Differentiable Programming for Modeling and Control of Energy Systems

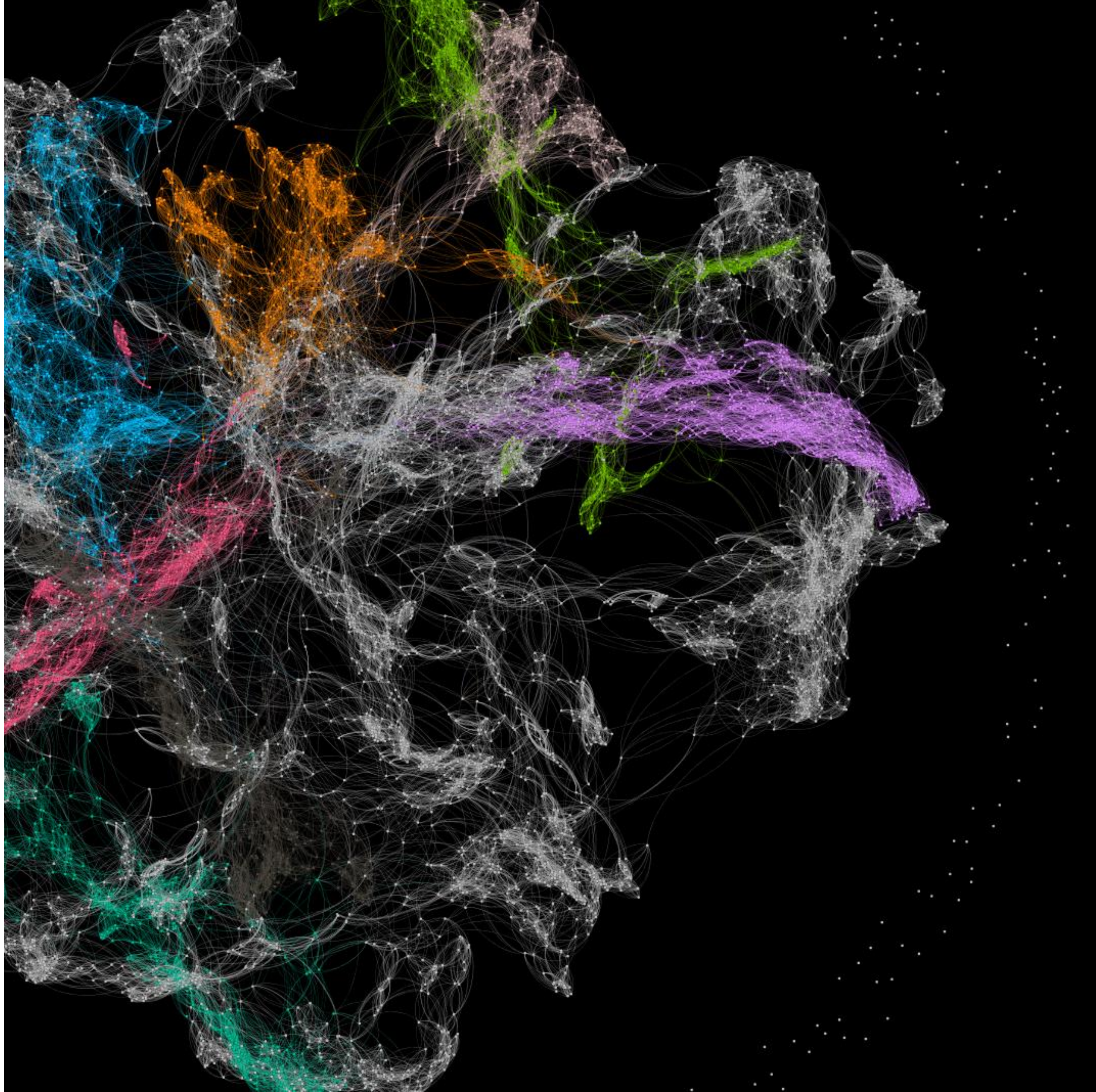
American Control Conference 2023, San Diego

June 6, 2023

Ján Drgoňa

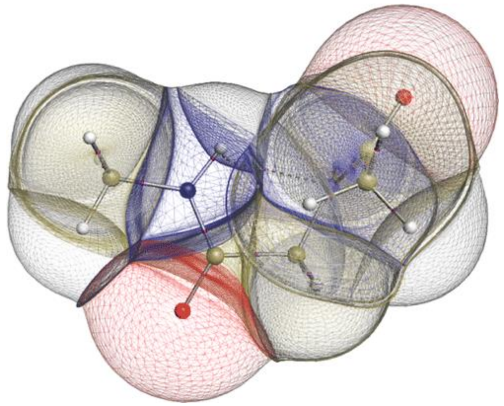
Collaborators: Aaron Tuor, James Koch, Shrirang Abhyankar, Ethan King, Wenceslao Shaw Cortez, Soumya Vasisht, Sayak Mukherjee, Mahantesh Halappanavar, Draguna Vrabie

PNNL is operated by Battelle for the U.S. Department of Energy



Challenges of Dynamical Systems Modeling and Control

- **Simulations** are crucial for many areas of decision-making and scientific discovery
- **Need**: Improve computational efficiency and scalability for heterogenous scientific simulations
- **Challenges**:
  - Data-driven predictive modeling and verification
  - Simulation of complex multi-scale systems of systems
  - Optimal control and design of complex systems
- **Emerging solution**:
  - Scientific Machine Learning connecting physics and AI domains



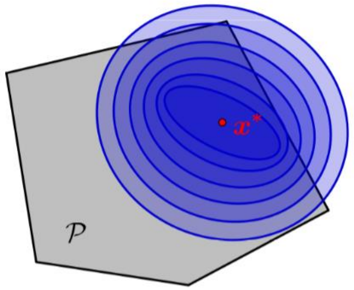
Latest Neural Nets Solve World's Hardest Equations Faster Than Ever Before

Quantamagazine

Landscape of Solution Methods

Constrained optimization

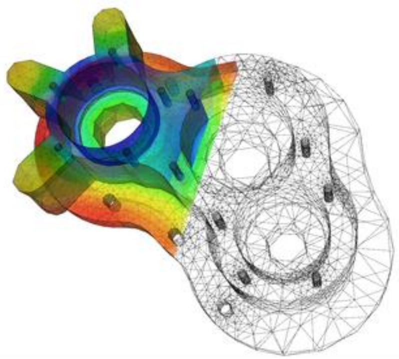
$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & b(x) \geq 0 \\ & c(x) = 0. \end{aligned}$$



- Requires prior knowledge of objective function and constraints

Differential equations

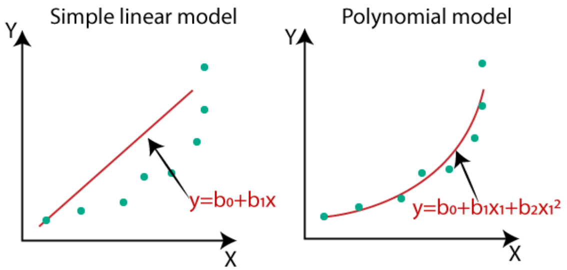
$$\frac{dy}{dx} = f(x)$$



- Requires prior knowledge of the physics to be modeled

Supervised Learning

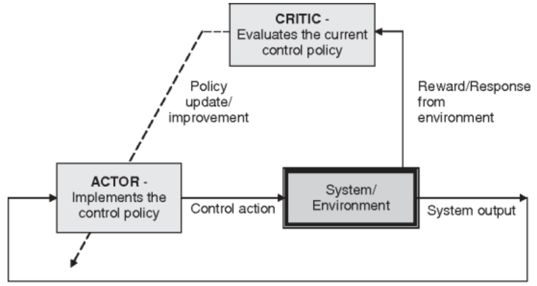
$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(y^i, f(x^i, \theta))$$



- Requires large labeled datasets

Reinforcement Learning

$$\begin{aligned} & \min_{\Theta} \sum_{i=1}^{m} \mathbf{r}(\mathbf{x}, \Theta) \\ \text{s.t. } & \mathbf{Bellman}(\mathbf{x}, \Theta) = \mathbf{0}, \\ & \mathbf{environment}(\mathbf{x}, \Theta) = \mathbf{0} \\ & \mathbf{x} \in \Xi \end{aligned}$$



- Requires environment model to sample

More domain knowledge

Less domain knowledge

Pacific Northwest
NATIONAL LABORATORY

Landscape of Solution Tools

Constrained optimization	Differential Equations	Supervised Learning	Reinforcement Learning

PYOMO
JUMP
GEKKO DYNAMIC OPTIMIZATION
CVXPY
CasADi
GUROBI OPTIMIZATION

DifferentialEquations.jl
MATLAB®
SciPy
PETSc TAO
NWCHEM

PyTorch
TensorFlow
flux
Gym

More domain knowledge

Less domain knowledge

Pacific Northwest NATIONAL LABORATORY

Landscape of Solution Tools

Online optimization	Differential Equations	Supervised Learning	Reinforcement Learning
PYOMO	DifferentialEquations.jl	PyTorch	
JUMP	MATLAB	TensorFlow	
GEKKO DYNAMIC OPTIMIZATION	SciPy	flux	
CVXPY	PETSc TAO	JAX	Gym
CasADi	NWCHEM		
GUROBI OPTIMIZATION			

What comes next? … Differentiable programming (DP): a unifying approach for data-driven modeling and optimization of complex systems based on automatic differentiation (AD)

Pacific Northwest NATIONAL LABORATORY

Differentiable Programming Enables Scientific Machine Learning

- **Differentiable Programming**
  - M. Innes, et al., *A Differentiable Programming System to Bridge Machine Learning and Scientific Computing*, 2019
- **Physics-informed Neural Networks**
  - M. Raissi, et al., *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, 2019
- **Neural Differential Equations**
  - R. T. Q. Chen, et al., *Neural Ordinary Differential Equations*, 2019
  - C. Rackauckas , et al., *Universal Differential Equations for Scientific Machine Learning*, 2021
- **Differentiable Optimization**
  - A. Agrawal, et al., Differentiable Convex Optimization Layers, 2019
  - P. Donti, et al., *DC3: A learning method for optimization with hard constraints*, 2021
  - S. Gould, et al., *Deep Declarative Networks: A New Hope*, 2020
  - J. Kotary, et al., *End-to-End Constrained Optimization Learning: A Survey*, 2021
- **Differentiable Control**
  - B. Amos, et al., *Differentiable MPC for End-to-end Planning and Control*, 2019
  - S. East, et al., *Infinite-Horizon Differentiable Model Predictive Control*, 2020

NeuroMANCER: a Scientific Machine Learning Library

1. Mathematical formulation

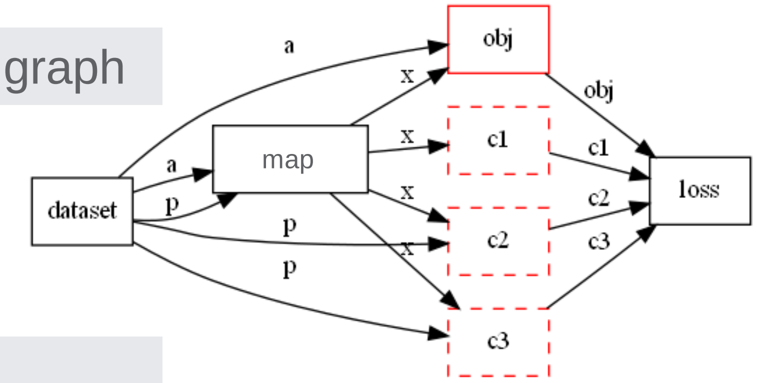
$$\min_{\Theta} \; (1-\mathbf{x})^2 + \boldsymbol{p}(\mathbf{y}-\mathbf{x}^2)^2$$
$$\text{s.t.} \; (\boldsymbol{p}/2)^2 \leq \mathbf{x}^2 + \mathbf{y}^2 \leq \boldsymbol{p}^2, \quad \mathbf{x} \geq \mathbf{y}$$
$$\mathbf{x} = \boldsymbol{\pi}_{\Theta}(\boldsymbol{p})$$

2. Python code interface

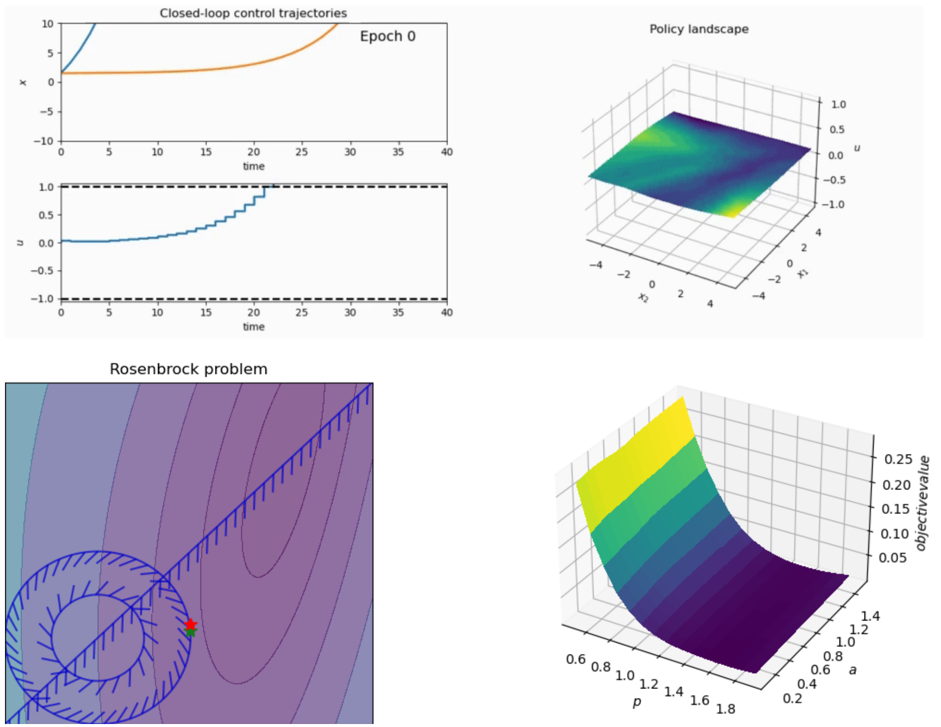
PyTorch

```
import neuromancer as nm

p = nm.constraints.variable(‘p’)
x = nm.constraints.variable(‘x’)
y = nm.constraints.variable('y’)

obj = ((1-x)**2 + p*(y-x**2)**2).minimize(weight=1.0, name='obj’)
c1 = (p/2)**2 <= x**2 + y**2
c2 = x**2 + y**2 <= p**2
c3 = x >= y

net = nm.blocks.MLP(insize=2, outsize=2, hsizes=[80]*4)
map = nm.maps.Map(net, input_keys=['p’], output_keys=[‘x’,‘y’])

loss = nm.loss.PenaltyLoss([obj], [c1, c2, c3])
problem = nm.problem.Problem([map], loss)
optimizer = torch.optim.AdamW(problem.parameters())
trainer = nm.Trainer(problem,data,optimizer)
best_model = trainer.train()
```

3. Problem graph

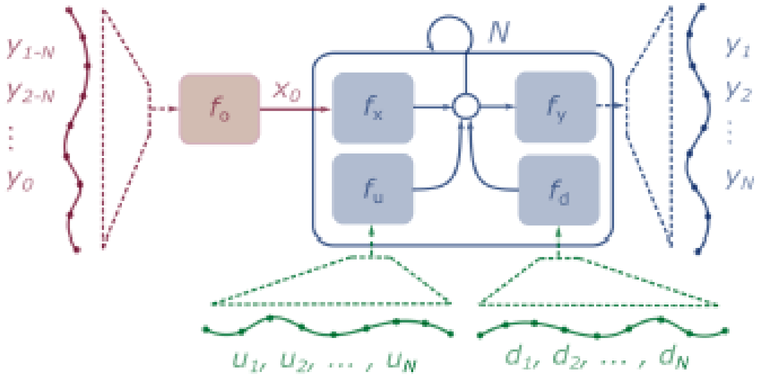


4. Results

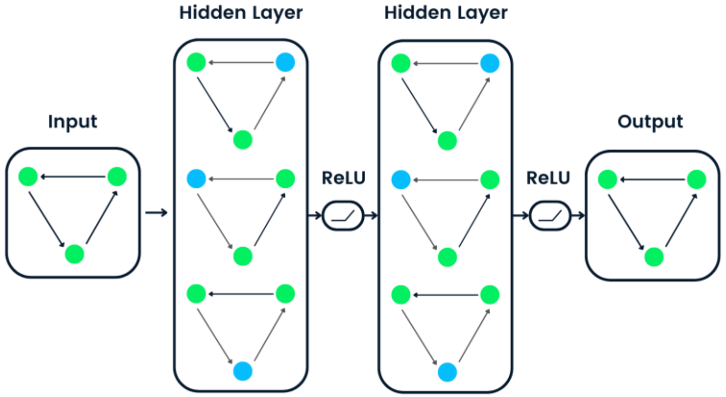


Data-driven Modeling Capabilities in Neuromancer

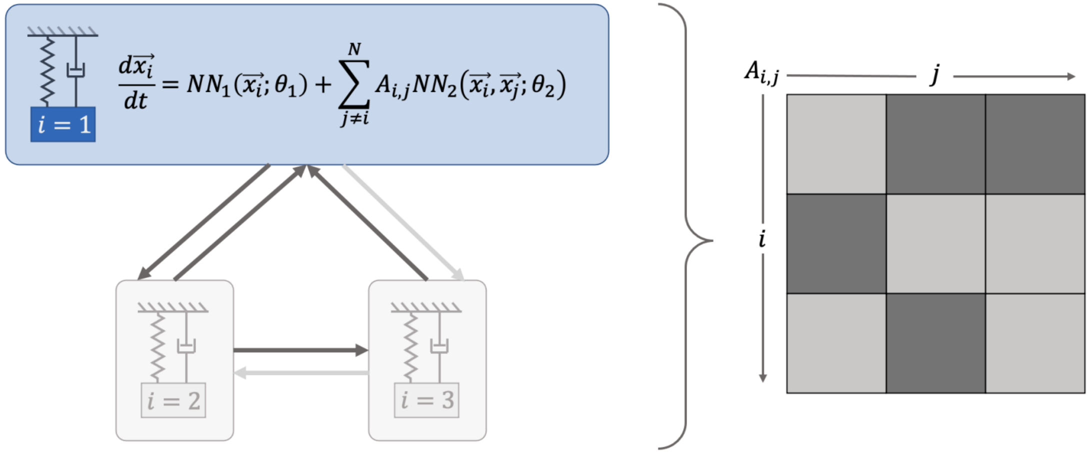
Component-based Physics-informed Machine Learning



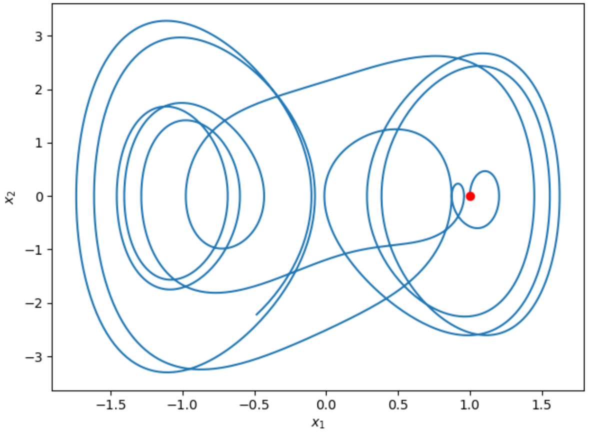
Graph Neural Networks



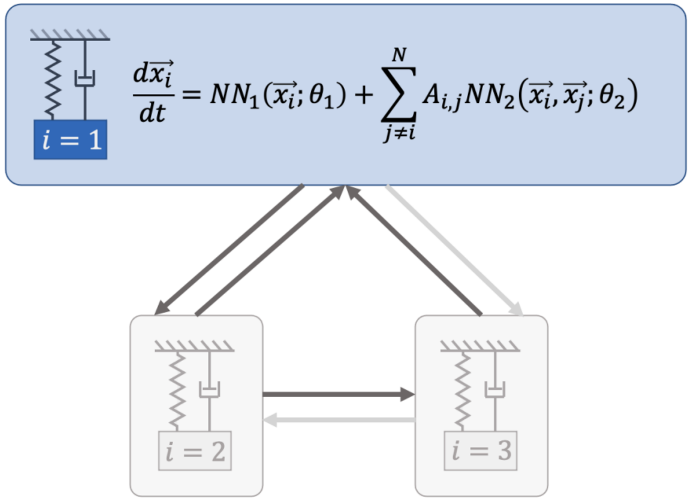
Networked Dynamical systems



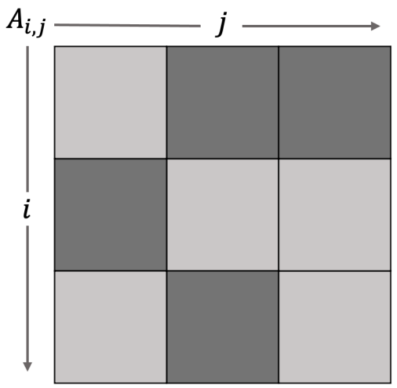
Differential Equation Solvers



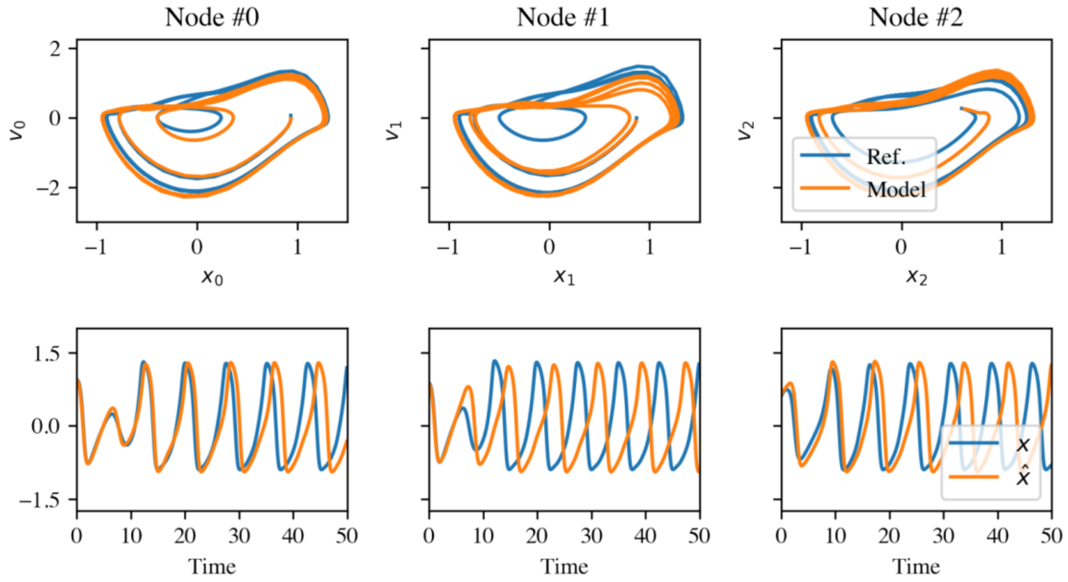
Networked Dynamical Systems via Universal Differential Equations



Network of unknown oscillators



Coupling adjacency



Ground truth Kuramoto system:

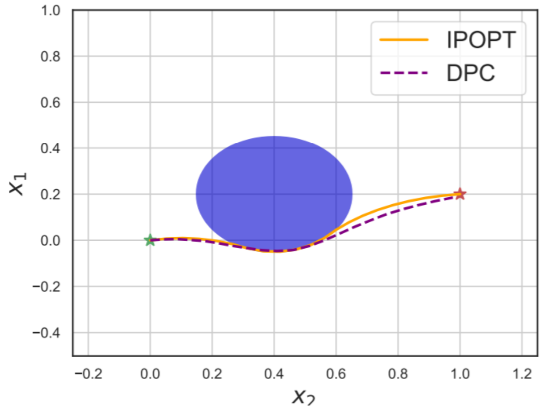
$$\frac{d\theta_i}{dt} = \omega + \frac{K}{N}\sum_{j=1}^{N} \sin(\theta_i - \theta_j).$$

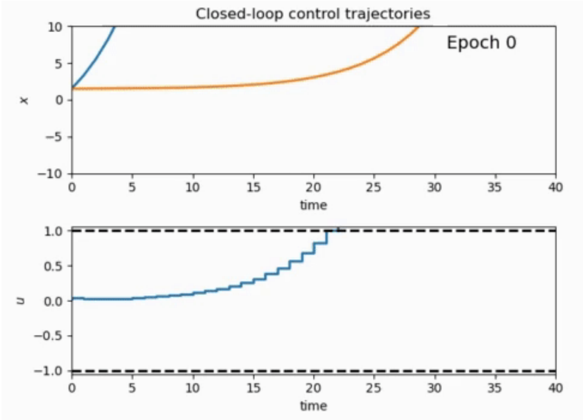
Generalization of learned dynamics on never-seen network topology.

James Koch, et al., **Structural Inference of Networked Dynamical Systems with Universal Differential Equations**, Chaos: An interdisciplinary Journal of Nonlinear Science, doi.org/10.1063/5.0109093, 2023

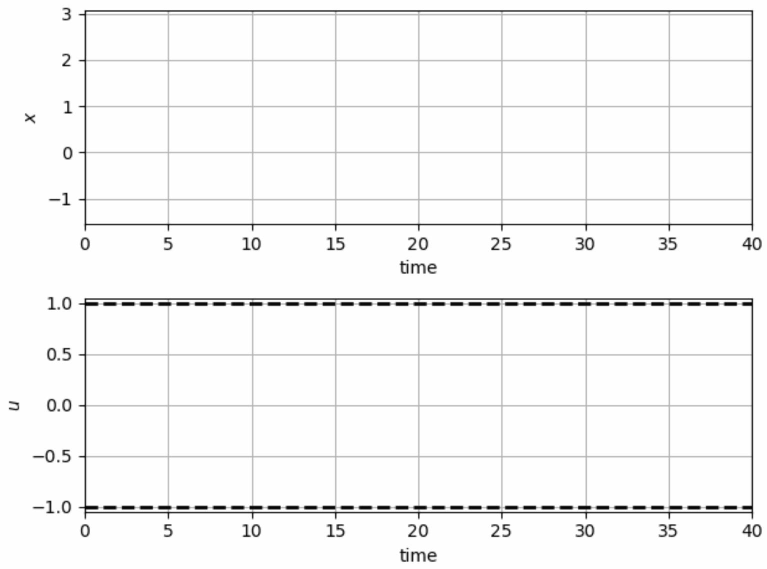
Optimal Control Capabilities in Neuromancer

Trajectory optimization and obstacle avoidance

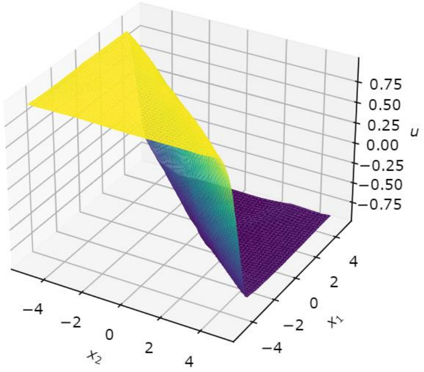


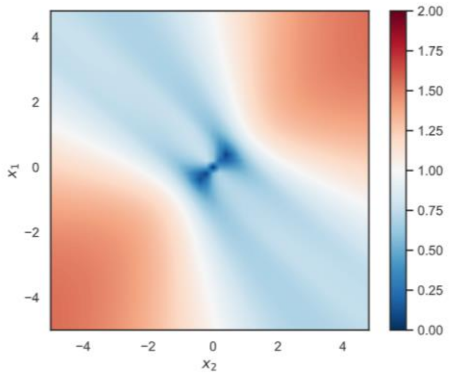


Learning stabilizing controllers



Code generation and edge deployment – under development





Differentiable Predictive Control (DPC)

From data to optimized explicit predictive control policy

MPC

Cost: $$\min_{\mathbf{u}_0,\ldots,\mathbf{u}_{N-1}} \sum_{k=0}^{N-1} \ell_{\text{MPC}}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{r}_k)$$

Model: $$\text{s.t. } \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \ k \in \mathbb{N}_0^{N-1}$$

Constraints: $$g(\mathbf{x}_k) \le \mathbf{0}, \ k \in \mathbb{N}_0^{N-1}$$
$$h(\mathbf{u}_k) \le \mathbf{0}, \ k \in \mathbb{N}_0^{N-1}$$
$$\mathbf{x}_0 = \mathbf{x}(t),$$

DPC

$$\min_{\boldsymbol{W}} \frac{1}{mN} \sum_{i=1}^{m} \sum_{k=0}^{N-1} \Big(\ell_{\text{MPC}}(\boldsymbol{x}_k^i, \boldsymbol{u}_k^i, \boldsymbol{r}_k^i) + p_x(c(\boldsymbol{x}_k^i, \boldsymbol{p}_{c_k}^i)) + p_u(g(\boldsymbol{u}_k^i, \boldsymbol{p}_{g_k}^i))\Big)$$
$$\begin{aligned} \text{s.t. } & \boldsymbol{x}_{k+1}^i = \boldsymbol{A}\boldsymbol{x}_k^i + \boldsymbol{B}\boldsymbol{u}_k^i, \\ & \boldsymbol{u}_k^i = \pi_{\mathbf{W}}(\boldsymbol{x}_k^i, \boldsymbol{\xi}_k^i) \\ & \boldsymbol{x}_0^i \in \mathbb{X} \subset \mathbb{R}^{n_x} \\ & \boldsymbol{\xi}_k^i = \{\boldsymbol{r}_k^i, \boldsymbol{p}_{c_k}^i, \boldsymbol{p}_{g_k}^i\} \in \Xi \subset \mathbb{R}^{n_\xi} \end{aligned}$$

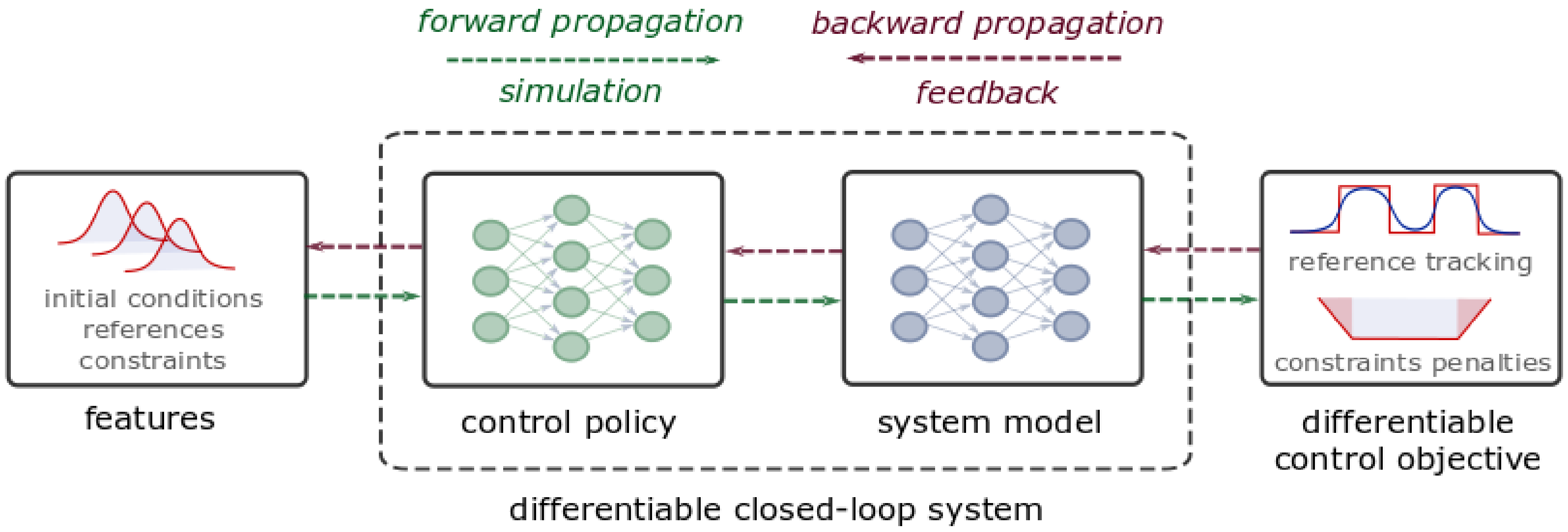
- Online optimization
- Optimize for control actions
- Implicit solver-based control policy

- Offline optimization
- Optimize for control policy parameters
- Explicit neural control policy

Jan Drgona, et al., **Differentiable Predictive Control: An MPC Alternative for Unknown Nonlinear Systems using Constrained Deep Learning**, Journal of Process Control, 2022

Differentiable Predictive Control (DPC)

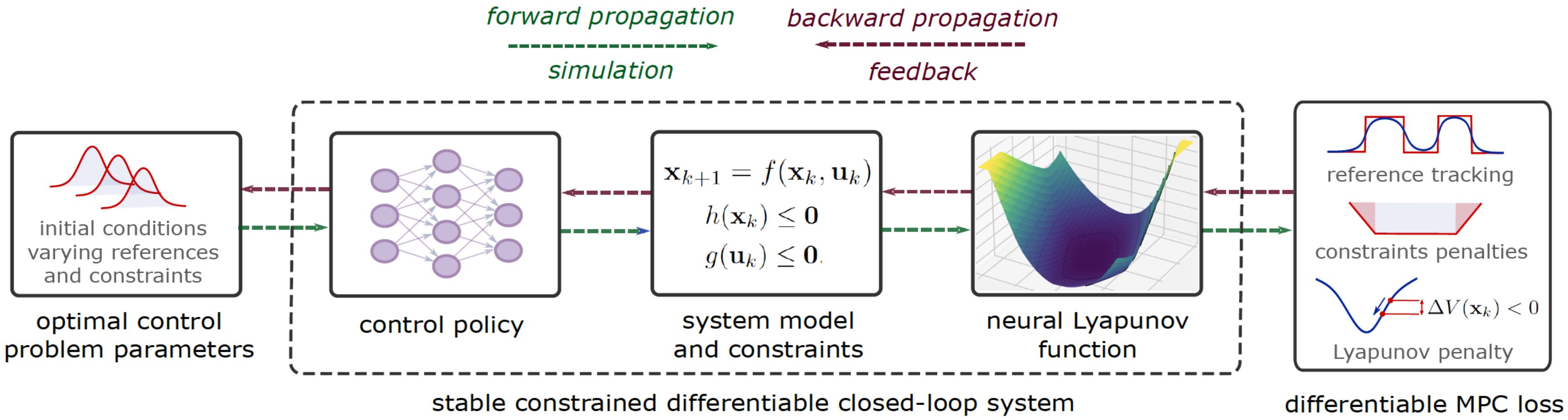
Learning-based control using differentiable system models

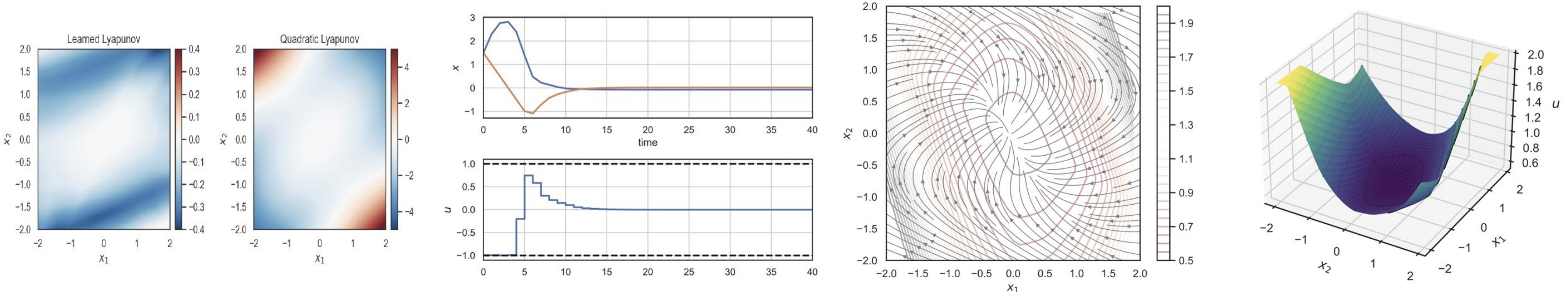


- Optimization over **design parameters** representing different **scenarios**
- Differentiable system model allows for **counterfactual reasoning** based on forward simulations
- Evaluate **quality of the soliton** based on domain specific metrics

Jan Drgona, et al., **Differentiable Predictive Control: An MPC Alternative for Unknown Nonlinear Systems using Constrained Deep Learning**, Journal of Process Control, 2022

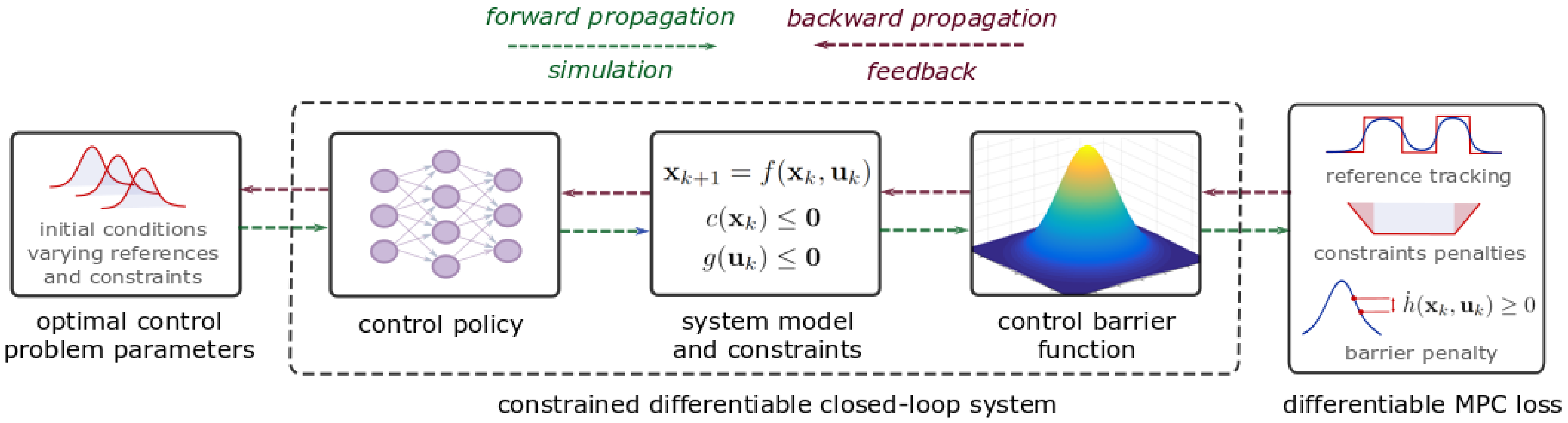
Learning Stable Policies with Neural Lyapunov Functions

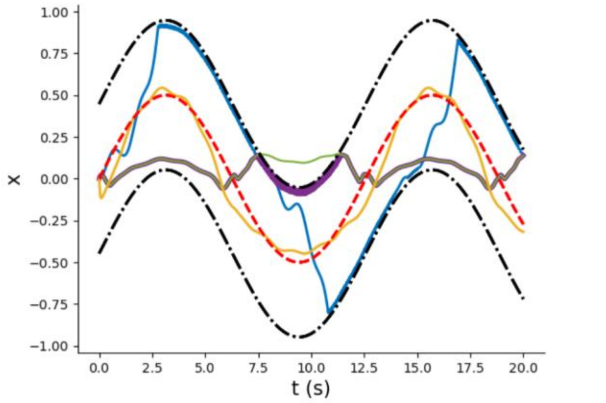




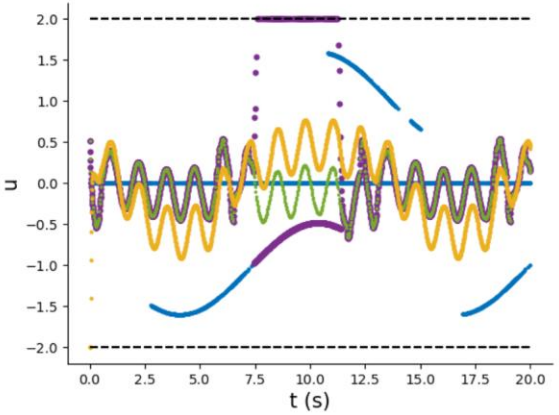
Sayak Mukherjee, Ján Drgoňa, Aaron Tuor, Mahantesh Halappanavar, Draguna Vrabie, **Neural Lyapunov Differentiable Predictive Control**, Conference on Decision and Control (CDC), 2022

Learning Safe Policies with Control Barrier Functions

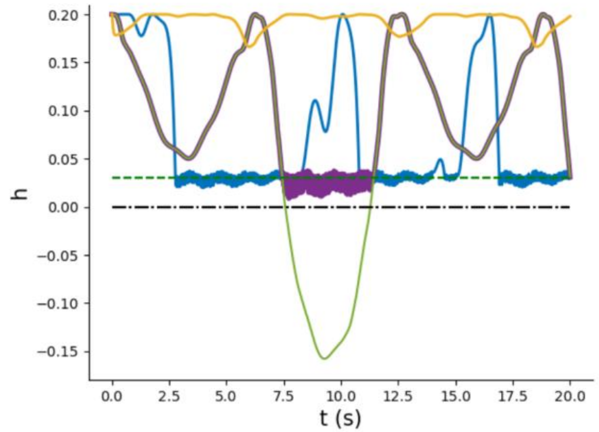




(a) State trajectories with reference (red dashed curve) and safe set boundary, $\partial\mathcal{C}(t)$ (black dashed-dotted curves).



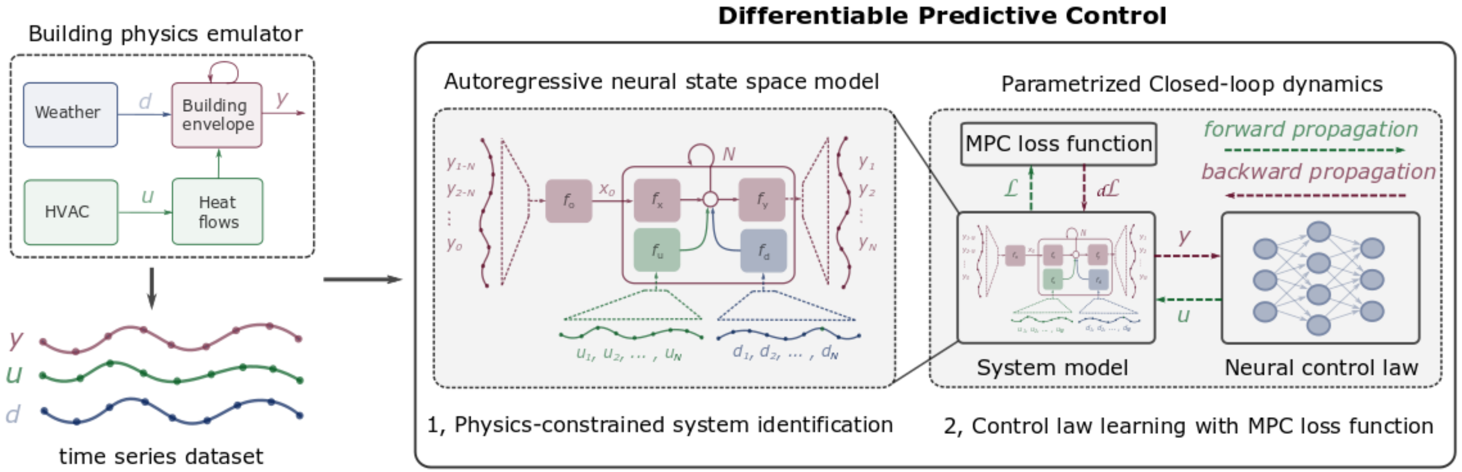
(b) Input trajectories with input bounds (black dashed lines).

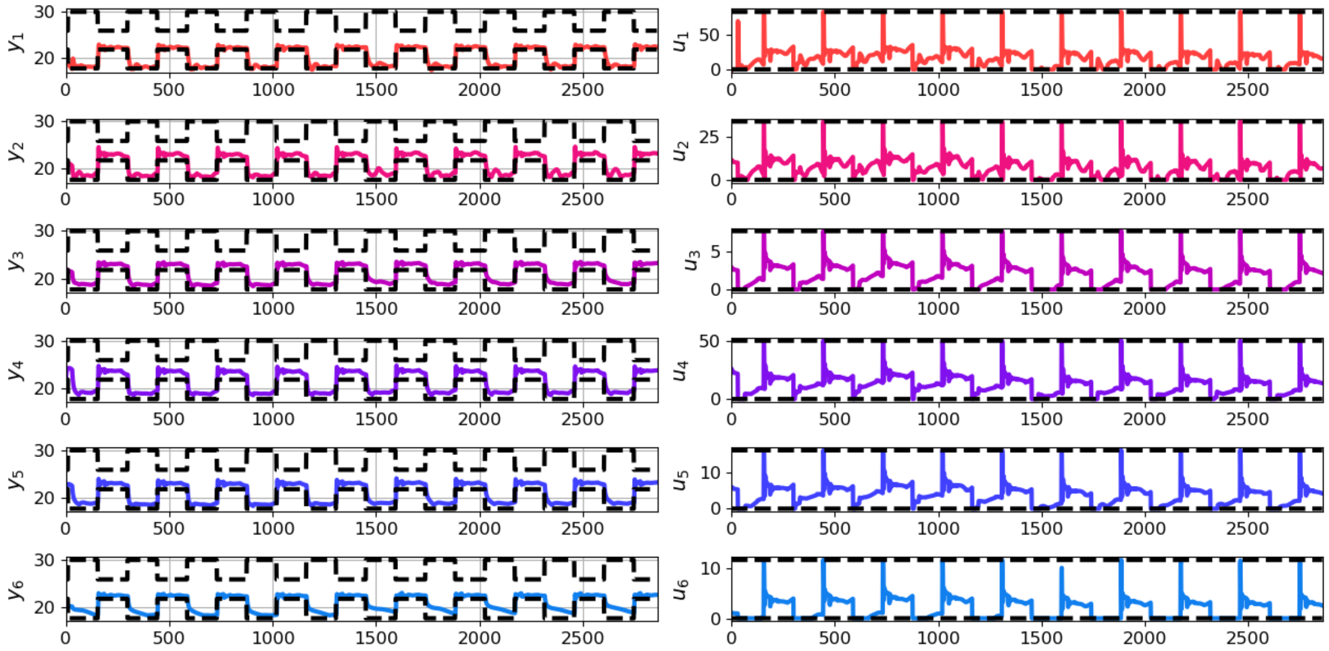


(c) Barrier trajectories with $h = a$ (green dashed line).

Wenceslao Shaw Cortez, et al., **Differentiable Predictive Control with Safety Guarantees: A Control Barrier Function Approach**, Conference on Decision and Control (CDC), 2022

Economic Differentiable Predictive Control of Building Energy System





Architectures

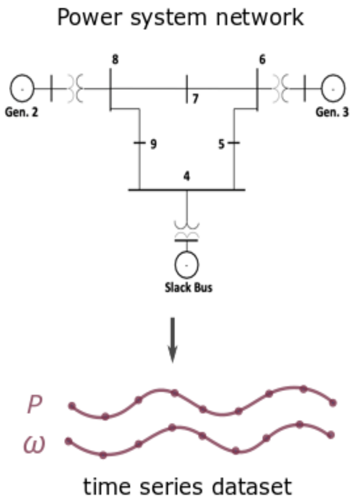
- Neural state space dynamics model
- MLP control policy architecture

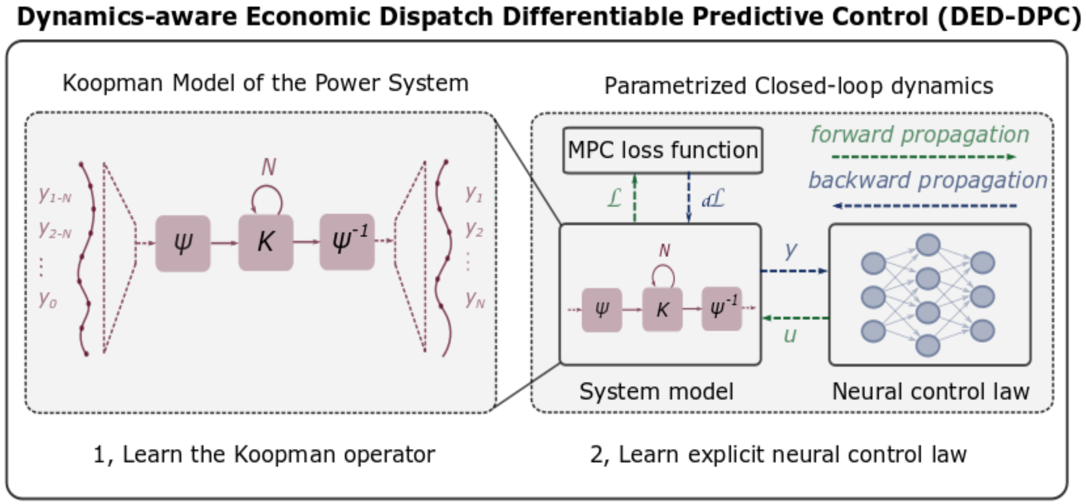
Results

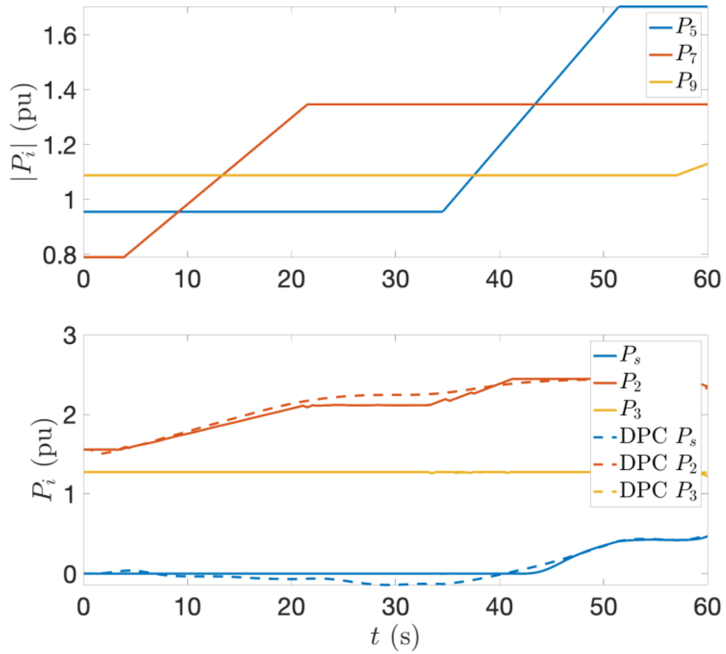
- Multi-input multi-output system dynamics
- Dynamic economic control objective
- Parametric constraints

Jan Drgona, Aaron Tuor, Elliott Skomski, Soumya Vasisht, Draguna Vrabie, **Deep Learning Explicit Differentiable Predictive Control Laws for Buildings**, IFAC NMPC 2021

Dynamics-Aware Economic Redispatch via Differentiable Predictive Control







Problem: Current redispatch processes do not incorporate system dynamics concerns. Incorporating dynamics in redispatch is too complex and/or time-consuming

Solution: Machine-learning based dynamics-aware redispatch. Learn system dynamics and control policies for faster assessment.

Ethan King, Ján Drgoňa, Aaron Tuor, Shrirang Abhyankar, Craig Bakker, Arnab Bhattacharya, Draguna Vrabie, **Koopman-based Differentiable Predictive Control for the Dynamics-Aware Economic Dispatch Problem**, 2022 American Control Conference

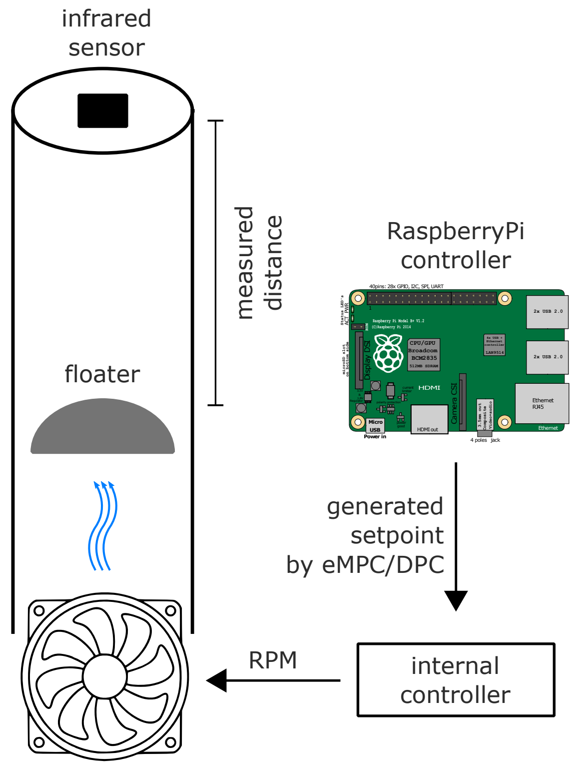
Architectures

- Koopman dynamics model
- CNN control policy

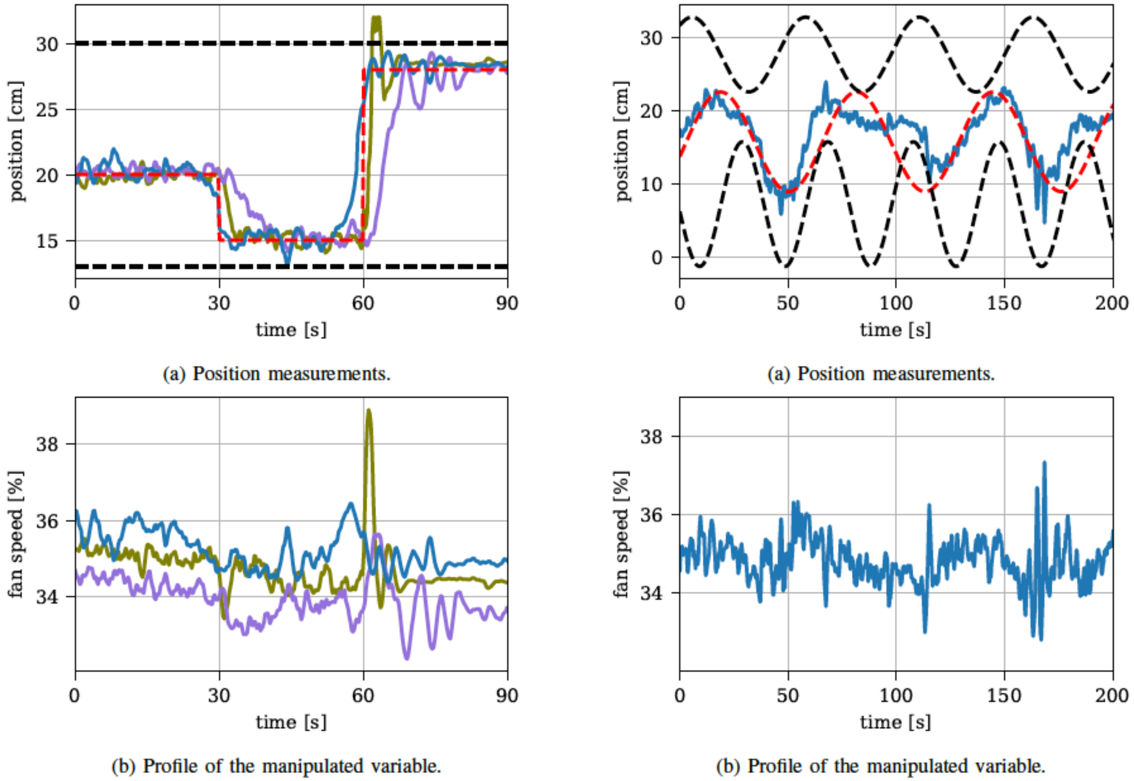
Results

- **5 orders of magnitude speed-up**
- Near optimal performance

Edge Deployment of Differentiable Predictive Control



Flexy-Air system sketch with Raspberry-Pi controller



(a) Position measurements.

(b) Profile of the manipulated variable.

(a) Position measurements.

(b) Profile of the manipulated variable.

Jan Drgona, Karol Kis, Aaron Tuor, Draguna Vrabie, Martin Klauco, **Differentiable Predictive Control: An MPC Alternative for Unknown Nonlinear Systems using Constrained Deep Learning**; Journal of Process Control, 2022

Proof of concept for AI on the edge platform.

NeuroMANCER

Open-source scientific machine learning (SciML) toolbox in PyTorch for integrating deep learning, constrained optimization, and physics-based modeling

- **Physics-informed machine learning**
- **Data-driven modeling of dynamical systems**
- **Model-based policy optimization**
- **Parametric constrained optimization**

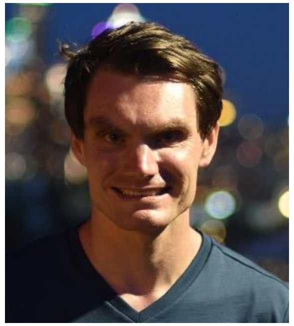
github.com/pnnl/neuromancer

Acknowledgements


Aaron Tuor


James Koch


Zhao Chen


Stefan Dernbach


Christian M. Legaard


Ethan King


Sayak Mukherjee


Wenceslao Shaw Cortez


Soumya Vasisht


Shrirang Abhyankar


Mahantesh Halappanavar


Draguna Vrabie

U.S. DEPARTMENT OF ENERGY

Pacific Northwest NATIONAL LABORATORY