

Optimizing Experiments with Pyomo.DoE

dowlinglab.github.io/pyomo-doe

Alexander (Alex) Dowling, Ph.D.

Daniel Laky, Ph.D.

Chemical and Biomolecular Engineering

University of Notre Dame

University of Notre Dame

March 17 and 21, 2025

Contributors to Pyomo.DoE and ParmEst: Jialu Wang (ND), Dan Laky (ND), Hailey Lynch (ND), John Siirola (SNL), Bethany Nicholson (SNL), Miranda Mundt (SNL), Shawn Martin (SNL), Katherine Klise (SNL)



Power of Adaptive Sequential Optimal Experiments

Self-Driving Laboratories

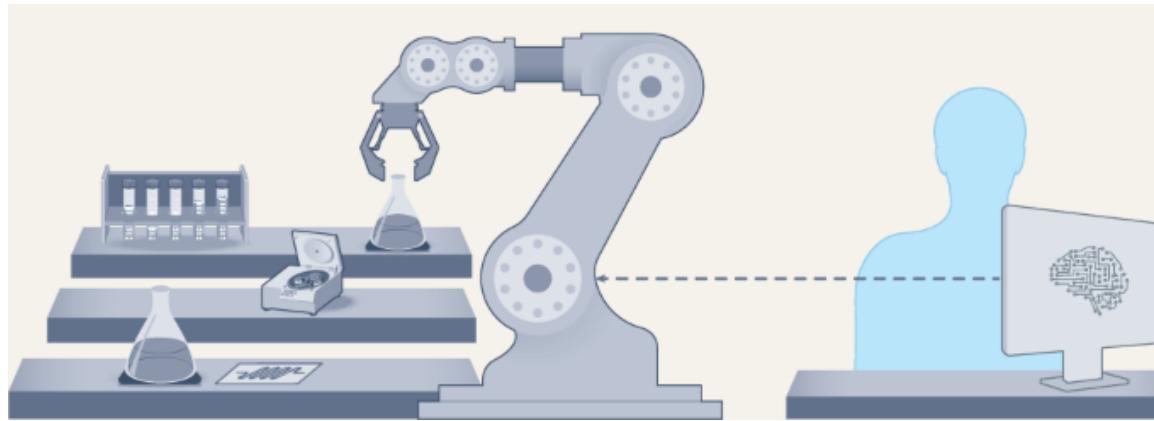


Figure: Abolhasani & Kumacheva (2023), *Nature Syn.*

Epps et al. (2022), *Advanced Materials*

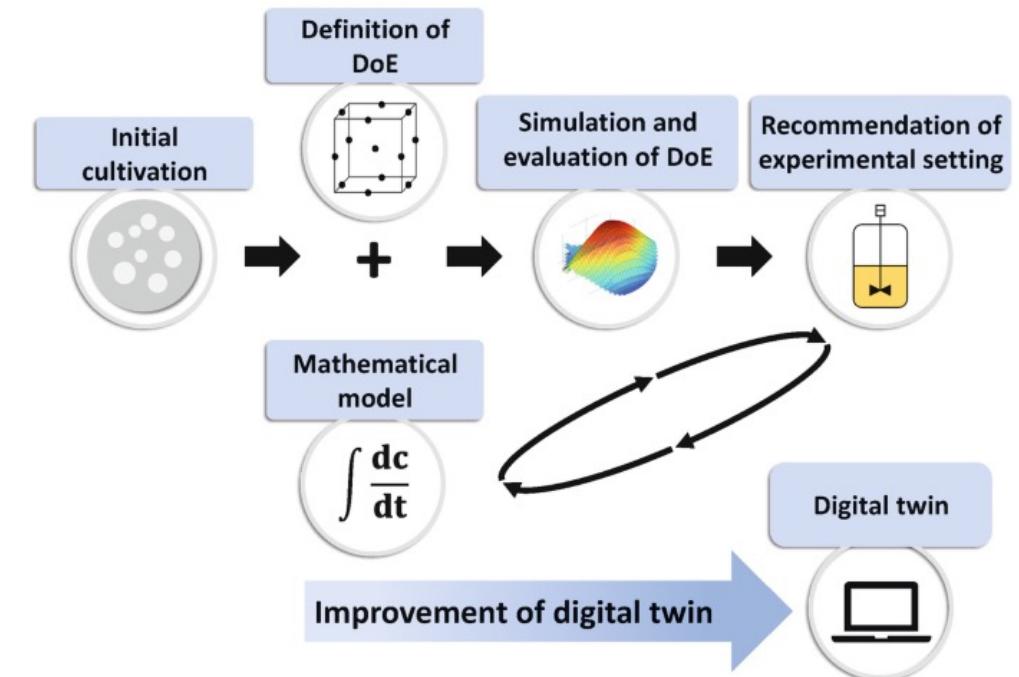
MacLeod et al. (2020), *Science Advances*

MacLeod et al. (2022), *Nature Communications*

Hase, Roch, Aspuru-Guzik (2019), *Trends in Chemistry*

Seifrid et al. (2022), *Acc. Chem. Res.*

Automation + Model-Based Design of Experiments

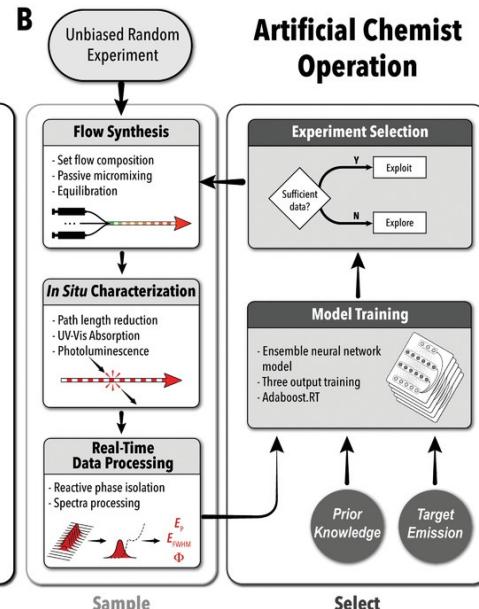
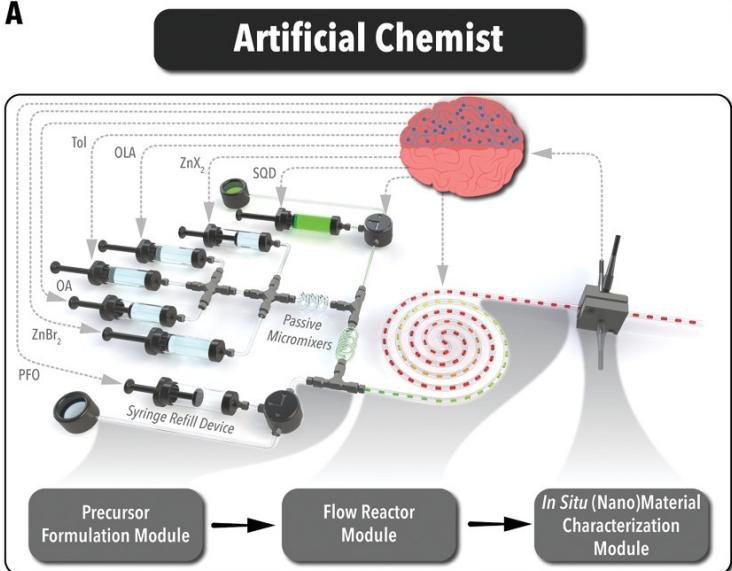


Kuchemuller et al. (2020), *Digital Twins*

Many Recent Examples of Sequential Optimal Experiments

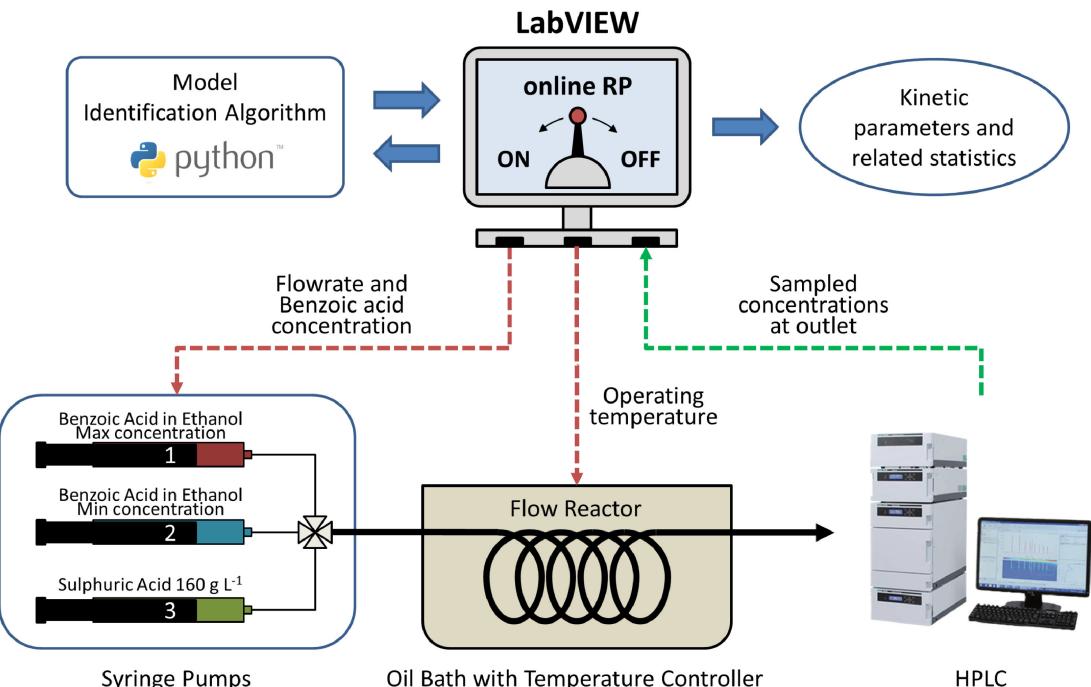
Quantum Dots (Machine Learning)

A



Epps et al. (2022), *Advanced Materials*

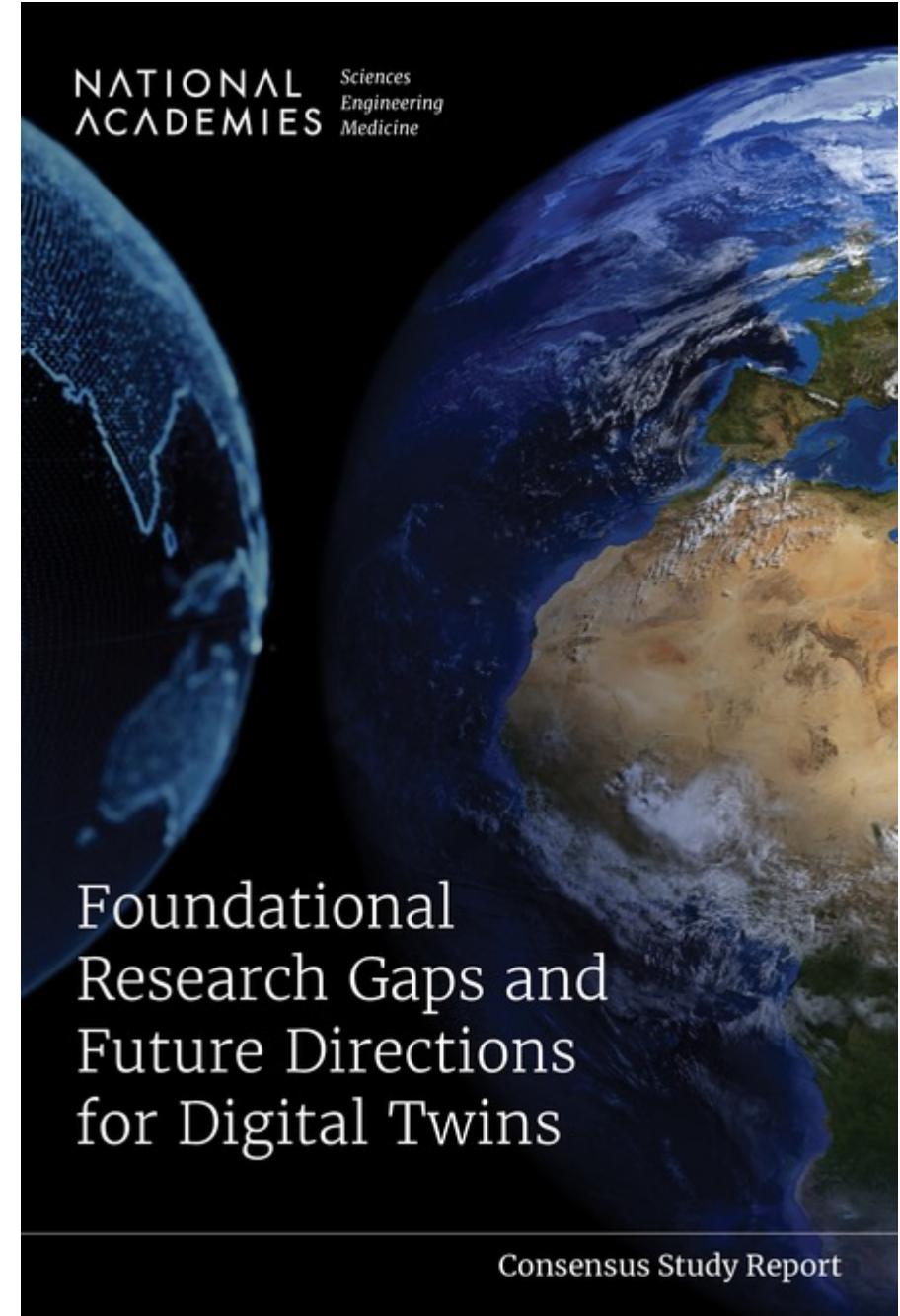
Reaction Engineering (Science-based Models)



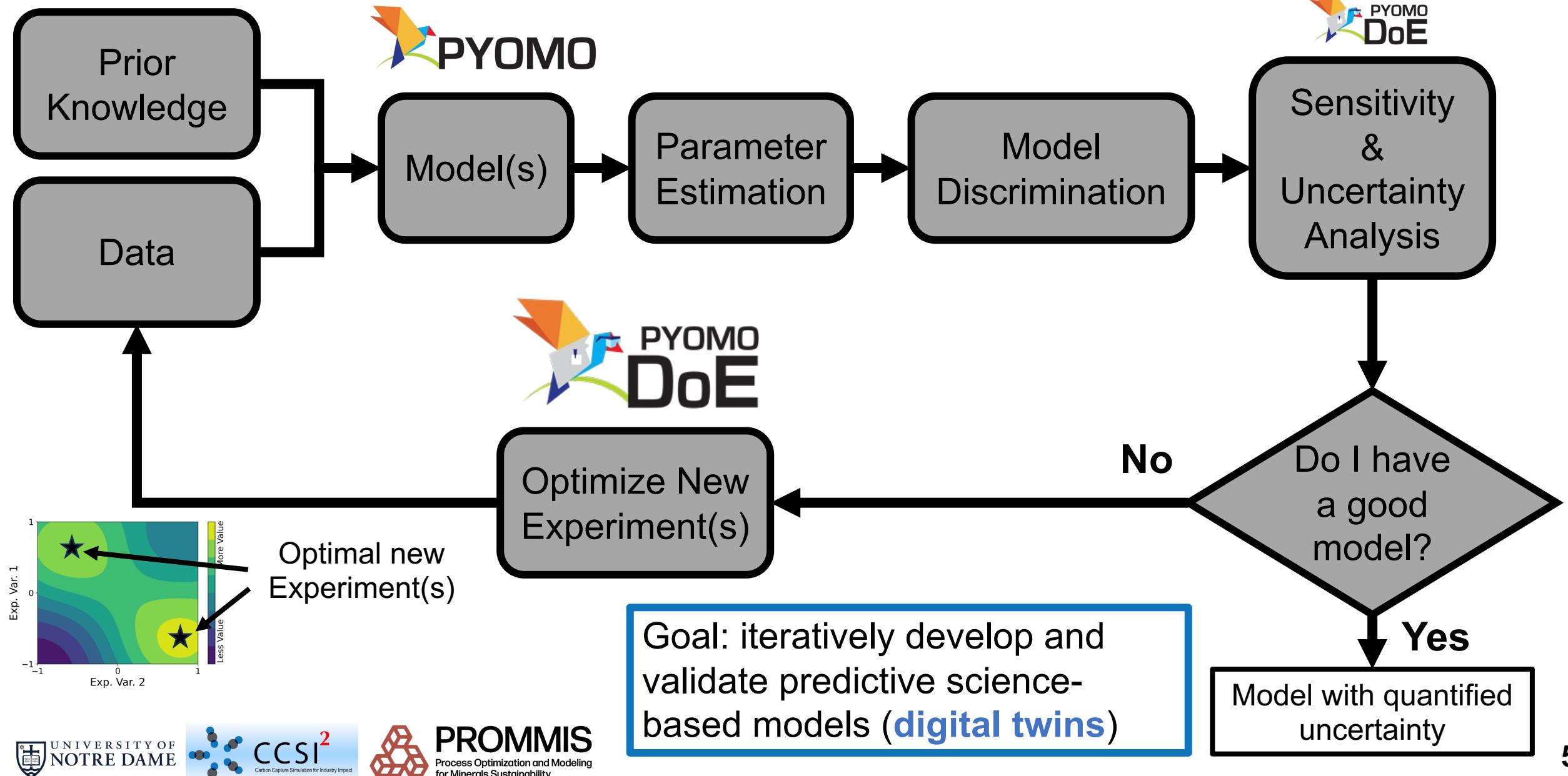
Quaglio et al. (2019), *Comp. & Chem. Eng.*

What is a Digital Twin?

A **digital twin** is a set of **virtual information constructs** that mimics the structure, context, and behavior of a **natural, engineered, or social system** (or **system-of-systems**), is **dynamically updated** with data from its physical twin, has a **predictive capability**, and **informs decisions** that realize value. The **bidirectional interaction** between the virtual and the physical is central to the digital twin.

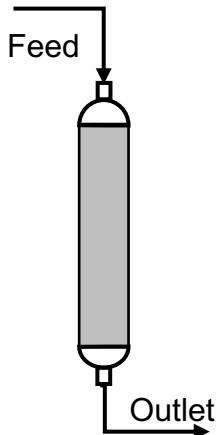


Science-based Data Analytics Workflow



SBDoe (a.k.a. MBDoE) Facilitates Collaborations

CO₂ Capture

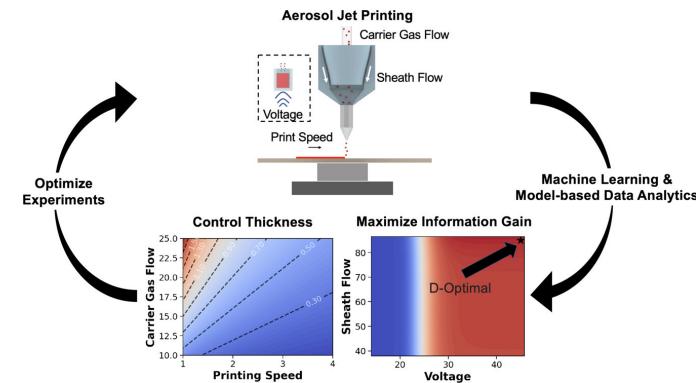


Jialu
Wang



Wang, J. and Dowling, A.W.
(2022), *AIChE J.* e17813.

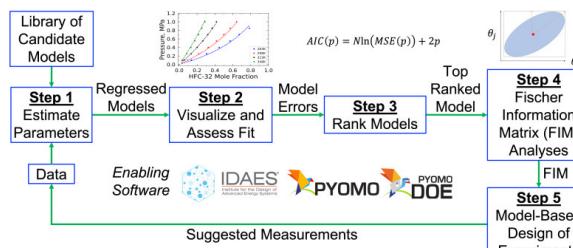
Additive Manufacturing of Thermoelectric Devices



Ke
Wang

Wang K., Zhang M., Wang, J., Shang, W.,
Zhang, Y., Luo, T., Dowling, A.W. (2023),
Digital Chemical Engineering

Thermodynamic Modeling (Refrigerants)

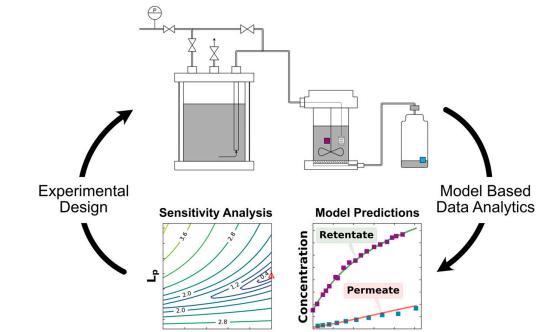


Dr. Bridgette
Befort



Befort, B.J., Garciadiego, A., Wang, J.,
Wang, K., Maginn, E.J., Dowling, A.W.
(2023), *Fluid Phase Equilibria*.

Rapid/Automated Membrane Characterization



Xinhong
Liu



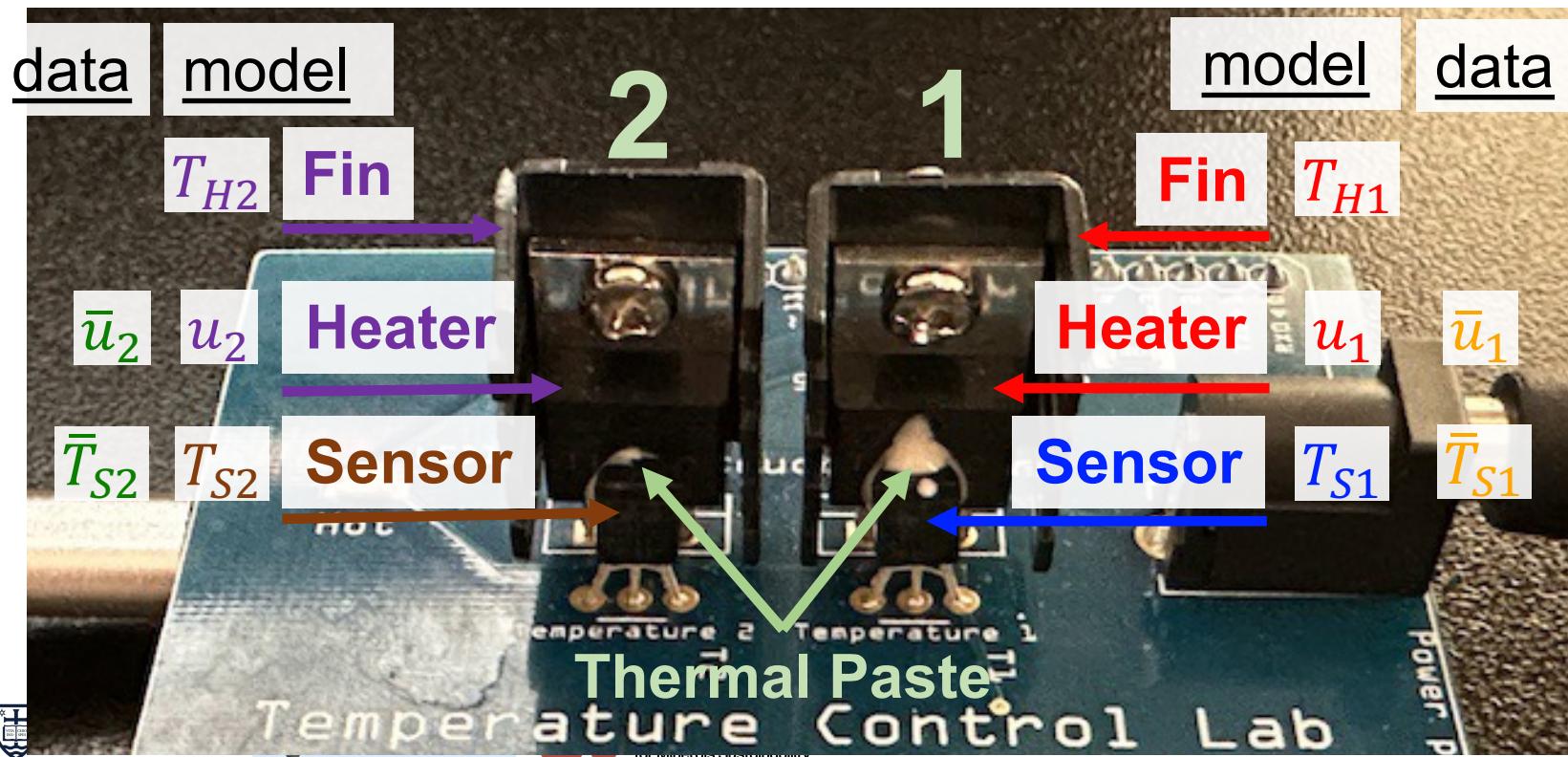
Ouimet, J.A., Xinhong, L.,
Brown, D.J., Eugene, E.A.,
Popps, T., Muetzel, Z.W.,
Dowling, A.W., Phillip, W.A.,
(2022). *J. Membrane Science*.

Pyomo.DoE Example: Temperature Control Lab (TC Lab)

dowlinglab.github.io/pyomo-doe/notebooks/tclab_model.html

$$C_p^H \frac{dT_{H,1}}{dt} = U_a(T_{amb} - T_{H,1}) + U_b(T_{S,1} - T_{H,1}) + \alpha P_1 u_1$$

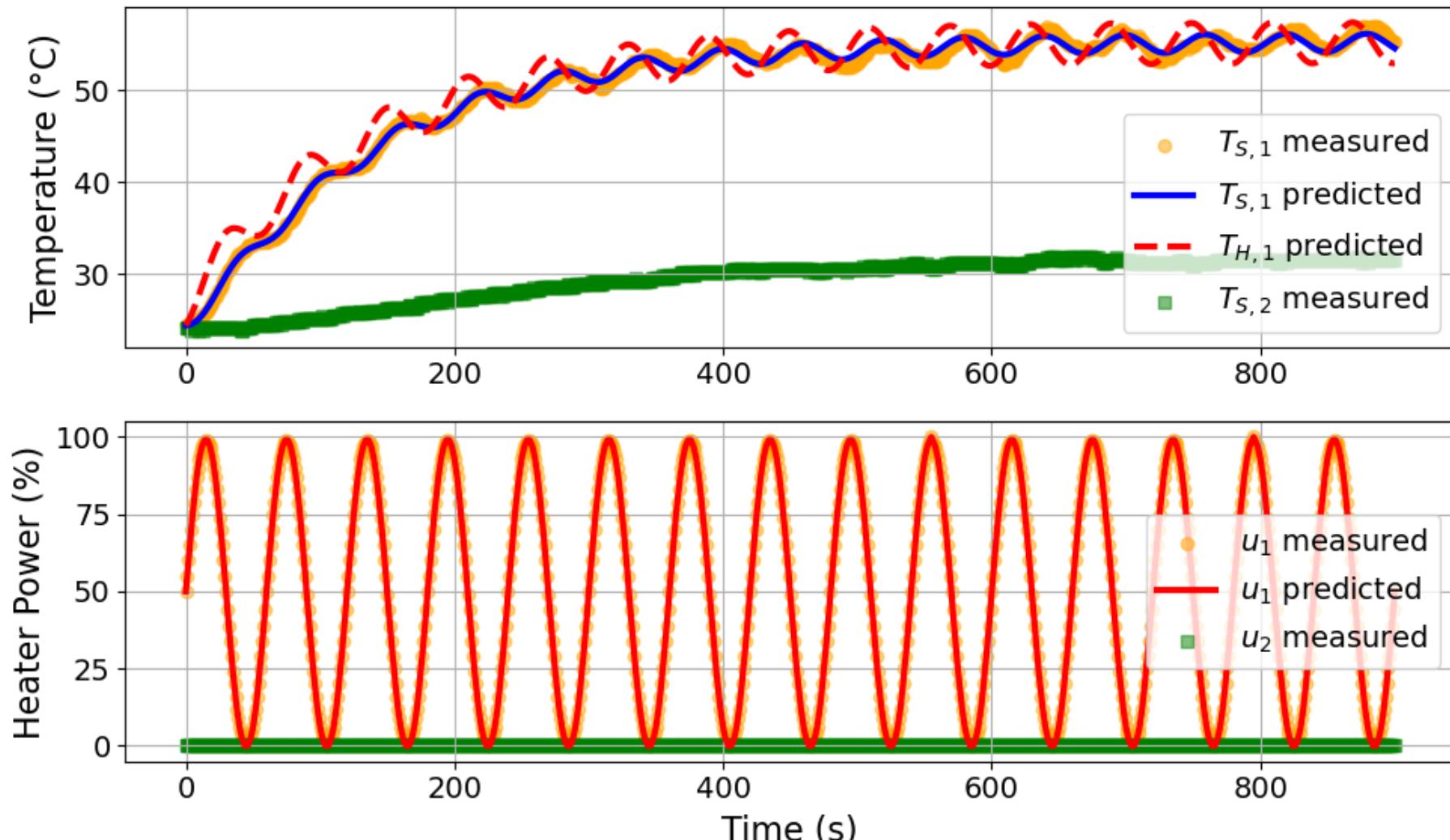
$$C_p^S \frac{dT_{S,1}}{dt} = U_b(T_{H,1} - T_{S,1}), \quad \theta = (U_a, U_b, C_p^H, C_p^S)^\top$$



Thank you to Prof. Jeff Kantor
(1954-2023) for the TCLab
example and so much more.

TC Lab: Data and Parameter Estimation

Hands-On Tutorial: dowlinglab.github.io/pyomo-doe



Notebooks at dowlinglab.github.io/pyomo-doe



TCLab Mathematical Model

We will use the TCLab as a motivating example for this workshop.

The code below sets default font sizes.

```
import matplotlib.pyplot as plt

SMALL_SIZE = 14
MEDIUM_SIZE = 16
BIGGER_SIZE = 18

plt.rc('font', size=SMALL_SIZE) # controls default text sizes
plt.rc('axes', titlesize=SMALL_SIZE) # fontsize of the axes title
plt.rc('axes', labelsize=MEDIUM_SIZE) # fontsize of the x and y labels
plt.rc('xtick', labelsize=SMALL_SIZE) # fontsize of the tick labels
plt.rc('ytick', labelsize=SMALL_SIZE) # fontsize of the tick labels
plt.rc('legend', fontsize=SMALL_SIZE) # legend fontsize
plt.rc('figure', titlesize=BIGGER_SIZE) # fontsize of the figure title
plt.rc('lines', linewidth=3)
```

Notebooks at dowlinglab.github.io/pyomo-doe



Modeling with Pyomo

This page is [adapted from our process control class](#) at Notre Dame; it was developed by Prof. Jeff Kantor.

```
# Install Pyomo and solvers for Google Colab
import sys

if "google.colab" in sys.modules:
    !wget "https://raw.githubusercontent.com/IDAES/idaes-pse/main/scripts/colab_helper.py"
    import colab_helper

    colab_helper.install_idaes()
    colab_helper.install_ipopt()

# Set plotting defaults
import matplotlib.pyplot as plt

SMALL_SIZE = 14
MEDIUM_SIZE = 16
BIGGER_SIZE = 18

plt.rc('font', size=SMALL_SIZE) # controls default text sizes
# for larger fonts, increase SMALL_SIZE by 2 or 4
```

TC Lab: Dynamic Optimization in Pyomo

dowlinglab.github.io/pyomo-doe/notebooks/pyomo_simulation.html

$$\min_{u(t)} \int_{t_0}^{t_f} \| SP(t) - T_H(t) \|^2 dt$$

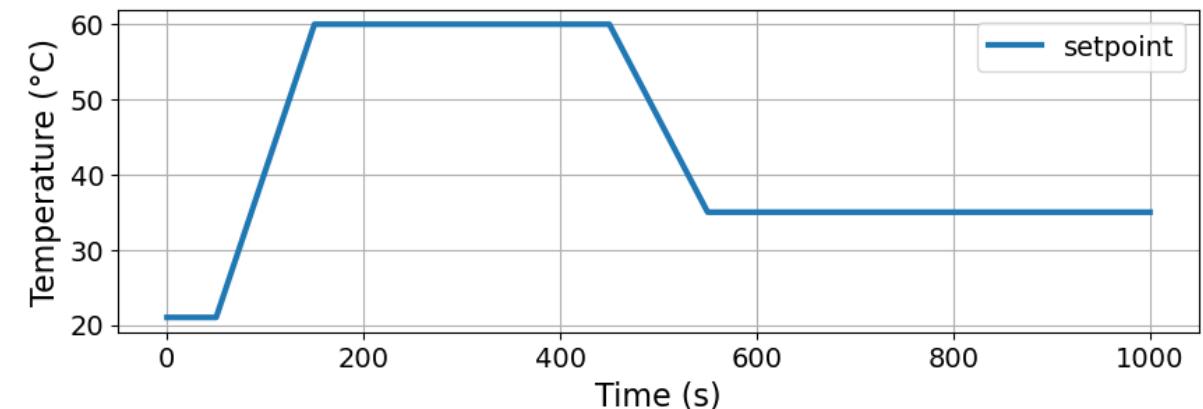
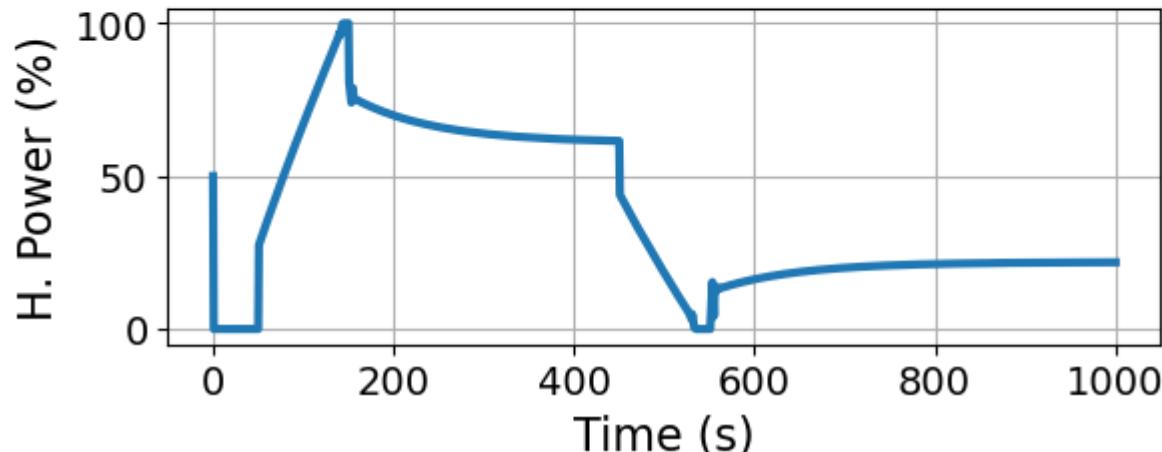
$$\text{s. t. } C_p^H \frac{dT_H}{dt} = U_a(T_{amb} - T_H) + U_b(T_S - T_H) + \alpha P u(t)$$

$$C_p^S \frac{dT_S}{dt} = U_b(T_H - T_S)$$

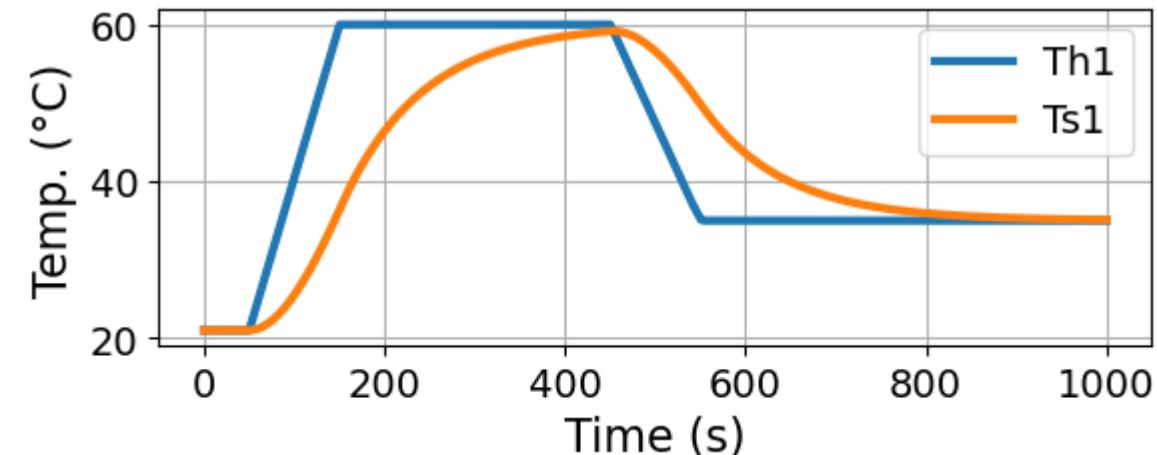
$$T_H(t_0) = T_{amb}$$

$$T_S(t_0) = T_{amb}$$

Optimal control profile



T_{h1} matches the setpoint using optimal control



End Session 1

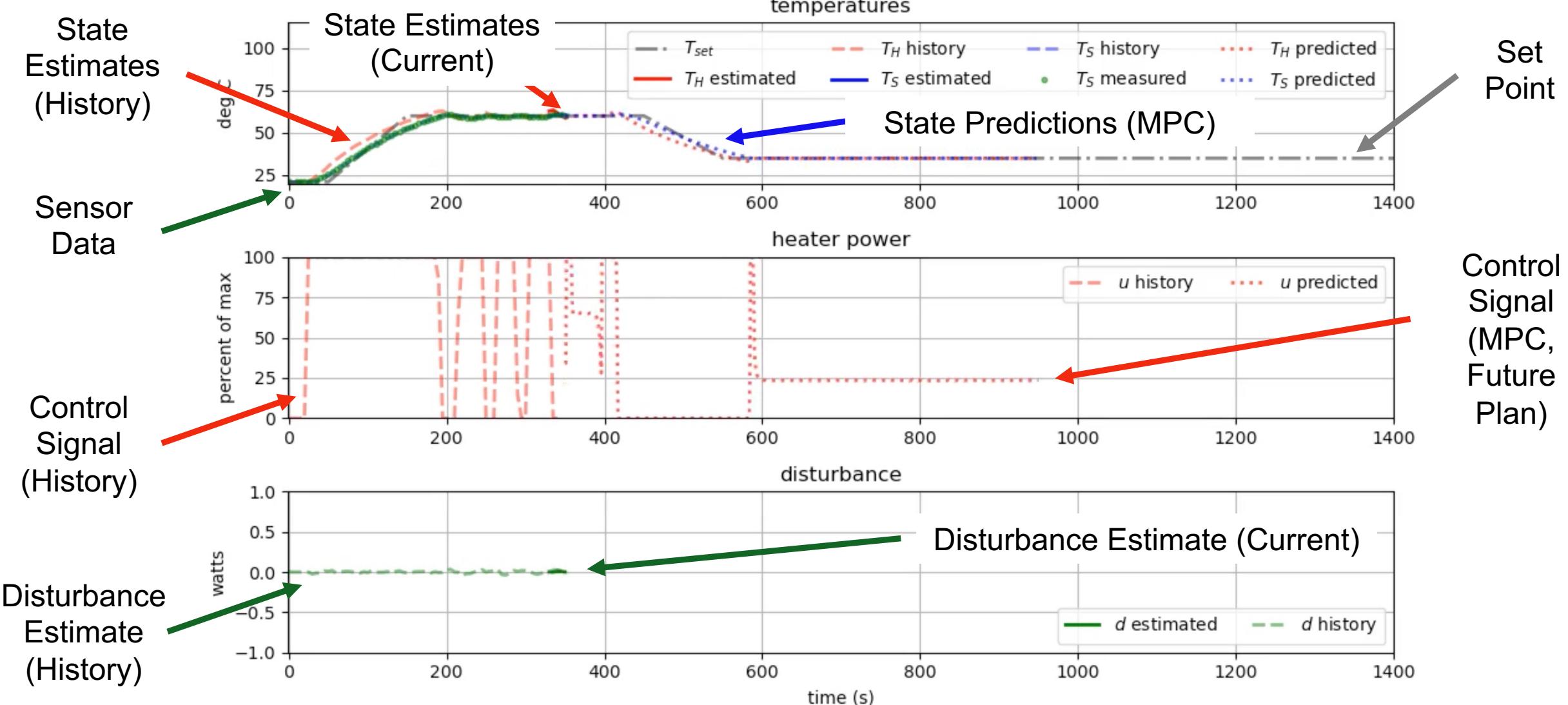
Recap Session 1:

- Motivation: sequential optimal experiments, digital twins
- Temperature control lab (TCLab) example
- Hands-on with Pyomo

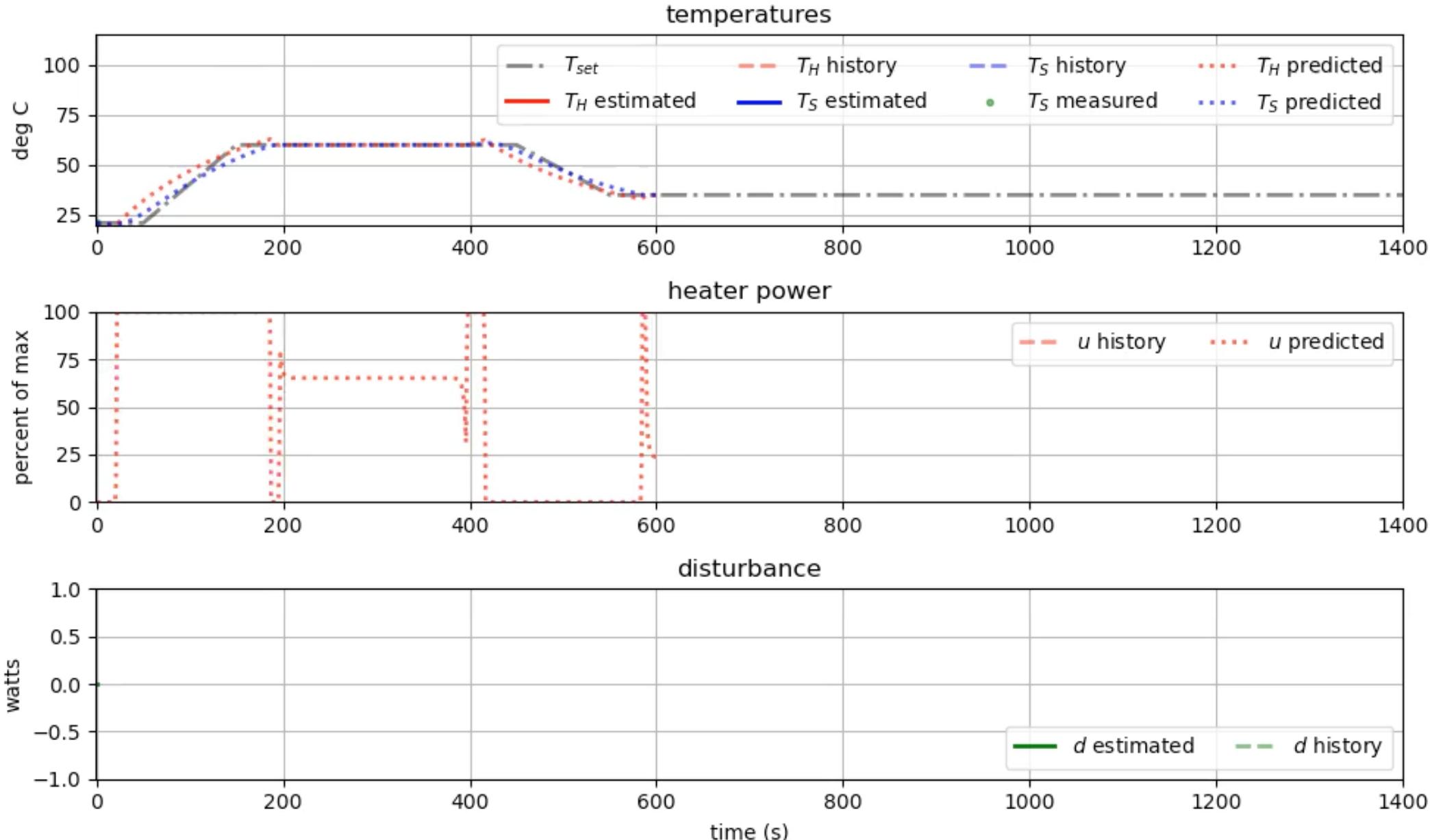
Next week, in **Session 2**, we will discuss:

- Parameter estimation
- Optimal experiment design

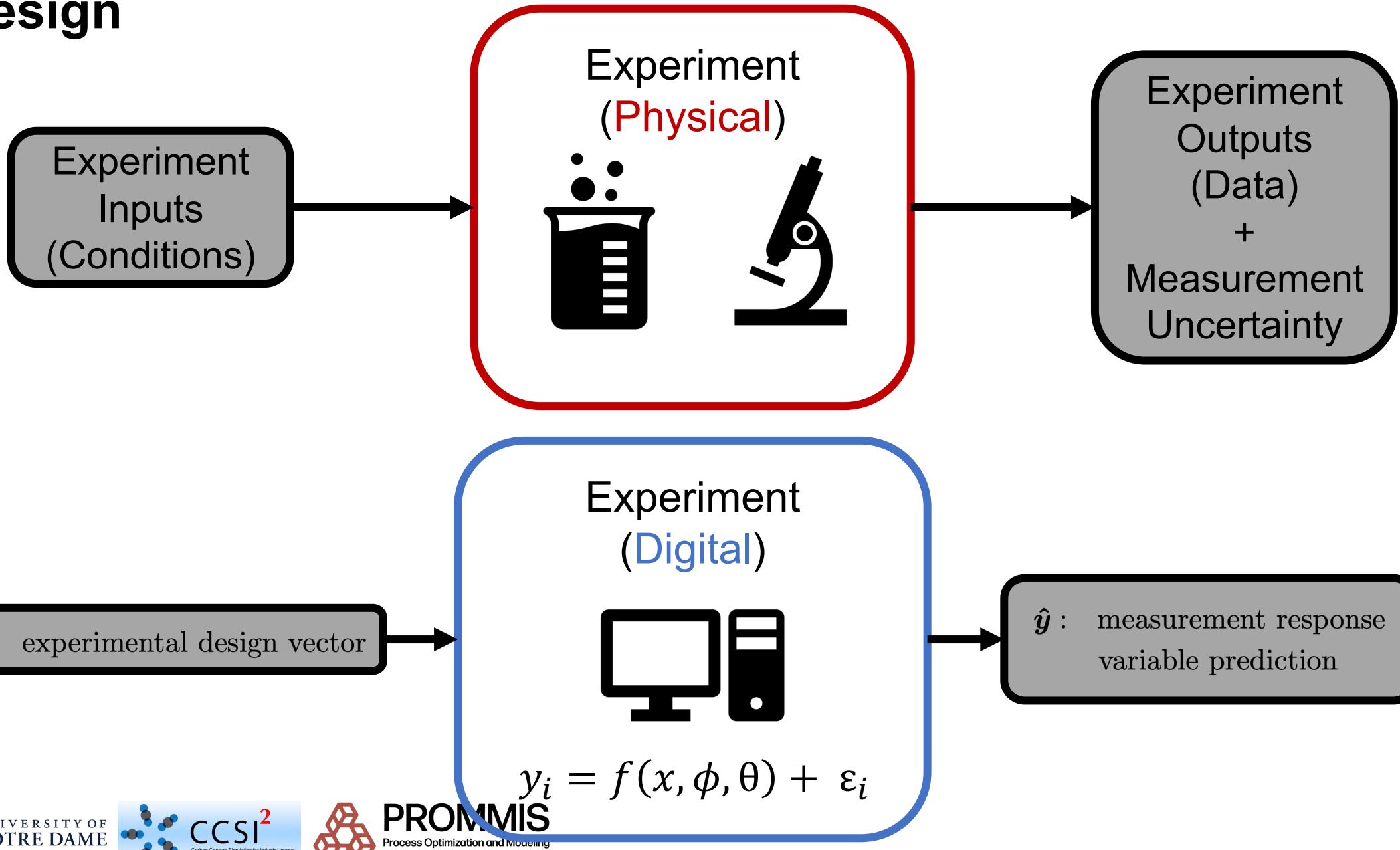
Teaching Digital Twins (MPC, State Estimation)



Teaching Digital Twins (MPC, State Estimation)



“Experiment” Abstraction Streamlines Closed-Loop Experiment Design



Dr. Bethany
Nicholson



Dr. John
Siirola



Dr. Shawn
Martin



Katherine
Klise

“Experiment” Abstraction Streamlines Closed-Loop Experiment Design

Known Variables

ϕ : experimental design vector
 y : measurement response variable



Experiment (Digital)



$$y_i = f(x, \phi, \theta) + \varepsilon_i$$

Unknown Variables

θ : unknown model parameters

$$\min_{\theta} \sum_i (y_i - \hat{y}_i)^2$$



Dr. Bethany Nicholson



Dr. John Siirola



Dr. Shawn Martin



Katherine Klise

θ : model parameters

Experiment (Digital)



$$y_i = f(x, \phi, \theta) + \varepsilon_i$$

ϕ : experimental design vector

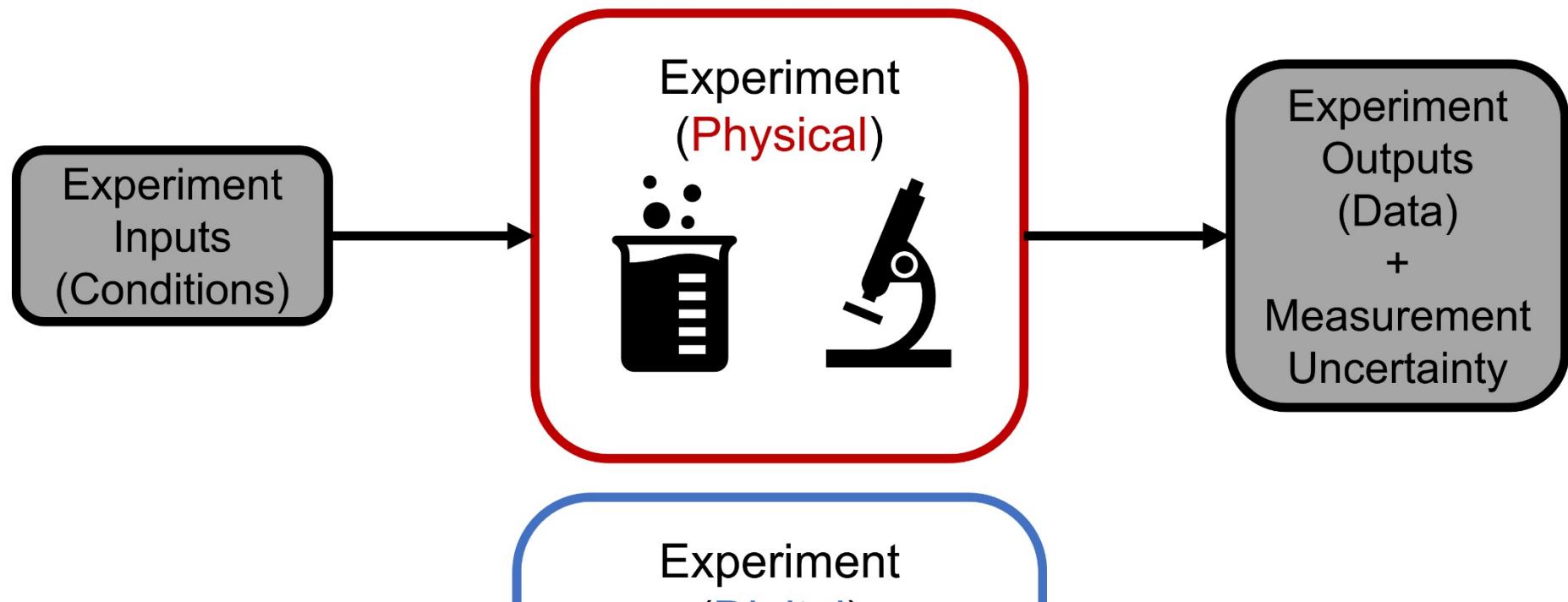
$$\max_{\phi_l \leq \phi \leq \phi_u} \Psi(\text{FIM}(\theta, \phi))$$

Notebooks at dowlinglab.github.io/pyomo-doe



Experiment Abstraction

The fundamental basis to parameter estimation and to optimal, science-based design of experiments (SBDoE) is an experiment. This concept is also borrows from the concept of a physical experiment:



Notebooks at dowlinglab.github.io/pyomo-doe



Nonlinear Regression with ParmEst

```
import sys

# If running on Google Colab, install Pyomo and Ipopt via IDAES
on_colab = "google.colab" in sys.modules
if on_colab:
    !wget "https://raw.githubusercontent.com/dowlinglab/pyomo-doe/main/notebooks/tclab_pyomo.py"

# import TCLab model, simulation, and data analysis functions
from tclab_pyomo import (
    TC_Lab_data,
    TC_Lab_experiment,
    extract_results,
    extract_plot_results,
)

# set default number of states in the TCLab model
number_tclab_states = 2
```

Parameter Estimation and Uncertainty Basics

Assume a model and error structure:

$$y_i = m(\mathbf{x}_i, \boldsymbol{\theta}) + \epsilon_i$$

measurement input observation
↓ variables error (i.i.d.)
model parameters

What values of model parameters $\boldsymbol{\theta}$ best fit the data X and y ?

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \Psi := \frac{1}{2} \sum_i [y_i - m(\mathbf{x}_i, \boldsymbol{\theta})]^2$$

best fit estimates

Bard (1974)
Bates and Watts (1988)
Pirnay, Lopez-Negrete, Biegler (2012)

How sensitive are the least-squares objective Ψ to perturbations in $\boldsymbol{\theta}$?

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 \Psi}{\partial \theta_1^2} & \cdots & \frac{\partial^2 \Psi}{\partial \theta_n \partial \theta_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \Psi}{\partial \theta_1 \partial \theta_m} & \cdots & \frac{\partial^2 \Psi}{\partial \theta_m^2} \end{bmatrix}$$

Hessian matrix

$$\mathbf{H} \approx \mathbf{Q}^T \mathbf{Q}$$

sensitivity matrix

How does measurement uncertainty ϵ propagate into uncertainty about the regressed parameters $\hat{\boldsymbol{\theta}}$?

covariance matrix for $\hat{\boldsymbol{\theta}}$

$$\mathbf{V}_{\hat{\boldsymbol{\theta}}} \approx \sigma_\epsilon^2 \mathbf{H}^{-1} \approx \sigma_\epsilon^2 (\mathbf{Q}^T \mathbf{Q})^{-1}$$

Fisher information matrix for $\hat{\boldsymbol{\theta}}$

$$\mathbf{M}_{\hat{\boldsymbol{\theta}}} \approx \mathbf{V}_{\hat{\boldsymbol{\theta}}}^{-1} \approx \frac{1}{\sigma_\epsilon^2} (\mathbf{Q}^T \mathbf{Q})$$

Extensions not shown: sophisticated error structures, Bayesian or MLE inference, ...

TCLab: Eigendecomposition of the Fisher Information Matrix

ParmEst: dowlinglab.github.io/pyomo-doe/notebooks/parmest.html

FIM: dowlinglab.github.io/pyomo-doe/notebooks/doe_exploratory_analysis.html

FIM:

```
[[517225.40941304  1360.01262476 -66404.72541298 -1002.47319402]
 [ 1360.01262476  5004.3737258  12379.2662576  5238.40389773]
 [-66404.72541298 12379.2662576  65481.16908635 14190.01468139]
 [-1002.47319402  5238.40389773  14190.01468139  5526.94375493]]
```

eigenvalues:

```
[5.26802218e+05 6.26035823e+04 3.83207978e+03 1.61037063e-02]
```

eigenvectors:

U_a	[-9.89752804e-01 -1.35949591e-01 4.36702406e-02 -7.52086327e-05]	U_a
U_b	[8.63262440e-04 -2.26164575e-01 -6.85698047e-01 -6.91857665e-01]	U_b
$1/C_p^H$	[1.42671125e-01 -9.31600001e-01 3.33329462e-01 -2.56487437e-02]	$1/C_p^H$
$1/C_p^S$	[5.79584008e-03 -2.49977462e-01 -6.45602485e-01 7.21578207e-01]	$1/C_p^S$

Difficult to uniquely estimate U_b and C_p^S with this single experiment!

Model-Based DoE Optimization Formulation

$$\begin{aligned}
 \max_{\varphi} \quad & \Psi[M(\hat{\theta}, \varphi)] \\
 \text{s. t.} \quad & \dot{x}(t) = f(x(t), z(t), u(t), \bar{w}, \hat{\theta}) \\
 & g(x(t), z(t), u(t), \bar{w}, \hat{\theta}) = 0 \\
 & y(t) = h(x(t), z(t), \hat{\theta}) \\
 & f^0(\dot{x}(t_0), x(t_0), z(t_0), u(t_0), \bar{w}, \hat{\theta}) = 0 \\
 & g^0(x(t_0), z(t_0), u(t_0), \bar{w}, \hat{\theta}) = 0 \\
 & y^0(t_0) = h(x(t_0), z(t_0), \hat{\theta})
 \end{aligned}
 \left. \begin{array}{l} \text{DAE System} \\ \text{Initial Conditions} \end{array} \right\} \quad m(x(t), y(t), z(t), u(t), \bar{w}, \hat{\theta}) = 0$$

y Measurements (model responses)

$\hat{\theta}$ Estimated parameters

x Time-dependent differential state variables

z Time-dependent algebraic state variables

u Time-varying control variables

\bar{w} Time-invariant control variable

Fisher information matrix (FIM):

$$M \approx V_{\hat{\theta}}^{-1} \approx \sigma_{\epsilon}^{-2} H \approx \sigma_{\epsilon}^{-2} Q^T Q$$

MBDoE Decisions:

$$\varphi = (u(t), x(t_0), z(t_0), \bar{w}, t)$$

Alphabetic Design Criteria Measure Information Content

Figure adapted from: Franceschini, G., & Macchietto, S. (2008). *Chem. Eng. Sci.*, 63(19), 4846-4872.

A-optimality

max trace(\mathbf{M})

enclosing box volume

poor choice for highly correlated θ

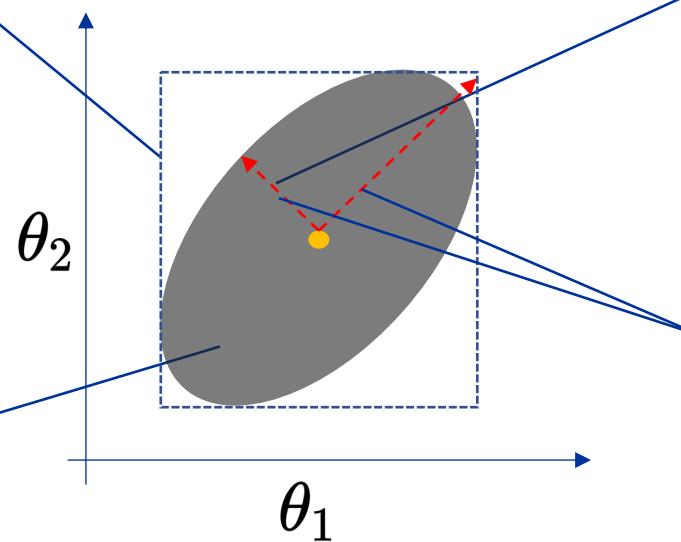
D-optimality

max det(\mathbf{M})

ellipsoid volume

robust to linear transformations

confidence ellipsoid for covariance matrix $\mathbf{V} = \mathbf{M}^{-1}$



E-optimality

max min(eig(\mathbf{M}))

major axis

recommended if \mathbf{M} is ill-conditioned

ME-optimality

$\min \kappa(\mathbf{M}) = \max(\text{eig}(\mathbf{M})) / \min(\text{eig}(\mathbf{M}))$

ratio of major to minor axes

recommended if \mathbf{M} is ill-conditioned

Model Discrimination

Hunter, W.G. and Reiner, A.M., 1965. Designs for discriminating between two rival models. *Technometrics*, 7(3), pp.307-323.

Buzzi-Ferraris, G. and Forzatti, P., 1983. A new sequential experimental design procedure for discriminating among rival models. *Chemical engineering science*, 38(2), pp.225-232.

Ferraris, G.B., Forzatti, P., Emig, G. and Hofmann, H., 1984. Sequential experimental design for model discrimination in the case of multiple responses. *Chemical engineering science*, 39(1), pp.81-85.

Joint Parameter Precision and Model Discrimination

Alberton, A.L., Schwaab, M., Lobão, M.W.N. and Pinto, J.C., 2011. Experimental design for the joint model discrimination and precise parameter estimation through information measures. *Chemical Engineering Science*, 66(9), pp.1940-1952.

Galvanin, F., Cao, E., Al-Rifai, N., Gavriilidis, A. and Dua, V., 2016. A joint model-based experimental design approach for the identification of kinetic models in continuous flow laboratory reactors. *Computers & Chemical Engineering*, 95, pp.202-215.

Galvanin, F., Cao, E., Al-Rifai, N., Dua, V. and Gavriilidis, A., 2015. Optimal design of experiments for the identification of kinetic models of methanol oxidation over silver catalyst. *Chimica Oggi-Chemistry Today*, 33(3), pp.51-56.

Pankajakshan, A., Waldron, C., Quaglio, M., Gavriilidis, A. and Galvanin, F., 2019. A Multi-Objective Optimal Experimental Design Framework for Enhancing the Efficiency of Online Model Identification Platforms. *Engineering*, 5(6), pp.1049-1059.

Pyomo.DoE Formulation: MBDoE as 2-Stage Program

max $\log \det(\mathbf{M}(\hat{\boldsymbol{\theta}}, \boldsymbol{\varphi})) = 2 \sum_{i=1}^{N_p} \log L_{ii}$ D-optimality

s.t.

$$\mathbf{M} = \sum_r \sum_{r'} \tilde{\sigma}_{r,r'} \mathbf{Q}_r^T \mathbf{Q}_{r'}$$

$\mathbf{M} = \mathbf{L}\mathbf{L}^T, \quad L_{ii} \geq \epsilon$ Cholesky factorization

$$q_{r,p}(t) = \frac{y_{r,p}^+(t) - y_{r,p}^-(t)}{2\epsilon_p}$$
 Central finite difference
$$\mathbf{m}(\mathbf{x}_p^+(t), \mathbf{y}_p^+(t), \mathbf{z}_p^+(t), \mathbf{u}(t), \bar{\mathbf{w}}, \boldsymbol{\theta}_p^+) = \mathbf{0}$$
 Two model evaluations
$$\mathbf{m}(\mathbf{x}_p^-(t), \mathbf{y}_p^-(t), \mathbf{z}_p^-(t), \mathbf{u}(t), \bar{\mathbf{w}}, \boldsymbol{\theta}_p^-) = \mathbf{0}$$

$$\boldsymbol{\theta}_p^+ = \hat{\boldsymbol{\theta}} + \mathbf{e}_p \epsilon_p$$
 Up and down perturbations
$$\boldsymbol{\theta}_p^- = \hat{\boldsymbol{\theta}} - \mathbf{e}_p \epsilon_p$$

Stage 2

$\forall p \in \{1, \dots, N_p\}$

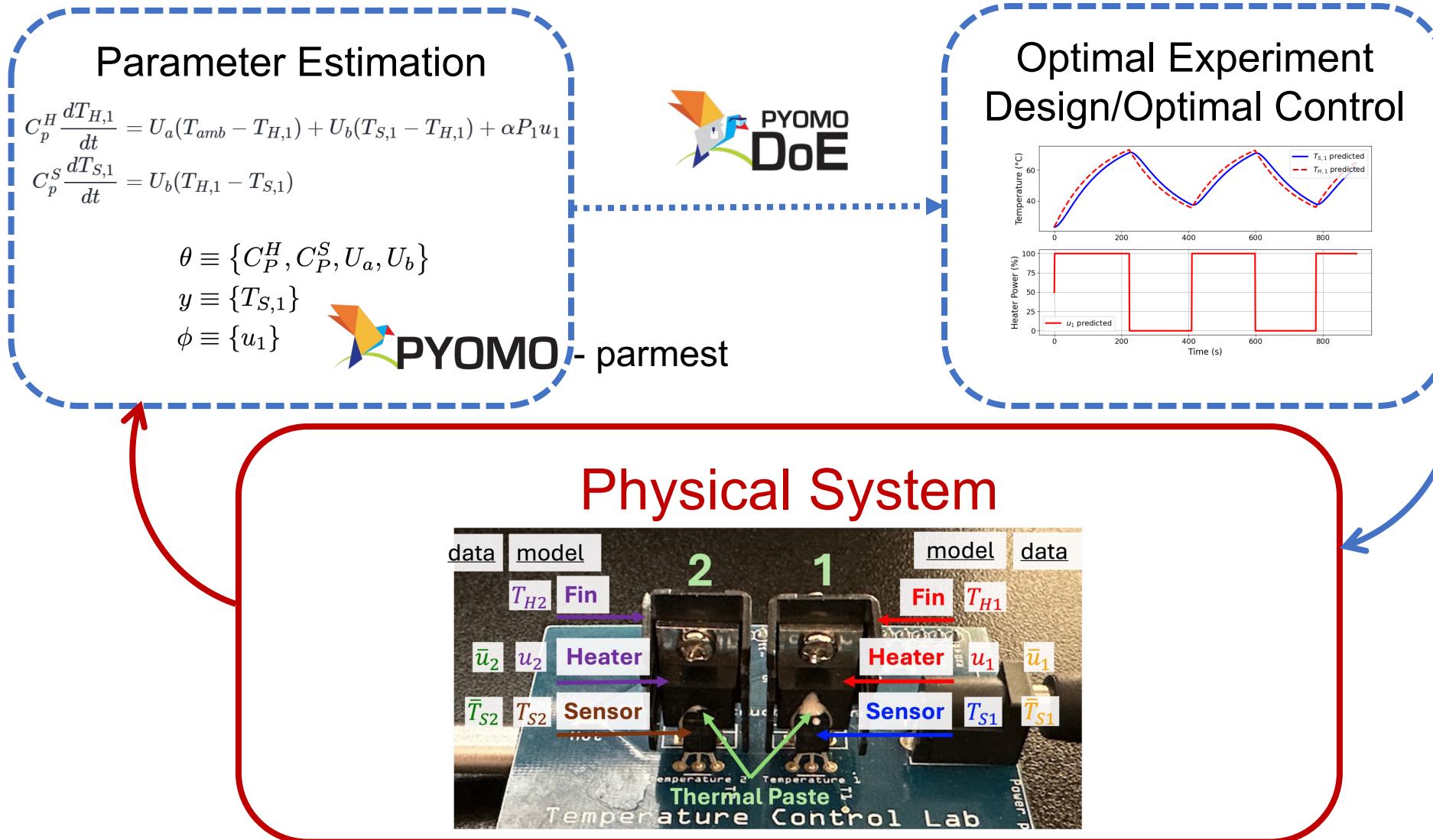
Model Sensitivity

$$\mathbf{Q}_r = \begin{bmatrix} \frac{\partial y_r(t_1)}{\partial \theta_1} & \dots & \frac{\partial y_r(t_1)}{\partial \theta_{N_p}} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_r(t_n)}{\partial \theta_1} & \dots & \frac{\partial y_r(t_n)}{\partial \theta_{N_p}} \end{bmatrix} = [\mathbf{q}_{r,1} \quad \dots \quad \mathbf{q}_{r,N_p}]$$

$$\mathbf{q}_{r,p} = \left[\frac{\partial y_r(t_1)}{\partial \theta_p} \quad \dots \quad \frac{\partial y_r(t_n)}{\partial \theta_p} \right]^T$$

\mathbf{y}	Measurements (model responses)
\mathbf{Q}_r	Dynamic sensitivity for response r
$\mathbf{m}()$	DAE model
$\hat{\boldsymbol{\theta}} \in \mathbb{R}^P$	Estimate for parameters
$\mathbf{M} \in \mathbb{R}^{P \times P}$	Fisher information matrix
$\mathbf{L} \in \mathbb{R}^{P \times P}$	Lower triangular Cholesky factorization
ϵ_p	Small perturbation for parameter p
$\mathbf{e}_p \in \mathbb{R}^P$	Unit vector with “1” in position p

Temperature Control Lab (TC-Lab) – Closed-Loop Experimental Design with New Experiment Abstraction



Notebooks at dowlinglab.github.io/pyomo-doe



Pyomo.DoE: Exploratory Analysis

```
import matplotlib.pyplot as plt

SMALL_SIZE = 14
MEDIUM_SIZE = 16
BIGGER_SIZE = 18

plt.rc('font', size=SMALL_SIZE) # controls default text sizes
plt.rc('axes', titlesize=SMALL_SIZE) # fontsize of the axes title
plt.rc('axes', labelsize=MEDIUM_SIZE) # fontsize of the x and y labels
plt.rc('xtick', labelsize=SMALL_SIZE) # fontsize of the tick labels
plt.rc('ytick', labelsize=SMALL_SIZE) # fontsize of the tick labels
plt.rc('legend', fontsize=SMALL_SIZE) # legend fontsize
plt.rc('figure', titlesize=BIGGER_SIZE) # fontsize of the figure title
plt.rc('lines', linewidth=3)
```

```
import sys

# If running on Google Colab, install Pyomo and Ipopt via IDAES
on_colab = "google.colab" in sys.modules
```

TC Lab: DoE, Exploratory Analysis

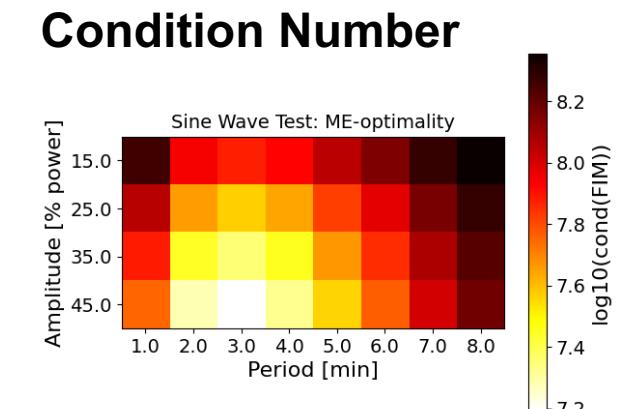
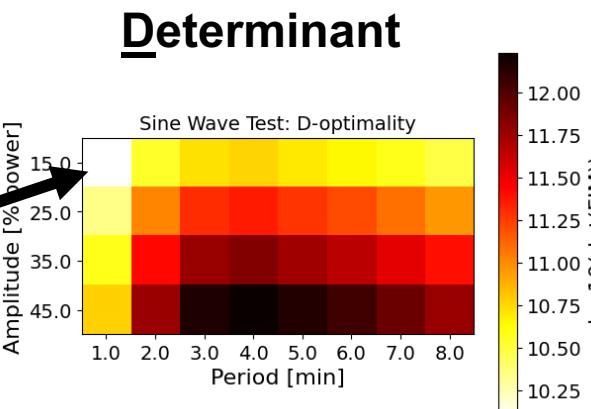
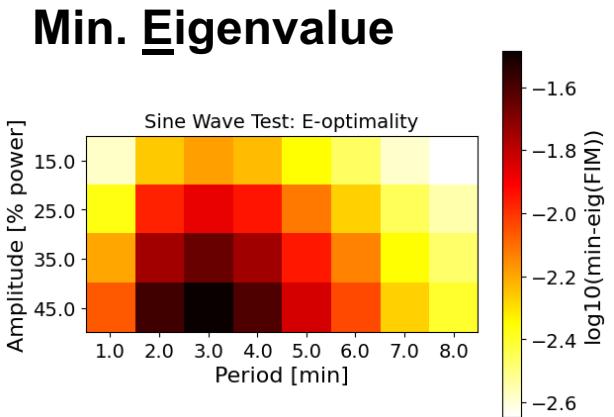
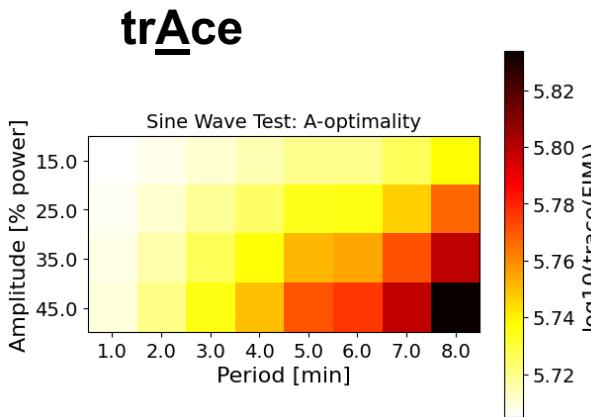
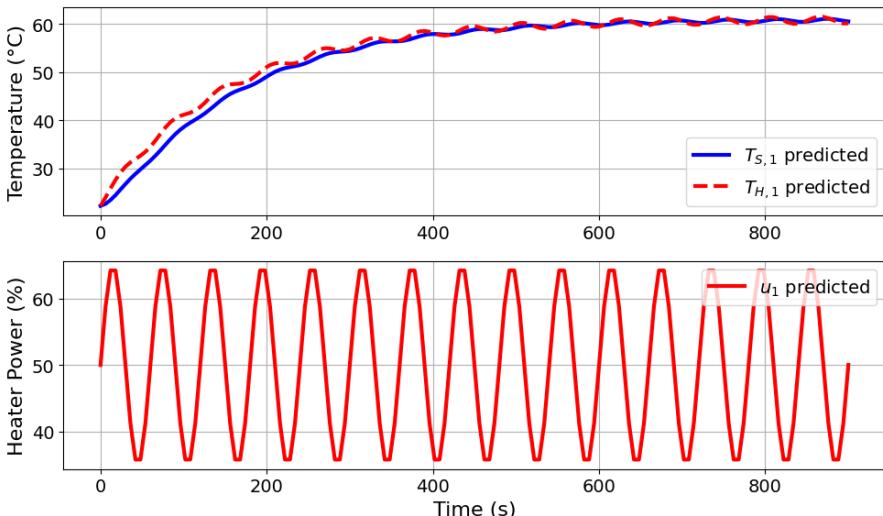
dowlinglab.github.io/pyomo-doe/notebooks/doe_exploratory_analysis.html

Sensitivity of the FIM to experimental design.

Example: Sine Wave

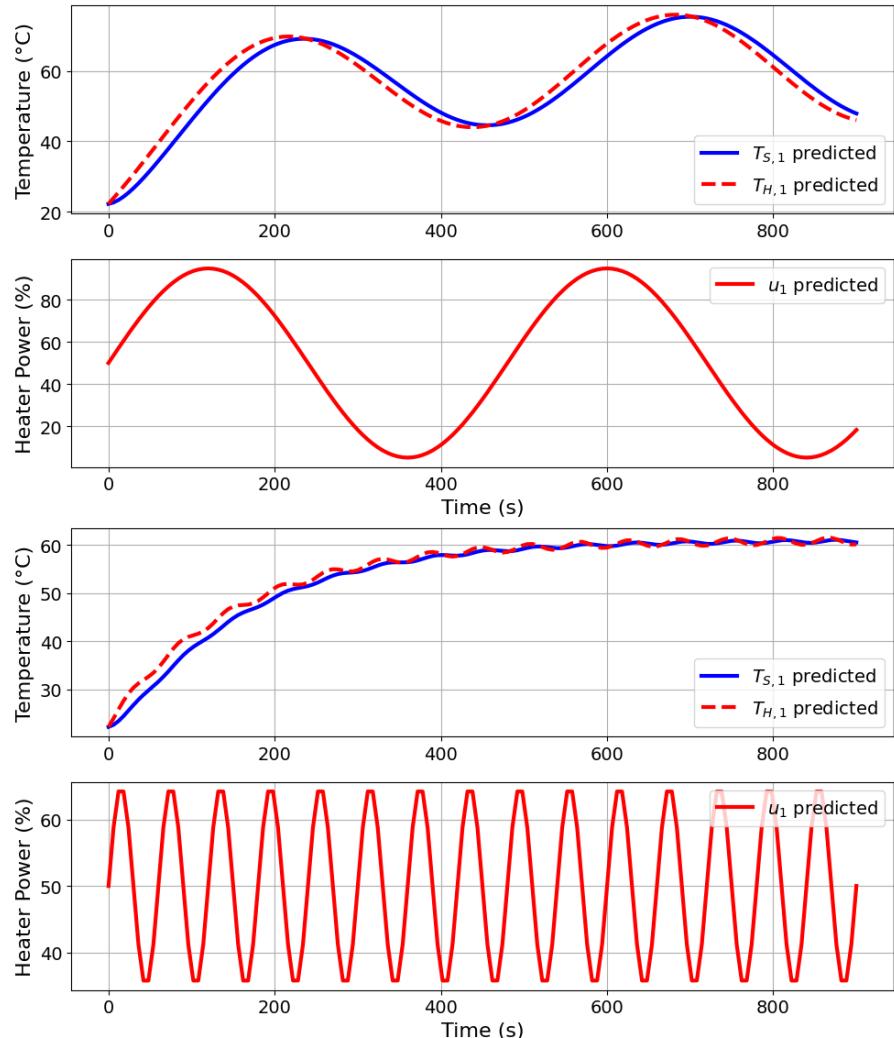
Vary:

- period (from 1 to 8 minutes)
- amplitude (from 15% to 50%)

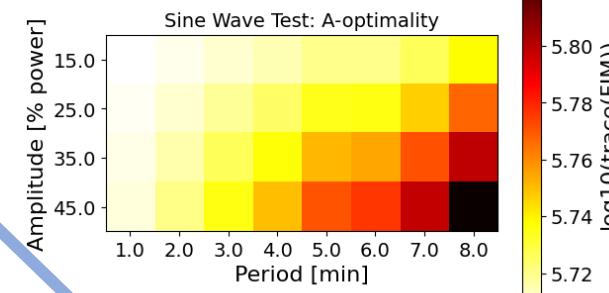


TC Lab: DoE, Exploratory Analysis

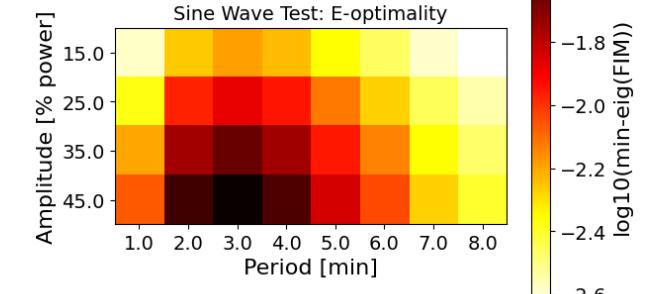
dowlinglab.github.io/pyomo-doe/notebooks/doe_exploratory_analysis.html



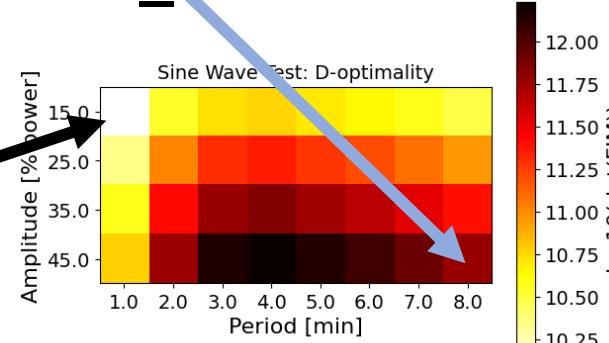
trAce



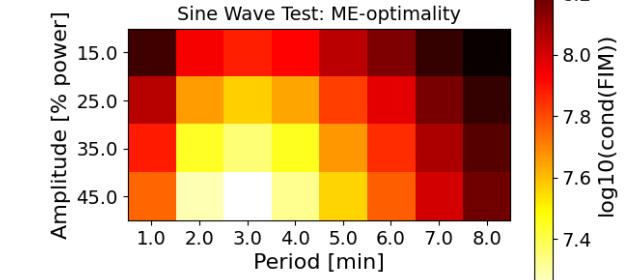
Min. Eigenvalue



Determinant



Condition Number



Notebooks at dowlinglab.github.io/pyomo-doe



Pyomo.DoE: Optimization

Our [earlier exploratory analysis](#) showed the sine wave experiment alone is rank deficient. What if instead of optimizing the sine wave parameters a and p , we directly optimize $u(t)$. In other words, we will formulate model-based design of experiments as an [optimal control problem](#).

Maximize a scalar-valued function $\psi(\cdot)$ of the Fisher information matrix \mathbf{M} :

$$\max_u \psi(\mathbf{M}(u) + \mathbf{M}_0)$$

$$\text{s. t. } C_p^H \frac{dT_H}{dt} = U_a(T_{amb} - T_H) + U_b(T_S - T_H) + \alpha P u(t)$$

$$C_p^S \frac{dT_S}{dt} = U_b(T_H - T_S)$$

$$0\% \leq u(t) \leq 100\%$$

$$T_H(t_0) = T_{amb}$$

$$T_S(t_0) = T_{amb}$$

TC Lab: A-Optimal Next Experiment

dowlinglab.github.io/pyomo-doe/notebooks/doe_optimize.html

$$\max_u \log \text{trace}(\mathbf{M}(u) + \mathbf{M}_0)$$

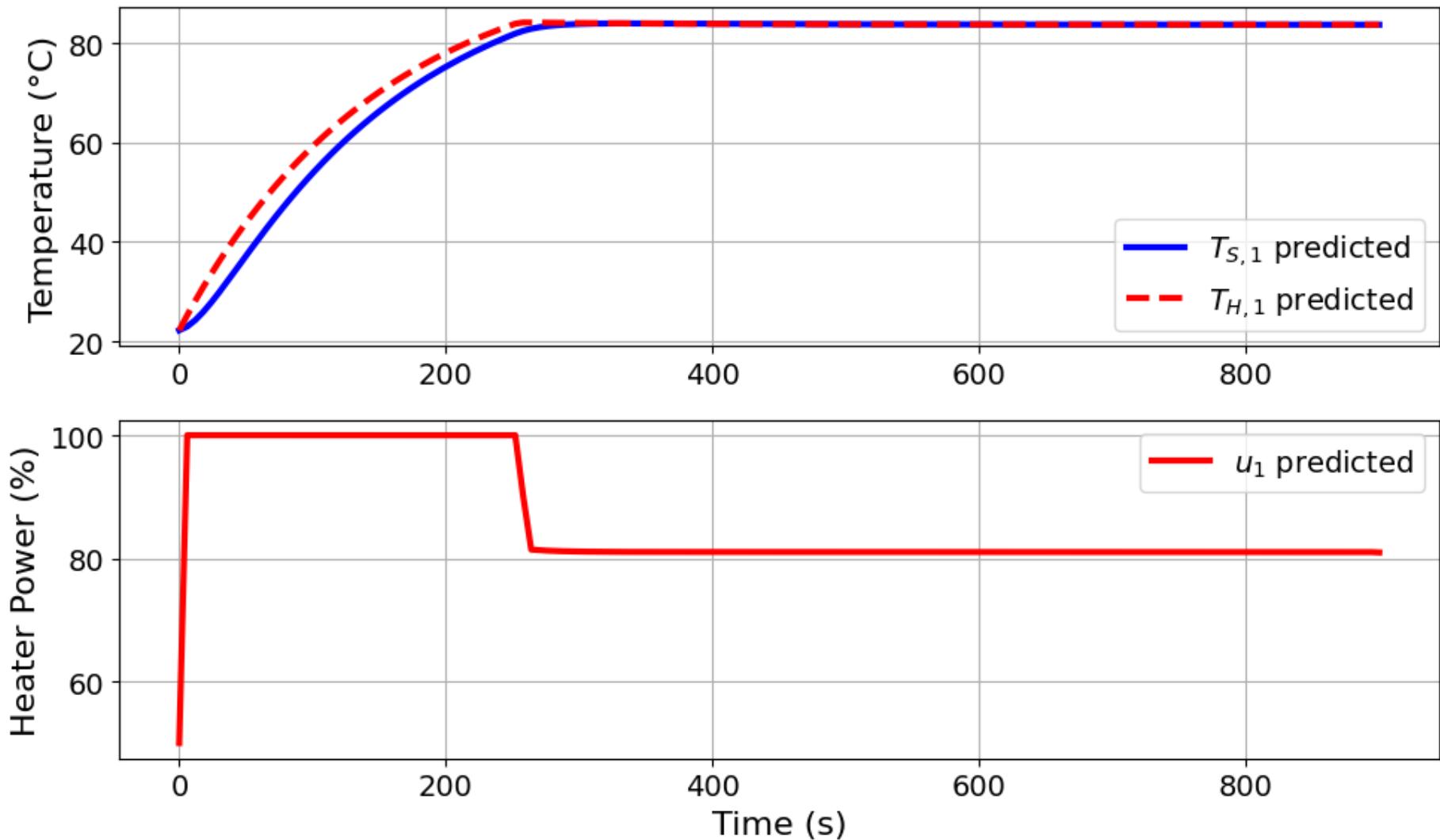
$$\text{s.t. } C_p^H \frac{dT_H}{dt} = \dots$$

$$C_p^S \frac{dT_S}{dt} = \dots$$

$$0\% \leq u(t) \leq 100\%$$

$$T_H(t_0) = T_{amb}$$

$$T_S(t_0) = T_{amb}$$



TC Lab: D-Optimal Next Experiment

dowlinglab.github.io/pyomo-doe/notebooks/doe_optimize.html

Best Sine Wave

$$\log_{10}(\det(M)) = 12$$

Optimal Control

$$\log_{10}(\det(M)) = 14.6$$

$$\max_u \log \det(\mathbf{M}(u) + \mathbf{M}_0)$$

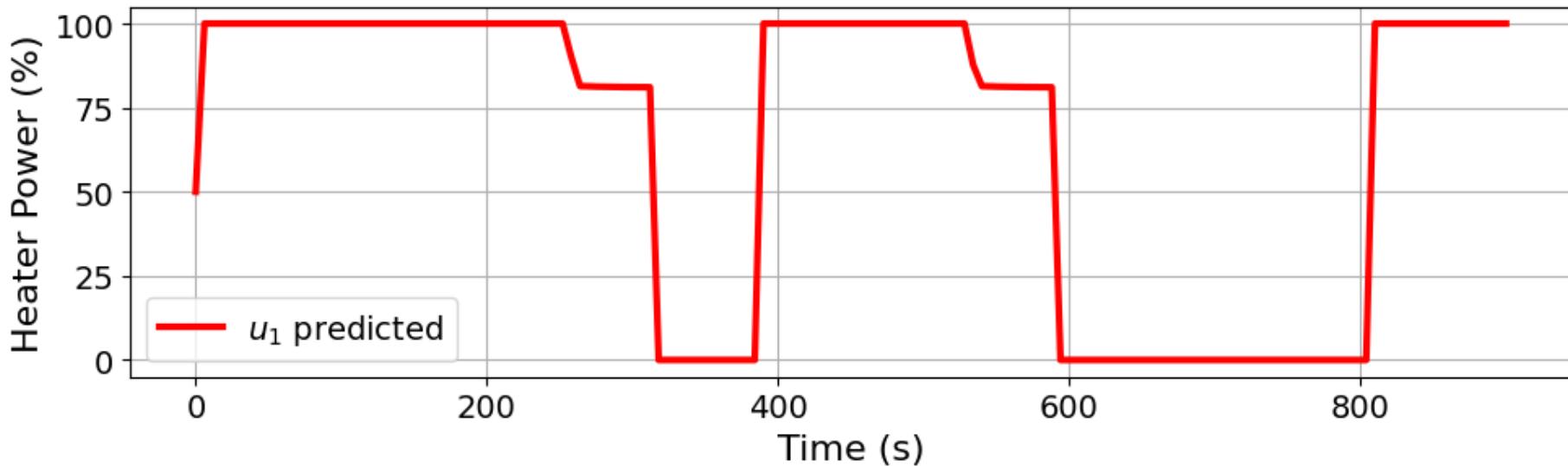
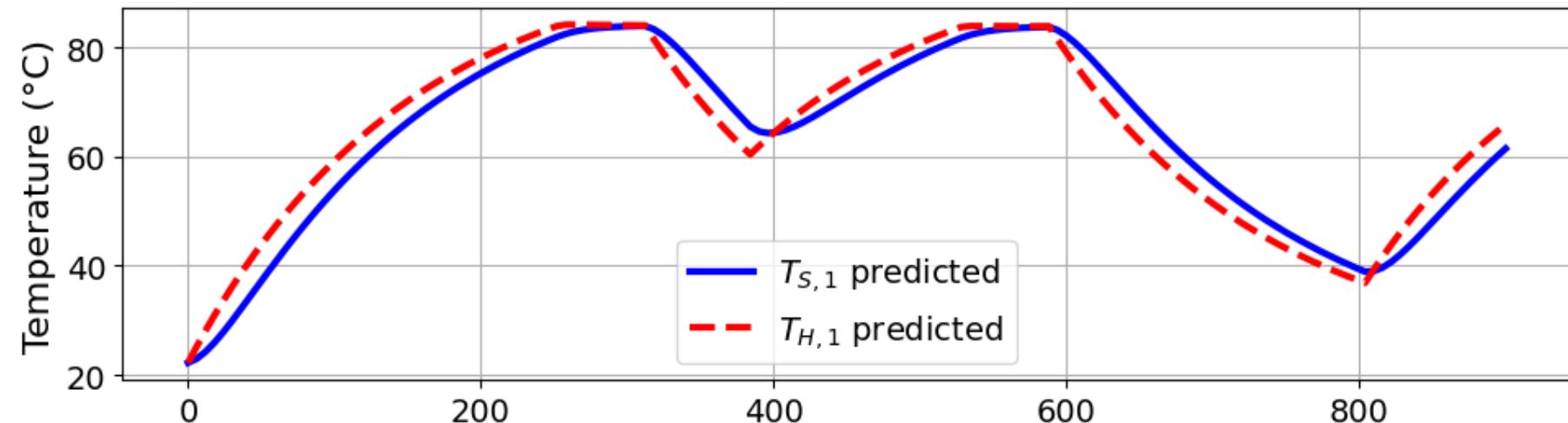
$$\text{s.t. } C_p^H \frac{dT_H}{dt} = \dots$$

$$C_p^S \frac{dT_S}{dt} = \dots$$

$$0\% \leq u(t) \leq 100\%$$

$$T_H(t_0) = T_{amb}$$

$$T_S(t_0) = T_{amb}$$



Getting Started with Pyomo.DoE

Documentation: https://pyomo.readthedocs.io/en/stable/contributed_packages/doe/doe.html

Community Detection for Pyomo models

- Pyomo.DoE
 - Methodology Overview
 - Pyomo.DoE Required Inputs
- Pyomo.DoE Solver Interface
- Pyomo.DoE Usage Example

GDPopt logic-based solver

Infeasible Irreducible System (IIS) Tool

Incidence Analysis

MindtPy Solver

MPC

Multistart Solver

Nonlinear Preprocessing Transformations

Parameter Estimation with `paramest`

PyNumero

PyROS Solver

Sensitivity Toolbox

Trust Region Framework Method Solver

MC++ Interface

z3 SMT Sat Solver Interface

[Read the Docs](#)

v: stable ▾

🏠 / Third-Party Contributions / Pyomo.DoE

[Edit on GitHub](#)

Pyomo.DoE

Pyomo.DoE (Pyomo Design of Experiments) is a Python library for model-based design of experiments using science-based models.

Pyomo.DoE was developed by **Jialu Wang** and **Alexander W. Dowling** at the University of Notre Dame as part of the [Carbon Capture Simulation for Industry Impact \(CCSI2\)](#) project, funded through the U.S. Department Of Energy Office of Fossil Energy.

If you use Pyomo.DoE, please cite:

[Wang and Dowling, 2022] Wang, Jialu, and Alexander W. Dowling. "Pyomo.DOE: An open-source package for model-based design of experiments in Python." AIChE Journal 68.12 (2022): e17813. <https://doi.org/10.1002/aic.17813>

Methodology Overview

Model-based Design of Experiments (MBDoE) is a technique to maximize the information gain of experiments by directly using science-based models with physically meaningful parameters. It is one key component in the model calibration and uncertainty quantification workflow shown below:

```
graph LR; A([Prior knowledge, preliminary data]) --> B[Model]; B --> C[Exploratory analysis]; C --> D[Parameter estimation]; D --> E[Uncertainty analysis]; E --> F([Model with quantified uncertainty]);
```

ParmEst and Pyomo.DoE Development Plans

Coming soon:

- Improved initialization
- Improved optimization performance
 - NLP decomposition
 - Grey-box objective calculations
- Improved modeling abstraction
 - Multiple experiments (e.g., planning batches)
 - Parameter uncertainty
- More applications, examples, and collaborations
- End-to-end uncertainty workflow (via interface with PyROS)



Acknowledgements

Pyomo.DoE development: We graciously acknowledge funding from the U.S. Department of Energy, Office of Fossil Energy and Carbon Management, through the Carbon Capture Program (CCSI²) and Office of Resource Sustainability (PrOMMiS).

This project was funded by the Department of Energy, National Energy Technology Laboratory an agency of the United States Government, through a support contract. Neither the United States Government nor any agency thereof, nor any of their employees, nor the support contractor, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.



Pyomo Team (SNL):

- Bethany Nicholson
- John Siirola
- Miranda Mundt

Contributors (ND):

- Dr. Jialu Wang
- Dr. Dan Laky
- Hailey Lynch



Thank you to Prof. Jeff Kantor (1954-2023) for the TCLab examples and so much more.