Optimizing Experiments with Pyomo.DoE

https://dowlinglab.github.io/pyomo-doe/



Daniel Laky[1,2], Shammah Lilonfe[1], Stephen Cini[1], Shuvashish Mondal[1], Shilpa Narasimhan[1], and Alexander Dowling[1]

[1]Chemical and Biomolecular Engineering,
University of Notre Dame, Notre Dame, IN

[2]Chemical Engineering,
Auburn University, Auburn, AL

American Control Conference

May 26, 2026



Other Contributors to ParmEst & Pyomo.DoE: ND: Jialu Wang, Hailey Lynch;
SNL: John Siirola, Bethany Nicholson, Miranda Mundt, Katherine Klise, Shawn Martin

UNIVERSITY OF NOTRE DAME · AUBURN UNIVERSITY · CCSI² Carbon Capture Simulation for Industry Impact · PROMMIS Process Optimization and Modeling for Minerals Sustainability

Agenda

Part 1 (12:15pm – 1:30pm): Welcome to Pyomo

- TCLab Mathematical Model
- Simulation and Dynamic Optimization in Pyomo
- Experiment Abstraction

Part 2 (1:45pm – 3pm): Parameter Estimation

- ParmEst Overview
- Uncertainty Quantification
- Multistart & Profile Likelihood
- Regularization

Part 3 (3:30pm – 5pm): Optimal Experiment Design

- Pyomo.DoE Overview
- OED Objectives
- Symbolic Derivatives
- Planning Multiple Experiments

UNIVERSITY OF NOTRE DAME · AUBURN UNIVERSITY · CCSI² Carbon Capture Simulation for Industry Impact · PROMMIS Process Optimization and Modeling for Minerals Sustainability

Power of Adaptive Sequential Optimal Experiments

Self-Driving Laboratories

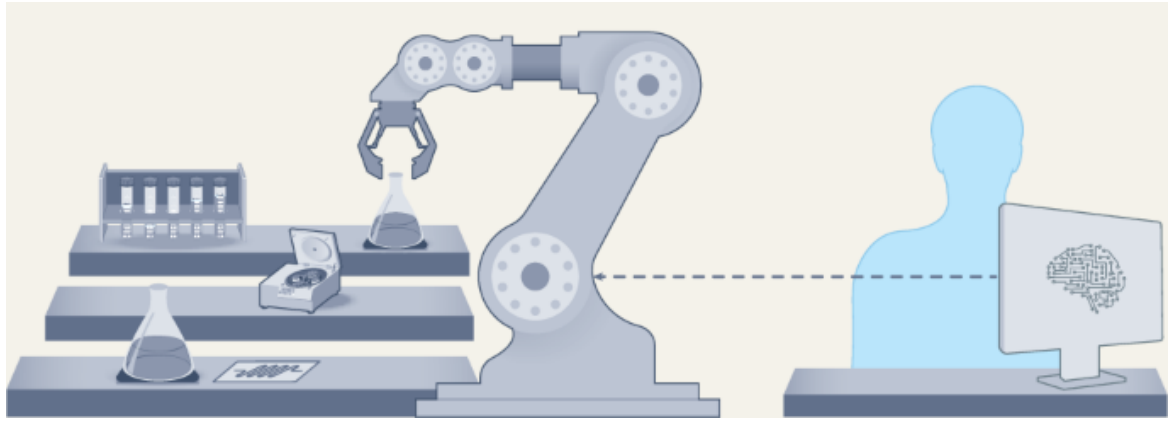
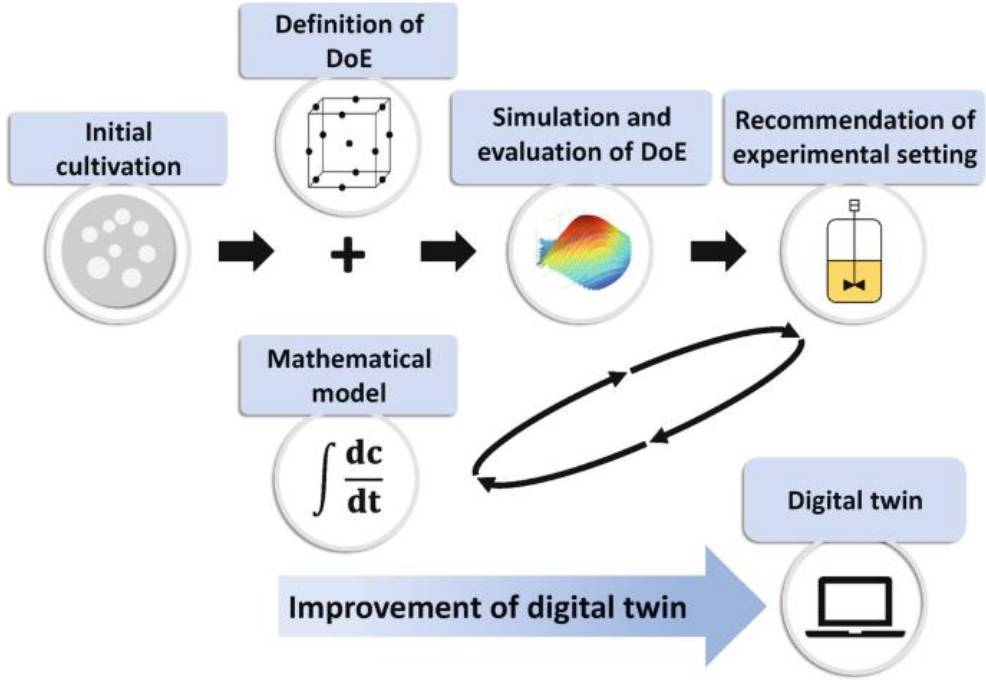


Figure: Abolhasani & Kumacheva (2023), *Nature Syn.*

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.*

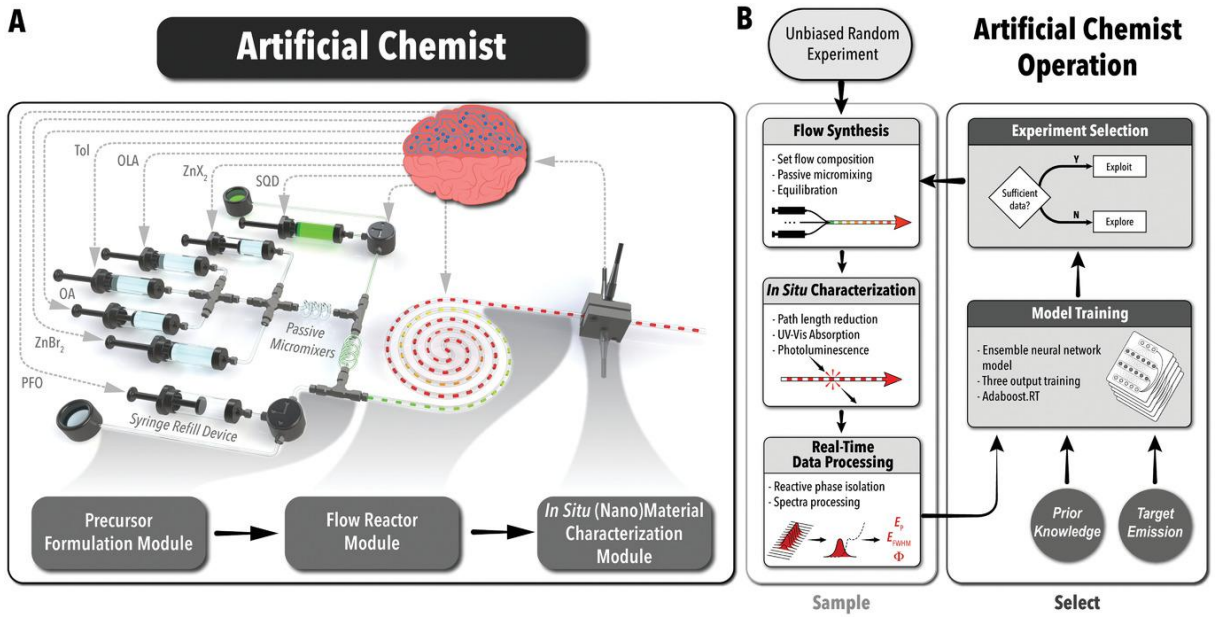
Automation + Model-Based Design of Experiments



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

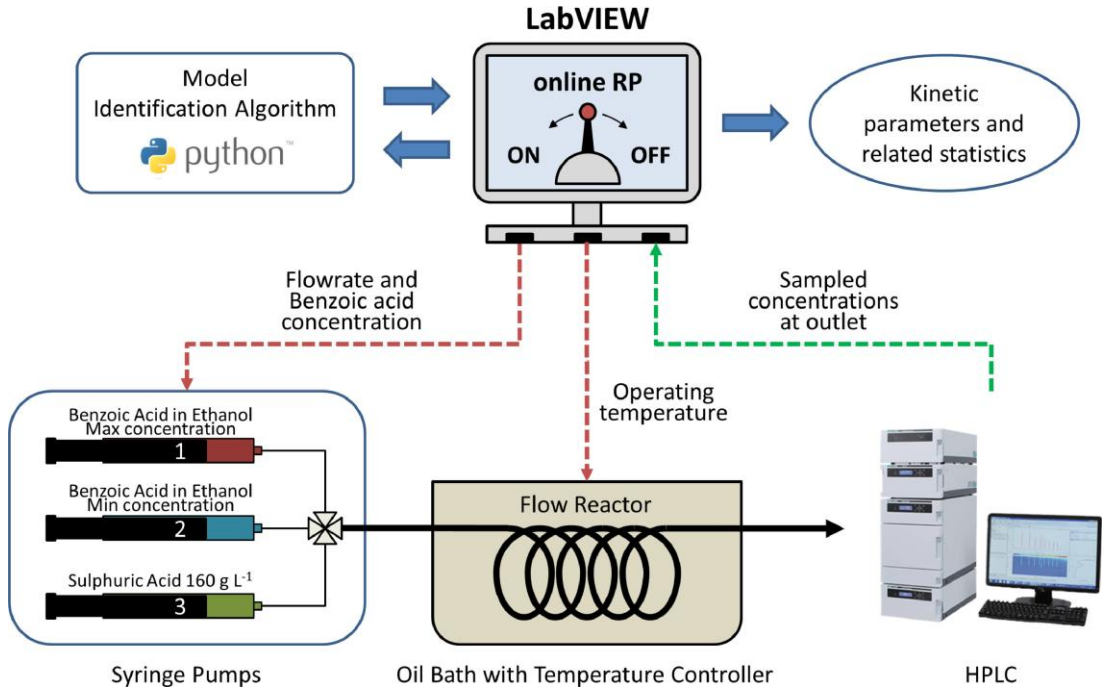
Many Recent Examples of Sequential Optimal Experiments

Quantum Dots
(Machine Learning)



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

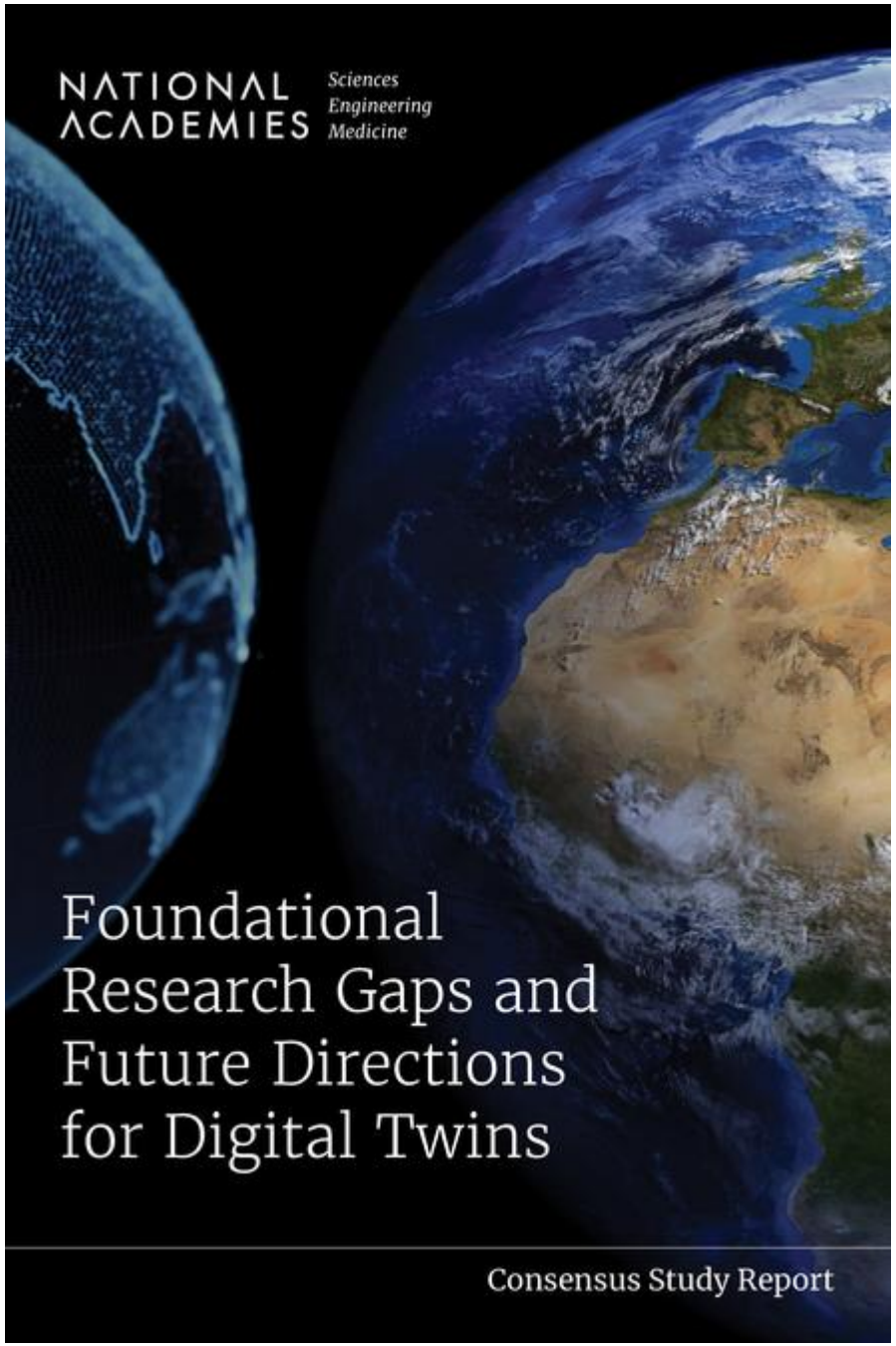
Reaction Engineering
(Science-based Models)



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

Prior Knowledge

Data

PYOMO

Model(s)

Parameter Estimation

Model Discrimination

PYOMO DoE

Sensitivity & Uncertainty Analysis

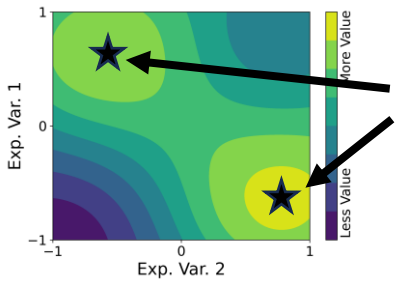
Do I have a good model?

No

Yes

PYOMO DoE

Optimize New Experiment(s)



Optimal new Experiment(s)

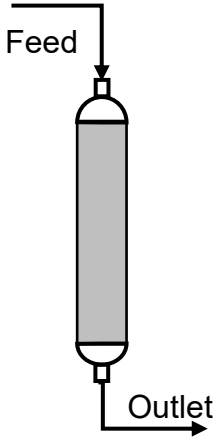
Goal: iteratively develop and validate predictive science-based models (**digital twins**)

Model with quantified uncertainty

UNIVERSITY OF NOTRE DAME · AUBURN UNIVERSITY · CCSI² Carbon Capture Simulation for Industry Impact · PROMMIS Process Optimization and Modeling for Minerals Sustainability

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

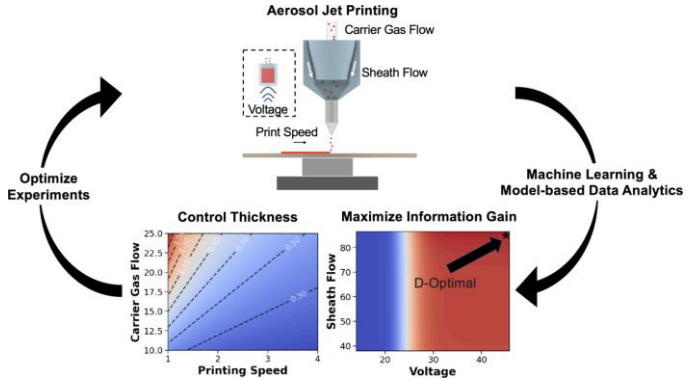
CO_2 Capture



Dr. Jialu Wang

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

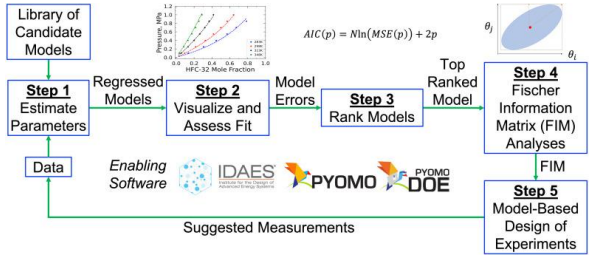
Additive Manufacturing of Thermoelectric Devices



Dr. Ke Wang

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

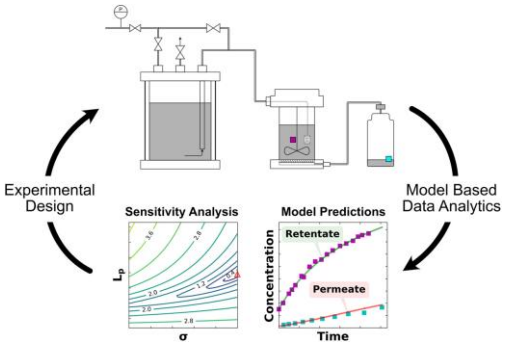
Thermodynamic Modeling (Refrigerants)



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.*

UNIVERSITY OF NOTRE DAME · AUBURN UNIVERSITY · CCSI² Carbon Capture Simulation for Industry Impact · PROMMIS Process Optimization and Modeling for Minerals Sustainability

Agenda

Part 1 (12:15pm – 1:30pm): Welcome to Pyomo

- TCLab Mathematical Model
- Simulation and Dynamic Optimization in Pyomo
- Experiment Abstraction

Part 2 (1:45pm – 3pm): Parameter Estimation

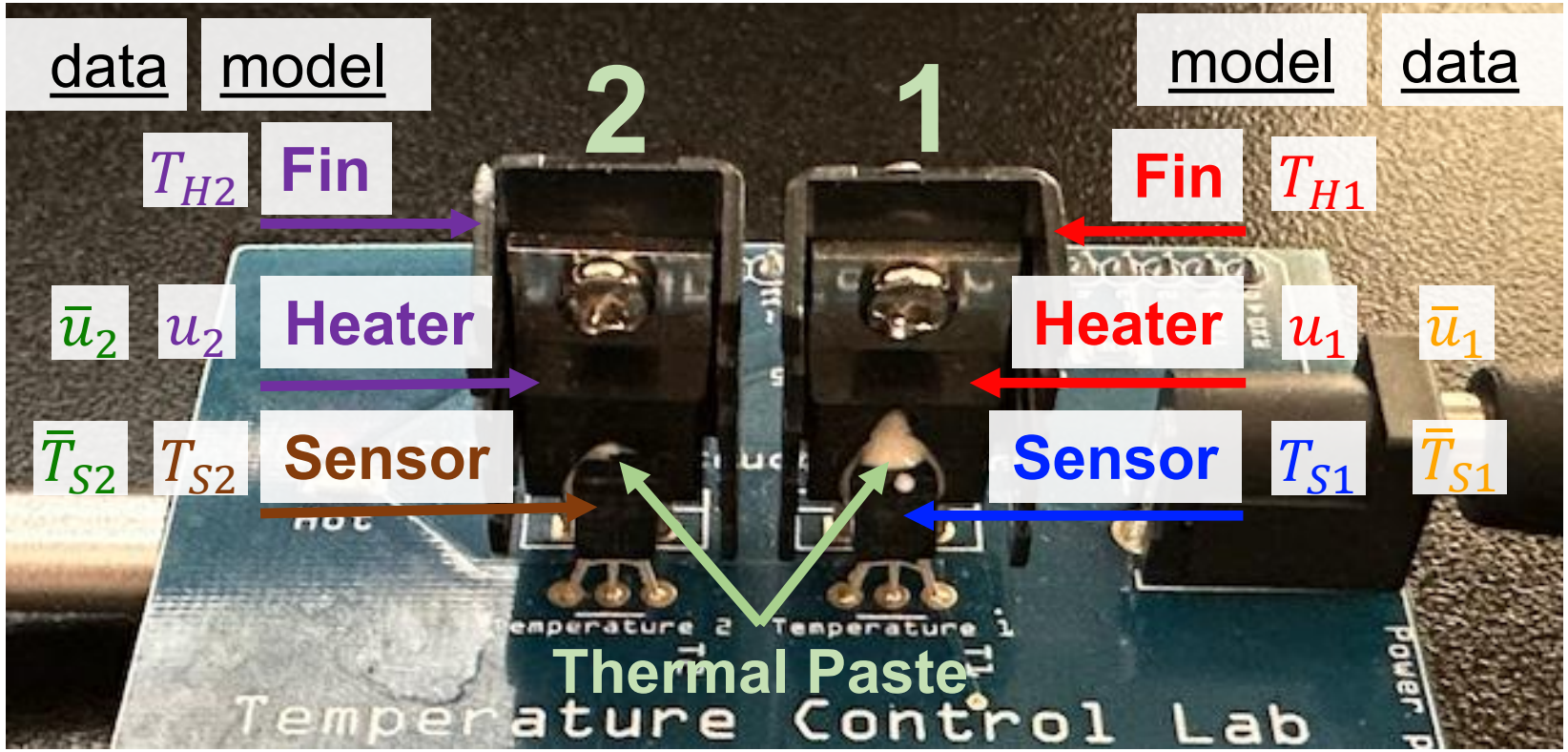
- ParmEst Overview
- Uncertainty Quantification
- Multistart & Profile Likelihood
- Regularization

Part 3 (3:30pm – 5pm): Optimal Experiment Design

- Pyomo.DoE Overview
- OED Objectives
- Symbolic Derivatives
- Planning Multiple Experiments

The Temperature Control Lab (TC Lab)

`https://ndcbe.github.io/controls`



Dowling et al. (2025). *Systems and Control Transactions*

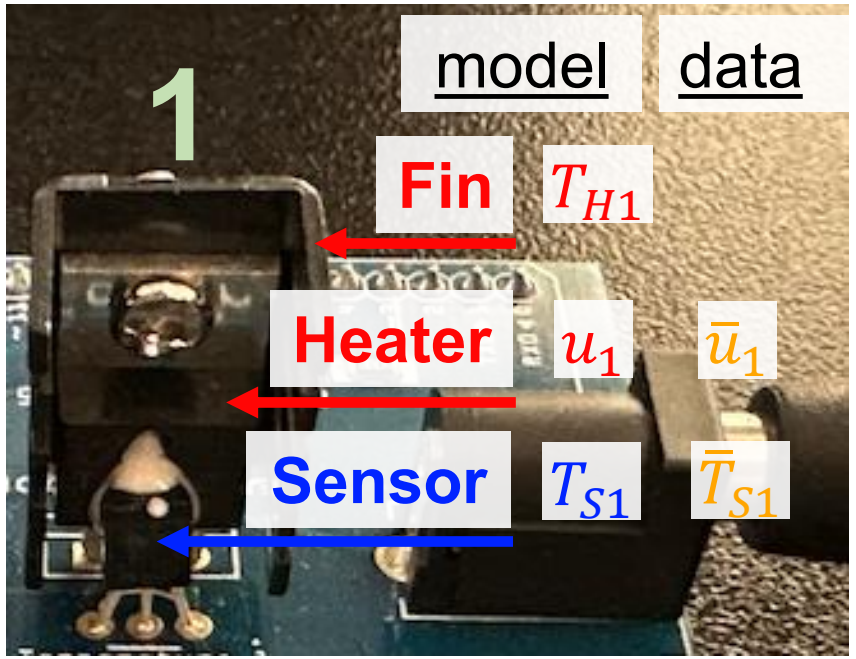
Thank you to Prof. Jeff Kantor (1954-2023) for the TCLab example and so much more.

TC Lab single heater model: two linear differential equations

Dowling et al. (2025). *Systems and Control Transactions*

Energy balance on the heater

$$\underbrace{C_{p,H}\frac{dT_{H,1}}{dt}}_{\text{Accumulation}} = \underbrace{\alpha P_1 u_1}_{\text{Input}} - \underbrace{\left(U_a\left(T_{H,1}-T_{amb}\right)+U_b\left(T_{H,1}-T_{S,1}\right)\right)}_{\text{Output}}$$



Energy balance on the sensor assuming no heat transfer to the ambient

$$\underbrace{C_{p,S}\frac{dT_{S,1}}{dt}}_{\text{Accumulation}} = \underbrace{U_b\left(T_{H,1}-T_{S,1}\right)}_{\text{Input}}$$

U_a = heater heat transfer coefficient in W/°C,
U_b = sensor heat transfer coefficient in W/°C,
$C_{p,H}$ = heat capacity of the heater in J/°C, $C_{p,S}$ = heat capacity of the sensor in J/°C,
$T_{H,1}$ = temperature of heater 1 in °C, $T_{S,1}$ = temperature of sensor 1 in °C,
T_{amb} = ambient temperature in °C, t = time in s,
u_1 = heater 1 power in %, P_1 = hardware-set maximum power limit in bit,
α = parameter for converting the unit of $P_1 u_1$ into W,

TC Lab single heater model: two linear differential equations

Model from energy balance

$$C_{p,H}\frac{dT_{H,1}}{dt} = \alpha P_1 u_1 - \left(U_a\left(T_{H,1} - T_{\text{amb}}\right) + U_b\left(T_{H,1} - T_{S,1}\right)\right), \qquad C_{p,S}\frac{dT_{S,1}}{dt} = U_b\left(T_{H,1} - T_{S,1}\right)$$

Rearranged TC Lab model

$$\frac{dT_{H,1}}{dt} = \frac{U_a\left(T_{\text{amb}} - T_{H,1}\right) + U_b\left(T_{S,1} - T_{H,1}\right) + \alpha P_1 u_1}{C_{p,H}}, \qquad \frac{dT_{S,1}}{dt} = \frac{U_b\left(T_{H,1} - T_{S,1}\right)}{C_{p,S}}$$

Unknown parameters

$$\boldsymbol{\theta} = \left(U_a,\ U_b,\ \left(C_{p,H}\right)^{-1},\ \left(C_{p,S}\right)^{-1}\right)^{\mathrm{T}}$$

Measured variables

$u_1,\ T_{S,1}$

Control variables

u_1

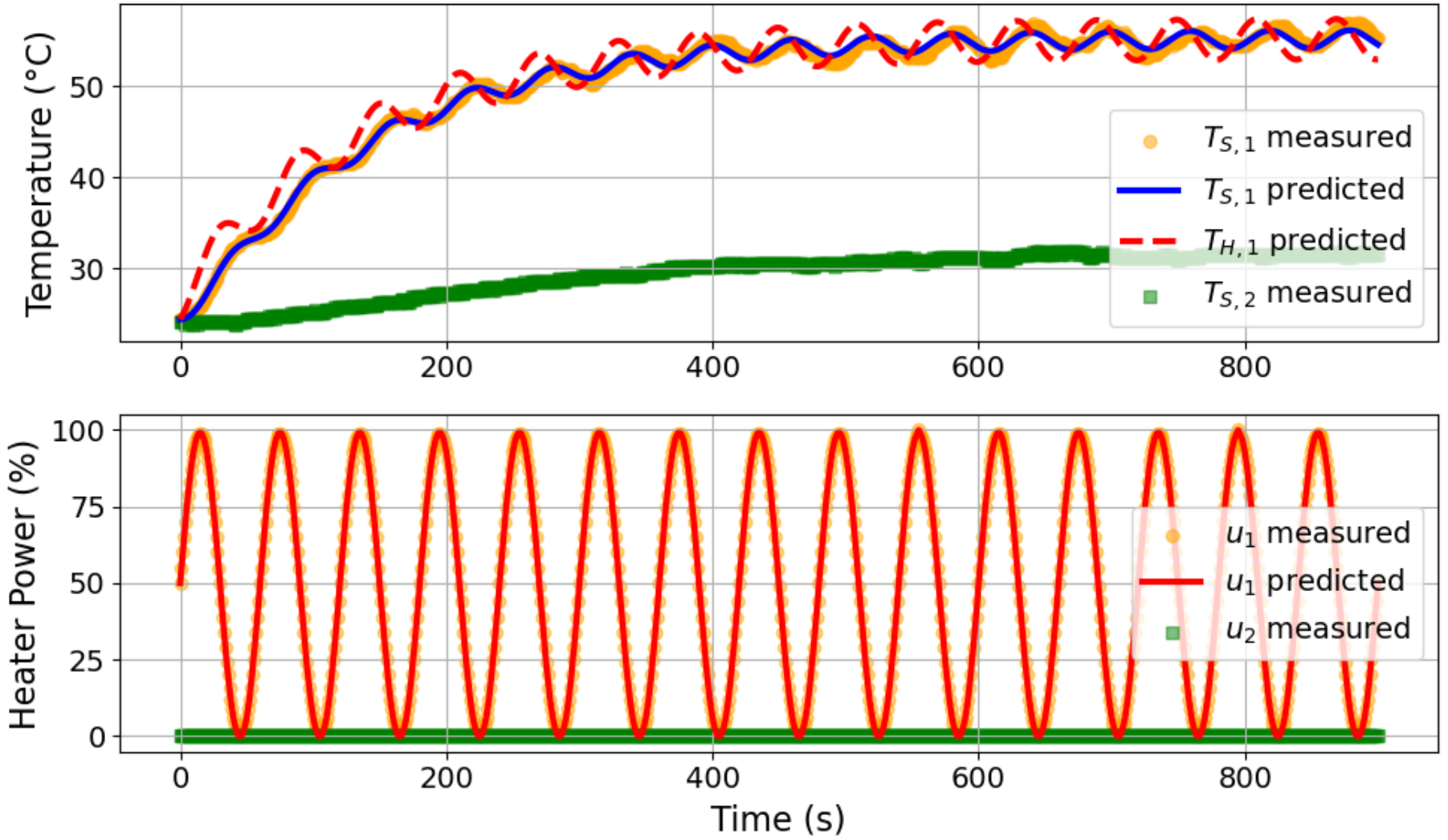
Data

$T_{\text{amb}},\ \alpha,\ P_1,\ u_1,\ T_{S,1}$

U_a = heater heat transfer coefficient in W/°C,
U_b = sensor heat transfer coefficient in W/°C,
$C_{p,H}$ = heat capacity of the heater in J/°C, $C_{p,S}$ = heat capacity of the sensor in J/°C,
$T_{H,1}$ = temperature of heater 1 in °C, $T_{S,1}$ = temperature of sensor 1 in °C,
T_{amb} = ambient temperature in °C, t = time in s,
u_1 = heater 1 power in %, P_1 = hardware-set maximum power limit in bit,
α = parameter for converting the unit of $P_1 u_1$ into W,

TC Lab: Data and Parameter Estimation

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



Notebooks at `dowlinglab.github.io/pyomo-doe`

CONTENTS

TCLab Mathematical Model

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

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
```

Notebooks at `dowlinglab.github.io/pyomo-doe`

Modeling with Pyomo

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

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

CONTENTS


```
# 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_he
    import colab_helper

    colab_helper.install_idaes()
    colab_helper.install_ipopt()

# Set plotting defaults
import matplotlib.pyplot as plt
```

TC Lab: Dynamic Optimization in Pyomo

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

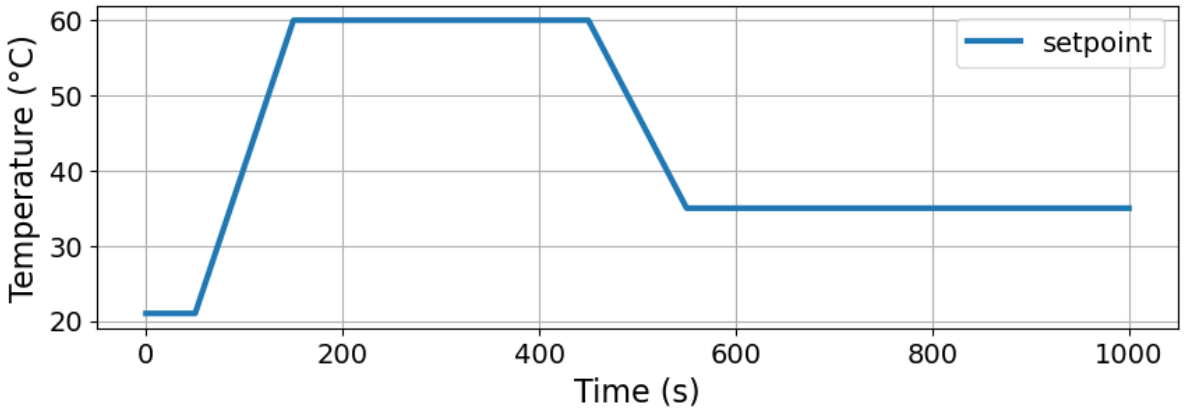
$$\min_{u(t)} \int_{t_0}^{t_f} \| SP(t) - T_H(t) \|^2 \, dt$$

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

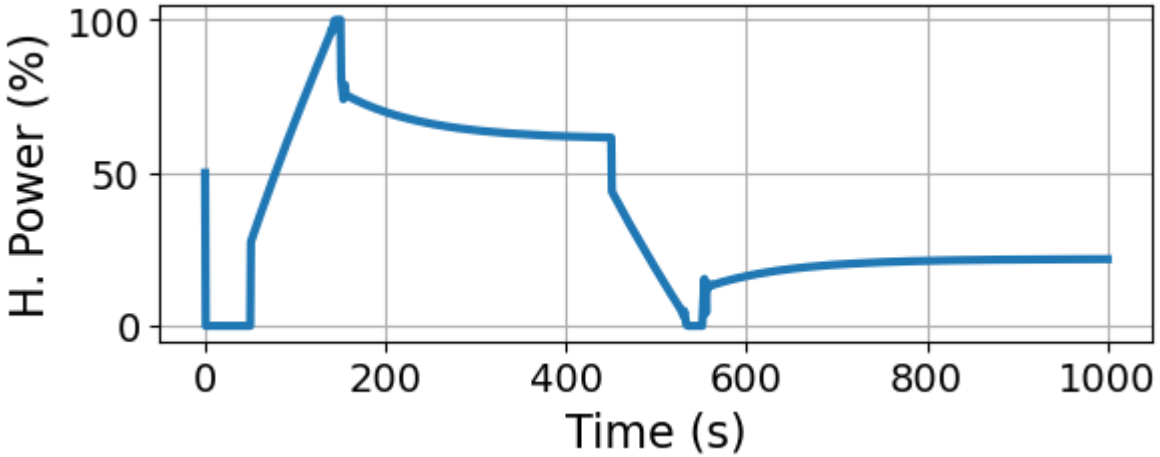
$$C_p^S \frac{dT_S}{dt} = U_b(T_H - T_S)$$

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

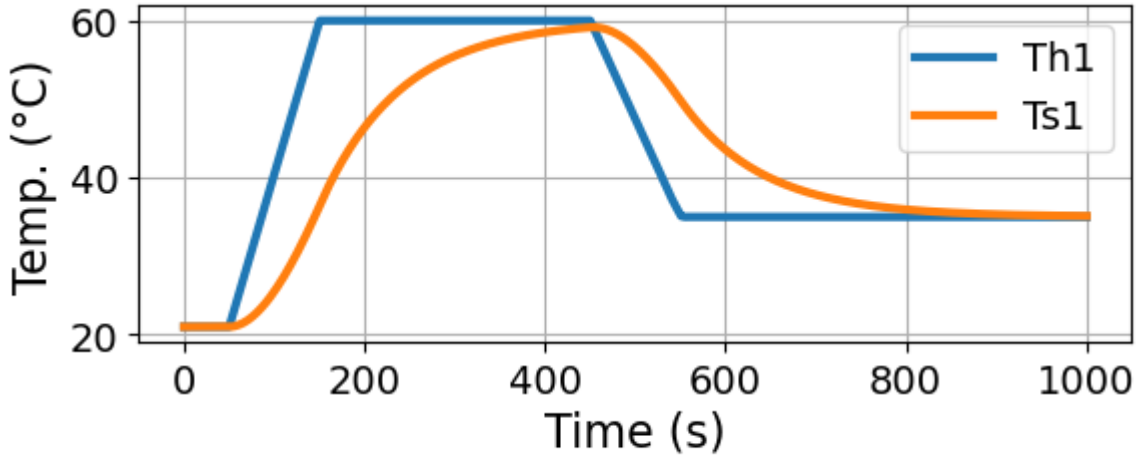
$$T_S(t_0) = T_{amb}$$



Optimal control profile



T_{h1} matches the setpoint using optimal control



TCLab & Process Controls UG Course at Notre Dame

Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Teaching Digital Twins in Process Control Using the Temperature Control Lab

Alexander W. Dowling*, Molly Dougher, Madelynn J. Watson, Hailey G. Lynch, Zhicheng Lu, and Daniel J. Laky

University of Notre Dame, Department of Chemical and Biomolecular Engineering, Notre Dame, IN 46556, United States
* Corresponding Author: adowling@nd.edu.

ABSTRACT

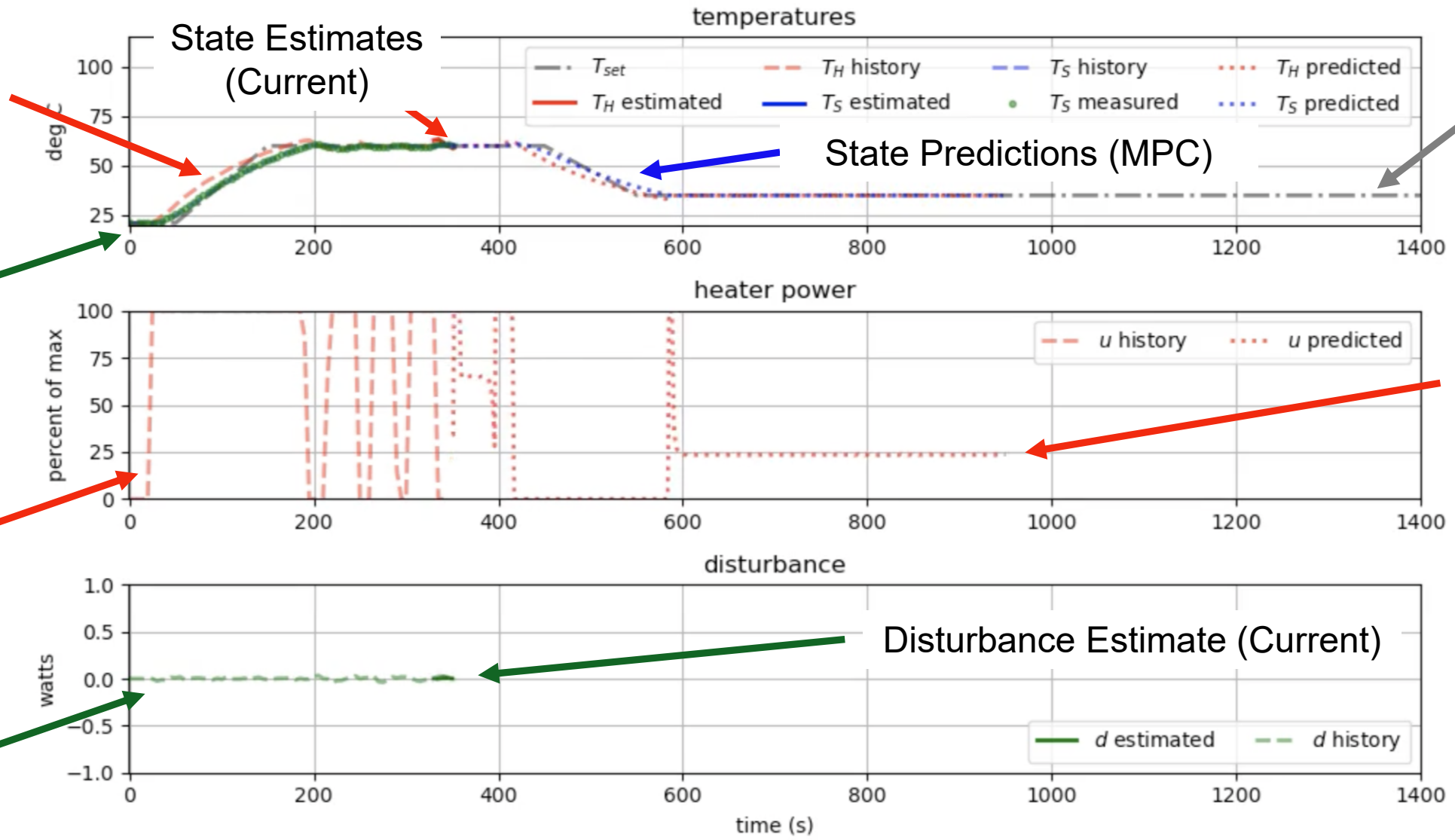
Process control can be one of the most exciting and engaging chemical engineering undergraduate courses! This paper describes our experience transforming *Chemical Process Control* into *Data Analytics, Optimization, and Control* at the University of Notre Dame (second semester required course in the junior year). Our modern course is built around six hands-on experiments in which students practice data-centric modeling and analysis using the Arduino-based Temperature Control Lab (TCLab) hardware. We argue that state-space dynamic modeling and optimization are more critical for educating modern chemical engineers than topics such as frequency domain analysis and controller synthesis emphasized in many classical undergraduate control courses. All the course material is available online at https://ndcbe.github.io/controls.

Keywords: Process Control, Education, Dynamic Modelling, System Identification, Model Predictive Control, Pyomo, Process Monitoring, Process Operations, Industry 4.0

Course Materials

Paper

Teaching Digital Twins (MPC, State Estimation)



State Estimates (History)

State Estimates (Current)

Set Point

State Predictions (MPC)

Sensor Data

Control Signal (MPC, Future Plan)

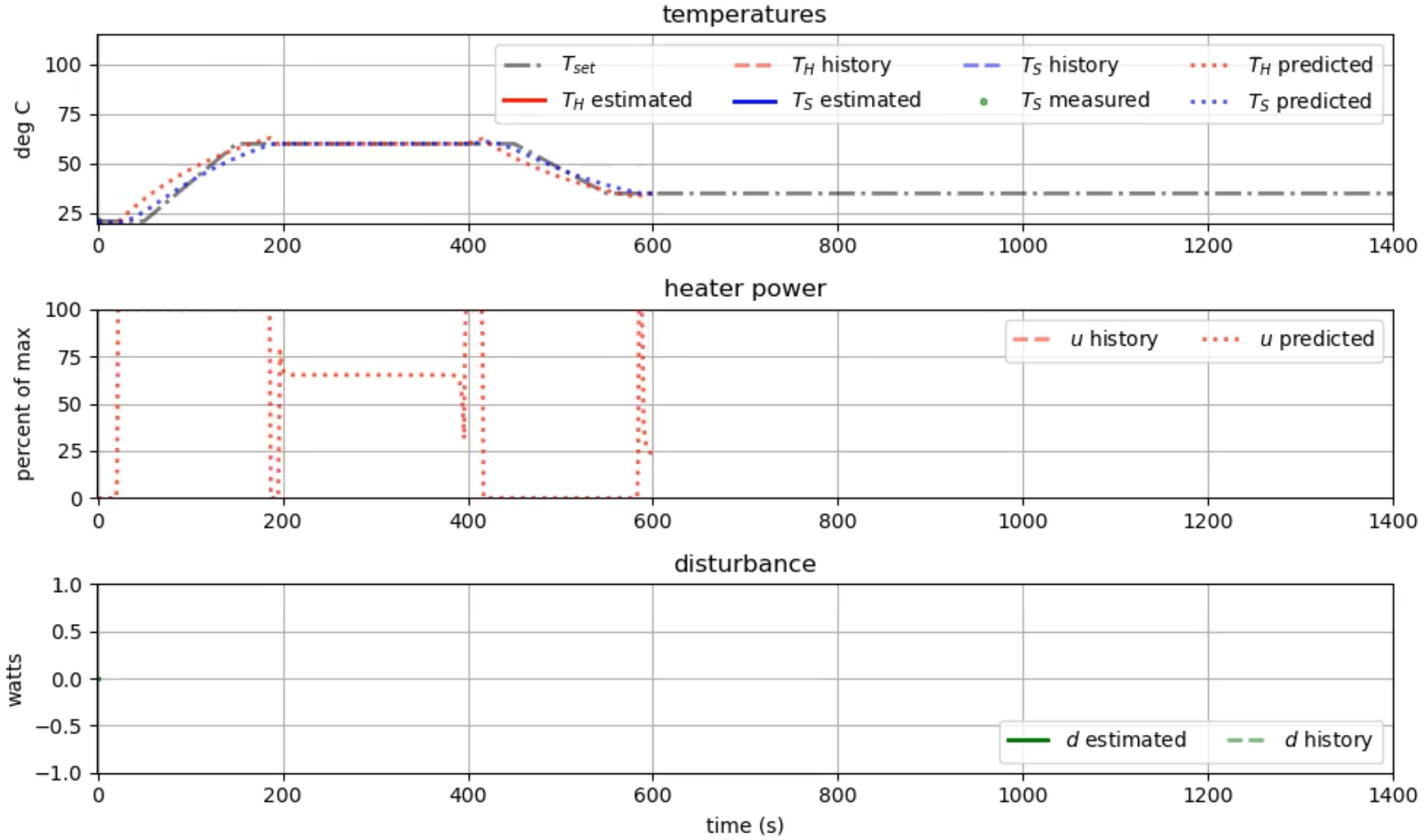
Control Signal (History)

Disturbance Estimate (Current)

Disturbance Estimate (History)

MPC = model predictive control

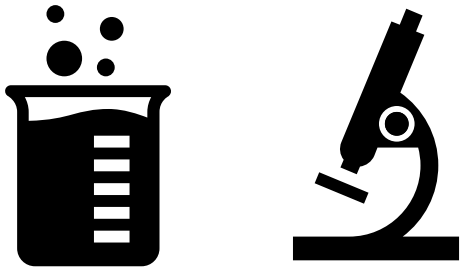
Teaching Digital Twins (MPC, State Estimation)



"Experiment" Abstraction Streamlines Closed-Loop Experiment Design

Experiment Inputs (Conditions)

Experiment (Physical)



Experiment Outputs (Data) + Measurement Uncertainty

Dr. Bethany Nicholson

Dr. John Siirola

Dr. Shawn Martin

Katherine Klise

ϕ : experimental design vector

Experiment (Digital)

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

$\hat{y}$: measurement response variable prediction

UNIVERSITY OF NOTRE DAME · AUBURN UNIVERSITY · CCSI² Carbon Capture Simulation for Industry Impact · PROMMIS Process Optimization and Modeling for Minerals Sustainability

"Experiment" Abstraction Streamlines Closed-Loop Experiment Design

PYOMO - parmest

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

PYOMO DoE

θ : model parameters

Experiment (Digital)

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

ϕ : experimental design vector

$$\max_{\phi_l \le \phi \le \phi_u} \Psi(\mathbf{FIM}(\theta, \phi))$$

Dr. Shawn Martin

Katherine Klise

PROMMIS

Process Optimization and Modeling for Minerals Sustainability

UNIVERSITY OF NOTRE DAME · AUBURN UNIVERSITY · CCSI² Carbon Capture Simulation for Industry Impact

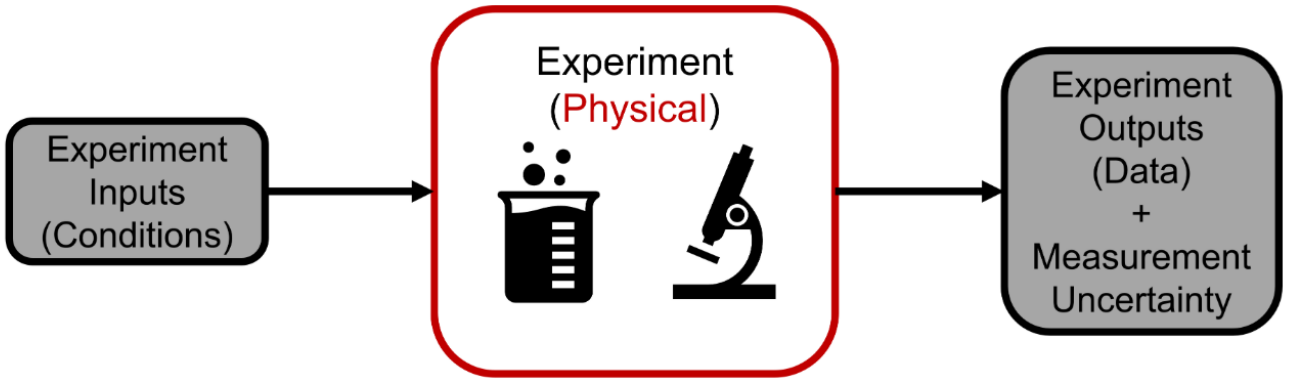
Notebooks at `dowlinglab.github.io/pyomo-doe`

CONTENTS

Experiment Abstraction

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

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:



Agenda

Part 1 (12:15pm – 1:30pm): Welcome to Pyomo

- TCLab Mathematical Model
- Simulation and Dynamic Optimization in Pyomo
- Experiment Abstraction

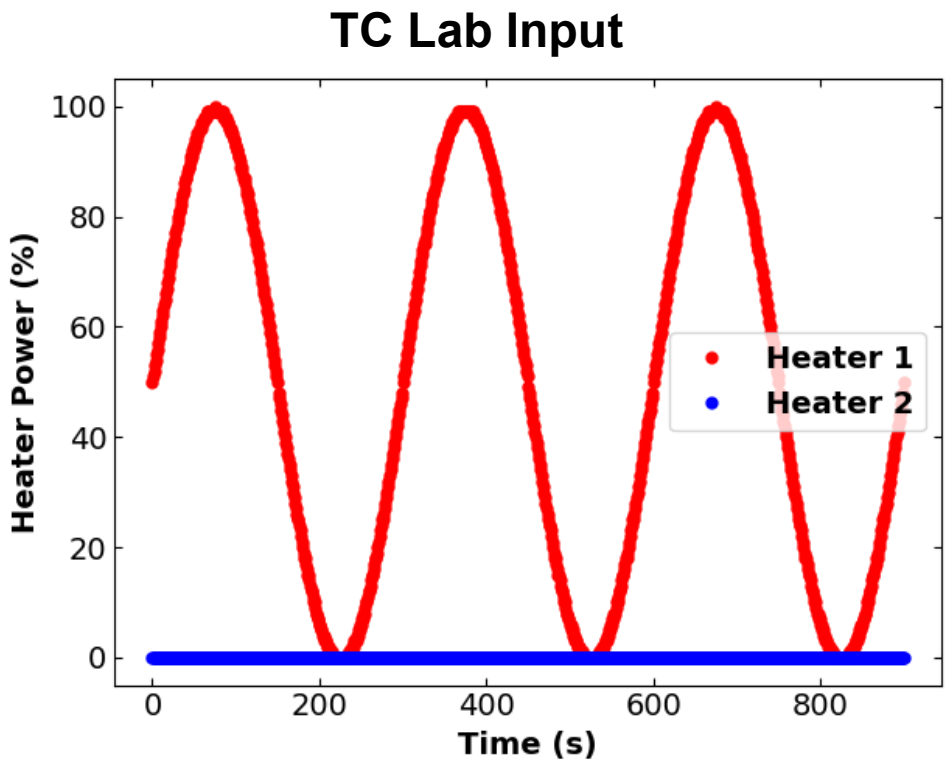
Part 2 (1:45pm – 3pm): Parameter Estimation

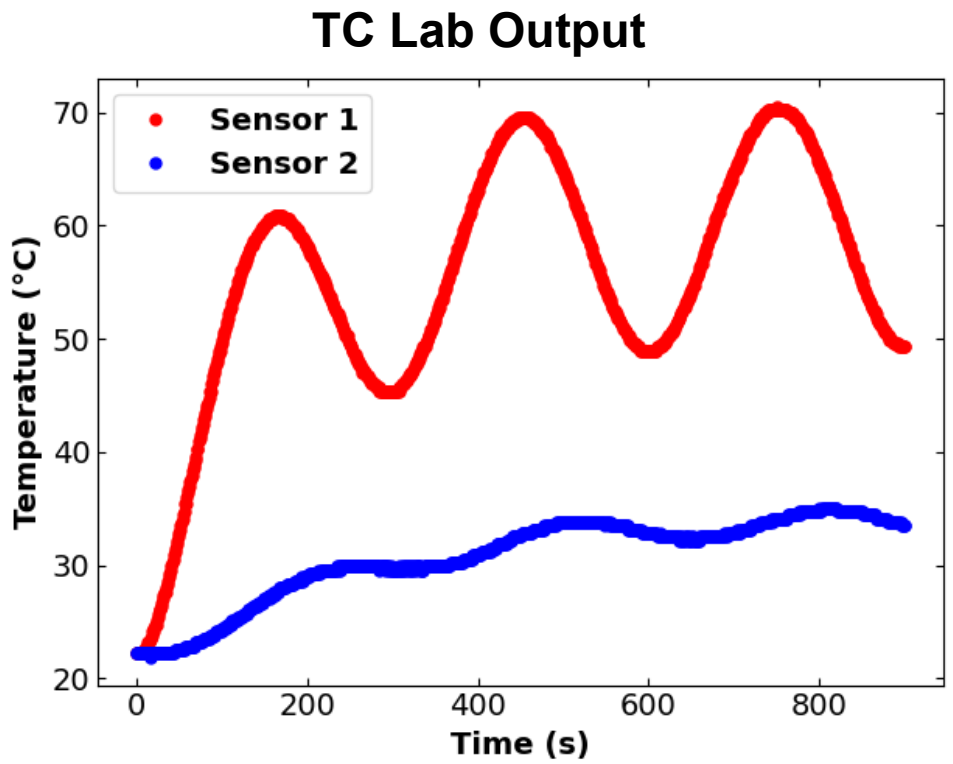
- ParmEst Overview
- Uncertainty Quantification
- Multistart & Profile Likelihood
- Regularization

Part 3 (3:30pm – 5pm): Optimal Experiment Design

- Pyomo.DoE Overview
- OED Objectives
- Symbolic Derivatives
- Planning Multiple Experiments

A sine test experiment was used to get the TC Lab data





TC Lab model:

$$\frac{dT_{H,1}}{dt} = \frac{U_a\left(T_{\text{amb}} - T_{H,1}\right) + U_b\left(T_{S,1} - T_{H,1}\right) + \alpha P_1 u_1}{C_{p,H}}, \qquad \frac{dT_{S,1}}{dt} = \frac{U_b\left(T_{H,1} - T_{S,1}\right)}{C_{p,S}}$$

Problem statement: Can the four unknown parameters $\left(U_a,\ U_b,\ \left(C_{p,H}\right)^{-1},\ \left(C_{p,S}\right)^{-1}\right)$ be reliably estimated from the data?

Estimating the four parameters follows an iterative approach

This talk and the following: Parameter estimation

Prior Knowledge

(Preliminary) Data

Model(s)

Pyomo Model

$y = f(\varphi; \theta) + \epsilon$

ParmEst

Parameter Estimation

$$\min_{\theta \in \Theta} \sum_{i=1}^{n} (y_i - f(\varphi_i; \theta))^{\mathrm{T}} \Sigma_y^{-1} (y_i - f(\varphi_i; \theta))$$

Parameter Estimate

$\hat{\theta}$

ParmEst

Sensitivity & Uncertainty Analysis

$$V_\theta = \left(\sum_{i=1}^{n} Q_i^{\mathrm{T}} \Sigma_y^{-1} Q_i \right)^{-1}_{\theta=\hat{\theta}}$$

Uncertainty Acceptable?

No

(Science-Based) Design of Experiments

$\max_{\varphi} \psi[\mathbf{FIM}(\theta, \varphi)]$

New data

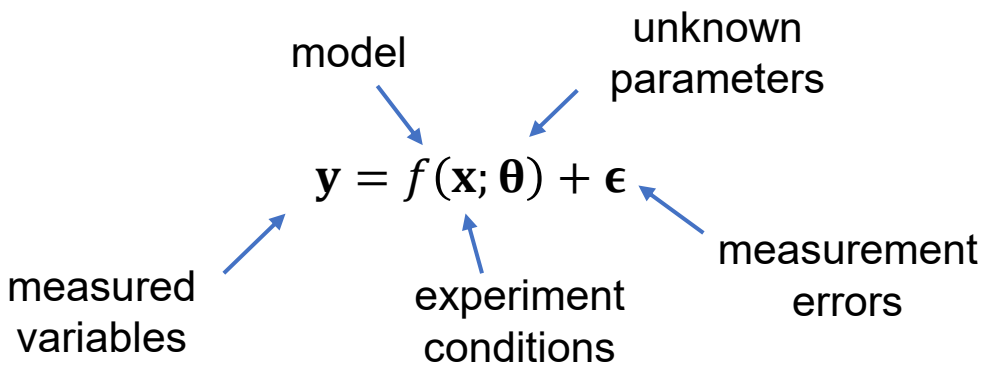
Yes

Model with Quantified Uncertainty

Pyomo is an open-source Python package for optimization modeling, **ParmEst** is a Pyomo tool for parameter estimation, **Pyomo.DoE** is a Pyomo tool for design of experiments, θ = unknown parameters, y = observation of measured variables, $\mathbf{FIM}$ = Fisher information matrix, V_θ = parameter covariance matrix, φ = design vector, ψ = scalar measure (e.g., determinant), Q = Jacobian matrix, ϵ = measurement error, Σ_y = measurement error covariance matrix, and n = number of experiments

Parameter estimation background

1. Express the data in the mathematical form:



2. Define measurement error model:

e.g., assume $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\mathbf{y})$

multivariate normal distribution

measurement covariance

3. Construct the (log) likelihood function

$$l(\boldsymbol{\theta}; y_1 \cdots y_n) = -\frac{n}{2}\left(\log(2\pi) + \log\left|\det \boldsymbol{\Sigma}_\mathbf{y}\right|\right) - \frac{1}{2}\sum_{i=1}^{n}\left(\mathbf{y}_i - \boldsymbol{f}(\mathbf{x}_i;\boldsymbol{\theta})\right)^{\mathrm{T}}\boldsymbol{\Sigma}_\mathbf{y}^{-1}\left(\mathbf{y}_i - \boldsymbol{f}(\mathbf{x}_i;\boldsymbol{\theta})\right)$$

constant c when $\boldsymbol{\Sigma}_\mathbf{y}$ is known

WSSE

4. Compute the maximum likelihood estimator

maximum likelihood estimate

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \mathrm{WSSE}$$

$\mathbf{y}$ = measured variables, $\boldsymbol{\epsilon}$ = measurement errors, l = log-likelihood function, n = number of experiments, $\boldsymbol{\theta}$ = parameters, WSSE = weighted sum of squared errors, $\mathbf{x}$ = experimental conditions, $\boldsymbol{\Sigma}_\mathbf{y}$ = measurement error covariance matrix

Notebooks at `dowlinglab.github.io/pyomo-doe`

Getting Started: Parameter Estimation Basics

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

The goal of parameter estimation is to estimate values for a vector, $\boldsymbol{\theta}$, from experimental data (observations) to use in the functional form:

$$\boldsymbol{y}_i = \boldsymbol{f}\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) + \boldsymbol{\varepsilon}_i \quad \forall\, i \in \{1, \ldots, n\} \tag{1}$$

where $\boldsymbol{y}_i \in \mathbb{R}^m$ are observations of the measured or output variables, $\boldsymbol{f}(\cdot)$ is the model function, $\boldsymbol{x}_i \in \mathbb{R}^q$ are the decision or input variables, $\boldsymbol{\theta} \in \mathbb{R}^p$ are the model parameters, $\boldsymbol{\varepsilon}_i \in \mathbb{R}^m$ are the measurement errors, and n is the number of experiments.

The following least squares objective can be used to estimate parameter values assuming that the measurement errors follow a Gaussian distribution:

CONTENTS

Covariance matrix quantifies parameter uncertainty

5. Define the Fisher information matrix:

$$\mathbf{FIM}(\boldsymbol{\theta}) = -\mathbb{E}_{\boldsymbol{\theta}}\left[\frac{\partial^2 l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right]$$

Information about parameters $\boldsymbol{\theta}$ contained in measured variable $\mathbf{y}$ from experiment design $\mathbf{x}$

6. Define the parameter covariance matrix:

For a consistent and asymptotically normal estimator: $\mathrm{Cov}(\hat{\boldsymbol{\theta}}) = \mathbf{FIM}^{-1}$

7. Compute the parameter covariance matrix:

With $l(\boldsymbol{\theta}) = -\mathrm{WSSE} + c$: $\quad \mathbf{V}_{\boldsymbol{\theta}} \approx -\left(\frac{\partial^2 l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right)^{-1}_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$

Method 1: *reduced_hessian*

$$\mathbf{V}_{\boldsymbol{\theta}} = \left(\frac{\partial^2 \mathrm{WSSE}}{\partial \boldsymbol{\theta}^2}\right)^{-1}_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$$

Computes reduced Hessian via PyNumero

Method 2: *finite_difference*

$$\frac{\partial^2 \mathrm{WSSE}}{\partial \boldsymbol{\theta}^2} \approx \sum_{i=1}^{n}\left(\frac{\partial f(\mathbf{x}_i;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^{\mathrm{T}} {\boldsymbol{\Sigma}_{\mathbf{y}}}^{-1}\left(\frac{\partial f(\mathbf{x}_i;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)$$

$$\mathbf{V}_{\boldsymbol{\theta}} = \left(\sum_{i=1}^{n} \mathbf{Q}_i^{\mathrm{T}} {\boldsymbol{\Sigma}_{\mathbf{y}}}^{-1} \mathbf{Q}_i\right)^{-1}_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$$

Uses central finite difference to compute $\mathbf{Q}_i$

Method 3: *automatic_differentiation_kaug*

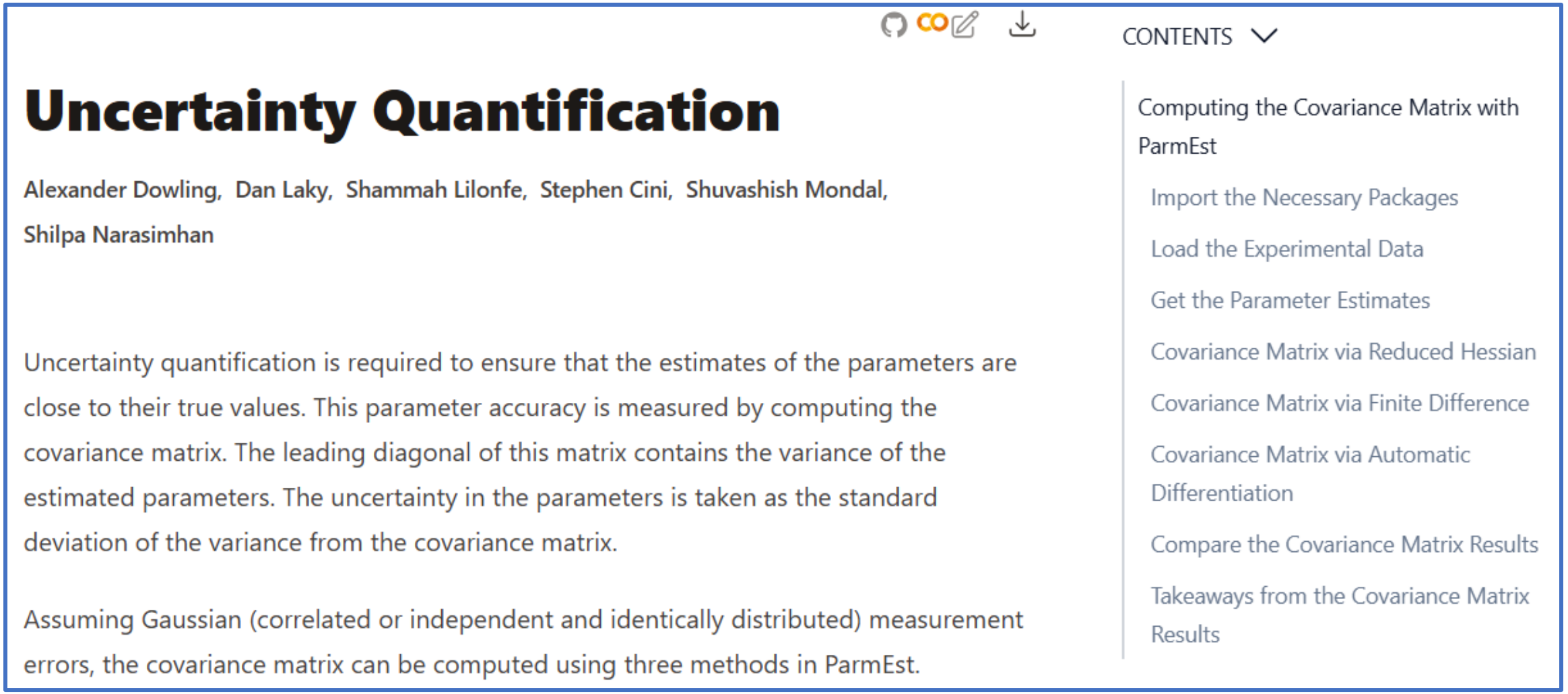
$$\mathbf{Q}_{\mathrm{kaug},\,i} = \frac{\partial f(\mathbf{x}_i;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \frac{\partial f(\mathbf{x}_i;\boldsymbol{\theta})}{\partial \mathbf{x}_i}\frac{\partial \mathbf{x}_i}{\partial \boldsymbol{\theta}}$$

$$\mathbf{V}_{\boldsymbol{\theta}} = \left(\sum_{i=1}^{n} \mathbf{Q}_{\mathrm{kaug},\,i}^{\mathrm{T}} {\boldsymbol{\Sigma}_{\mathbf{y}}}^{-1} \mathbf{Q}_{\mathrm{kaug},\,i}\right)^{-1}_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$$

Uses implicit differentiation and KKT conditions to compute $\mathbf{Q}_{\mathrm{kaug},\,i}$

$\mathbf{y}$ = measured variables, l = log-likelihood function, n = number of experiments, WSSE = weighted sum of squared errors, $\boldsymbol{\Sigma}_{\mathbf{y}}$ = measurement error covariance matrix, $\mathbf{Q}$ = Jacobian, $\boldsymbol{\theta}$ = parameters, $\mathbf{x}$ = experimental conditions, $\mathbf{V}_{\boldsymbol{\theta}}$ = parameter covariance matrix, $\mathbf{FIM}$ = Fisher information matrix

Notebooks at `dowlinglab.github.io/pyomo-doe`



CONTENTS

Uncertainty Quantification

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

Uncertainty quantification is required to ensure that the estimates of the parameters are close to their true values. This parameter accuracy is measured by computing the covariance matrix. The leading diagonal of this matrix contains the variance of the estimated parameters. The uncertainty in the parameters is taken as the standard deviation of the variance from the covariance matrix.

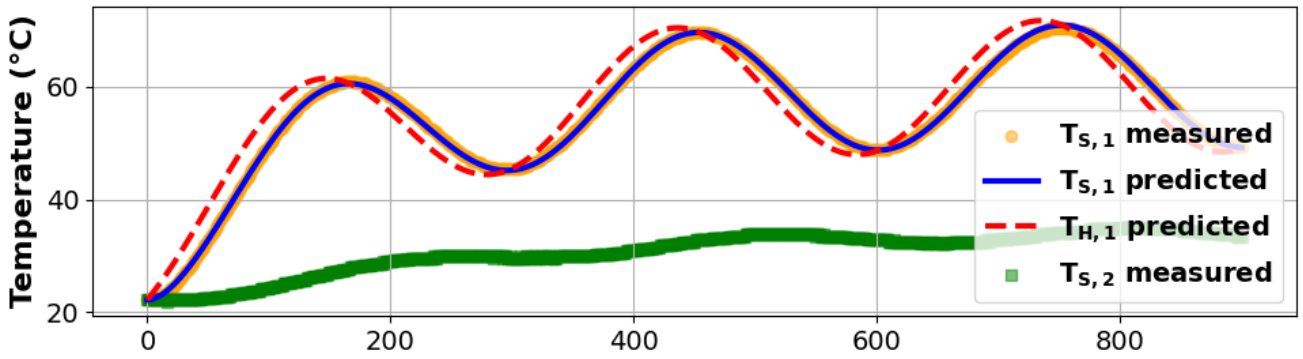
Assuming Gaussian (correlated or independent and identically distributed) measurement errors, the covariance matrix can be computed using three methods in ParmEst.

Parameter estimates (from ParmEst) have high uncertainty

$U_a = 0.042 \pm 3.3 \times 10^{-5}$ W/°C , $U_b = 0.017 \pm \mathbf{1.7 \times 10^{3}}$ W/°C, $(C_{p,H})^{-1} = 0.171 \pm \mathbf{1.1 \times 10^{3}}$ °C/J,
$(C_{p,S})^{-1} = 3.138 \pm \mathbf{3.3 \times 10^{5}}$ °C/J

Covariance matrix (`finite_difference` method)

	U_a	U_b	$(C_{p,H})^{-1}$	$(C_{p,S})^{-1}$
U_a	$\mathbf{1.1 \times 10^{-9}}$	5.2×10^{-2}	3.5×10^{-2}	-1.0×10^{1}
U_b	5.2×10^{-2}	$\mathbf{2.9 \times 10^{6}}$	1.9×10^{6}	-5.6×10^{8}
$(C_{p,H})^{-1}$	3.5×10^{-2}	1.9×10^{6}	$\mathbf{1.3 \times 10^{6}}$	-3.8×10^{8}
$(C_{p,S})^{-1}$	-1.0×10^{1}	-5.6×10^{8}	-3.8×10^{8}	$\mathbf{1.1 \times 10^{11}}$



Eigenvalues of V_θ:

$[1.1 \times 10^{11} \quad 9.2 \times 10^{3} \quad 1.4 \times 10^{-10} \quad 1.5 \times 10^{3}]$

Eigenvectors of V_θ :

[0.0000	-0.0000	1.0000	0.0000]	U_a
[0.0051	0.9929	0.0000	0.1186]	U_b
[0.0034	0.1186	0.0000	-0.9929]	$(C_{p,H})^{-1}$
[-1.0000	0.0055	0.0000	-0.0028]	$(C_{p,S})^{-1}$

Condition number: 6.7×10^{20}

Four ways to possibly improve estimates:

- Multi-start initialization optimization
- Regularization using prior information
- More data collection via experimental design
- Alternative model structures / formulations

Notebooks at `dowlinglab.github.io/pyomo-doe`

Multistart Optimization with ParmEst

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

Parameter estimation problems are often **nonconvex**, meaning that the objective function $g(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ may contain multiple local minima. As a result, the solution obtained by a nonlinear optimization solver can depend strongly on the initial guess for $\boldsymbol{\theta}$.

Multistart optimization is a practical strategy to address this challenge. Instead of solving the problem once from a single initial point, we solve it multiple times from different initializations and compare the resulting solutions.

CONTENTS

Multistart and Estimability Analysis Further Characterize the Objective Function

Identifiability describes a model or model parameter's ability to be uniquely estimated.

- **Structural** identifiability – **model** focused
- **Practical** identifiability or **estimability** – **data quality** focused

8. Perform multistart optimization

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \frac{1}{2}\left(\mathbf{y}-\mathbf{f}(\mathbf{x};\boldsymbol{\theta})\right)^{\mathrm{T}}\boldsymbol{\Sigma}_y^{-1}\left(\mathbf{y}-\mathbf{f}(\mathbf{x};\boldsymbol{\theta})\right),$$

$$\boldsymbol{\Theta} = \left\{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_p \middle| \boldsymbol{\theta}_l \le \boldsymbol{\theta}_s \le \boldsymbol{\theta}_u, s=1,\ldots,p\right\}$$

where $\boldsymbol{\Theta}$ is randomly sampled using multiple methods (random uniform, Latin Hypercube Sampling (LHS), or Sobol sampling)

$\mathbf{y}$ = measured variables, L = likelihood function, $\boldsymbol{\epsilon}$ = measurement errors, $\boldsymbol{\theta}$ = parameters, l = log-likelihood function, n = number of experiments, WSSE = weighted sum of squared errors, $\mathbf{x}$ = experiment conditions, $\mathbf{V}$ = covariance of the measurement error

9. Evaluate estimability with profile likelihood

$$\mathrm{PL_j}\left(\theta_j^*\right) = \min_{\boldsymbol{\theta}:\theta_j=\theta_j^*} \mathrm{WSSE}(\boldsymbol{\theta})$$

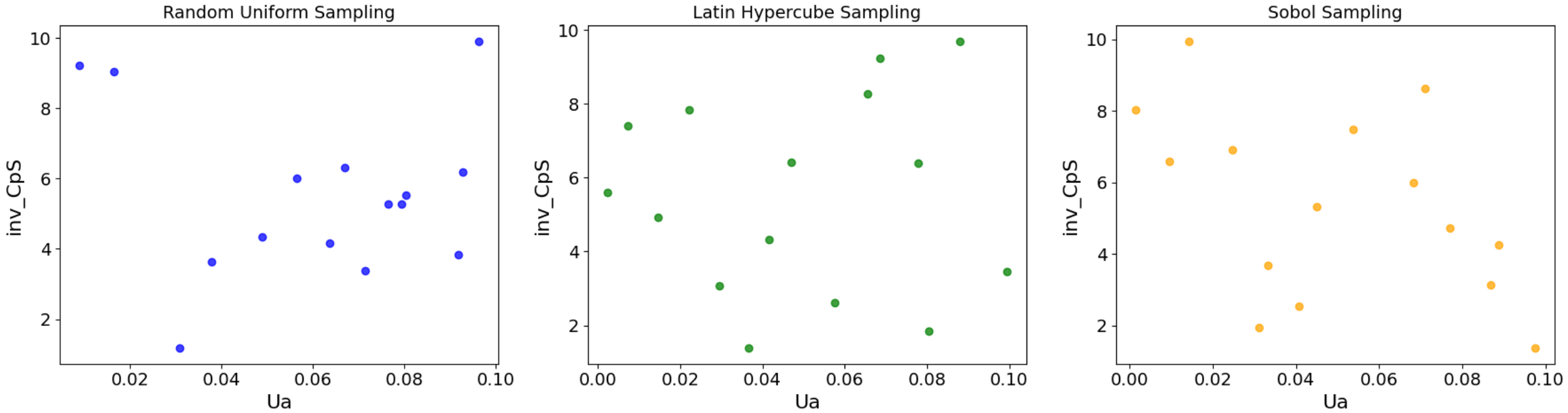
$$2[\mathrm{PL_j}\left(\theta_j^*\right) - \mathrm{WSSE}(\hat{\boldsymbol{\theta}})] \le \chi^2_{1,\alpha}$$

where one θ, θ_j^*, is fixed and sampled within its bounds, while all the others are estimated, to see how the profile of the objective changes

Profile likelihood pseudocode

```
profiles = []

for pname in profiled_theta_list:
    for g in grid_values[pname]:

        # set the profiled parameter value
        init_theta[pname] = float(g)

        # fix pname at g and re-optimize all other parameters
        obj_val, theta_conv, term = self._Q_opt(
        theta_vals=init_theta,
        fixed_theta_values={pname: float(g)},
        )

        # compute profile likelihood statistic
        profile_value = 2.0 * (obj_val - obj_hat)
```

TCLab: Results from multistart sampling methods



	random_ uniform	latin_ hypercube	sobol_ sampling
Ua	0.041705	0.041705	0.041705
Ub	0.017065	0.017199	0.017104
inv_CpH	0.170701	0.170791	0.170727
inv_CpS	3.137848	3.111786	3.130236
objective	53.773993	53.773993	53.773993

Takeaways:

- Stratified sampling techniques (LHS and Sobol) cover more of sampling space and avoid multiple samples in same region
- Other than Ua, values differ between techniques, but same objective value.
- Similar to with covariance, this points to potential issue with estimability.

Notebooks at `dowlinglab.github.io/pyomo-doe`

Profile Likelihood with ParmEst

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

After estimating parameter values, it is important to assess whether those parameters are supported by the available experimental data. In other words, we would like to understand not only which parameter values minimize the objective function, but also how much confidence we should have in those values.

Two related concepts are especially important in parameter estimation:

1. **Identifiability**

   A parameter is identifiable if it can, in principle, be uniquely determined from ideal noise-free data and the specified model structure.

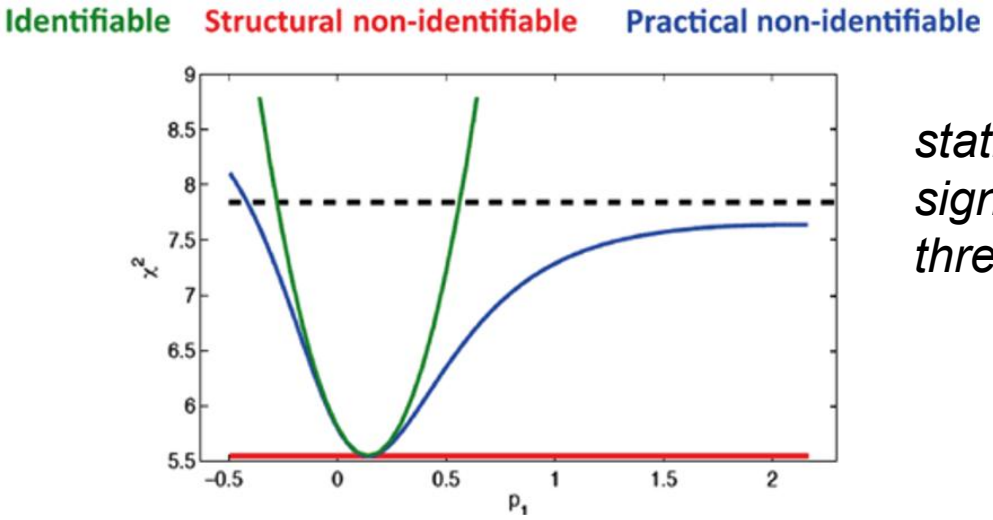
   Structural identifiability is a property of the model itself. If a parameter is structurally unidentifiable, then no amount of additional data from the same

CONTENTS

Please open the notebook in Colab and run the first cell (takes a few minutes).

Activity: Discuss estimability of parameters based on profile likelihood results

Analyzing profile likelihood:

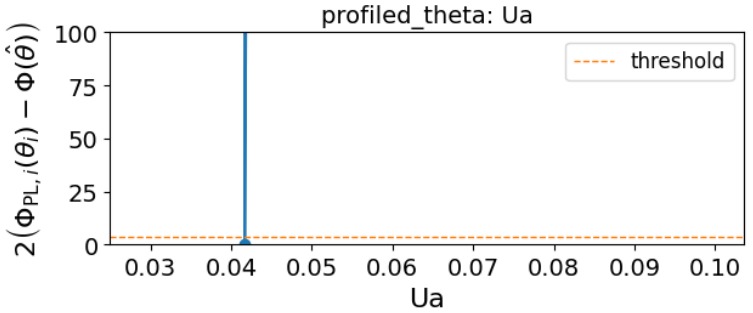


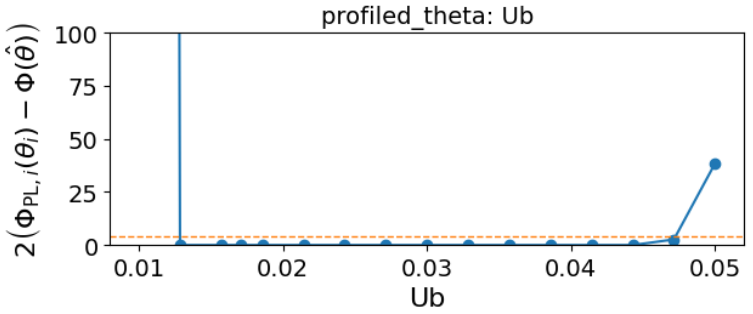
statistical significance threshold

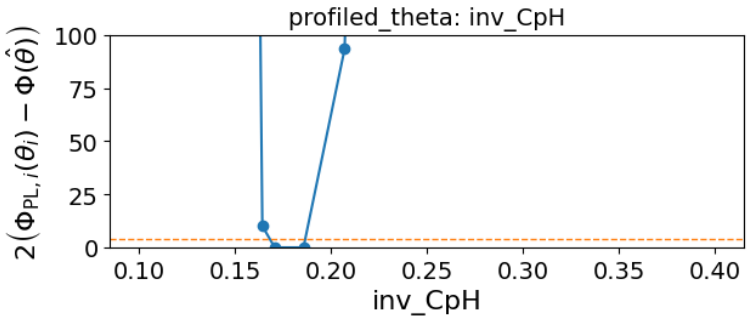
Bachmann et al (2011), *Journal of Internal Medicine*

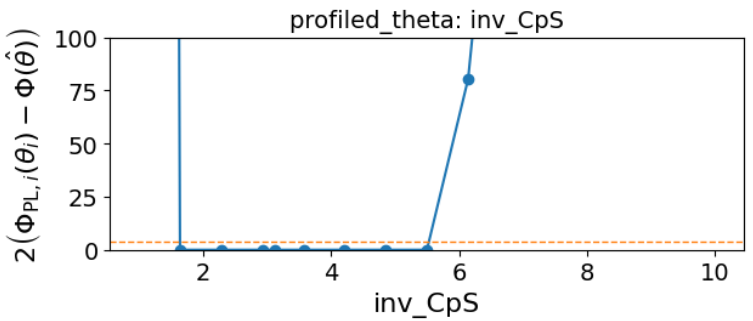
From results (used ylims to zoom in):

- U_a is most likely estimable (practically identifiable)
- U_b, CpH^{-1}, and CpS^{-1} are most likely inestimable
- U_b and CpS^{-1} may be structurally nonidentifiable









Notebooks at `dowlinglab.github.io/pyomo-doe`

L2 Regularization with ParmEst

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

Parameter estimation problems can be challenging when the available data are limited, noisy, or weakly informative for some parameters. In these cases, the least squares or maximum likelihood objective may have broad valleys, multiple local optima, or parameter estimates that are physically unreasonable.

One way to incorporate additional information is through **regularization**. Regularization augments the parameter estimation objective with an additional term that penalizes deviations from prior parameter values.

CONTENTS

Regularization Leverages Prior Parameter Information

Define the Fisher information matrix

$$\mathbf{M}(\boldsymbol{\theta}) = -\mathbb{E}\left(\frac{\partial^2 l(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}} \mid \boldsymbol{\theta}\right)$$

Information contained in experiment design **X** and measurements **y** about parameters $\boldsymbol{\theta}$ in model $\mathbf{f}(\cdot,\cdot)$

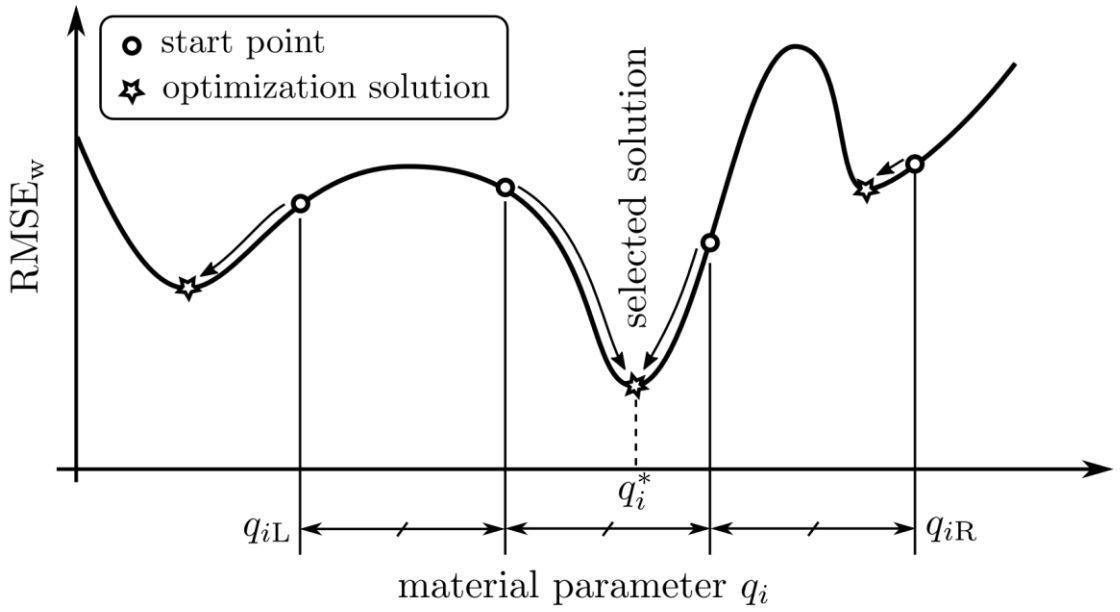
Use previous estimate (or physical intuition) $\mathbf{M}$ and $\hat{\boldsymbol{\theta}}$ for regularization

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathrm{WSSE} + \frac{1}{2}\left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\mathbf{0}}\right)^{\mathrm{T}} \mathbf{M}\left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\mathbf{0}}\right)$$

L2 Regularization
(Gaussian Prior)

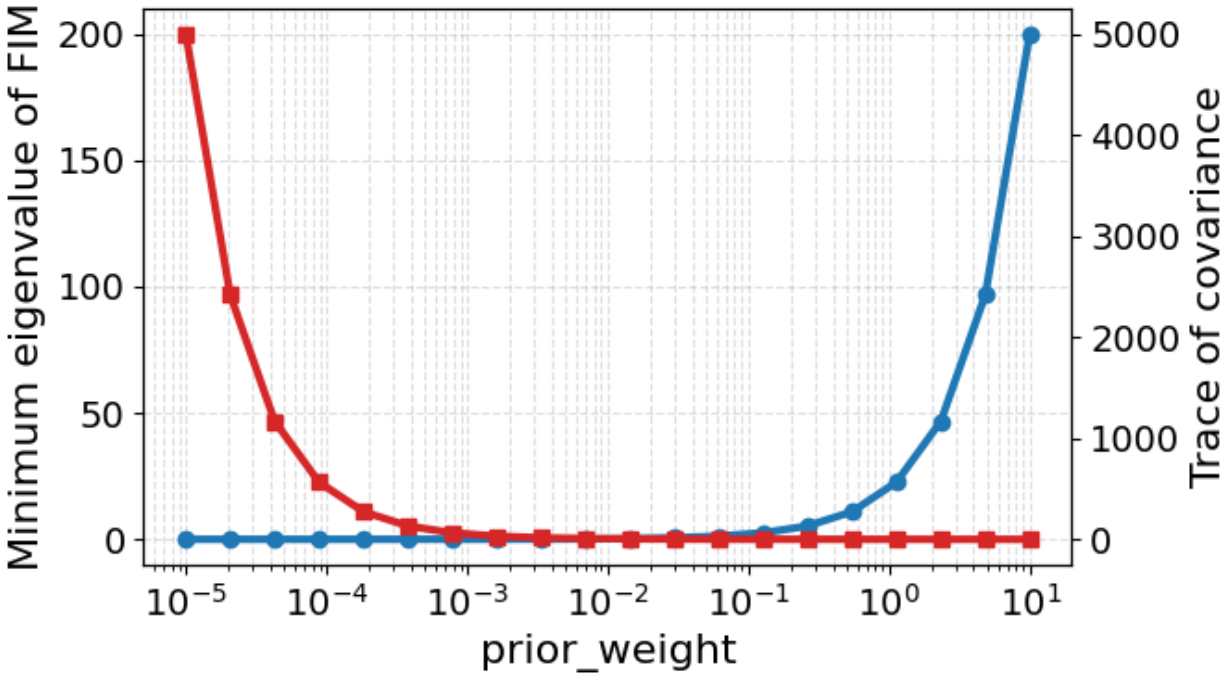
Adding regularization makes the problem equivalent to **Maximum A Posteriori (MAP) estimation**, because we are incorporating more prior information about parameters.

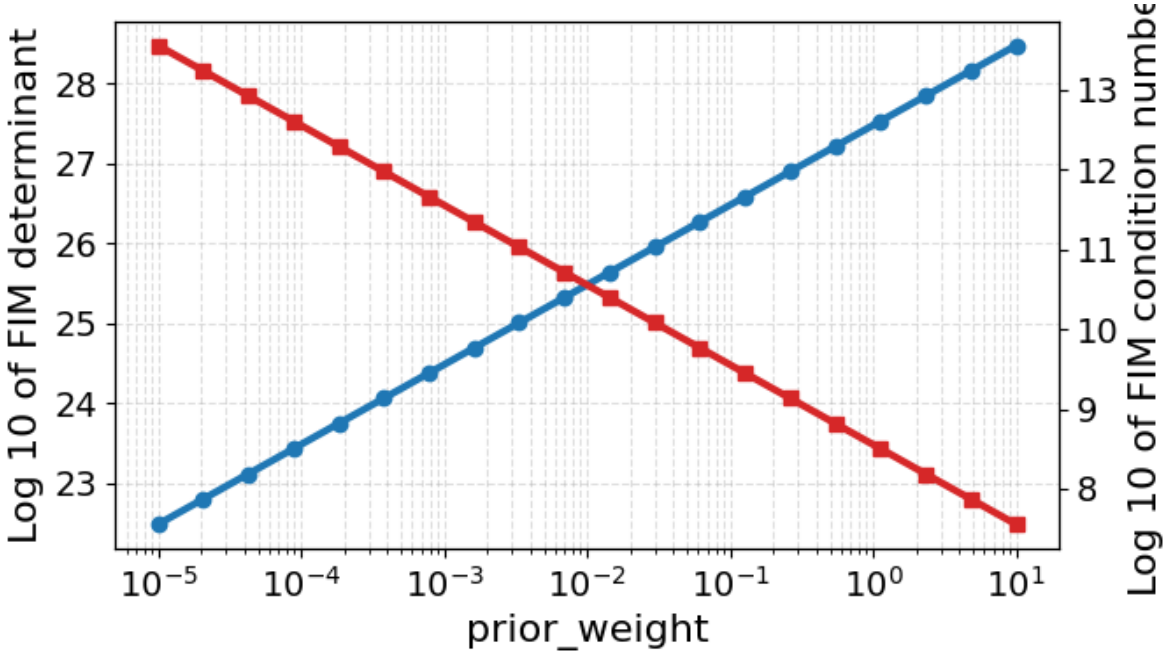
$\mathbf{y}$ = measured variables, L = likelihood function, $\boldsymbol{\epsilon}$ = measurement errors, $\boldsymbol{\theta}$ = parameters, l = log-likelihood function, n = number of experiments, WSSE = weighted sum of squared errors, $\mathbf{x}$ = experiment conditions, $\mathbf{V}$ = covariance of the measurement error



Reisinger et al. *Biomechanics and Modeling in Mechanobiology* (2020)

Activity: How does the strength of the prior affect the parameter estimation?





Takeaways: increasing the regularization weight…

- Raises minimum eigenvalue and determinant of FIM (**blue**)
- Lowers the FIM condition number and trace of covariance matrix (**red**)

Notebooks at `dowlinglab.github.io/pyomo-doe`

Exercise: ParmEst

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

In the main workshop tutorial, the four heat transfer parameters, U_a, U_b, C_p^H, C_p^S, were estimated from the sine test data using the following TC Lab model:

$$C_p^H \frac{dT_H}{dt} = U_a(T_{\text{amb}} - T_H) + U_b(T_S - T_H) + \alpha P u(t), \quad (1)$$

$$C_p^S \frac{dT_S}{dt} = U_b(T_H - T_S) \quad (2)$$

where T_H and T_S are the heater and sensor temperatures in $^\circ\text{C}$, respectively, T_{amb} is the ambient temperature in $^\circ\text{C}$, C_p^H and C_p^S are the heat capacity of the heater and sensor in $\text{J}/^\circ\text{C}$, respectively, U_a and U_b are the heat transfer coefficients from the heater to the sensor and the sensor to ambient (in $\text{W}/^\circ\text{C}$), respectively, P is the maximum power limit in bit, u is the heater power in $\%$, and α is constant that converts the unit of $Pu(t)$ into W.

CONTENTS

Exercise: Linear model reformulation (w.r.t. parameters) can improve estimability

Original TC Lab model

$$\frac{dT_{H,1}}{dt} = \frac{U_a(T_{\text{amb}} - T_{H,1}) + U_b(T_{S,1} - T_{H,1}) + \alpha P_1 u_1(t)}{C_{p,H}},$$

$$\frac{dT_{S,1}}{dt} = \frac{U_b(T_{H,1} - T_{S,1})}{C_{p,S}}$$

Recover "physical" model parameters θ and covariance

$$U_a = \frac{\alpha P_1 \beta_1}{\beta_4}, \quad U_b = \frac{\alpha P_1 \beta_2}{\beta_4},$$

$$C_{p,H} = \frac{\alpha P_1}{\beta_4}, \quad C_{p,S} = \frac{\alpha P_1 \beta_2}{\beta_3 \beta_4}$$

$$\mathbf{V}_{\boldsymbol{\theta}} = \left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\beta}}\right) \mathbf{V}_{\boldsymbol{\beta}} \left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\beta}}\right)^{\mathrm{T}}$$

Reformulated TC Lab model

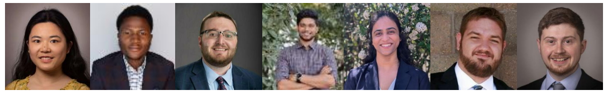
$$\frac{dT_{H,1}}{dt} = \beta_1(T_{amb} - T_{H,1}) + \beta_2(T_{S,1} - T_{H,1}) + \beta_4 u_1(t),$$

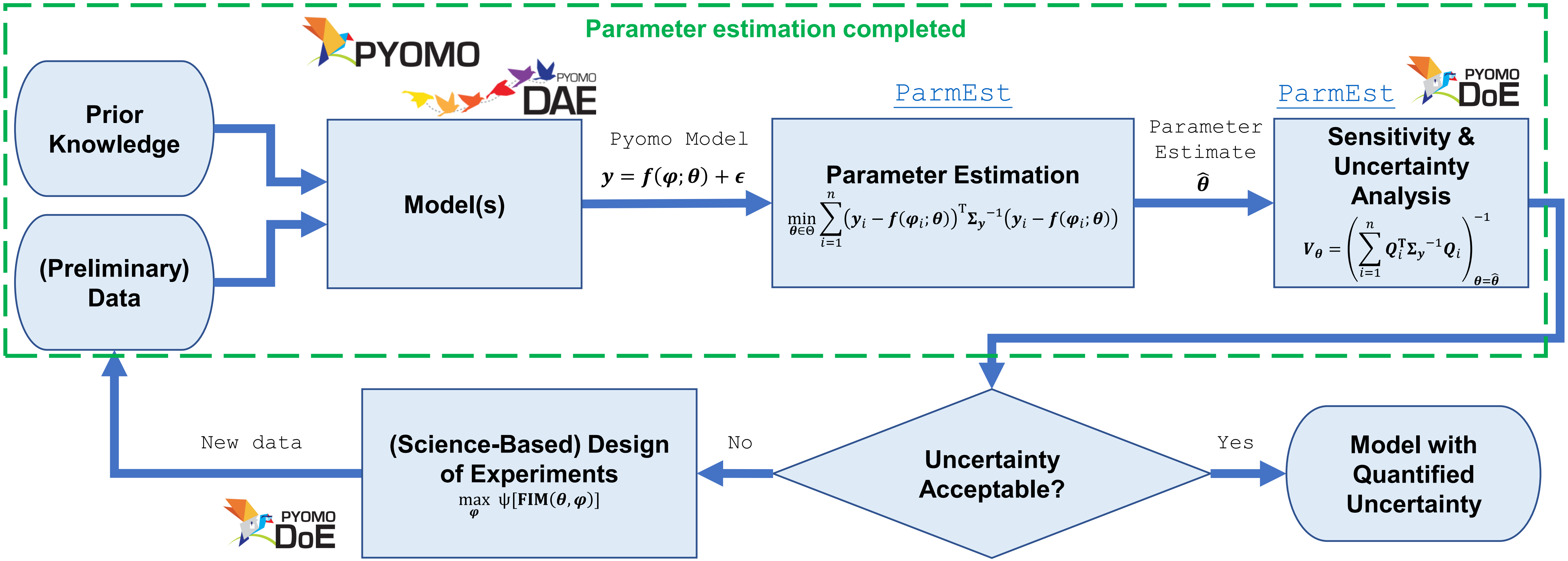
$$\frac{dT_{S,1}}{dt} = \beta_3(T_{H,1} - T_{S,1})$$

$$\beta_1 = \frac{U_a}{C_{p,H}}, \quad \beta_2 = \frac{U_b}{C_{p,H}}, \quad \beta_3 = \frac{U_b}{C_{p,S}}, \quad \beta_4 = \frac{\alpha P_1}{C_{p,H}}$$

U_a = heater heat transfer coefficient in W/°C,
U_b = sensor heat transfer coefficient in W/°C,
$C_{p,H}$ = heat capacity of the heater in J/°C, t = time in s,
$T_{H,1}$ = temperature of heater 1 in °C,
$T_{S,1}$ = temperature of sensor 1 in °C,
T_{amb} = ambient temperature in °C,
$C_{p,S}$ = heat capacity of the sensor in J/°C,
u_1 = heater 1 power in %,
P_1 = hardware-set maximum power limit in bit,
α = known constant for converting the unit of $P_1 u_1$ into W

Estimating the four parameters follows an iterative approach





Pyomo is an open-source Python package for optimization modeling, **ParmEst** is a Pyomo tool for parameter estimation, **Pyomo.DoE** is a Pyomo tool for design of experiments, θ = unknown parameters, y = observation of measured variables, $\mathbf{FIM}$ = Fisher information matrix, V_θ = parameter covariance matrix, φ = design vector, ψ = scalar measure (e.g., determinant), Q = Jacobian matrix, ϵ = measurement error, Σ_y = measurement error covariance matrix, and n = number of experiments

UNIVERSITY OF NOTRE DAME · AUBURN UNIVERSITY · CCSI² Carbon Capture Simulation for Industry Impact · PROMMIS Process Optimization and Modeling for Minerals Sustainability

Agenda

Part 1 (12:15pm – 1:30pm): Welcome to Pyomo

- TCLab Mathematical Model
- Simulation and Dynamic Optimization in Pyomo
- Experiment Abstraction

Part 2 (1:45pm – 3pm): Parameter Estimation

- ParmEst Overview
- Uncertainty Quantification
- Multistart & Profile Likelihood
- Regularization

Part 3 (3:30pm – 5pm): Optimal Experiment Design

- Pyomo.DoE Overview
- OED Objectives
- Symbolic Derivatives
- Planning Multiple Experiments

Notebooks at `dowlinglab.github.io/pyomo-doe`

Getting Started with Pyomo.DoE

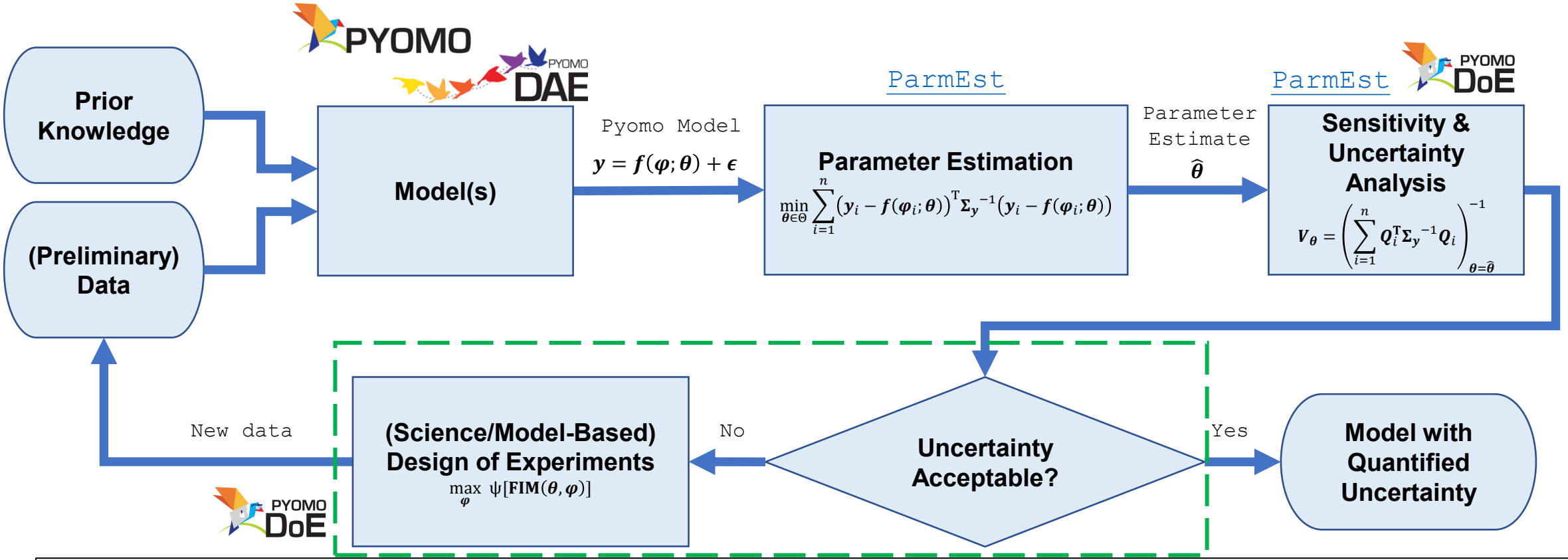
Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

Model-based design uses physics-based models. It starts with prior knowledge and iterates through the loop until a satisfactory model is found. We have already discussed parameter estimation in the first part of the workshop. If we have multiple models, we can discriminate among them to find the best model that fits the data. However, that is not a part of this workhsop.

Please open the notebook in Colab and run the first cell (takes a few minutes).

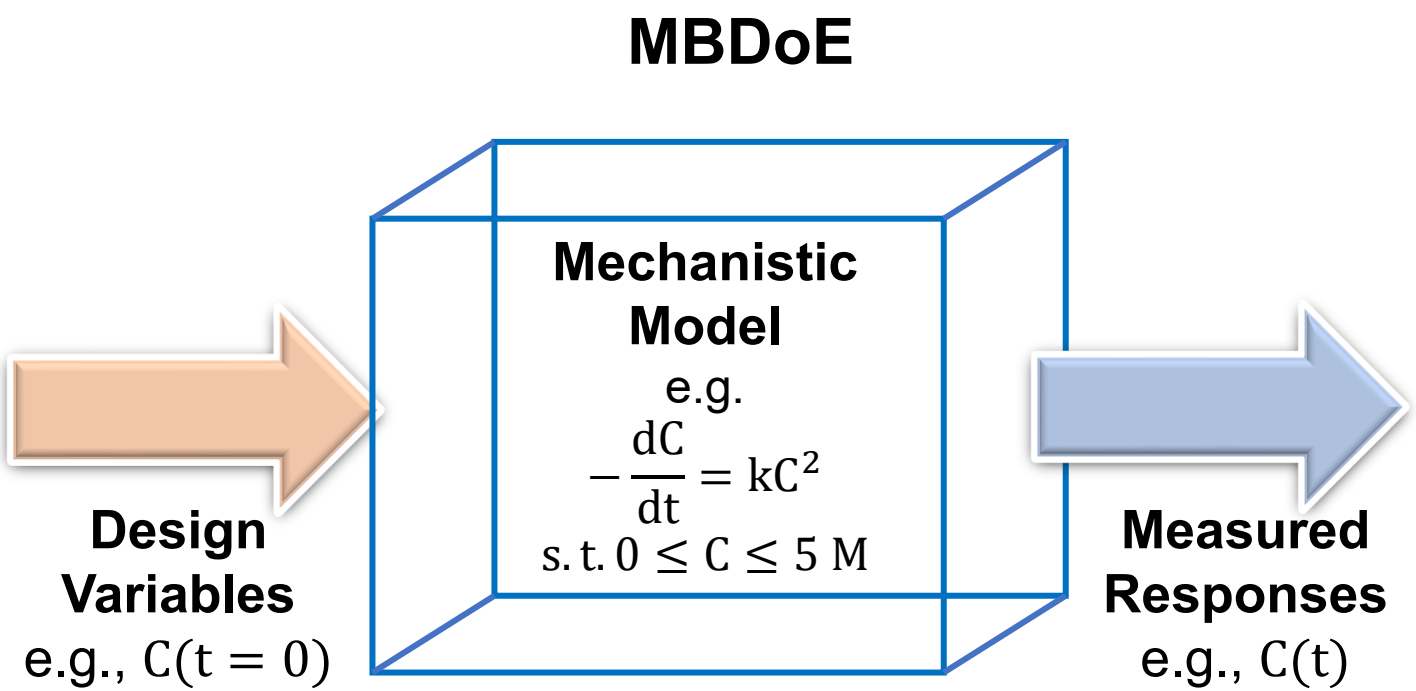
CONTENTS

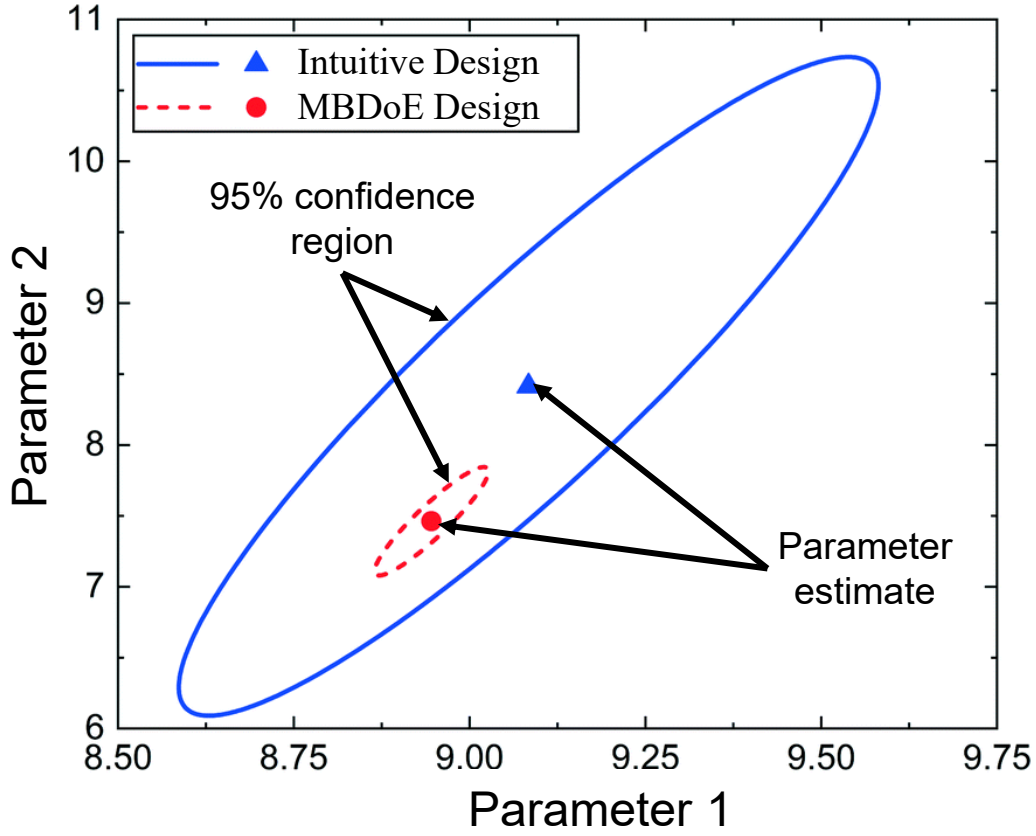
Optimizing new experiments



Pyomo is an open-source Python package for optimization modeling, **ParmEst** is a Pyomo tool for parameter estimation, **Pyomo.DoE** is a Pyomo tool for design of experiments, $\boldsymbol{\theta}$ = unknown parameters, $\boldsymbol{y}$ = observation of measured variables, $\mathbf{FIM}$ = Fisher information matrix, $\boldsymbol{V_\theta}$ = parameter covariance matrix, $\boldsymbol{\varphi}$ = design vector, ψ = scalar measure (e.g., determinant), $\boldsymbol{Q}$ = Jacobian matrix, $\boldsymbol{\epsilon}$ = measurement error, $\boldsymbol{\Sigma_y}$ = measurement error covariance matrix, and n = number of experiments

Model-based design of experiments (MBDoE) optimizes information gained from next experiment(s)





Waldron et al. (2020), *Reaction Chemistry & Engineering*

MBDoE uses a mechanistic model to optimize experimental conditions that improve parameter precision[1]

MBDoE can reduce parameter uncertainty relative to intuitive designs

[1] Franceschini and Macchietto (2008), *Chem. Eng. Sci.*

Model-Based DoE Optimization Formulation

$$\max_{\boldsymbol{\varphi}} \quad \Psi[\, \boldsymbol{M}(\, \hat{\boldsymbol{\theta}}, \boldsymbol{\varphi}\,)\,]$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\, \boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \bar{\boldsymbol{w}}, \hat{\boldsymbol{\theta}}\,)$$

$$\boldsymbol{g}(\, \boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \bar{\boldsymbol{w}}, \hat{\boldsymbol{\theta}}\,) = \boldsymbol{0}$$

$$\boldsymbol{y}(t) = \boldsymbol{h}(\, \boldsymbol{x}(t), \boldsymbol{z}(t), \hat{\boldsymbol{\theta}})$$

DAE System

$$\boldsymbol{f^0}(\dot{\boldsymbol{x}}(t_0), \boldsymbol{x}(t_0), \boldsymbol{z}(t_0), \boldsymbol{u}(t_0), \bar{\boldsymbol{w}}, \hat{\boldsymbol{\theta}}) = \boldsymbol{0}$$

$$\boldsymbol{g^0}(\, \boldsymbol{x}(t_0), \boldsymbol{z}(t_0), \boldsymbol{u}(t_0), \bar{\boldsymbol{w}}, \hat{\boldsymbol{\theta}}) = \boldsymbol{0}$$

$$\boldsymbol{y^0}(t_0) = \boldsymbol{h}(\, \boldsymbol{x}(t_0), \boldsymbol{z}(t_0), \hat{\boldsymbol{\theta}})$$

Initial Conditions

$$\boldsymbol{m}(\, \boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \bar{\boldsymbol{w}}, \hat{\boldsymbol{\theta}}\,) = \boldsymbol{0}$$

We will start with **pseudo-A optimality**:

$$\Psi(\mathbf{M}) = \log_{10}(\text{trace}(\mathbf{M}))$$

$\boldsymbol{y}$	Measurements (model responses)
$\hat{\boldsymbol{\theta}}$	Estimated parameters
$\boldsymbol{x}$	Time-dependent differential state variables
$\boldsymbol{z}$	Time-dependent algebraic state variables
$\boldsymbol{u}$	Time-varying control variables
$\bar{\boldsymbol{w}}$	Time-invariant control variable

Fisher information matrix (FIM):

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

MBDoE Decisions:

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

Franceschini, G., & Macchietto, S. (2008). Model-based design of experiments for parameter precision: State of the art. *Chemical Engineering Science*, *63*(19), 4846-4872.

Scaling in Pyomo.DoE

1. Constant scaling (`scale_constant_value`)

Multiplies every element of the sensitivity matrix $\mathbf{Q} = \frac{\partial \hat{y}}{\partial \theta}\Big|_{\hat{\theta}}$ by a single uniform scalar.

Impact: Shifts the overall magnitude to avoid floating-point errors but **does not improve the matrix condition number**.

2. Nominal Parameter Scaling (`scale_nominal_param_value`)

Scales each column of the sensitivity matrix by its corresponding nominal parameter value.

Impact: Directly improves the condition number of the sensitivity matrix and the resulting FIM. This is the preferred choice when parameter ranges span multiple orders of magnitude.

$$\mathbf{Q}_{\text{scaled}} = \begin{bmatrix} \frac{\partial y_1}{\partial \theta_1}\theta_1 & \frac{\partial y_1}{\partial \theta_2}\theta_2 & \cdots & \frac{\partial y_1}{\partial \theta_p}\theta_p \\ \frac{\partial y_2}{\partial \theta_1}\theta_1 & \frac{\partial y_2}{\partial \theta_2}\theta_2 & \cdots & \frac{\partial y_2}{\partial \theta_p}\theta_p \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial \theta_1}\theta_1 & \frac{\partial y_n}{\partial \theta_2}\theta_2 & \cdots & \frac{\partial y_n}{\partial \theta_p}\theta_p \end{bmatrix}$$

If we use scaling, then we also need to ensure that the prior FIM is also scaled accordingly and after designing the experiments, we also need to unscale the FIM to make a fair comparison

TC Lab: Pseudo-A-Optimal Next Experiment

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

$$\max_{u} \ \log \text{trace}\left(\mathbf{M}(u) + \mathbf{M}_0\right)$$

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

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

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

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

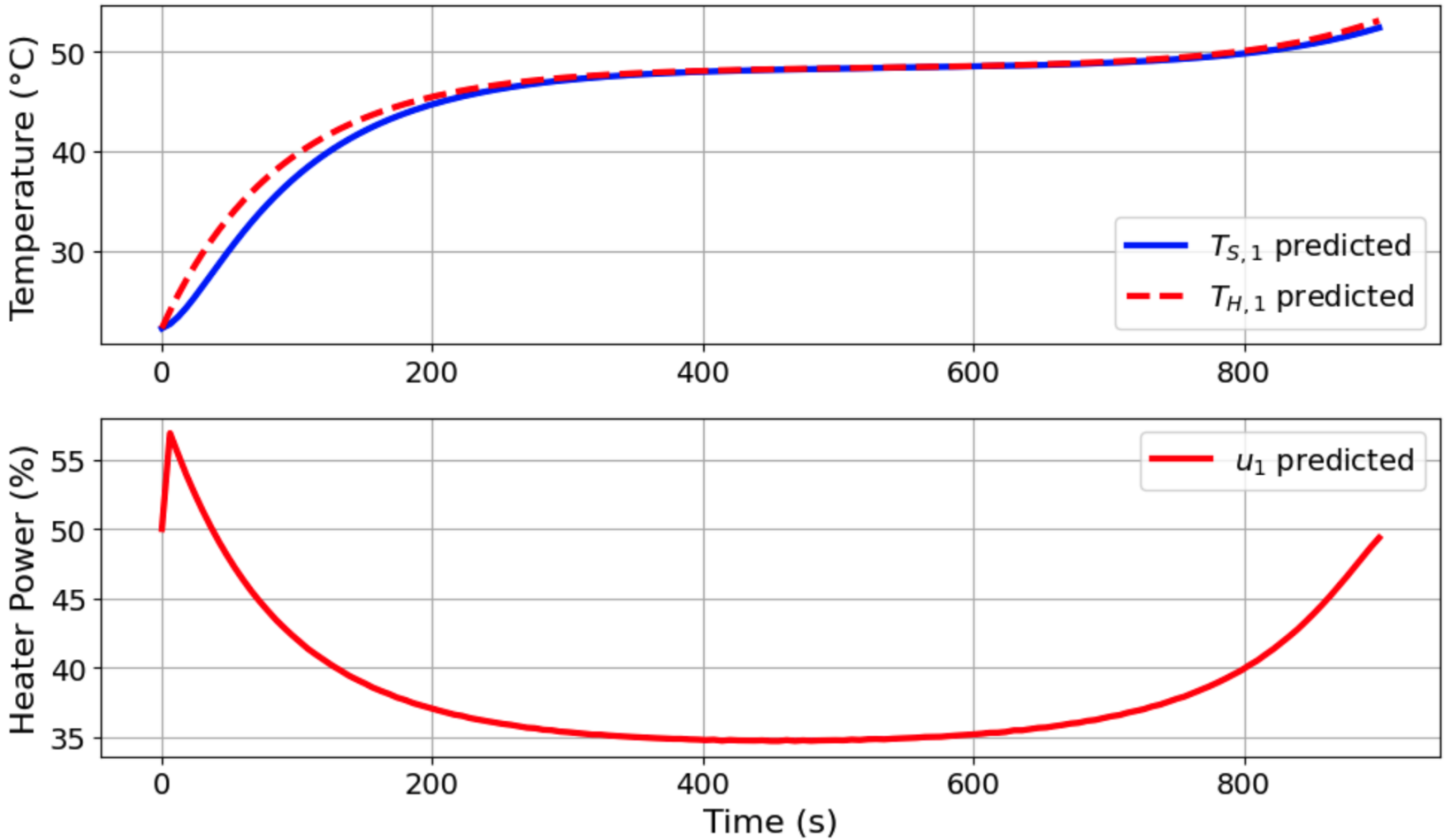
$$T_S(t_0) = T_{amb}$$

Parameters ± SD (Prior)

```
Ua: 4.170e-02 ± 1.363e-05
Ub: 1.201e-02 ± 4.030e-04
inv_CpH: 1.675e-01 ± 3.212e-04
inv_CpS: 4.545e+00 ± 1.581e-01
```

Parameters ± SD (Design)

```
Ua: 4.170e-02 ± 1.363e-05
Ub: 1.201e-02 ± 4.877e-05
inv_CpH: 1.675e-01 ± 2.064e-04
inv_CpS: 4.545e+00 ± 6.429e-04
```



pseudo-A optimal design improves the estimability of C_p^S

Notebooks at `dowlinglab.github.io/pyomo-doe`

Experiment Design Objectives in Pyomo.DoE

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

In the previous notebook, we formulated model-based design of experiments as an optimal control problem using one Fisher information matrix (FIM)-based objective.

In this workbook, we extend that idea to additional optimality criteria. We will use Pyomo.DoE to formulate, initialize, and solve experiment design problems using D-, A-, pseudo A-, E-, and ME-optimality objectives. This lets us compare how the choice of design criterion changes the optimized input profile, $u(t)$, and the information content of the resulting experiment.

CONTENTS

Please open the notebook in Colab and run the first cell (takes a few minutes).

Alphabetic Design Criteria Measure Information Content

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

A-optimality

min trace($\mathbf{V_\theta}$)

enclosing box volume

poor choice for highly correlated $\boldsymbol{\theta}$

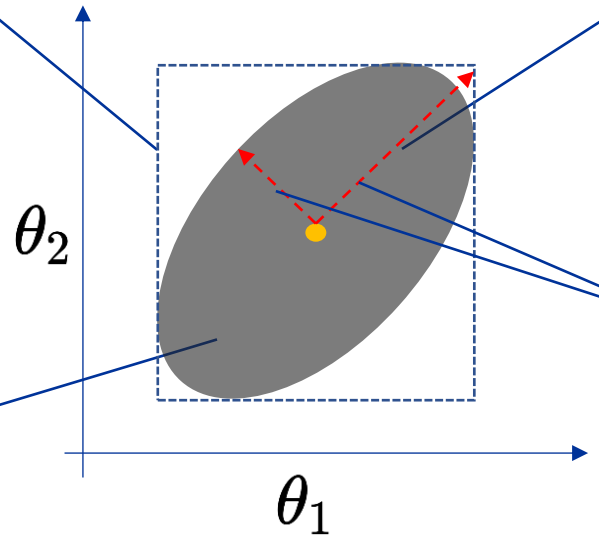
D-optimality

max det($\mathbf{M}$) = 1 / min det($\mathbf{V_\theta}$)

ellipsoid volume

robust to linear transformations

confidence ellipsoid for covariance matrix $\mathbf{V_\theta} \approx \mathbf{M}^{-1}$



E-optimality

max min(eig($\mathbf{M}$)) = min max eig(($\mathbf{V_\theta}$))

major axis

recommended if $\mathbf{M}$ is ill-conditioned

ME-optimality

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

UNIVERSITY OF NOTRE DAME · AUBURN UNIVERSITY · CCSI² Carbon Capture Simulation for Industry Impact · PROMM Process Optimization and M for Minerals Sustainability

Default Pyomo.DoE Formulation: MBDoE as 2-Stage Program

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

s.t. $\mathbf{M} = \sum_{r}\sum_{r'} \tilde{\sigma}_{r,r'} \mathbf{Q}_r^{\mathrm{T}} \mathbf{Q}_{r'}$ **Stage 1**

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

$q_{r,p}(t) = \frac{y^{+}_{r,p}(t) - y^{-}_{r,p}(t)}{2\,\epsilon_p}$ Central finite difference

$\boldsymbol{m}\left(\boldsymbol{x}_p^{+}(t), \boldsymbol{y}_p^{+}(t), \boldsymbol{z}_p^{+}(t), \boldsymbol{u}(t), \bar{\boldsymbol{w}}, \boldsymbol{\theta}_p^{+}\right) = \mathbf{0}$

$\boldsymbol{m}\left(\boldsymbol{x}_p^{-}(t), \boldsymbol{y}_p^{-}(t), \boldsymbol{z}_p^{-}(t), \boldsymbol{u}(t), \bar{\boldsymbol{w}}, \boldsymbol{\theta}_p^{-}\right) = \mathbf{0}$ Two model evaluations

$\boldsymbol{\theta}_p^{+} = \hat{\boldsymbol{\theta}} + \boldsymbol{e}_p \epsilon_p$

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

Stage 2

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

Model Sensitivity

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

$$\mathbf{q}_{r,p} = \begin{bmatrix} \frac{\partial y_r(t_1)}{\partial \theta_p} & \cdots & \frac{\partial y_r(t_n)}{\partial \theta_p} \end{bmatrix}^{\mathrm{T}}$$

$\boldsymbol{y}$	Measurements (model responses)
$\mathbf{Q_r}$	Dynamic sensitivity for response *r*
$\boldsymbol{m}(\,)$	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*
$\boldsymbol{e}_p \in \mathbb{R}^P$	Unit vector with "1" in position *p*

Wang and Dowling, 2022. *AIChE J.*

Improving Computational Performance by Simplifying the Optimal DoE Problem Formulation

$$\max_{\phi_l \le \phi \le \phi_u} \quad \mathbf{FIM} \rightarrow \boxed{\text{Grey Box}} \rightarrow \text{logdet}\,(\mathbf{FIM})$$

$$\text{s.t.} \quad \mathbf{FIM}\left(\hat{\boldsymbol{\theta}}, \boldsymbol{\phi}\right) = \sum_{r}^{N_r} \sum_{r'}^{N_r} \tilde{\sigma}_{(r,r')} \mathbf{Q}_r^{\mathrm{T}} \mathbf{Q}_{r'} + \mathbf{FIM}_{\mathrm{prior}}$$

$$\boldsymbol{q}_{r,p} = \frac{\boldsymbol{y}_{r,p}^{+} - \boldsymbol{y}_{r,p}^{-}}{2\epsilon_p} \quad \forall\, r \in \{1, ..., N_{\mathrm{meas}}\} \forall\, p \in \{1, ..., N_p\}$$

$$\boldsymbol{y}_{r,p}^{+} = m\left(x_p, \boldsymbol{\theta}_p^{+}\right) \quad \forall\, p \in \{1, ..., N_p\}$$

$$\boldsymbol{y}_{r,p}^{-} = m\left(x_p, \boldsymbol{\theta}_p^{-}\right) \quad \forall\, p \in \{1, ..., N_p\}$$

$$\boldsymbol{\theta}_p^{+} = \hat{\boldsymbol{\theta}} + \boldsymbol{e}_p \epsilon_p \quad \forall\, p \in \{1, ..., N_p\}$$

$$\boldsymbol{\theta}_p^{-} = \hat{\boldsymbol{\theta}} - \boldsymbol{e}_p \epsilon_p \quad \forall\, p \in \{1, ..., N_p\}$$

Jacobian $$\frac{\partial\, \text{logdet}\,(\mathbf{FIM})}{\partial\, \mathbf{FIM}} = \frac{1}{2}\left(\mathbf{FIM}^{-1} + \mathbf{FIM}^{-T}\right)$$

Hessian $$\frac{\partial}{\partial \mathbf{FIM}_{\mathrm{kl}}}\left(\frac{\partial \text{logdet}\,(\mathbf{FIM})}{\partial \mathbf{FIM}}\right)_{\mathrm{ij}} = (\mathbf{FIM}^{-1})_{\mathrm{il}} (\mathbf{FIM}^{-1})_{\mathrm{kj}}$$

Callback to a “grey box” function allows D-optimality calculation without Cholesky Factorization

This requires derivatives of the outputs: $\text{logdet}\,(\mathbf{FIM})$ with respect to the inputs: $\mathbf{FIM}$

Some Common Criteria Cannot Be Generally Posed in Equation-Oriented Formulations

Dan Laky

Laky et al (2026), Optimal Experimental Design using Eigenvalue-Based Criteria with Pyomo.DoE, *arXiv:2604.03354*

$$\max_{\phi_l \le \phi \le \phi_u} \quad \Psi\left(\mathbf{FIM}\left(\hat{\theta}, \phi\right)\right)$$

$$\text{s.t.} \quad \mathbf{FIM}\left(\hat{\theta}, \phi\right) = \sum_{r}^{N_r} \sum_{r'}^{N_r} \tilde{\sigma}_{(r,r')} \mathbf{Q}_r^{\mathrm{T}} \mathbf{Q}_{r'} + \mathbf{FIM}_{\mathrm{prior}}$$

$$\boldsymbol{q}_{r,p} = \frac{\boldsymbol{y}_{r,p}^{+} - \boldsymbol{y}_{r,p}^{-}}{2\epsilon_p} \quad \forall\, r \in \{1, ..., N_{\mathrm{meas}}\} \forall\, p \in \{1, ..., N_p\}$$

$$\boldsymbol{y}_{r,p}^{+} = m\left(x_p, \boldsymbol{\theta}_p^{+}\right) \quad \forall\, p \in \{1, ..., N_p\}$$

$$\boldsymbol{y}_{r,p}^{-} = m\left(x_p, \boldsymbol{\theta}_p^{-}\right) \quad \forall\, p \in \{1, ..., N_p\}$$

$$\boldsymbol{\theta}_p^{+} = \hat{\boldsymbol{\theta}} + \boldsymbol{e}_p \epsilon_p \quad \forall\, p \in \{1, ..., N_p\}$$

$$\boldsymbol{\theta}_p^{-} = \hat{\boldsymbol{\theta}} - \boldsymbol{e}_p \epsilon_p \quad \forall\, p \in \{1, ..., N_p\}$$

Eigenvalue-based metrics cannot be algebraically expressed beyond 4 parameters

E-optimality

max min(eig(**M**))

major axis

recommended if **M** is ill-conditioned

ME-optimality

min κ(**M**) = max(eig(**M**)) / min(eig(**M**))

ratio of major to minor axes

recommended if **M** is ill-conditioned

PYOMO DoE

UNIVERSITY OF NOTRE DAME · AUBURN UNIVERSITY · CCSI² Carbon Capture Simulation for Industry Impact · PROMMIS Process Optimization and Modeling for Minerals Sustainability

GreyBox to Evaluate Eigenvalue Metrics

Dan Laky

$$\max_{\phi_l \le \phi \le \phi_u} \quad \Psi\left(\mathbf{FIM}\left(\hat{\theta}, \phi\right)\right)$$

$\mathbf{FIM}$ → Grey Box → $\Psi\left(\mathbf{FIM}\left(\hat{\theta}, \phi\right)\right)$

$$\text{s.t.} \quad \mathbf{FIM}\left(\hat{\theta}, \phi\right) = \sum_{r}^{N_r} \sum_{r'}^{N_r} \tilde{\sigma}_{(r,r')} \mathbf{Q}_r^{\mathrm{T}} \mathbf{Q}_{r'} + \mathbf{FIM}_{\text{prior}}$$

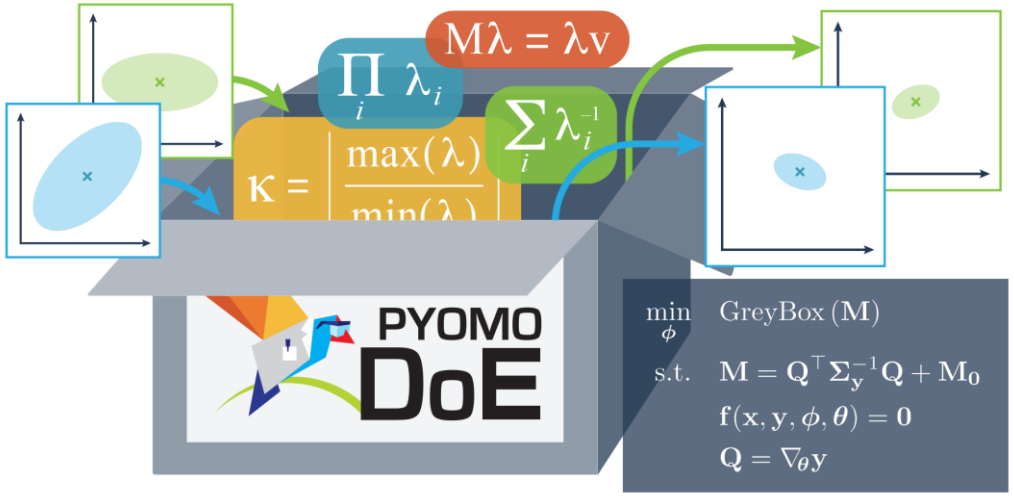
$$\boldsymbol{q}_{r,p} = \frac{\boldsymbol{y}_{r,p}^{+} - \boldsymbol{y}_{r,p}^{-}}{2\epsilon_p} \quad \forall\, r \in \{1, ..., N_{\text{meas}}\} \forall\, p \in \{1, ..., N_p\}$$

$$\boldsymbol{y}_{r,p}^{+} = m\left(x_p, \boldsymbol{\theta}_p^{+}\right) \quad \forall\, p \in \{1, ..., N_p\}$$

$$\boldsymbol{y}_{r,p}^{-} = m\left(x_p, \boldsymbol{\theta}_p^{-}\right) \quad \forall\, p \in \{1, ..., N_p\}$$

$$\boldsymbol{\theta}_p^{+} = \hat{\boldsymbol{\theta}} + \boldsymbol{e}_p \epsilon_p \quad \forall\, p \in \{1, ..., N_p\}$$

$$\boldsymbol{\theta}_p^{-} = \hat{\boldsymbol{\theta}} - \boldsymbol{e}_p \epsilon_p \quad \forall\, p \in \{1, ..., N_p\}$$

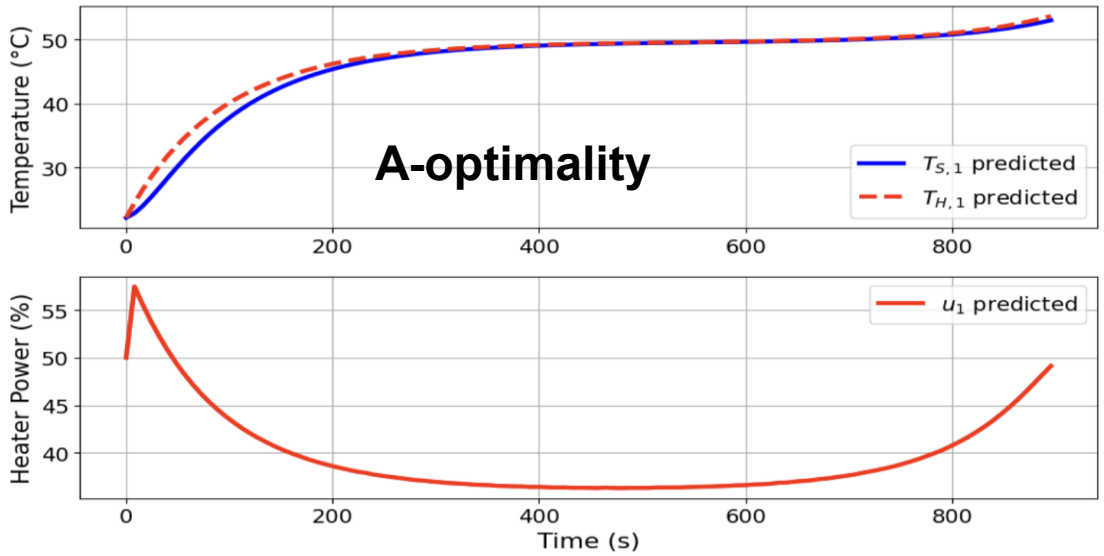


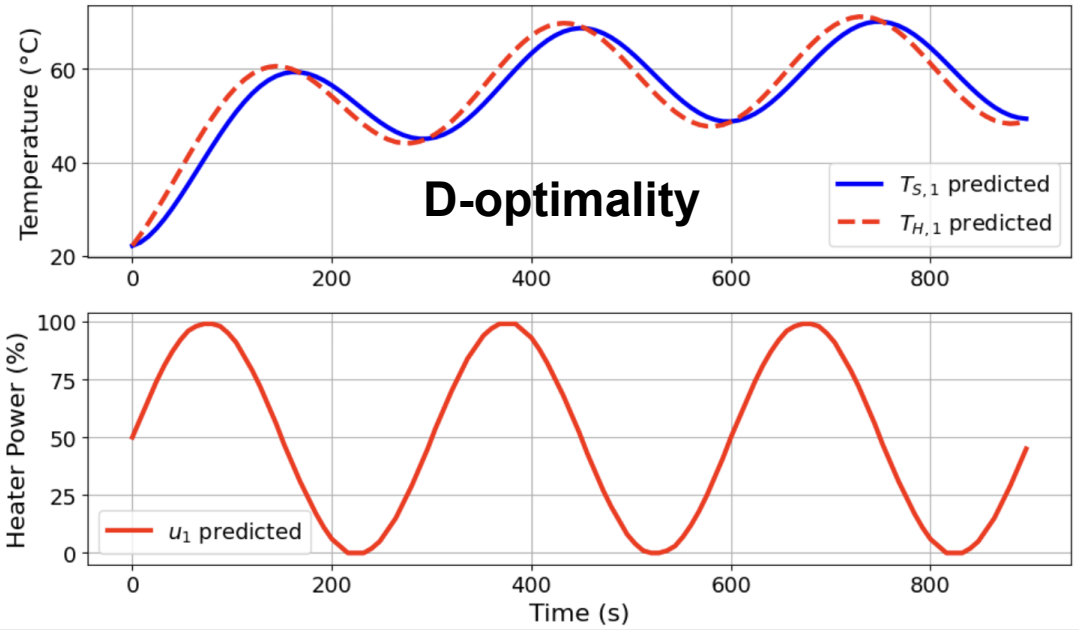
arXiv:2604.03354

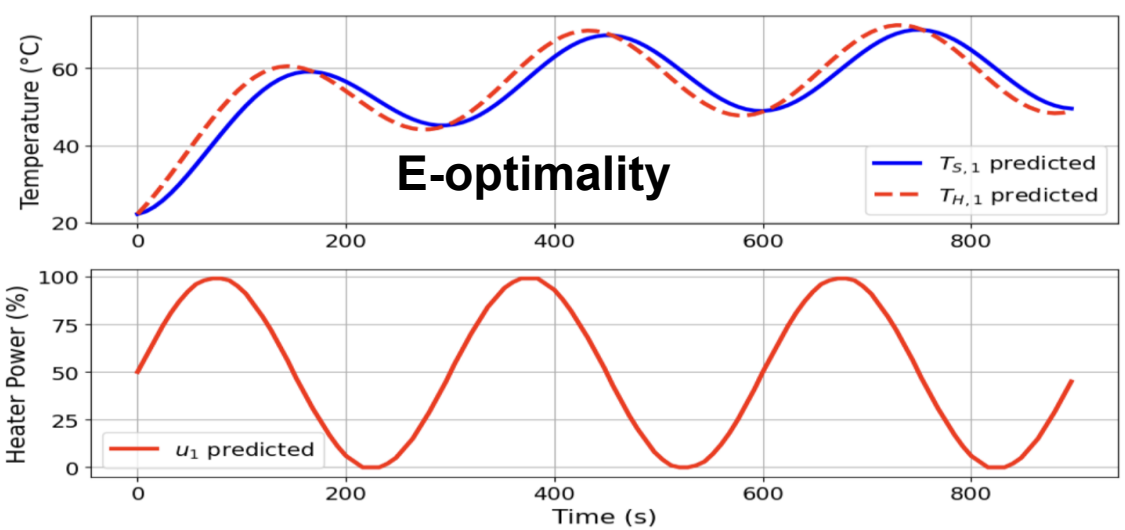
Key Contribution: Derivation of exact first and second derivative formulas of metrics Ψ with respect to elements of **M**

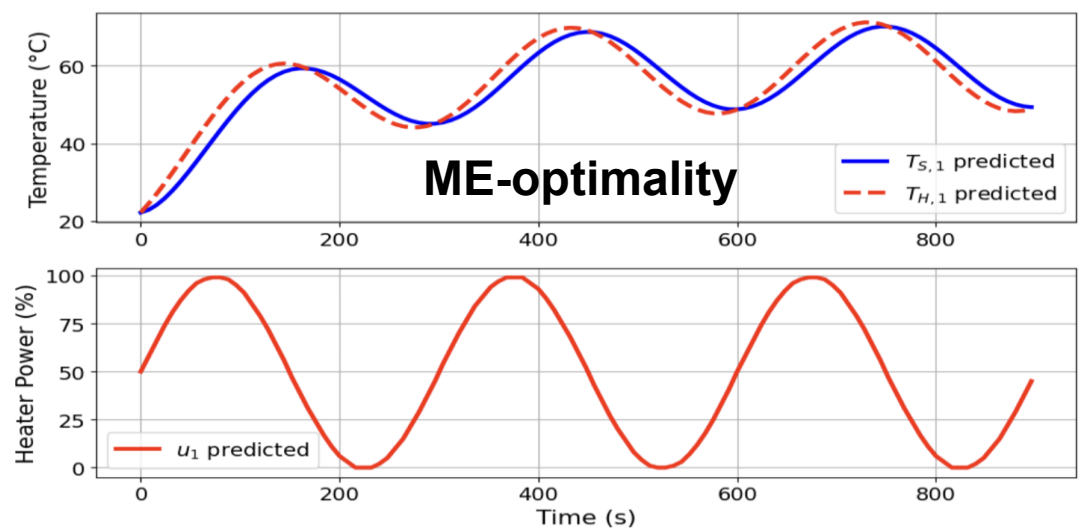
Take Away: Eigenvalue objectives now available for many unknown parameters within an equation-oriented optimization framework (Pyomo)

Optimized Experiments from Different Design Criteria









Cross-Evaluation of Optimized Experiment Designs

Experiment Design	$\log_{10}(\mathrm{M(u)} + \mathrm{M_0})^{-1}$	$\log_{10}\det(\mathrm{M(u)} + \mathrm{M_0})$	$\log_{10}\lambda_{\min}(\mathrm{M(u)} + \mathrm{M_0})$	$\log_{10}\kappa\,(\mathrm{M(u)} + \mathrm{M_0})$	Design decision
Prior FIM	-1.6	27.78	1.6	8.25	N/A
A-optimal	-7.34	36.12	7.36	2.59	Maximize overall information
D-optimal	-4.66	30.84	4.66	5.20	Maximize average uncertainty
E-optimal	-4.66	30.84	4.66	5.20	Improve weak direction
ME-optimal	-4.72	30.90	4.72	5.14	Better conditioned than E

ME reduces eigenvalue spread relative to E-opt

Notebooks at `dowlinglab.github.io/pyomo-doe`

Computing Sensitivity Matrix in Pyomo.DoE

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

Recall that `Pyomo.DoE` utilizes the sensitivity matrix defined as the derivative of the estimates of the measured outputs $\hat{y}$ with respect to the parameter to be estimated $\hat{\theta}$:

$$Q = \nabla_{\hat{\theta}} \hat{y} \tag{1}$$

to compute the FIM as:

$$M = Q^T \Sigma_y^{-1} Q \tag{2}$$

where Σ_y is the measurement covariance matrix

CONTENTS

Please open the notebook in Colab and run the first cell (takes a few minutes).

Sensitivity Computation Using Symbolic Derivatives

Sensitivity matrix $\boldsymbol{Q} = \nabla_{\hat{\theta}}\hat{\boldsymbol{y}}$ feeds the FIM: $\boldsymbol{M} = \boldsymbol{Q}^{T}\boldsymbol{\Sigma}_{y}^{-1}\boldsymbol{Q}$ where $\boldsymbol{\Sigma}_{y}$ is the measurement covariance matrix

1. Central Finite Difference

$$\boldsymbol{Q} \approx \frac{\hat{\boldsymbol{y}}(\hat{\boldsymbol{\theta}} + \Delta\hat{\boldsymbol{\theta}}) - \hat{\boldsymbol{y}}(\hat{\boldsymbol{\theta}} - \Delta\hat{\boldsymbol{\theta}})}{2\Delta\hat{\boldsymbol{\theta}}}$$

where $\Delta\hat{\theta}$ is the perturbation step size

Requires $2\mathrm{n_p} + 1$ copies of the base model to solve

Optimization problem size significantly **increases with number of parameters**

Default/legacy approach in Pyomo.DoE

2. Automatic Differentiation (PyNumero)

Implicit model of the system:

$$\mathrm{F}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{\theta}}, \boldsymbol{u}) = 0$$

PyNumero computes derivatives of F: $\nabla_{\hat{x}}\mathrm{F}$, $\nabla_{\hat{y}}\mathrm{F}$, $\nabla_{\hat{\theta}}\mathrm{F}$.
These are rearranged to obtain:

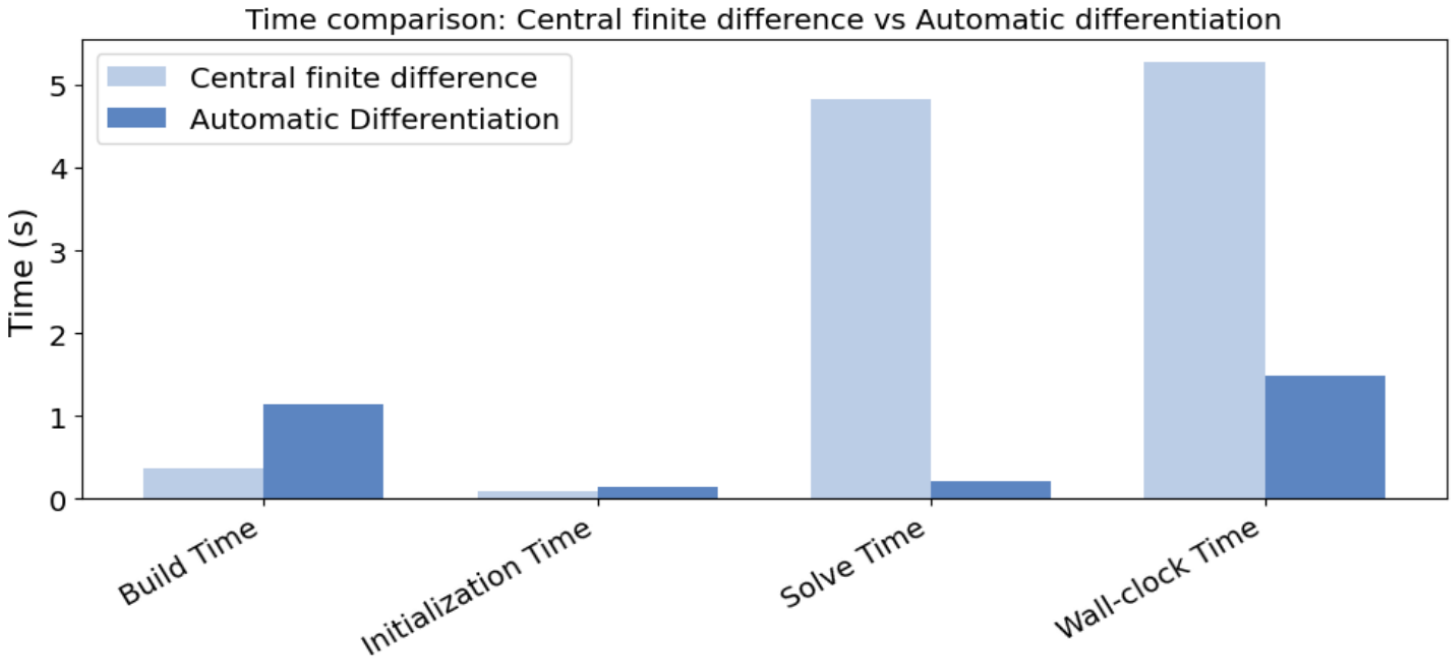
$$\nabla_{\hat{\theta}}\mathrm{F} = \begin{bmatrix} \nabla_{\hat{y}}\mathrm{F} \\ \nabla_{\hat{x}}\mathrm{F} \end{bmatrix} \begin{bmatrix} \nabla_{\hat{\theta}}\hat{\boldsymbol{y}} := \boldsymbol{Q} \\ \nabla_{\hat{\theta}}\hat{x} \end{bmatrix}$$

The output sensitivity needed for the FIM is: $Q = \nabla_{\hat{\theta}}\hat{\boldsymbol{y}}$

No repeated parameter-perturbation solves are required.

New feature in Pyomo.DoE

Sensitivity computation within Pyomo.DoE: A Comparison



Time Component	Finite Difference	Symbolic (Automatic Differentiation (AD))
Build Time	0.4 s	1.2 s
Solve Time	0.1 s	**0.15 s**
Initialization Time	4.8 s	0.2 s
Wall-clock Time	5.3 s	**1.5 s**

Key Takeaways

- Symbolic (AD) build time longer due to derivative setup
- Symbolic (AD) greatly reduces solve times by avoiding repeated parameter perturbations
- Overall speedup: approximately 3.5 times faster based on wall-clock time
- Symbolic (AD) is likely beneficial for large problems with many parameters

Notebooks at `dowlinglab.github.io/pyomo-doe`

Planning Multiple Experiments with Pyomo.DoE

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

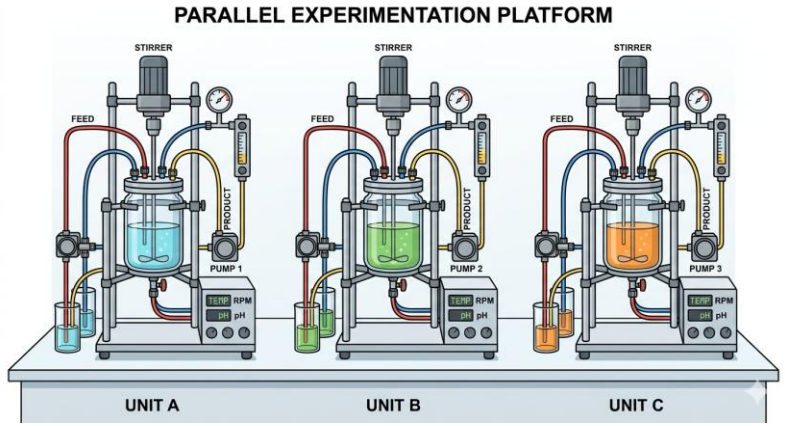
Multi-experiment design is useful when several experiments can be run in parallel. Instead of designing one experiment at a time, we optimize multiple experiments together so that they provide complementary information about the model parameters. In Pyomo.DoE, this is done by maximizing a scalar function of the sum of the Fisher information matrices from each experiment, together with any prior information.

CONTENTS

Please open the notebook in Colab and run the first cell (takes a few minutes).

Why design multiple experiments?

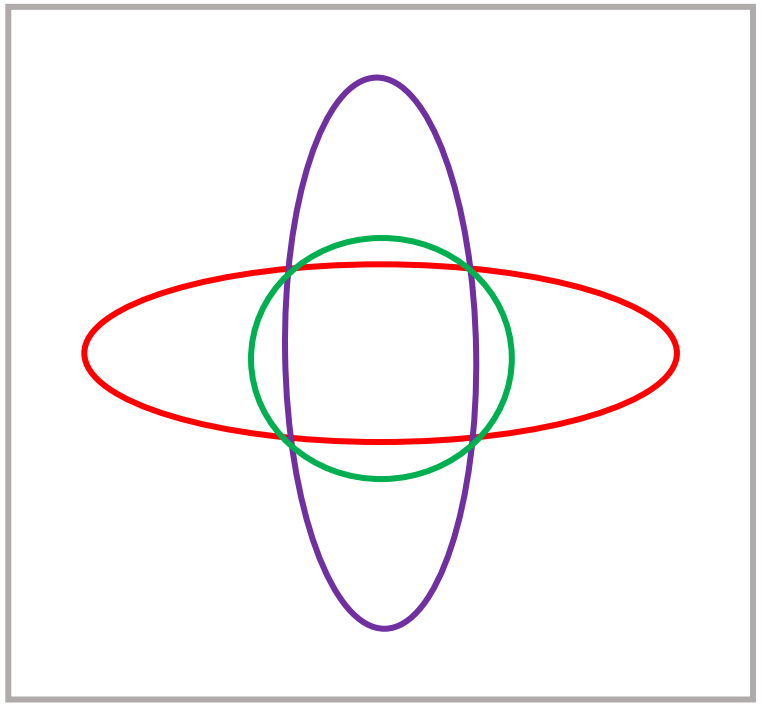
Multiple experimental setups[1]



Gemini-generated image

https://ucallmlabs.com

Complementary Information[2]



Confidence ellipse:
– exp 1 only
– exp 2 only
– exp 1 and 2 combined

[1] Galvanin et al. (2007), *Industrial & Eng. Chem. Research*
[2] Ballan et al. (2012), *Computer Aided Chem. Eng.*

Multi-experiment design formulation

$$\max_{\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_{N_{exp}}} \quad \psi(\mathbf{M})$$

Choose all experiment designs simultaneously

$$\text{s.t. } \mathbf{M} = \mathbf{M}_0 + \sum_{k\in E} \mathbf{M}_k(\hat{\boldsymbol{\theta}}, \boldsymbol{\phi}_k)$$

Combine information

$$\mathbf{M}_k(\hat{\boldsymbol{\theta}}, \boldsymbol{\phi}_k) = \mathbf{Q}_k^T \Sigma_y^{-1} \mathbf{Q}_k; \quad \forall k \in E$$

Information from experiment k

$$\mathbf{Q}_k(\hat{\boldsymbol{\theta}}) = \frac{\partial \mathbf{h}(\mathbf{x}_k(t), \mathbf{z}_k(t), \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} |_{\hat{\boldsymbol{\theta}}}; \quad \forall k \in E$$

Sensitivity to parameters

$$\phi_{primary,1} \leq \cdots \leq \phi_{primary,N_{exp}}$$

Avoid equivalent experiment permutations

Each experiment contributes information; multi-experiment design optimizes the total information from the set

E : Set of experiments, $\{1,\ldots,N_{exp}\}$; $\mathbf{M}$: Fisher information matrix; $\mathbf{M}_0$: Prior Fisher information matrix; $\hat{\mathbf{y}}_k$: Model prediction for experiment k; $\hat{\boldsymbol{\theta}}$: Parameter estimate; $\boldsymbol{\phi}$: Design vector; $\phi_{primary}$: A selected variable from the design vector $\boldsymbol{\phi}$

Galvanin et al. (2007), Industrial & Eng. Chem. Research

TC Lab: Multi-experiment Design with D-optimality

$$\max_{u_1,..,u_{N_{exp}}} \quad \psi\left(\mathbf{M}_0 + \sum_{k\in E} \mathbf{M}_k(u_k)\right)$$

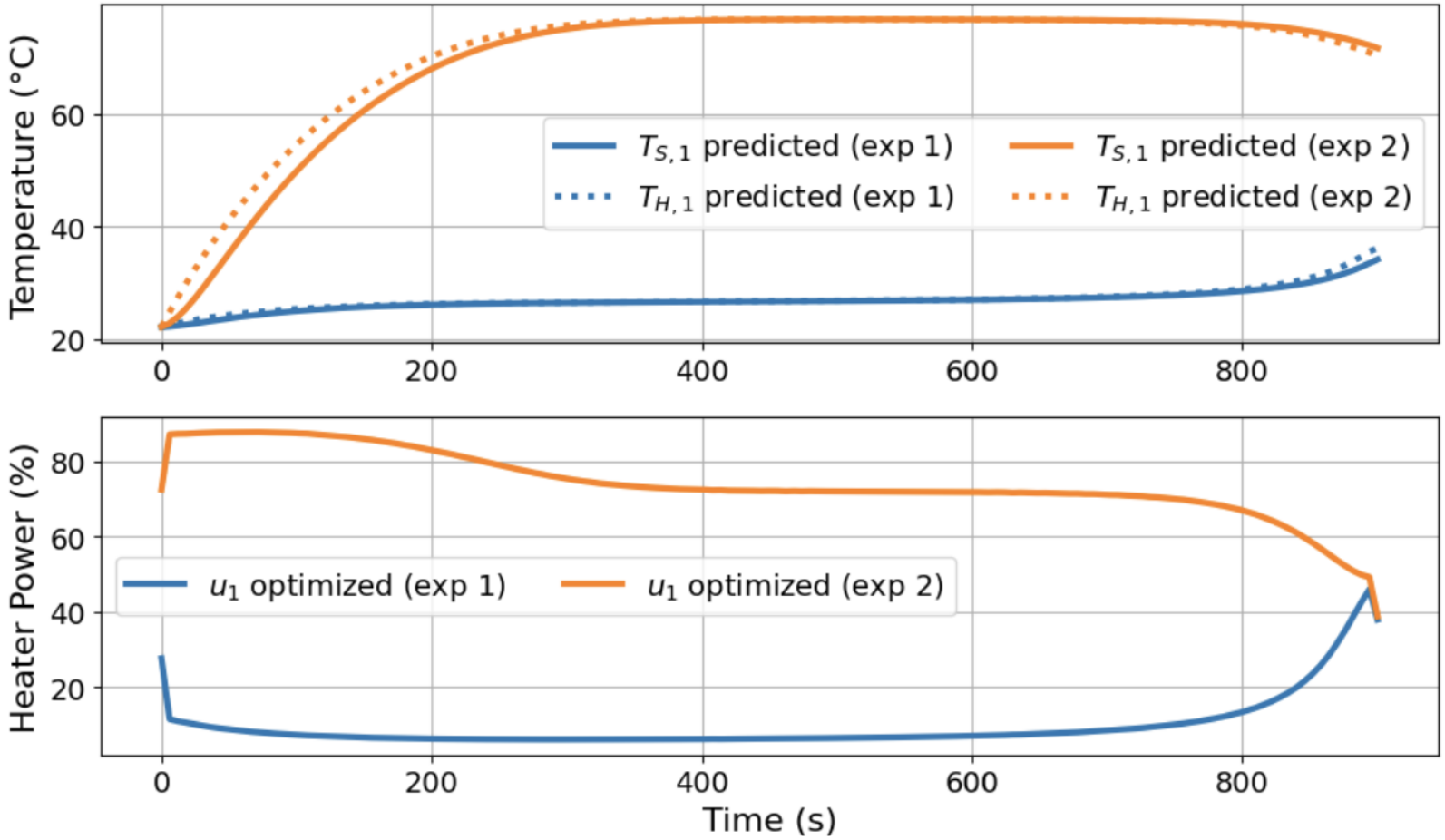
$$\text{s.t.} \quad C_p^H \frac{dT_{H,k}}{dt} = U_a(T_{amb,k} - T_{H,k}) + \cdots$$

$$C_p^S \frac{dT_{S,k}}{dt} = U_b(T_{H,k} - T_{S,k}); \ \forall k \in E$$

$$0\% \le u_k(t) \le 100\%; \ \forall k \in E$$

$$T_{H,k}(t_0) = T_{amb,k}; \ \forall k \in E$$

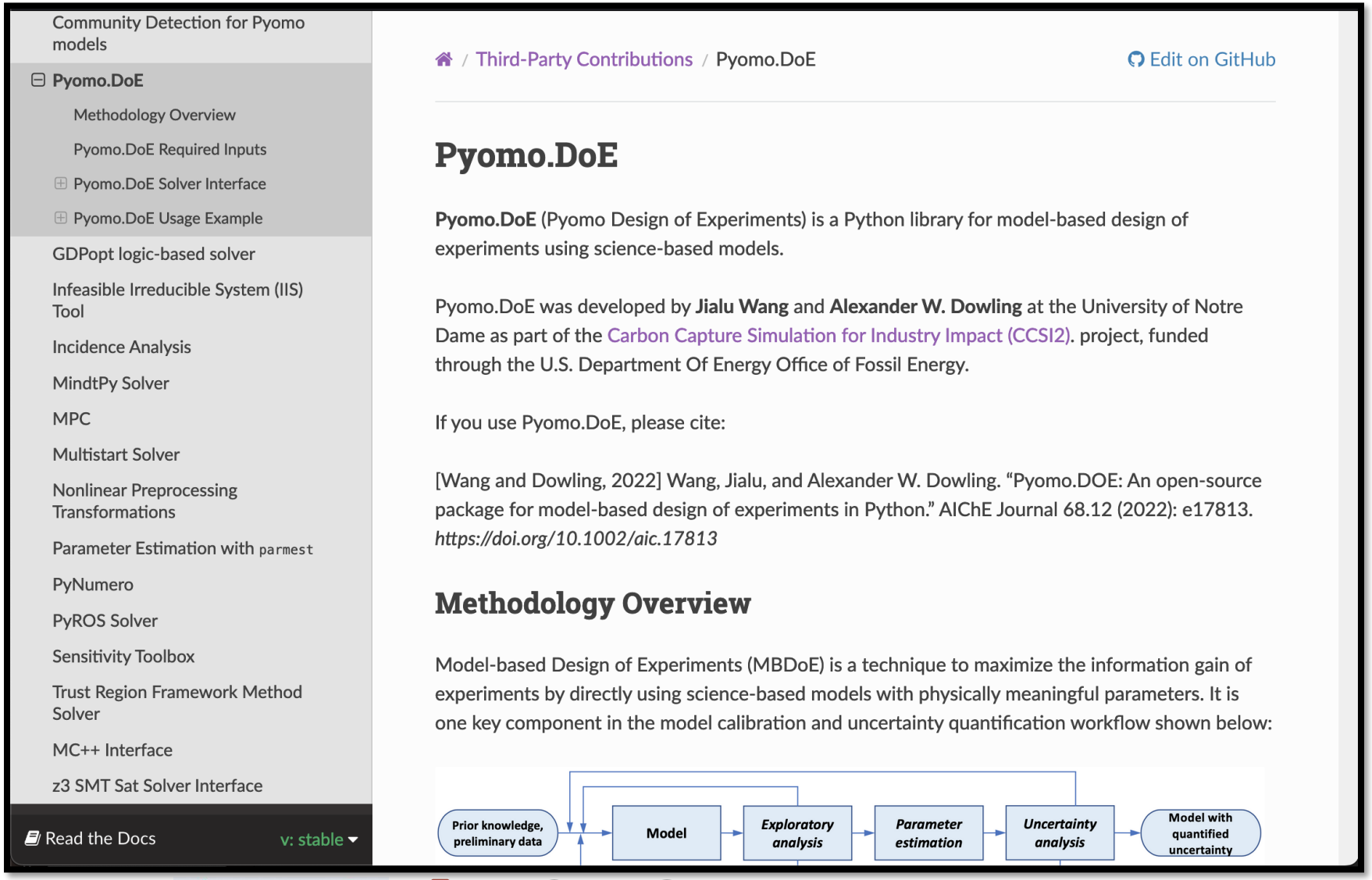
$$T_{S,k}(t_0) = T_{amb,k}; \ \forall k \in E$$



Exp 1 yields information along the coordinate axes of internal parameters (Ub, CpS, CpH). Exp 2 yields information along the axis of ambient dissipation (Ua).

Getting Started with Pyomo.DoE

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



Notebooks at `dowlinglab.github.io/pyomo-doe`

Exercise: Pyomo.DoE

Alexander Dowling, Dan Laky, Shammah Lilonfe, Stephen Cini, Shuvashish Mondal, Shilpa Narasimhan

In this notebook, you will use Pyomo.DoE to compute the A- and D-optimal experiments from the TCLab. In our previous notebook, we used the sine test as a starting point. We will do the same here as well.

Recall, we can computing the next best experiment assuming we already completed one prior experiment. Thus it is important to confirm our optimal experiment design does not change if we change the prior experiment of optimization initial point.

```
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/t
else:
    import os
```

CONTENTS

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).*

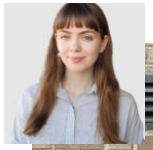
This project was funded by the Department of Energy, National Energy Technology Laboratory an agency of the United States Government, through a support contract. Neither the United States Government nor any agency thereof, nor any of their employees, nor the support contractor, nor any of their employees, 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
- Hailey Lynch



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