Technical Risk Reduction:
Model Based Design of Experiments and Robust Optimization

Chrysanthos Gounaris and Alex Dowling

Jason Sherman, Jialu Wang, Anca Ostace, Douglas Allen, Miguel Zamarripa, Andrew Lee, John Siirola, Joshua Morgan

2023 FECM / NETL
Carbon Management Research Project Review Meeting
August, 31, 2023

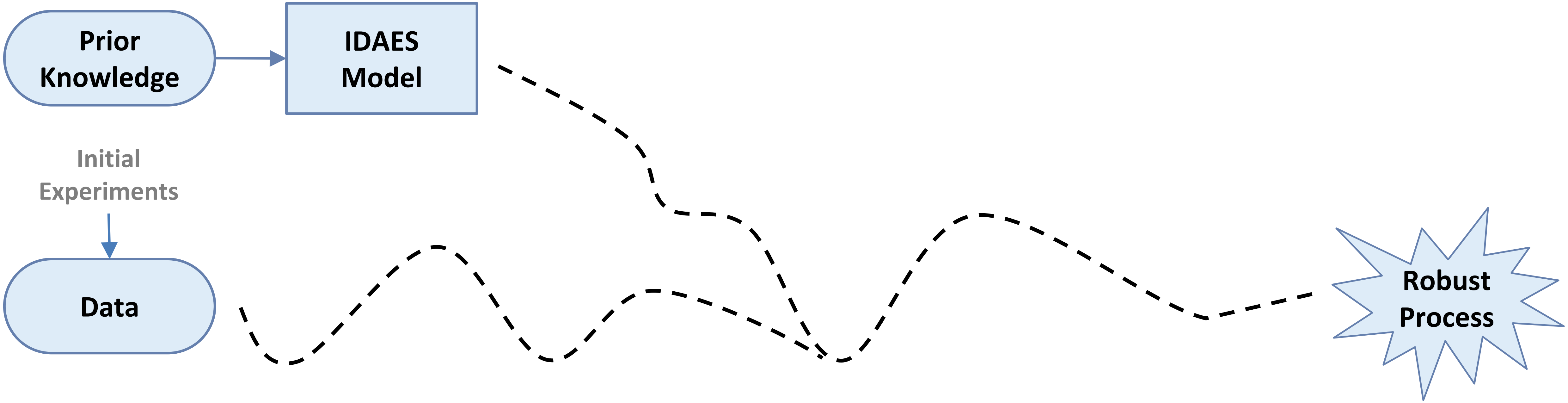
Introduction

- **Design decisions are always subject to technical risk**
  - incomplete “science” (e.g., uncertainty around material properties)
  - use of simplified models (for tractability)
- Practitioners compensate by **over-designing** (often “ad-hoc”)
- Designing plants with **operational flexibility** can help with robustness
- **Need advanced design frameworks** that factor in our “knowledge” of uncertainty
  - Interaction with “knowledge gathering” from pilot tests

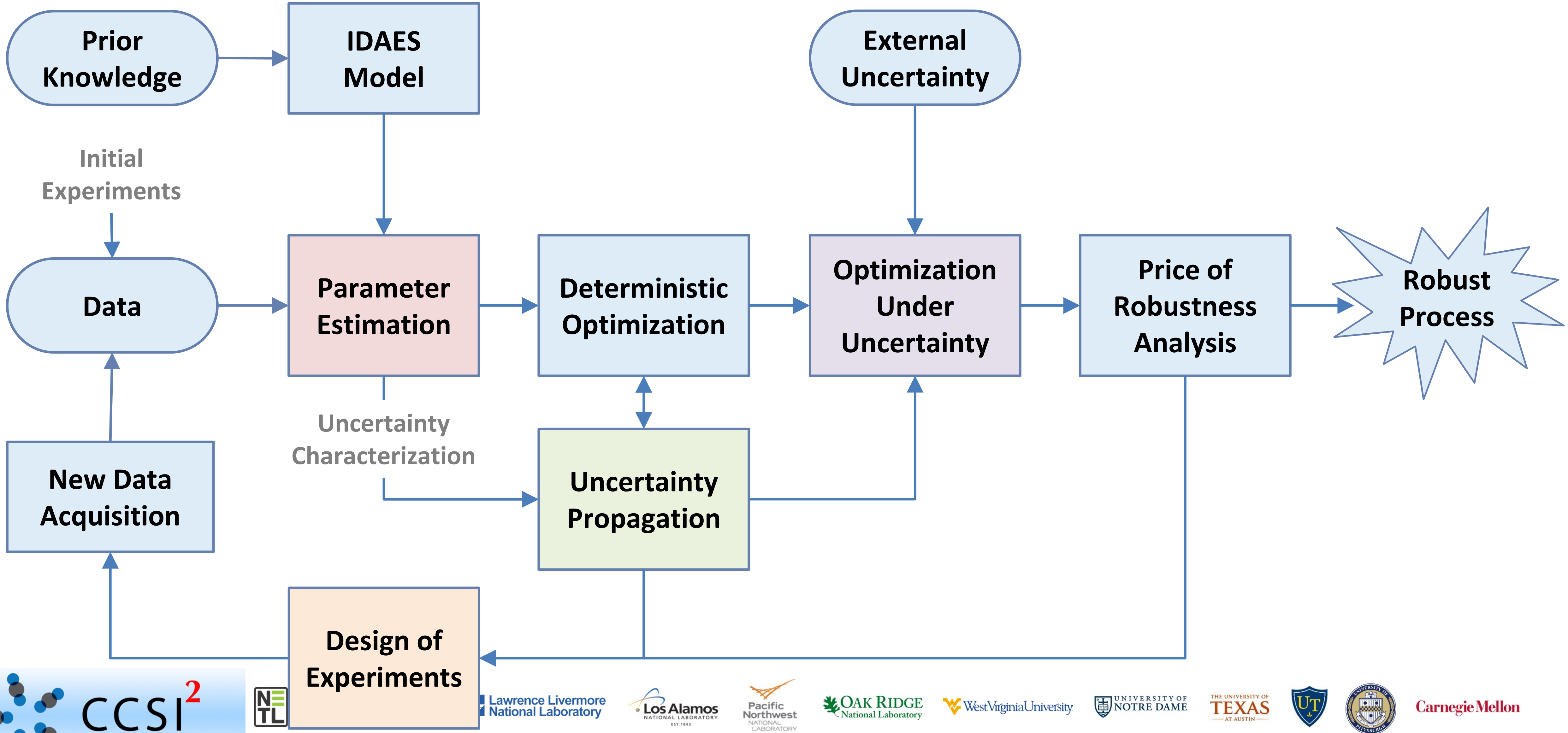
Outline

- Present our computational framework for risk-averse process design
- Cover advances in Robust Nonlinear Optimization (CMU)
- Cover advances in Science-Based Design of Experiments (ND)

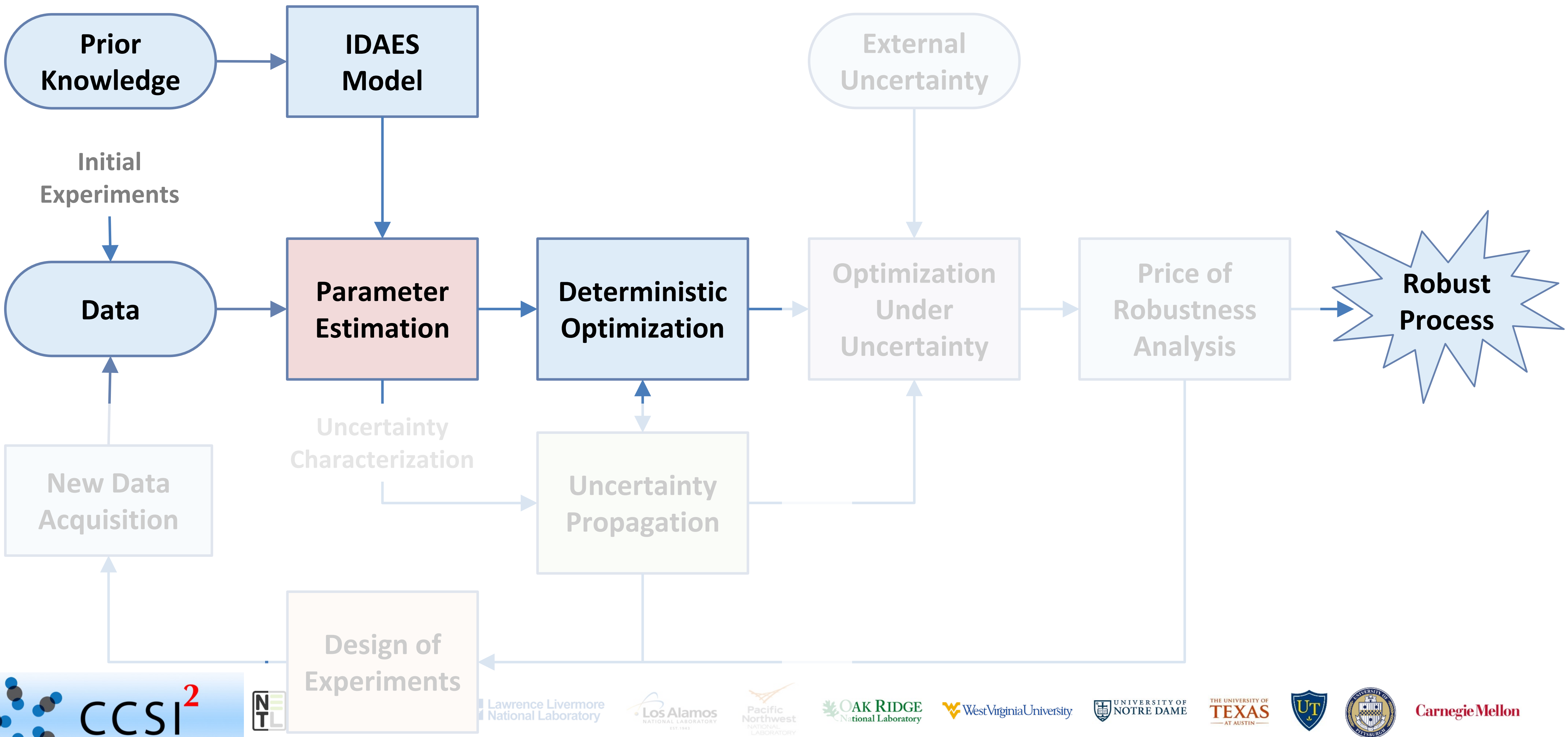
End-to-End Framework for Risk-Averse Process Design



End-to-End Framework for Risk-Averse Process Design



End-to-End Framework for Risk-Averse Process Design



MEA-based CO_2 Absorption Column Model

Degrees of Freedom:

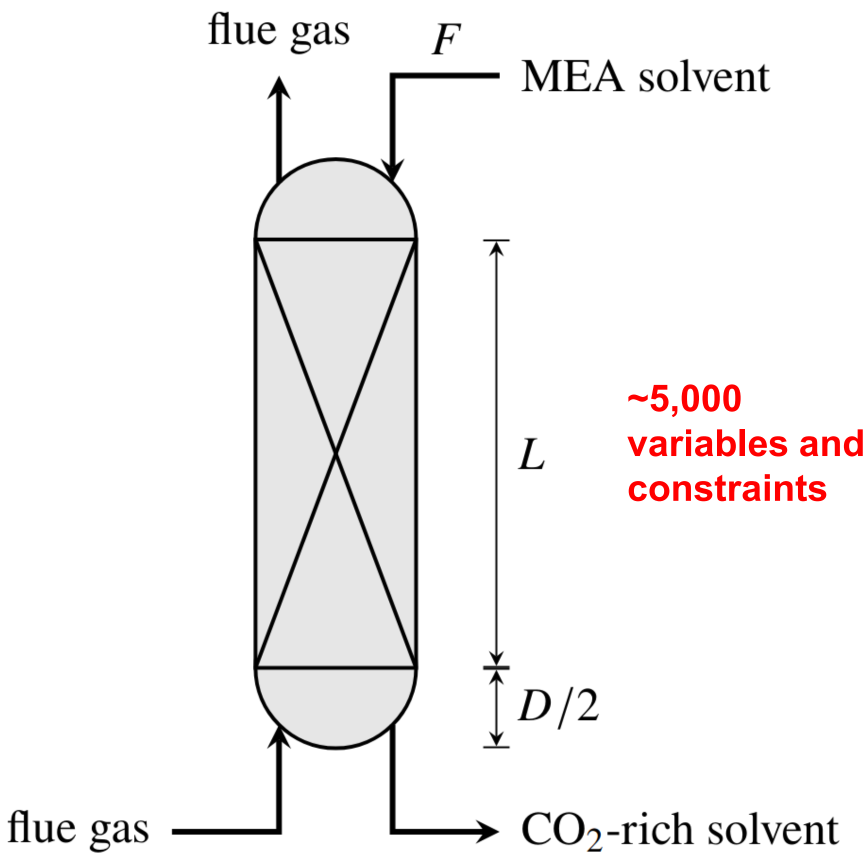
- **Column length (L)**
- **Column diameter (D)**
- **Solvent recirculation rate (F)**
  - adjustable during operation

Minimize:

- **Proxy cost objective** combining column size (CAPEX) and MEA flowrate (OPEX)

Subject to:

- **Process equality constraints**
  - thermodynamic and transport equations
- **Sizing constraints**
  - bounds on the *L*/*D* ratio (1.2–30 used)
- **Performance constraints**
  - CO_2 capture rate requirement
  - Flooding fraction bound constraints (simplified after rigorous analysis)

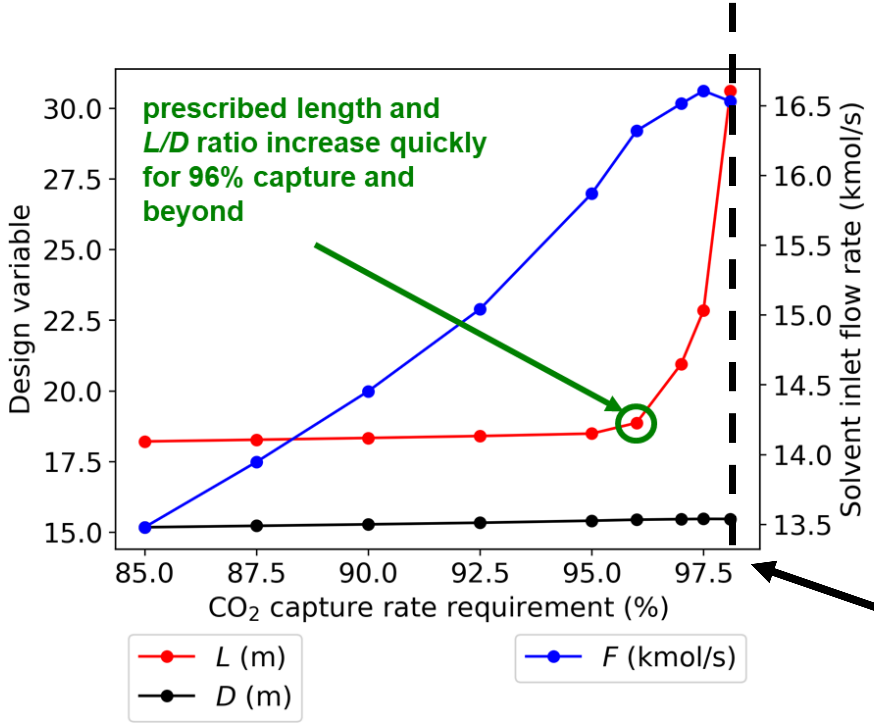


~5,000 variables and constraints

MEA Absorption Column Model – Deterministic Solutions

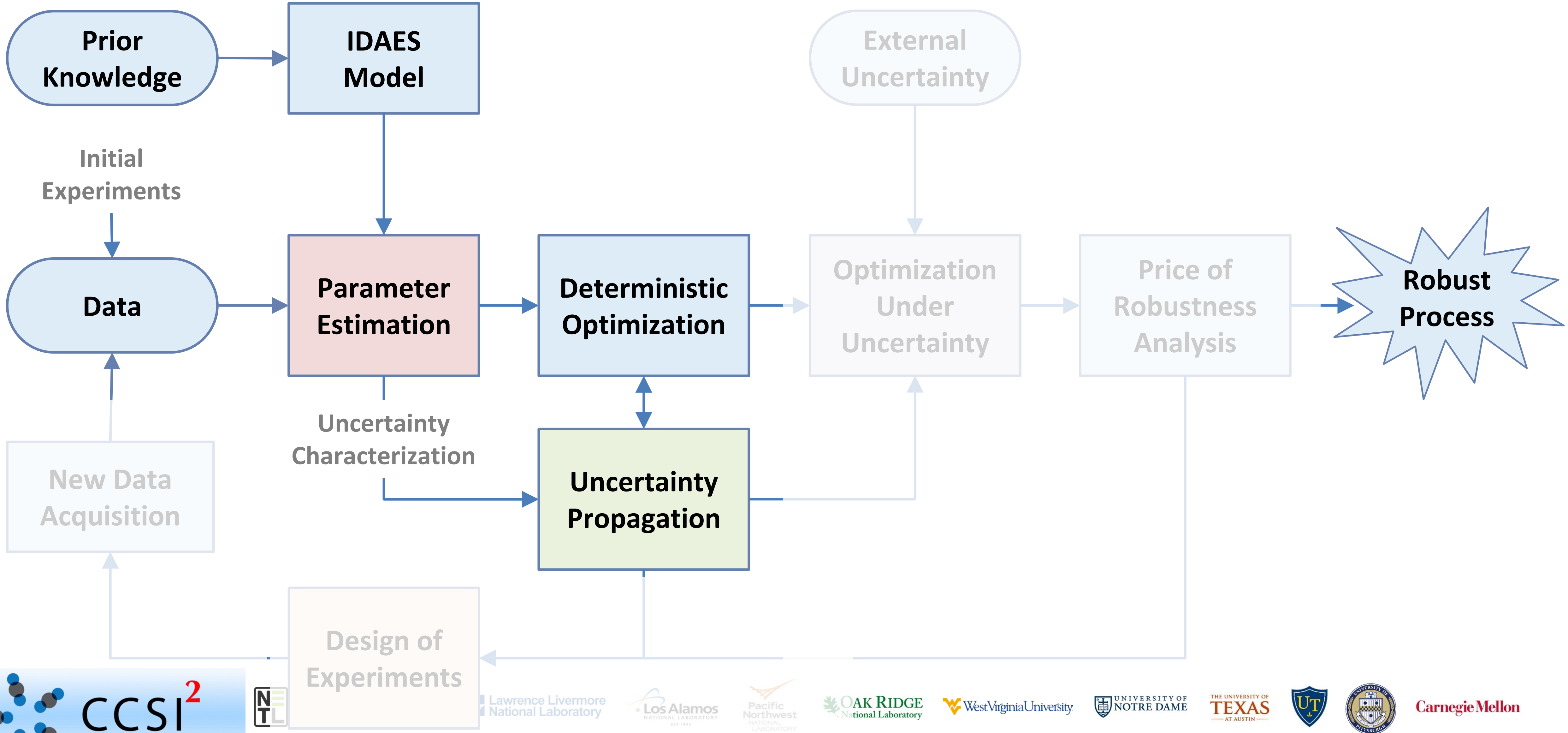
- **Deterministic optimization can be used to obtain minimal cost designs for different levels of capture**

% Capture Requirement	Cost
85.0	13.69
87.5	14.00
90.0	14.32
92.5	14.70
95.0	15.21
97.0	16.06



Model predicts **max possible capture of 98.2%**

End-to-End Framework for Risk-Averse Process Design



Parameter Estimation and UQ in MEA Absorption Property Models

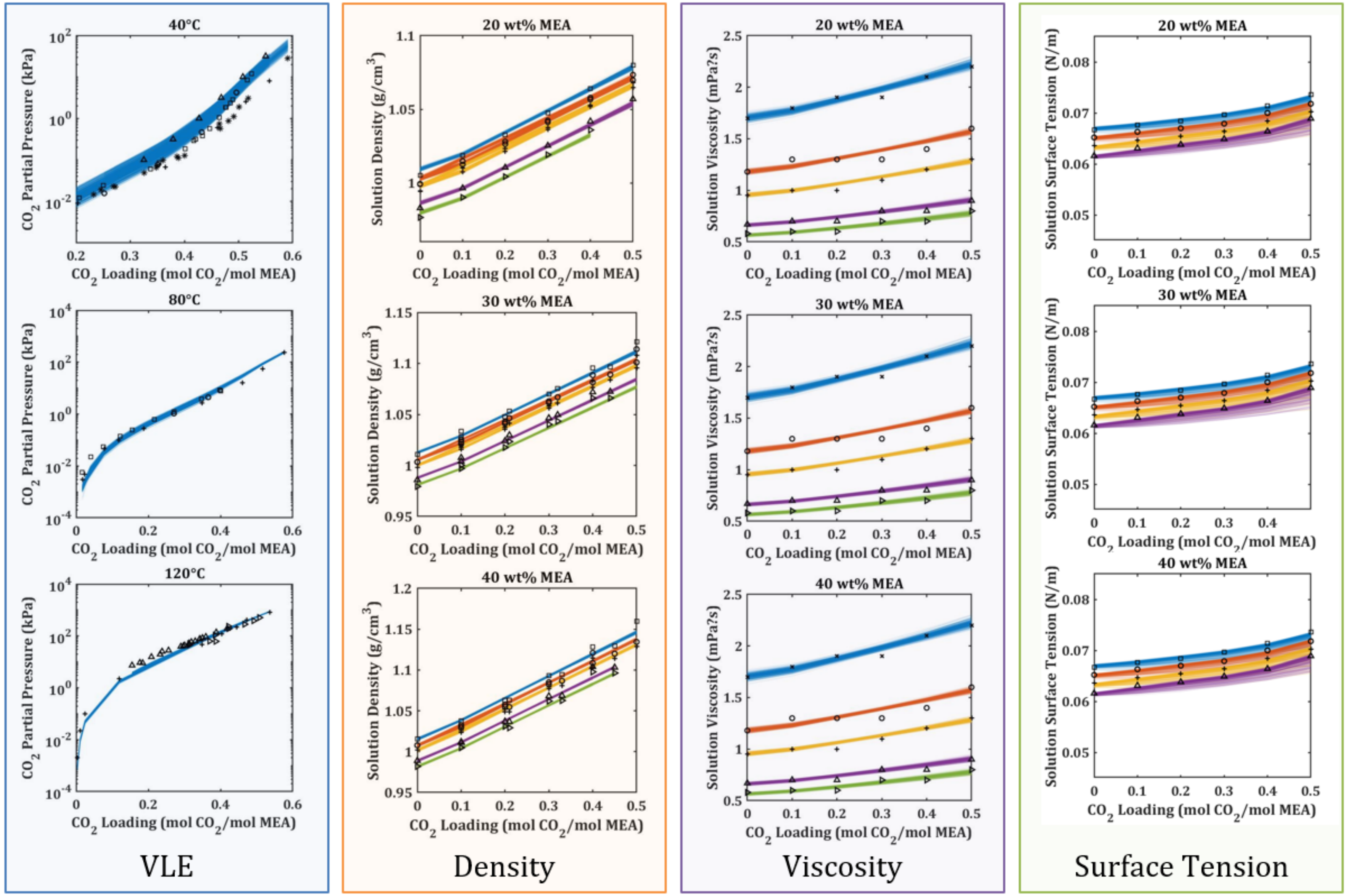
(Anca Ostace, Alex Dowling, Joshua Morgan)

Uncertainty Characterization:

Used *parmest*[1] to identify **point estimates** and **covariances** in:

- vapor-liquid equilibrium,
- solution density,
- viscosity, and
- surface tension

parameters.



Model predictions (200 parameter realizations, transparent lines) vs. experimental data (symbols)

[1] Klise, Nicholson, Staid, Woodruff. *Computer Aided Chemical Engineering*, 47 (2019): 41-46.

Parameter Estimation and UQ in MEA Absorption Property Models

(Anca Ostace, Alex Dowling, Joshua Morgan)

Uncertainty Characterization:

Used *parmest*[1] to identify **point estimates** and **covariances** in:

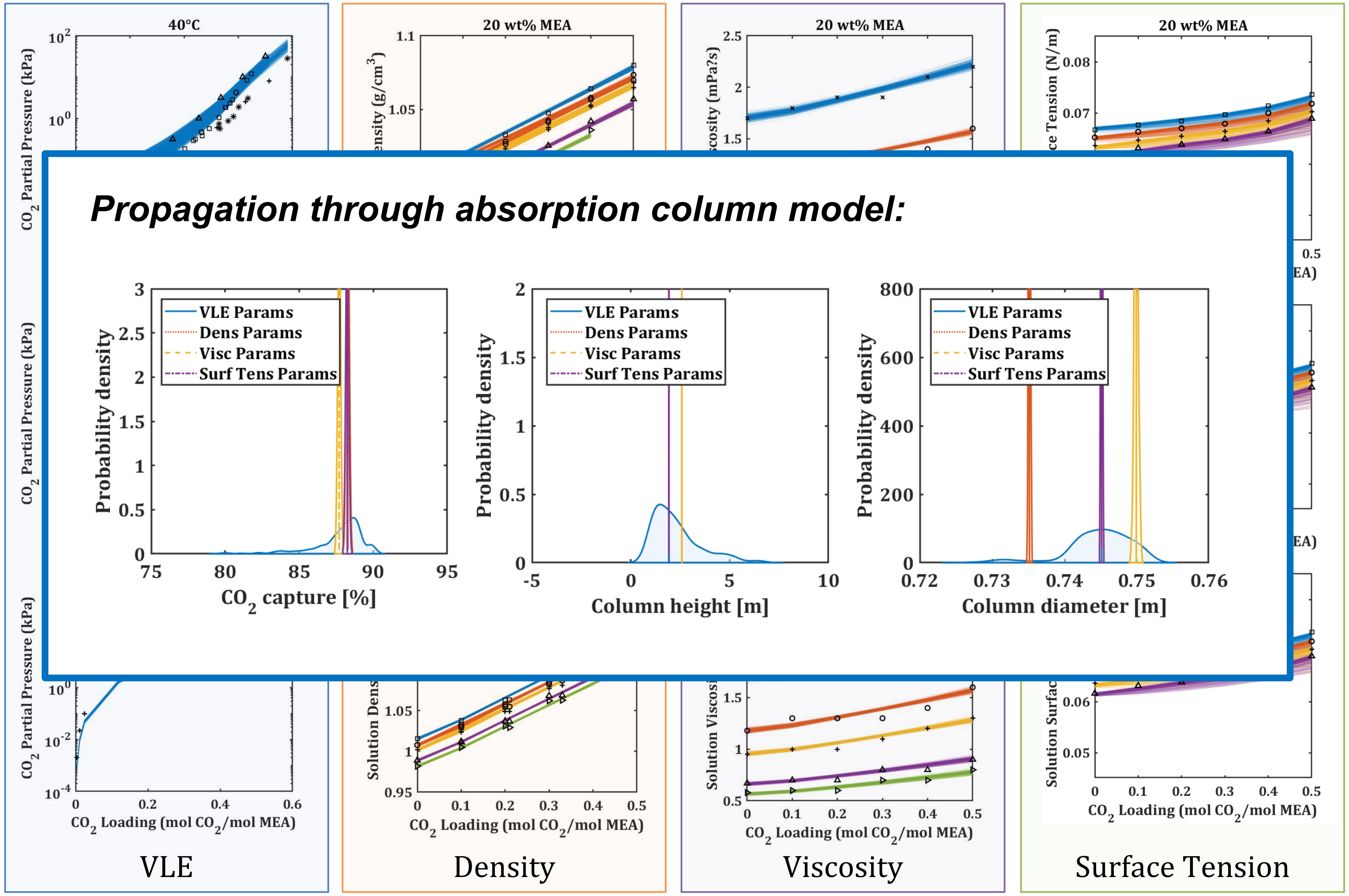
- vapor-liquid equilibrium,
- solution density,
- viscosity, and
- surface tension

parameters.

Main takeaways:

(1) Uncertainty most pronounced in VLE parameters

(2) Prediction from property models is not very sensitive to the remaining parameters



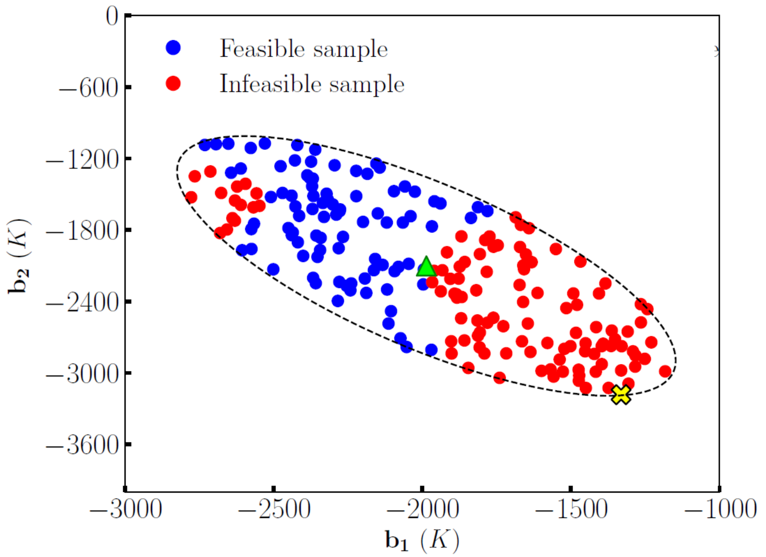
Model predictions (200 parameter realizations, transparent lines) vs. experimental data (symbols)

[1] Klise, Nicholson, Staid, Woodruff. *Computer Aided Chemical Engineering*, 47 (2019): 41-46.

MEA Absorption Column Model –
Robustness of Deterministic Model Solutions

- **Deterministic optimization can be used to obtain minimal cost designs for different levels of capture**
- **Considering only nominal property values leads to non-robust designs**

% Capture Requirement	**Cost**	**# Scenarios Feasible (out of 200)**
85.0	13.69	81
87.5	14.00	75
90.0	14.32	83
92.5	14.70	87
95.0	15.21	82
97.0	16.06	82



Example uncertainty set
(6D set used in this study)

MEA Absorption Column Model – Performance in Light of Altered Capture Targets

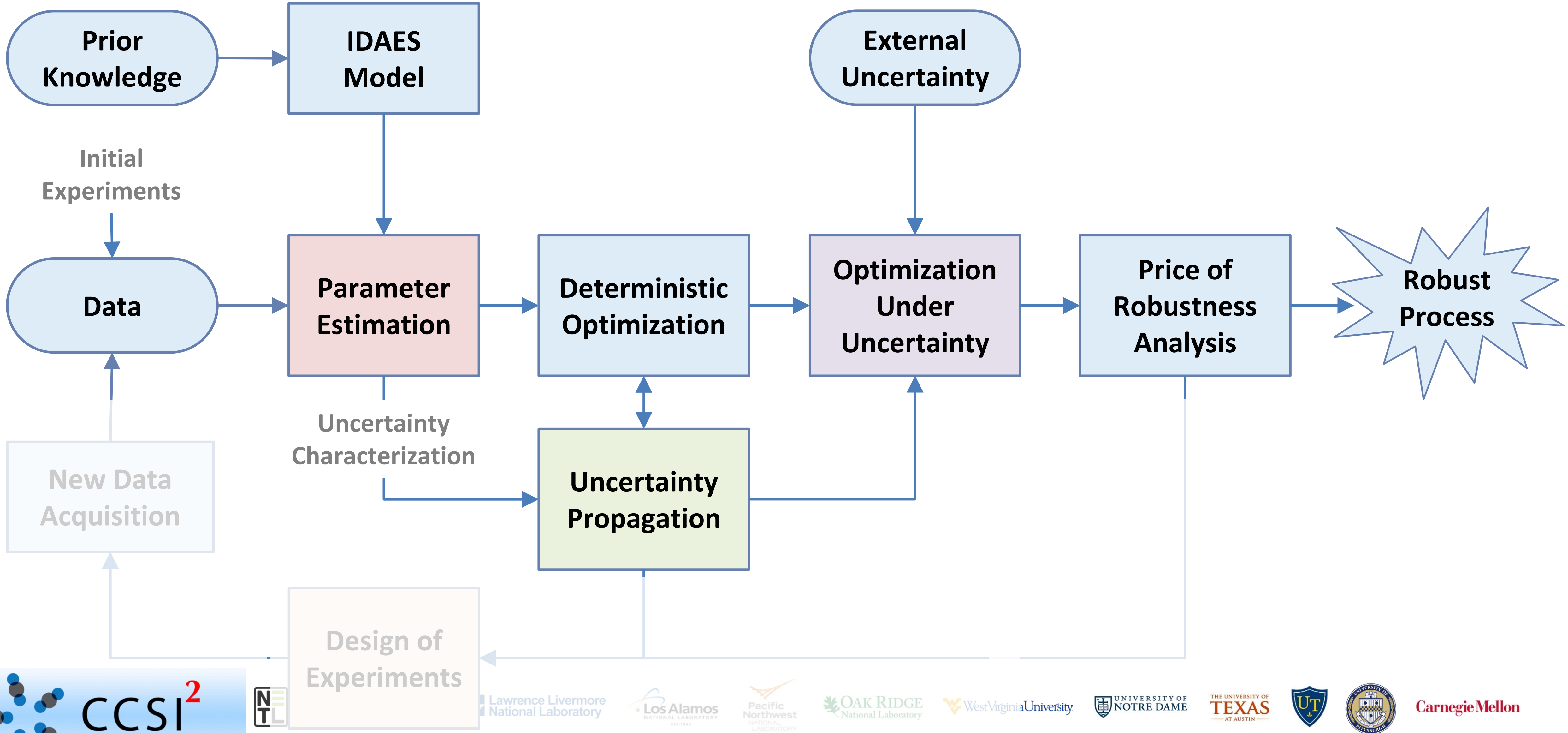
- **Non-robust designs are less likely to adapt to increased capture targets**
- **A significant level of over-design required to establish guarantees**

Nominally optimal for increasing % capture

% Capture Requirement during Design	Cost	# Scenarios Feasible (out of 200) subject to Off-Spec % Capture Requirement					
		85.0	**87.5**	**90.0**	**92.5**	**95.0**	**97.0**
85.0	13.69	81	8	0	0	0	0
87.5	14.00	143	75	10	0	0	0
90.0	14.32	180	136	83	23	0	0
92.5	14.70	189	170	132	87	26	3
95.0	15.21	190	186	166	132	82	45
97.0	16.06	189	185	184	172	127	82

Evaluating robustness for increasing capture rate requirement

End-to-End Framework for Risk-Averse Process Design



Nonlinear Optimization Under Uncertainty

(Jason Sherman, John Siirola, Chrysanthos Gounaris)

Two-Stage Decision-Making Framework

Commit upon **design decisions** (1st-stage DOFs) → Observe **uncertain parameters** (directly or via system response) → Adjust system via **control actions** (2nd-stage DOFs)

Two-Stage Robust Optimization-Based Design Capability

GIVEN

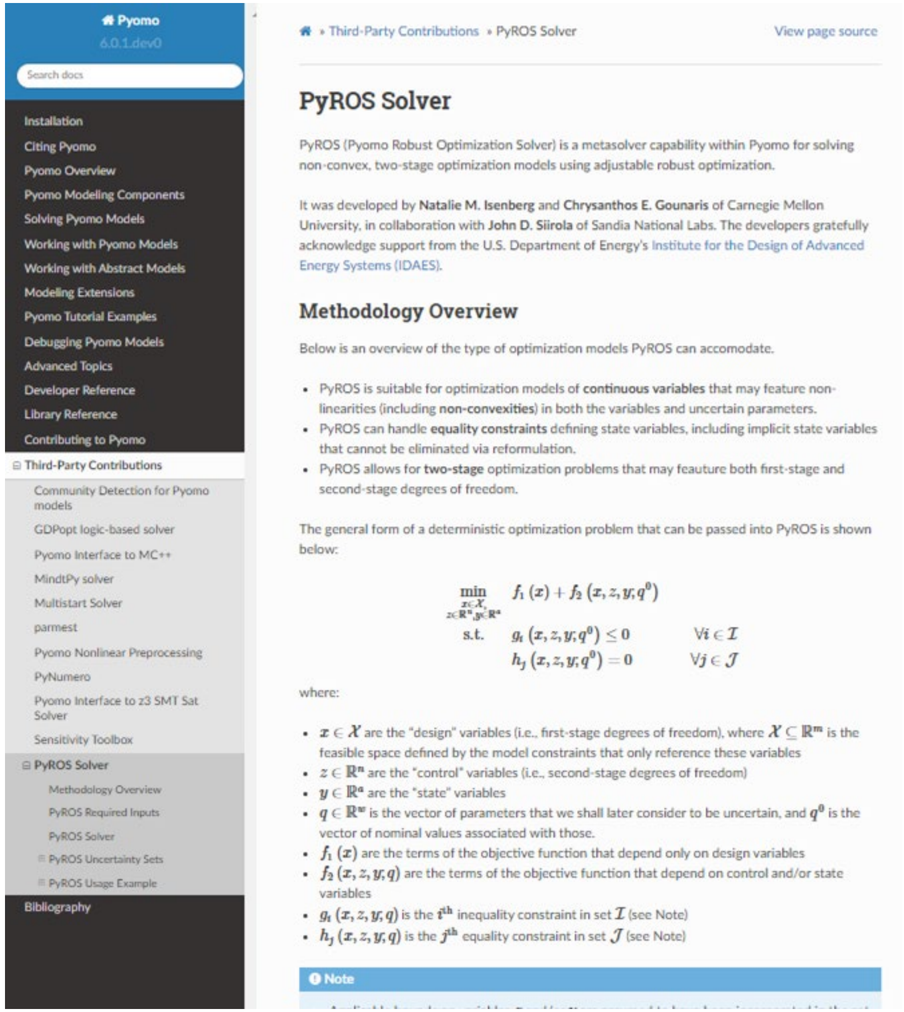
- **Deterministic model**
  - e.g., IDAES flowsheet model
- **Degree-of-freedom partitioning** into 1st-stage and 2nd-stage
- **Quantification of uncertainty** in form of uncertainty set
  - e.g., 95% confidence ellipsoid

DETERMINE

- System **design that is guaranteed to remain feasible** under all scenarios
- Accompanying control policy to perform any **operating adjustments** needed for system to achieve feasibility
- Optimality in light of a **combined economic objective** (CapEx+OpEx)

PyROS: a Pyomo Robust Optimization Solver

https://pyomo.readthedocs.io/en/stable/contributed_packages/pyros.html



Pyomo
6.0.1.dev0

Search docs

- Installation
- Citing Pyomo
- Pyomo Overview
- Pyomo Modeling Components
- Solving Pyomo Models
- Working with Pyomo Models
- Working with Abstract Models
- Modeling Extensions
- Pyomo Tutorial Examples
- Debugging Pyomo Models
- Advanced Topics
- Developer Reference
- Library Reference
- Contributing to Pyomo
- Third-Party Contributions
  - Community Detection for Pyomo models
  - GDPopt logic-based solver
  - Pyomo Interface to MC++
  - MindtPy solver
  - Multistart Solver
  - parmest
  - Pyomo Nonlinear Preprocessing
  - PyNumero
  - Pyomo Interface to z3 SMT Sat Solver
  - Sensitivity Toolbox
  - PyROS Solver
    - Methodology Overview
    - PyROS Required Inputs
    - PyROS Solver
    - PyROS Uncertainty Sets
    - PyROS Usage Example
- Bibliography

» Third-Party Contributions » PyROS Solver — View page source

PyROS Solver

PyROS (Pyomo Robust Optimization Solver) is a metasolver capability within Pyomo for solving non-convex, two-stage optimization models using adjustable robust optimization.

It was developed by **Natalie M. Isenberg** and **Chrysanthos E. Gounaris** of Carnegie Mellon University, in collaboration with **John D. Siirola** of Sandia National Labs. The developers gratefully acknowledge support from the U.S. Department of Energy's Institute for the Design of Advanced Energy Systems (IDAES).

Methodology Overview

Below is an overview of the type of optimization models PyROS can accomodate.

- PyROS is suitable for optimization models of **continuous variables** that may feature non-linearities (including **non-convexities**) in both the variables and uncertain parameters.
- PyROS can handle **equality constraints** defining state variables, including implicit state variables that cannot be eliminated via reformulation.
- PyROS allows for **two-stage** optimization problems that may feauture both first-stage and second-stage degrees of freedom.

The general form of a deterministic optimization problem that can be passed into PyROS is shown below:

$$\begin{aligned}\min_{\substack{x\in\mathcal{X},\\ z\in\mathbb{R}^n, y\in\mathbb{R}^a}} \quad & f_1(x) + f_2(x,z,y;q^0) & \\ \text{s.t.} \quad & g_i(x,z,y;q^0) \le 0 & \forall i \in \mathcal{I} \\ & h_j(x,z,y;q^0) = 0 & \forall j \in \mathcal{J}\end{aligned}$$

where:

- $x \in \mathcal{X}$ are the "design" variables (i.e., first-stage degrees of freedom), where $\mathcal{X} \subseteq \mathbb{R}^m$ is the feasible space defined by the model constraints that only reference these variables
- $z \in \mathbb{R}^n$ are the "control" variables (i.e., second-stage degrees of freedom)
- $y \in \mathbb{R}^a$ are the "state" variables
- $q \in \mathbb{R}^w$ is the vector of parameters that we shall later consider to be uncertain, and q^0 is the vector of nominal values associated with those.
- $f_1(x)$ are the terms of the objective function that depend only on design variables
- $f_2(x,z,y;q)$ are the terms of the objective function that depend on control and/or state variables
- $g_i(x,z,y;q)$ is the i^{th} inequality constraint in set $\mathcal{I}$ (see Note)
- $h_j(x,z,y;q)$ is the j^{th} equality constraint in set $\mathcal{J}$ (see Note)

Note

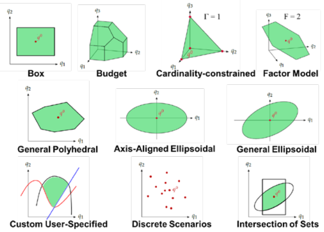
(Step 0) Build a deterministic Pyomo model

(Step 1) Define other required inputs

```
>>> first_stage_variables =[m.x5, m.x6, m.x19, m.x22, m.x23, m.x24, m.x31]
>>> second_stage_variables = [m.x1, m.x2, m.x3, m.x4, m.x20, m.x21]
>>> # === The remaining variables are implicitly designated to be state variables

>>> # === Specify which parameters are uncertain ===
>>> uncertain_parameters = [m.p] # We can pass IndexedParams this way to PyROS, or as a expanded list per

>>> # === Designate local and global NLP solvers. Here we use BARON as both the local and the global NLP s
>>> local_solver = pyo.SolverFactory('baron')
>>> global_solver = pyo.SolverFactory('baron')
```

(Step 2) Construct the uncertainty set

```
>>> # === Define our pertinent data
>>> percent_deviation = 0.15
>>> bounds = [(nominal_values[i] - percent_deviation*nominal_values[i],
              nominal_values[i] + percent_deviation*nominal_values[i])
              for i in range(4)]

>>> # === Construct the Box Set
>>> box_uncertainty_set = pyros.BoxSet(bounds=bounds)
```



(Step 3) Invoke PyROS as a Pyomo Solver

```
>>> # === Make the PyROS solver object ===
>>> pyros_solver = SolverFactory("pyros")

>>> # === Solve the uncertain optimization problem ===
>>> affine_results = pyros_solver.solve(model = m,
                                        first_stage_variables = first_stage_variables,
                                        second_stage_variables = second_stage_variables,
                                        uncertain_params = uncertain_parameters,
                                        uncertainty_set = box_uncertainty_set,
                                        local_solver = local_solver,
                                        global_solver = global_solver,
                                        options = {
                                          "objective_focus": pyros.ObjectiveType.worst_case,
                                          "decision_rule_order": 1,
                                          "solve_master_globally": True
                                        })
```

MEA Absorption Column Model – PyROS Results

- Robust designs are **more expensive** than their deterministic counterparts
- Cost increases **only as necessary for increased feasibility guarantees** (more scenarios factored in)
- Such robust design hierarchies establish an upper limit on the **$ worth spending to reduce uncertainty**

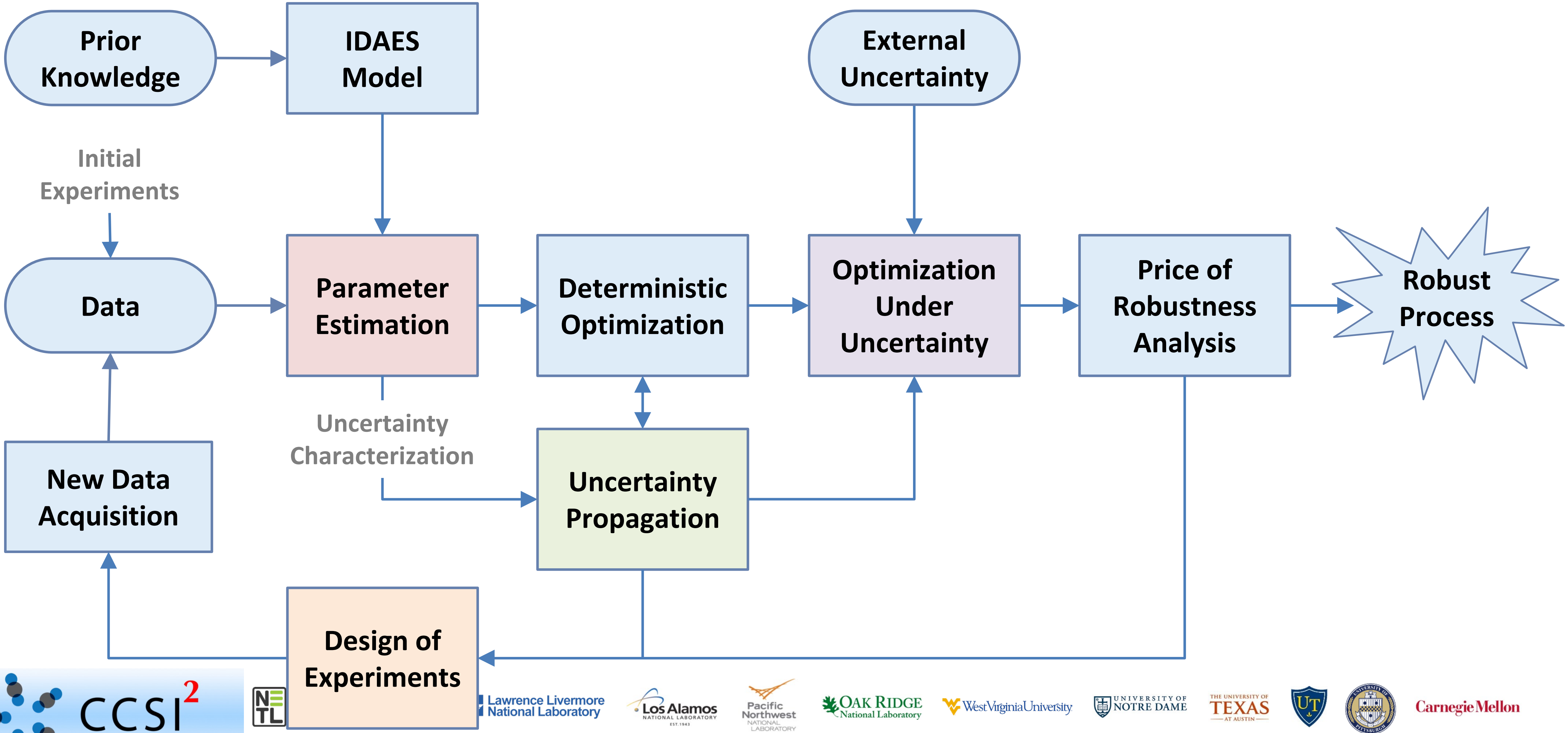
    e.g., shall we do more "science" to improve our property models?

Minimum Capture Rate (%)	Robust Column Proxy Cost and DOF (L, D, F) Values [m, m, kmol/s] for different Confidence Levels			
	0% (deterministic)	**90%**	**95%**	**99%**
90.0	14.32 (18.33, 15.28, 14.45)	17.19 (25.09, 15.51, 17.08)	17.30 (26.43, 15.52, 17.29)	17.75 (29.24, 15.55, 17.76)
92.5	14.70 (18.40, 15.33, 15.04)	18.05 (27.94, 15.54, 17.62)	18.48 (29.50, 15.56, 17.85)	19.40 (32.81, 15.59, 18.33)
95.0	15.21 (18.49, 15.41, 15.87)	19.37 (33.17, 15.58, 18.17)	19.92 (35.33, 15.60, 18.40)	21.14 (40.22, 15.63, 18.87)

In all cases, 4-5 PyROS iterations (~10 min. wall time) required

- *average deterministic solve time ~3 sec.*

End-to-End Framework for Risk-Averse Process Design



Why Sequential DoE?

SDoE: directly incorporate knowledge learned in previous stages
Result: strategic data collection across multiple stages

Feasibility study → Exploration of input space → Model building / refinement → Optimization → Confirmation

Feasibility study	Exploration of input space	Model building / refinement	Optimization	Confirmation
Is it possible to collect quality data? Proof of concept	Understand basic relationship between inputs and responses	Verify that the model captures patterns Add data for better model parameter estimation or prediction	Focus on region of maximum interest Close to most desirable operation conditions	Verify results for production or operational use Ability to duplicate results

SBDoE Determines the Next Best Experiment to Minimize Uncertainty in Estimated Parameters $\hat{\boldsymbol{\theta}}$

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

$$\text{s.t.} \quad \left.\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\, \boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \overline{\boldsymbol{w}}, \hat{\boldsymbol{\theta}}\,) \\ \boldsymbol{g}(\, \boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \overline{\boldsymbol{w}}, \hat{\boldsymbol{\theta}}\,) &= \boldsymbol{0} \\ \boldsymbol{y}(t) &= \boldsymbol{h}(\, \boldsymbol{x}(t), \boldsymbol{z}(t), \hat{\boldsymbol{\theta}}) \end{aligned}\right\} \text{DAE System}$$

$$\left.\begin{aligned} \boldsymbol{f}^{\boldsymbol{0}}\big(\dot{\boldsymbol{x}}(t_0), \boldsymbol{x}(t_0), \boldsymbol{z}(t_0), \boldsymbol{u}(t_0), \overline{\boldsymbol{w}}, \hat{\boldsymbol{\theta}}\big) &= \boldsymbol{0} \\ \boldsymbol{g}^{\boldsymbol{0}}\big(\, \boldsymbol{x}(t_0), \boldsymbol{z}(t_0), \boldsymbol{u}(t_0), \overline{\boldsymbol{w}}, \hat{\boldsymbol{\theta}}\big) &= \boldsymbol{0} \\ \boldsymbol{y}^{\boldsymbol{0}}(t_0) &= \boldsymbol{h}(\, \boldsymbol{x}(t_0), \boldsymbol{z}(t_0), \hat{\boldsymbol{\theta}}) \end{aligned}\right\} \text{Initial Conditions}$$

$\boldsymbol{y}$	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
$\overline{\boldsymbol{w}}$	Time-invariant control variable

Fisher information matrix (FIM):

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

SBDoE Decisions:

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

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

Pyomo.DoE Enables Non-Experts to Use SBDoE

`create_model`

Create Pyomo model for DAE
Compatible with parmest

`DesignVariables`

Specify the SBDoE degrees of freedom and their bounds

`MeasurementVariables`

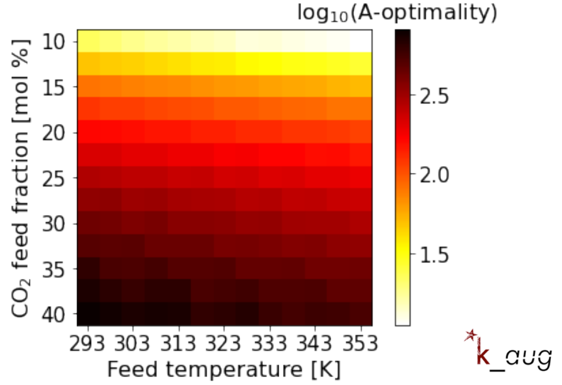
Specify the SBDoE measurement variables and observation error covariance matrix

`DesignOfExperiment`

PYOMO DOE

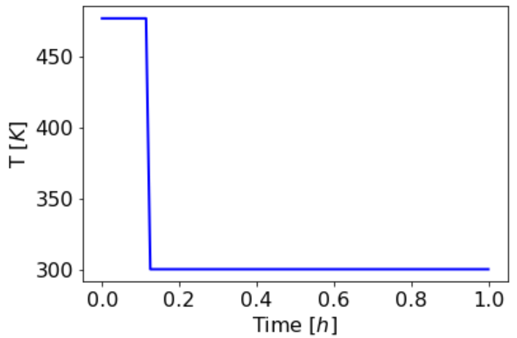
`compute_FIM`

Fast exploratory analysis

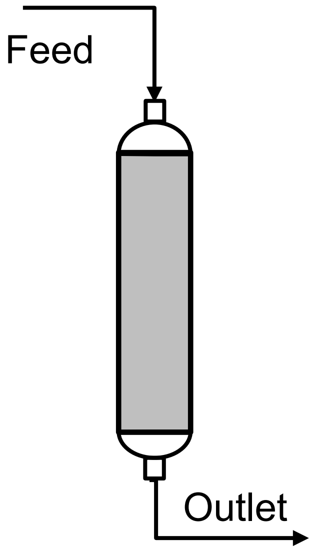


`stochastic_program`

Dynamic optimization

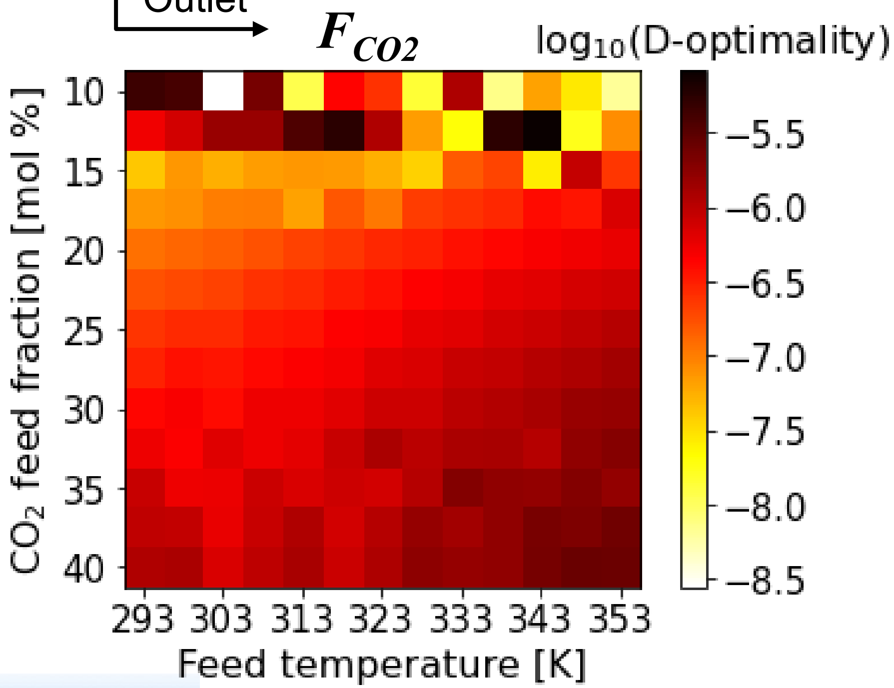


Optimize Fixed Bed Breakthrough Experiments to Characterize CO_2 Capture Sorbent dmpn-Mg_2(dobpdc)



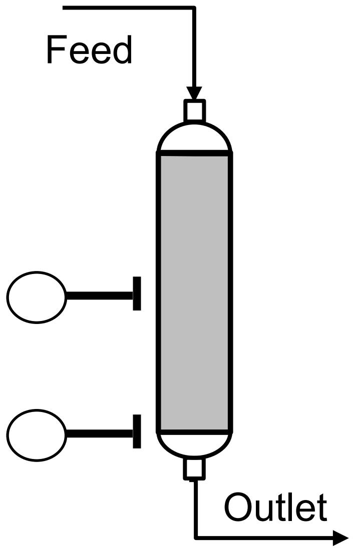
Measurements:

- F_{CO2}: CO_2 outlet flowrate, [mol/s]



Wang, J., and A.W. Dowling. "Pyomo. DOE: An open-source package for model-based design of experiments in Python." *AIChE J.* (2022)

Optimize Fixed Bed Breakthrough Experiments to Characterize CO_2 Capture Sorbent dmpn-Mg_2(dobpdc)

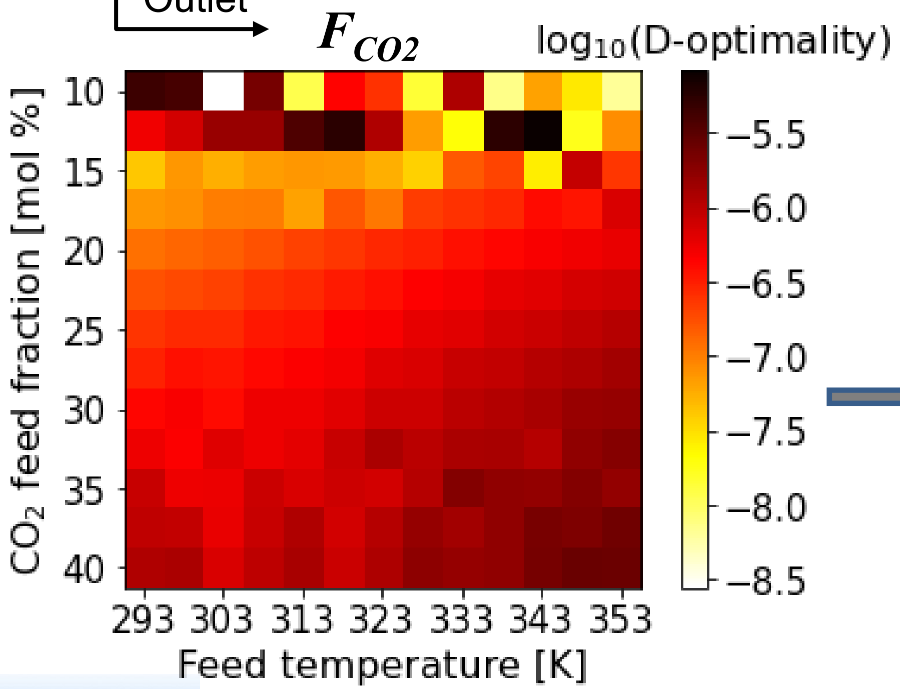


Measurements:

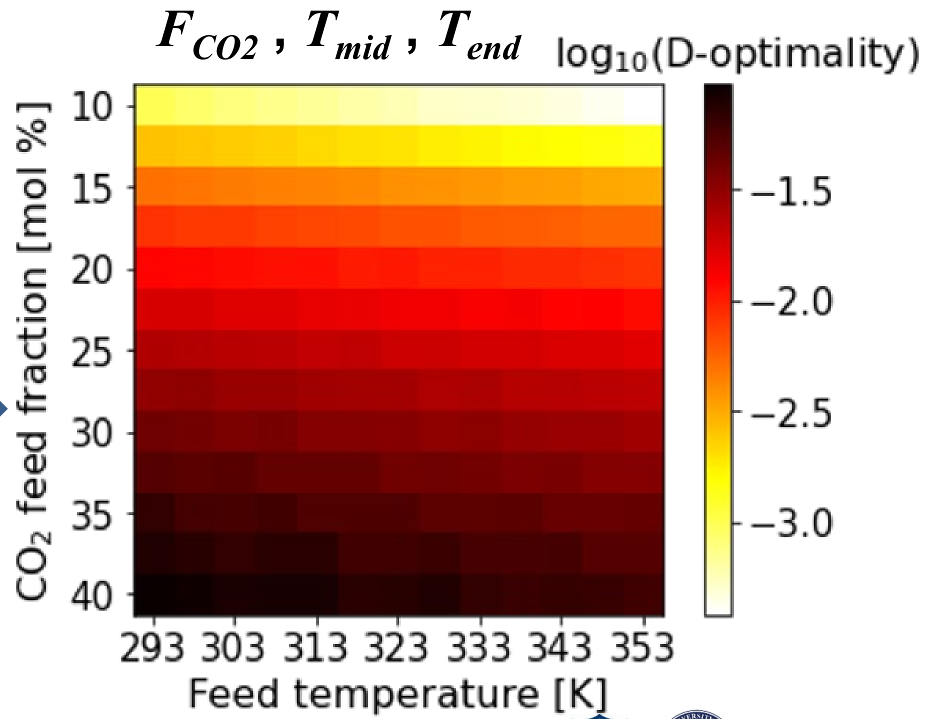
- F_{CO2}: CO_2 outlet flowrate, [mol/s]
- T_{mid}: Temperature in the middle part of the bed, [K]
- T_{end}: Temperature in the outlet end bed, [K]

Insight: Model is not identifiable with original configuration – measure outlet flowrate

Recommendation: Add two temperature measurements along the bed



Add two ***T*** measurements



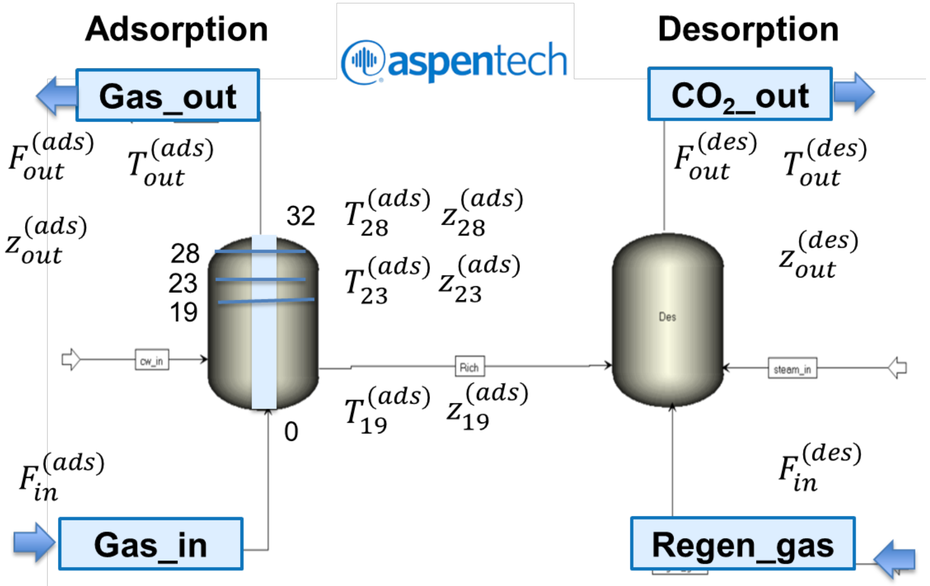
CCSI² Carbon Capture Simulation for Industry Impact

NATIONAL ENERGY TECHNOLOGY LABORATORY | Berkeley Lab | Lawrence Livermore National Laboratory | Los Alamos National Laboratory | Pacific Northwest National Laboratory | Oak Ridge National Laboratory | West Virginia University | University of Notre Dame | The University of Texas at Austin | UT | University of Pittsburgh | Carnegie Mellon

Wang, J., and A.W. Dowling. "Pyomo. DOE: An open-source package for model-based design of experiments in Python." *AIChE J.* (2022)

Measurement Optimization

- Often decide on measurements (instrumentation) months to years before any experiments are conducted
- How to rank measurements from most to least informative?



Dynamic Rotary Bed System

14 possible measurements:
- Flowrates (4)
- Temperatures (4)
- Compositions (5)

Need to balance measurement value (information) and cost

Science-Based Design of Experiments (SBDoE)

Objective Maximize measure of Fisher information matrix

Decisions Experimental conditions: control variables, initial states, number of experiments

Measurement Optimization (MO)

Objective Maximize measure of Fisher information matrix

Decisions

Decisions
Measurements:
What, *Where*, and *When*

Key Takeaways

- **End-to-end framework for risk-averse process design** enables decision-makers **avoid unnecessary over-designs** and **shorten development cycles**
  - leverages the **Pyomo ecosystem** of computational capabilities

- The **PyROS tool for two-stage robust nonlinear optimization** can be invoked to **determine designs with quantifiable insurances of feasibility** and performance guarantees
  - establishes **Pareto fronts between cost and risk** of designs

- The **Pyomo.DoE tool brings science-based design of experiments** capabilities to **support pilot test campaign efforts**
  - recent emphasis on **measurement optimization**

Acknowledgements

- Douglas Allen, NETL
- Andrew Lee, NETL
- Joshua Morgan, NETL
- Anca Ostace, NETL
- Jason Sherman, CMU
- John Siirola, SNL
- Jialu Wang, UND
- Miguel Zamarripa, NETL

We graciously acknowledge funding from the U.S. Department of Energy, Office of Fossil Energy and Carbon Management, through the Point Source Carbon Capture Program

CCSI²

Carbon Capture Simulation for Industry Impact

For more information

Michael.matuszewski@netl.doe.gov