovery of complex, hybrid physical laws from experi-

mental data.

The ability to distill natural laws from observational data into concise mathematical equations is a cornerstone of scientific progress. Recently, data-driven methods, particularly the Sparse Identification of Nonlinear Dynamics (SINDy) framework, have shown great promise in automating this discovery process (*1*). SINDy operates on the assumption of parsimony: that most physical systems are governed by equations with only a few active terms. However, the original SINDy formulation is best suited for explicit, ordinary differential equations in the form $\dot{\mathbf{x}} = f(\mathbf{x})$.

This has led to the development of specialized variants to handle different physical paradigms. For instance, Lagrangian SINDy focuses on identifying the scalar Lagrangian of a system, a more concise representation for many mechanical systems (*2, 3*). Other variants like SINDy-PI were developed to handle implicit dynamics ($\mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}) = 0$), which are common in constrained mechanical systems or models with rational dynamics, though they can be sensitive to noise (*4*). These methods, while powerful, exist as distinct algorithms, making it difficult to model hybrid systems that might involve, for example, a core Lagrangian structure with additional non-conservative forces best described in a classical framework. Furthermore, scaling these discovery methods to handle large datasets and enabling their use in modern applications like reinforcement learning requires a focus on computational efficiency.

Here, we introduce PY-XL-SINDY, a comprehensive and extensible Python framework that unifies these disparate approaches and addresses their limitations. Our main contributions are:

- **A Unified Hybrid Modeling Framework:** We introduce a modular catalog architecture that allows for the seamless combination of different physical models. Users can construct a single regression problem from ‘Lagrange‘, ‘Classical‘ (Newtonian), and ‘ExternalForces‘ components, enabling the discovery of hybrid dynamics.

- **Robust Implicit Dynamics Identification:** We implement a robust solver for implicit dynamics based on constrained optimization, inspired by SINDy-PI, and introduce a novel post-processing algorithm to automatically cluster and identify the true sparse solution from the resulting coefficient matrix.

- **High-Performance Parallelization:** The framework is integrated with JAX, enabling JIT-compilation and auto-vectorization (‘vmap‘) of the dynamics functions. This allows for massive parallelization of simulations, critical for large-scale hyperparameter searches, uncertainty quantification, and integration with machine learning pipelines.

- **End-to-End Validation Pipeline:** We present a complete workflow for generating data from the high-fidelity MuJoCo simulator, performing the system identification, and validating the discovered models against the simulator's trajectory, providing a robust metric for real-world performance.

A Unified Catalog for Hybrid Dynamics

We demonstrate the power of our unified framework by identifying the dynamics of a simulated cart-pole system with joint friction. The core dynamics are elegantly described by a Lagrangian, while the dissipative friction forces are non-conservative and best modeled as classical velocity-dependent terms. Our framework constructs a hybrid experimental matrix from a 'Lagrange' catalog and a 'Classical' catalog (Fig. 1). Using a hard-thresholding sparse regression, PY-XL-SINDY successfully identifies the correct terms for both the Lagrangian and the friction components simultaneously from a single dataset. The trajectory of the discovered hybrid model shows excellent agreement with the ground-truth data generated by the MuJoCo simulator, outperforming models derived from a purely Lagrangian or purely classical approach.

Robust Identification of Implicit Dynamics

Many complex mechanical systems, such as the double pendulum on a cart, feature implicit dynamics due to kinematic constraints. We tested our framework's implicit solver on this challenging system, using data generated without external forces. The algorithm formulates the problem as a constrained optimization problem, searching for sparse solutions in the null space of the dynamics library. Our novel post-processing technique, which clusters homothetic vectors in the resulting solution matrix, successfully identified the underlying physical law with high precision (Fig. 2). This demonstrates the framework's ability to robustly handle systems that are inaccessible to standard SINDy.

Performance Gains with JAX-based Parallelization

The computational cost of system identification can be prohibitive, especially when exploring large parameter spaces or using the model for downstream tasks like reinforcement learning. We benchmarked the performance of our JAX-compiled dynamics functions against standard NumPy implementations. For a double pendulum system, the JAX-based function, when vectorized using ‘vmap‘, can simulate tens of thousands of different initial states in parallel on a GPU. This results in a per-simulation time that is orders of magnitude faster than a sequential NumPy approach (Table 1), opening the door for data-intensive discovery and control applications.

Discussion

In this work, we presented PY-XL-SINDY, a novel framework that unifies and extends the capabilities of Sparse Identification of Nonlinear Dynamics. By introducing a modular, hybrid catalog system, we have bridged the gap between Lagrangian and Newtonian formulations, allowing for the discovery of more realistic physical models that include both conservative and dissipative forces. Our implementation of a robust implicit solver with automated post-processing expands the class of systems that can be identified to include those with complex constraints.

The integration with JAX for high-performance computing is a key practical advance, making it feasible to apply these discovery techniques to large-scale problems. Our end-to-end pipeline, which uses the MuJoCo simulator for data generation and validation, ensures that the discovered models are not just theoretically sound but are benchmarked against a realistic physics engine.

Limitations of our current framework include the need for the user to design the initial candidate library of functions, a common challenge in all SINDy-based methods. Future work will focus on automating this library construction process and integrating the high-performance dynamics models into model-based reinforcement learning algorithms, paving the way for agents that can not only learn to control a system but also discover its underlying physical laws.

Figure 1: **Validation of a discovered hybrid model for a cart-pole with friction.** (**A**) Comparison of trajectories for the generalized coordinates (position and angle) between the ground-truth MuJoCo simulation (black dashed line) and the model discovered by py-xl-sindy (blue solid line). (**B**) Residuals between the MuJoCo and discovered model trajectories, showing high fidelity. (**C**) The discovered coefficient vector, showing sparse non-zero terms corresponding to the correct Lagrangian and friction functions.

Table 1: **Performance benchmark of JAX-based parallelized dynamics function.** Comparison of average computation time per simulation step for a double pendulum model. The JAX implementation is benchmarked on both CPU and GPU, showing orders-of-magnitude speedup when simulating large batches in parallel.

Implementation	Batch Size	Hardware	Avg. Time per Step (ms)
NumPy (sequential)	1	CPU	~0.15
JAX (sequential)	1	CPU	~0.16
JAX (parallelized)	10,000	GPU	~0.0002

Figure 2: **Implicit identification of a double pendulum on a cart.** (**A**) The solution matrix ‘X‘ from the constrained optimization problem. Most columns are sparse. (**B**) Post-processing identifies a cluster of homothetic column vectors (highlighted in red) that correspond to the true physical law. (**C**) The final, single solution vector extracted from the cluster, accurately representing the system’s implicit dynamics.

References and Notes

1. S. L. Brunton, J. L. Proctor, J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* **113** (15), 3932–3937 (2016), doi:10.1073/pnas.1517384113.

2. H. K. Chu, M. Hayashibe, Discovering Interpretable Dynamics by Sparsity Promotion on Energy and the Lagrangian. *IEEE Robotics and Automation Letters* **5** (2), 2154–2161 (2020), doi: 10.1109/LRA.2020.2970626.

3. A. Purnomo, M. Hayashibe, Sparse identification of Lagrangian for nonlinear dynamical systems via proximal gradient method. *Scientific Reports* **13** (1), 7919 (2023), doi: 10.1038/s41598-023-34931-0.

4. K. Kaheman, J. N. Kutz, S. L. Brunton, SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **476** (2242), 20200279 (2020), doi:10.1098/rspa.2020.0279.

Acknowledgments

Funding: [TODO: List your funding sources here. Example: This work was supported by the JSPS Grant-in-Aid for Scientific Research (B) under Grant 18H01399.]

Author contributions: E.C. conceived the idea, developed the software, performed the experiments, and wrote the manuscript. A.C. assisted with... L.S. supervised the project... [TODO: Fill out contributions.]

Competing interests: The authors declare no competing interests.

Data and materials availability: All code for the PY-XL-SINDY framework and the scripts to reproduce the experiments are publicly available on GitHub at `https://github.com/Eymeric65/py-xl-sindy`. All data used in this study was generated using the provided code and the MuJoCo simulator.

Supplementary materials

Materials and Methods

Supplementary Text

Figs. S1 to S2

Tables S1 to S2

Supplementary Materials for

A Unified Framework for Robust Discovery of Hybrid Physical Dynamics from Data

Eymeric Chauchat,* A. Collaborator, Lead Scientist

*Corresponding author. Email: [Your Email Here]

This PDF file includes:

Materials and Methods

Materials and Methods

The Unified Catalog Framework

The core of PY-XL-SINDY is its modular catalog system, defined in ‘src/xlsindy/catalog.py‘. The abstract base class ‘CatalogCategory‘ defines a standard interface for different physical paradigms. We implemented three main categories:

- ***Lagrange*:** This category takes a library of candidate symbolic functions and constructs the experimental matrix by applying the Euler-Lagrange operator, $\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i}$.
- ***Classical*:** This category represents standard Newtonian or state-space models. It takes a library of functions of the state variables (e.g., polynomials, trigonometric functions) and constructs the corresponding columns in the experimental matrix directly.
- ***ExternalForces*:** This category explicitly models external forces, typically serving as the known right-hand-side of the final governing equation.

The ‘CatalogRepartition‘ class combines instances of these categories into a single, cohesive experimental matrix $\mathbf{\Theta}$. This allows for the construction of a hybrid equation, for example $\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau + \mathbf{F}_{friction}(\dot{\mathbf{q}})$, where the Lagrangian terms are handled by the ‘Lagrange‘ catalog and the friction terms by the ‘Classical‘ catalog.

Sparse Regression Techniques

The framework solves the sparse regression problem $\mathbf{b} = \mathbf{\Theta\Xi}$ to find the coefficient vector $\mathbf{\Xi}$. Two primary solvers are implemented in ‘src/xlsindy/optimization.py‘:

- **Hard Thresholding (STLSQ):** An iterative method where a standard least-squares solution is computed, and then coefficients below a certain threshold are pruned. The process is repeated on the reduced library until the solution converges.
- **Lasso Regression:** Utilizes the L1 regularizer to promote sparsity, implemented via Scikit-learn’s ‘LassoCV‘ to automatically select the optimal regularization parameter α.

Implicit Dynamics Identification

For implicit systems, where no term can be isolated on the left-hand side, we seek a sparse vector $\boldsymbol{\Xi}$ in the null space of $\boldsymbol{\Theta}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = 0$. Following the robust approach of SINDy-PI (*4*), we solve a constrained optimization problem using ‘cvxpy‘. The problem is formulated as:

$$\min_{\mathbf{X}} ||\boldsymbol{\Theta}\mathbf{X} - \boldsymbol{\Theta}||_F + \lambda||\mathbf{X}||_1 \quad \text{s.t.} \quad \mathrm{diag}(\mathbf{X}) = 0. \tag{S1}$$

The resulting solution matrix $\mathbf{X}$ ideally contains multiple column vectors that are sparse and homothetic (i.e., pointing in the same direction), representing the same physical law. Our post-processing algorithm in ‘_implicit_post_treatment‘ clusters these vectors based on angular similarity and extracts the principal direction via SVD to yield a single, robust solution vector.

Data Generation from MuJoCo Simulator

All training and validation data were generated using the MuJoCo physics simulator. The scripts in the ‘util/‘ directory manage this process.

1. **Data Generation (‘mujoco_generate_data.py‘):** A simulation is run for a specified duration (‘max_time‘). At each step, a randomly generated, time-varying external force is applied to the system’s actuators. The force is generated using a sum of sinusoids with varying frequencies and amplitudes to ensure sufficient exploration of the state space.

2. **Coordinate Transformation (‘xlsindy_gen.py‘ files):** MuJoCo often uses a different set of generalized coordinates than the standard textbook Lagrangian formulation (e.g., cumulative angles). Each experiment folder contains a ‘mujoco_transform‘ function to convert the simulator’s output (‘qpos‘, ‘qvel‘, ‘qacc‘) into the coordinate system used for the symbolic model.

3. **Data Storage:** The generated time-series data is saved to ‘.npz‘ files, while simulation metadata is saved to corresponding ‘.json‘ files. A validation database (‘validation_database.pkl‘) is created by subsampling from all generated experiments to test the generalization of discovered models.