**Course title:** Scientific Machine Learning for Modeling, Optimization, and Control of Dynamical Systems

**Description:**

This course covers a range of scientific machine learning (SciML) methods for modeling, optimization, and control of dynamical systems. Students will learn to systematically integrate physics-based models and constraints into deep learning architectures, and to leverage data-driven methods for accelerating the solution of large-scale optimization and optimal control problems. Key topics include physics-informed neural networks (PINNs), neural differential equations (NODEs), learning to optimize (L2O), feasibility restoration, and differentiable control. The course emphasizes the unification of theory and practice: lectures will provide the mathematical foundations, while hands-on labs and project-based assessments will give students experience implementing algorithms and applying them to real-world problems. Application domains include building energy management, robotics, networked systems, and power grids.

**Syllabus:**

* Week 1: Introduction to Scientific Machine Learning (SciML)
* Week 2: Introduction to differentiable programming
* Week 3: Physics-informed neural networks (PINNs)
* Week 4: Neural ordinary differential equations (NODEs)
* Week 5: Neural differential equations with constraints
* Week 6: Learning to optimize (L2O)
* Week 7: Feasibility restoration methods in L2O
* Week 8: Differentiable control
* Week 9: Safety filters for learning-based control
* Week 10: SciML applications
* Week 11: Project evaluation with presentations

**Prerequisites:**

* Differential calculus
* Linear algebra
* Deep learning fundamentals
* Constrained optimization fundamentals
* Dynamical systems fundamentals
* Optimal control fundamentals
* Solid Python programming skills

**Suggested reading:**

1. Karniadakis, G.E., Kevrekidis, I.G., Lu, L. et al. Physics-informed machine learning. Nat Rev Phys 3, 2021.
2. Thiyagalingam, J., Shankar, M., Fox, G. et al. Scientific machine learning benchmarks. Nature Reviews Physics 4, 413–420, 2022.
3. Nghiem T., Drgona J., et al. Physics-Informed Machine Learning for Modeling and Control of Dynamical Systems, ACC, 2023.
4. M. Innes, et al., A Differentiable Programming System to Bridge Machine Learning and Scientific Computing, 2019
5. Baydin, Atilim Gunes et al. Automatic differentiation in machine learning: a survey. Journal of Machine Learning Research, 2015
6. S. Scardapane, Alice's Adventures in a Differentiable Wonderland -- Volume I, A Tour of the Land, arXiv preprint 2404.17625, 2024.
7. M. Raissi, et al., *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,* 2019
8. R. T. Q. Chen, et al., *Neural Ordinary Differential Equations*, 2019
9. C. Rackauckas, et al., *Universal Differential Equations for Scientific Machine Learning*, 2021
10. J. Kotary, et al., *End-to-End Constrained Optimization Learning: A Survey*, 2021
11. Brandon Amos, Tutorial on amortized optimization, arXiv:2202.00665, 2025
12. Priya Donti, et al., DC3: A learning method for optimization with hard constraints, ICLR, 2021
13. Yoshua Bengio, et al., Machine learning for combinatorial optimization: a methodological tour d’horizon. European Journal of Operational Research, 2021
14. Dimitris Bertsimas and Bartolomeo Stellato. Online mixed-integer optimization in milliseconds. INFORMS Journal on Computing, 2022.
15. B. Amos, et al., *Differentiable MPC for End-to-end Planning and Control*, 2019
16. J. Drgoňa, A. Tuor and D. Vrabie, "Learning Constrained Parametric Differentiable Predictive Control Policies With Guarantees," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2024

**Core libraries**

* **NeuroMANCER**: <https://github.com/pnnl/neuromancer> — open-source SciML library for constrained learning and control
* **PyTorch**: <https://pytorch.org> — differentiable programming and deep learning framework
* **torchdiffeq**: <https://github.com/rtqichen/torchdiffeq> — Neural ODE solvers in PyTorch
* **CasADi**: <https://web.casadi.org/> — automatic differentiation framework for optimization and optimal control

**Course goals**

* Understand the fundamentals of differentiable programming, including forward- and reverse-mode automatic differentiation and computational graph construction.
* Formulate and implement physics-informed neural networks (PINNs) for ODE- and PDE-based modeling tasks.
* Develop and train neural ODE and differential algebraic equation (DAE) models, including constrained and projected dynamics.
* Apply learning-to-optimize (L2O) methods to accelerate solutions of constrained optimization problems.
* Implement feasibility restoration techniques via differentiable projected gradient descent, and convex optimization layers.
* Formulate and implement differentiable predictive control (DPC) to solve parametric optimal control problems.
* Implement safety filters and learn neural Lyapunov functions for safe learning-based control.
* Apply SciML techniques to real-world domains such as building energy management, networked systems, power systems, and battery management.
* Design and execute a project that applies existing SciML techniques to a real-world problem or develop novel SciML methods tackling some of the open challenges.

**Grading and course expectation**

Both the graduate and undergraduate students of this course will be graded using the same policy:

* Participation 10%
* Homework 30%
* Project proposal 10%
* Project progress report 10%
* Project final report 20%
* Project presentation 20%