MATLAB EXPO

다양한 엔지니어링 최적화 문제를 해결하는 효과적인 방법

최규웅 부장, 매스웍스코리아

MathWorks®

Agenda

Solving engineering optimization problems

- Parameter estimation with experiment data
  - Sensitivity Analyzer / Global Optimization Toolbox
- Optimal calibration with experiment data
  - Model-Based Calibration Toolbox
- Conclusions

Parameter estimation with experiment data

Problem statement

- **Goal:**
  To introduce an optimization workflow using “**Global Optimization Toolbox**” in physical modeling

  – An evaporator modeling example based on experiment data including

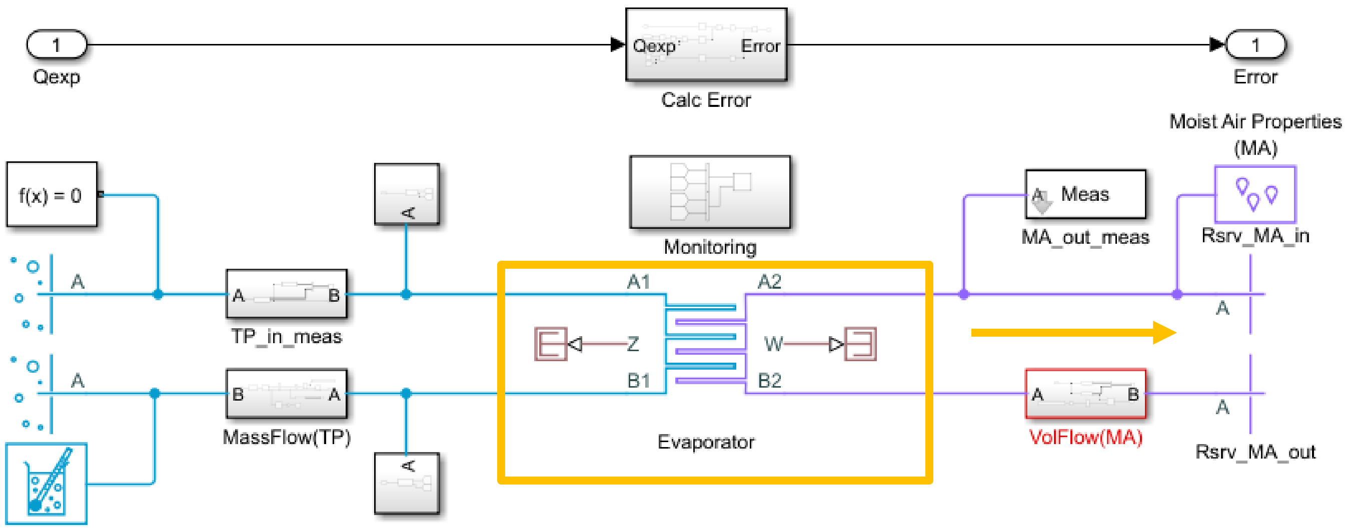
  – How to use **“Sensitivity Analyzer”** for more efficient parameter estimation

  – How to use **“ga”** solver of Global Optimization Toolbox

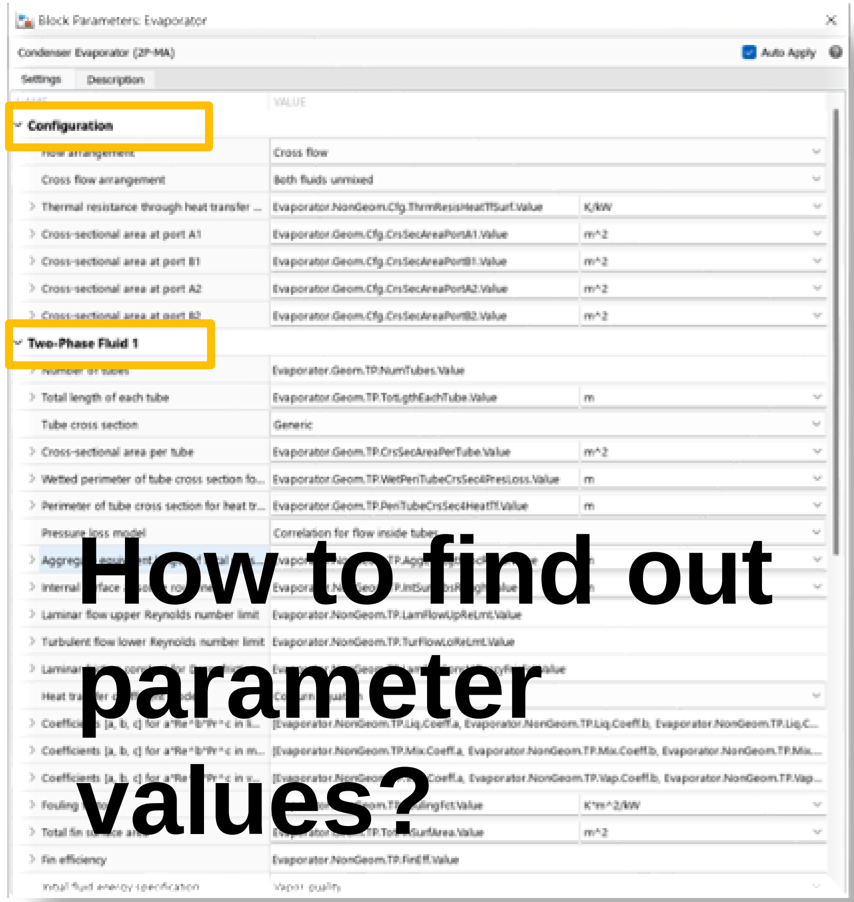
Parameter estimation with experiment data

Plant model review

- And evaporator model



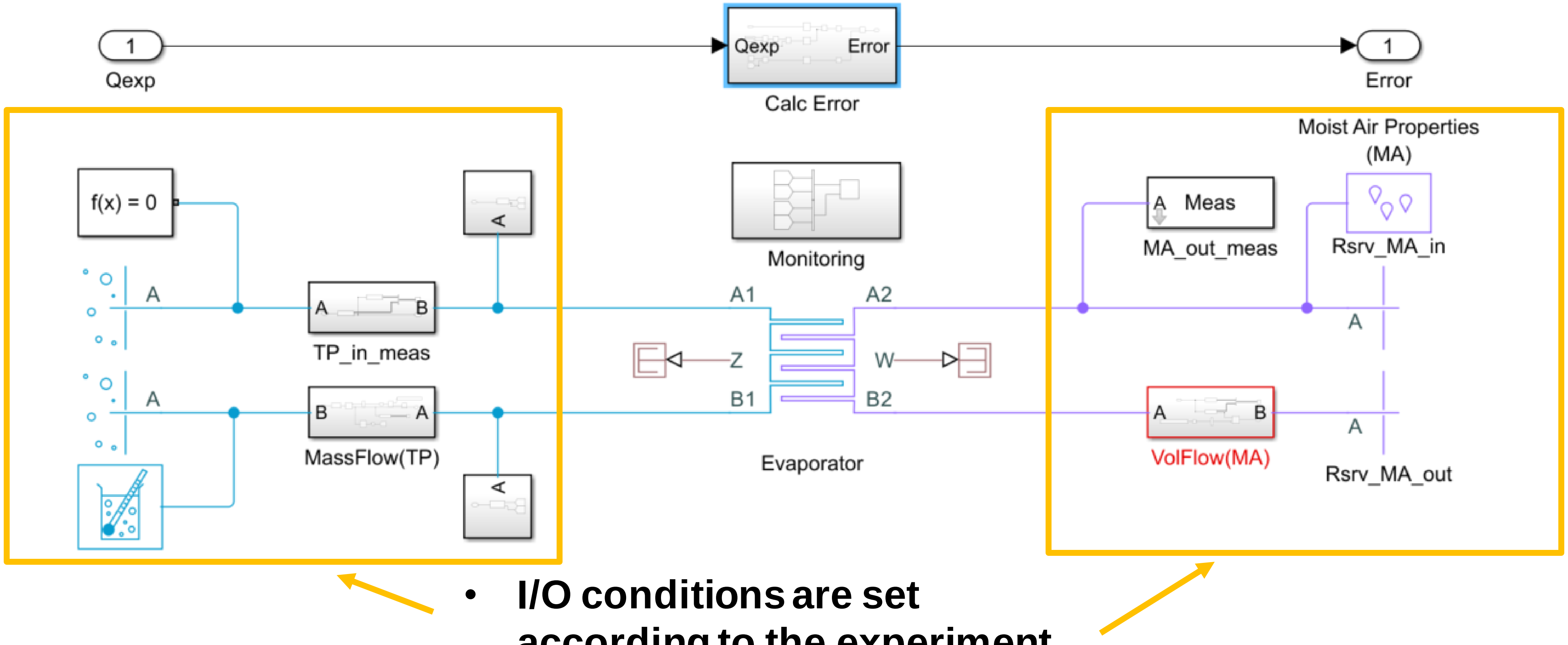
- **12 heat transfer equation coefficients**
- **5 pressure loss params**



How to find out parameter values?

Parameter estimation with experiment data

Plant model review

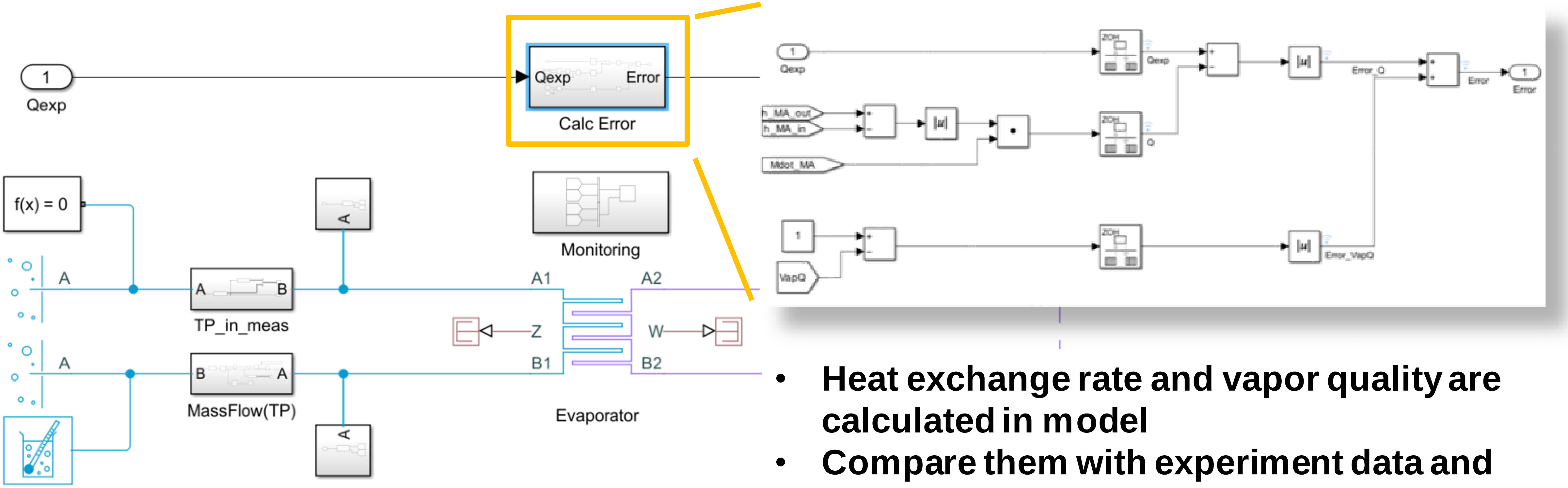


- **I/O conditions are set according to the experiment data**

Parameter estimation with experiment data

Plant model review

- Calculation error is designed for optimization objective function

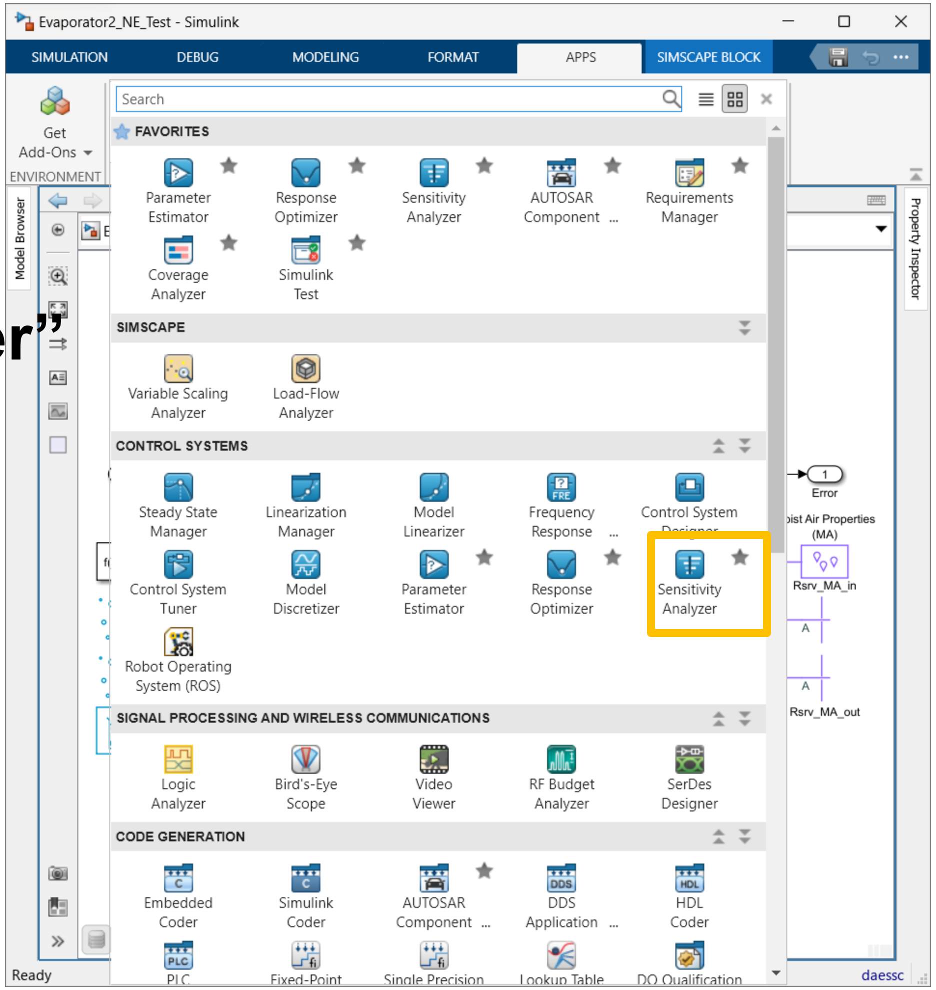


- **Heat exchange rate and vapor quality are calculated in model**
- **Compare them with experiment data and errors can be used directly in objective function**

Parameter estimation with experiment data

Sensitivity Analyzer

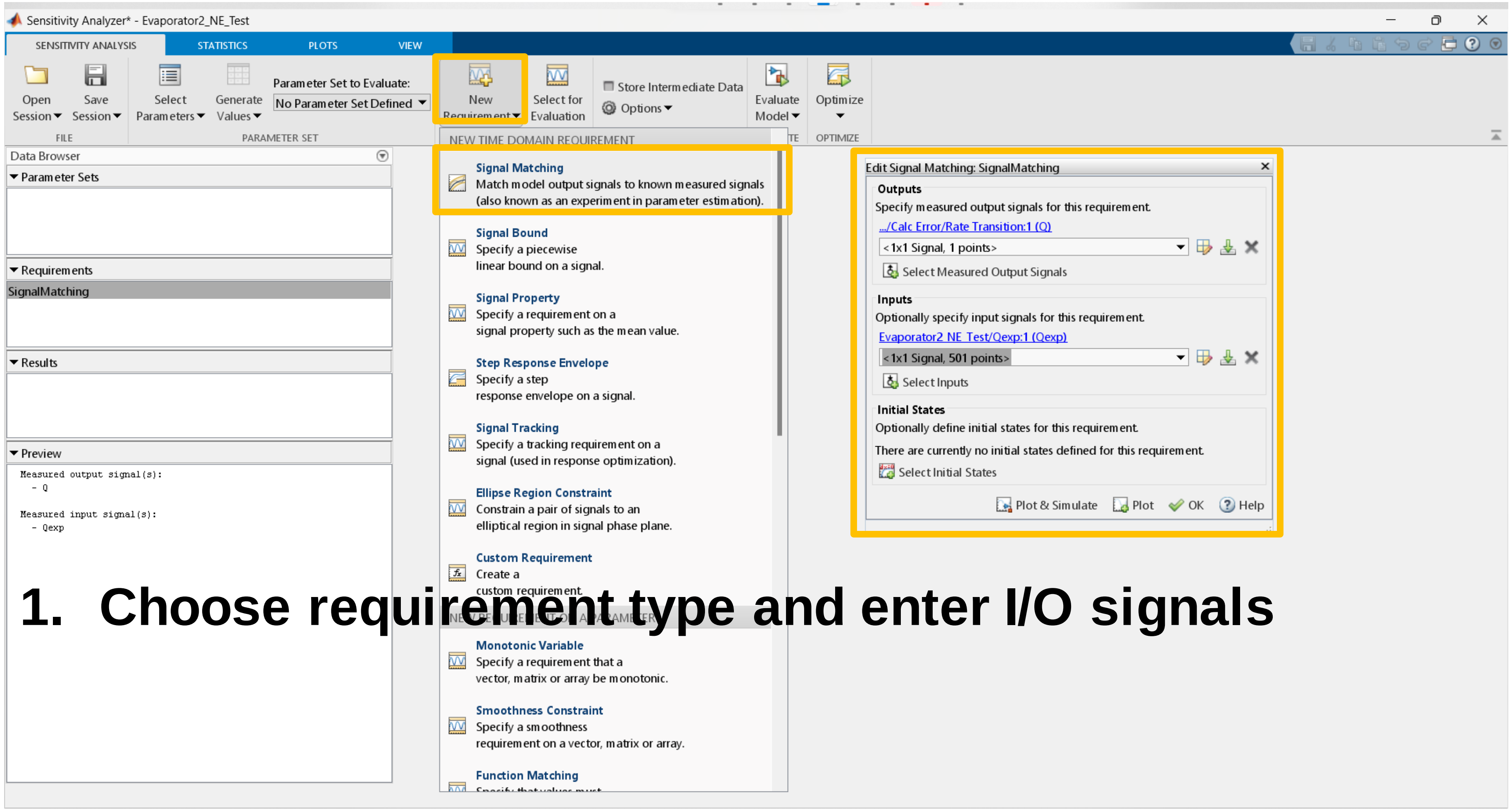
- Q: What parameters are to be tuned?
- A: The most impactive ones
- Q: What are the most impactive ones?
- A: If you don’t know, try **“Sensitivity Analyzer”** from Simulink Apps



- **“Sensitivity Analyzer”**
  - Explore design space and [determine most influential model parameters]

Parameter estimation with experiment data

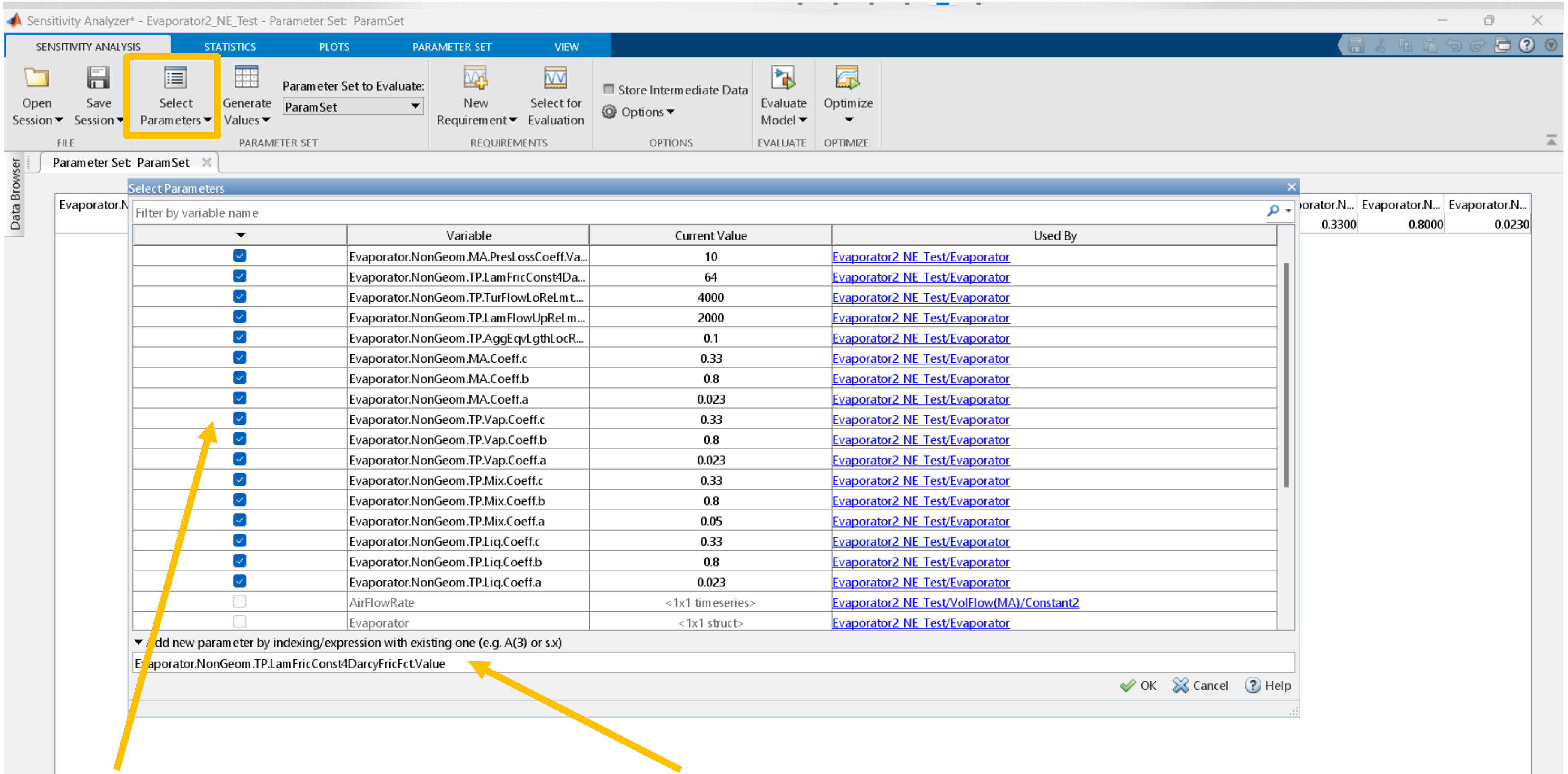
Sensitivity Analyzer



1. Choose requirement type and enter I/O signals

Parameter estimation with experiment data

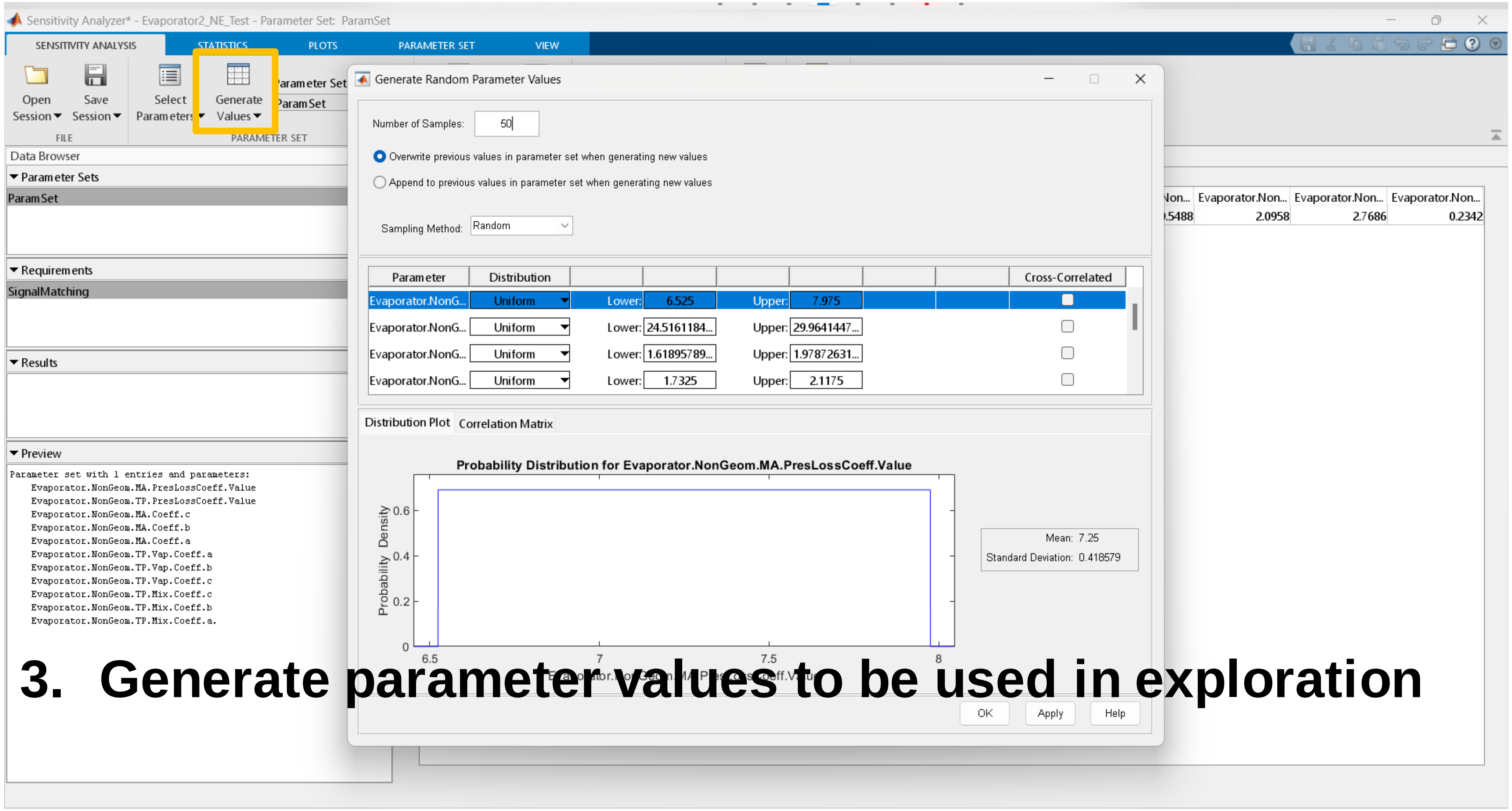
Sensitivity Analyzer



2. Choose parameters or enter parameter names

Parameter estimation with experiment data

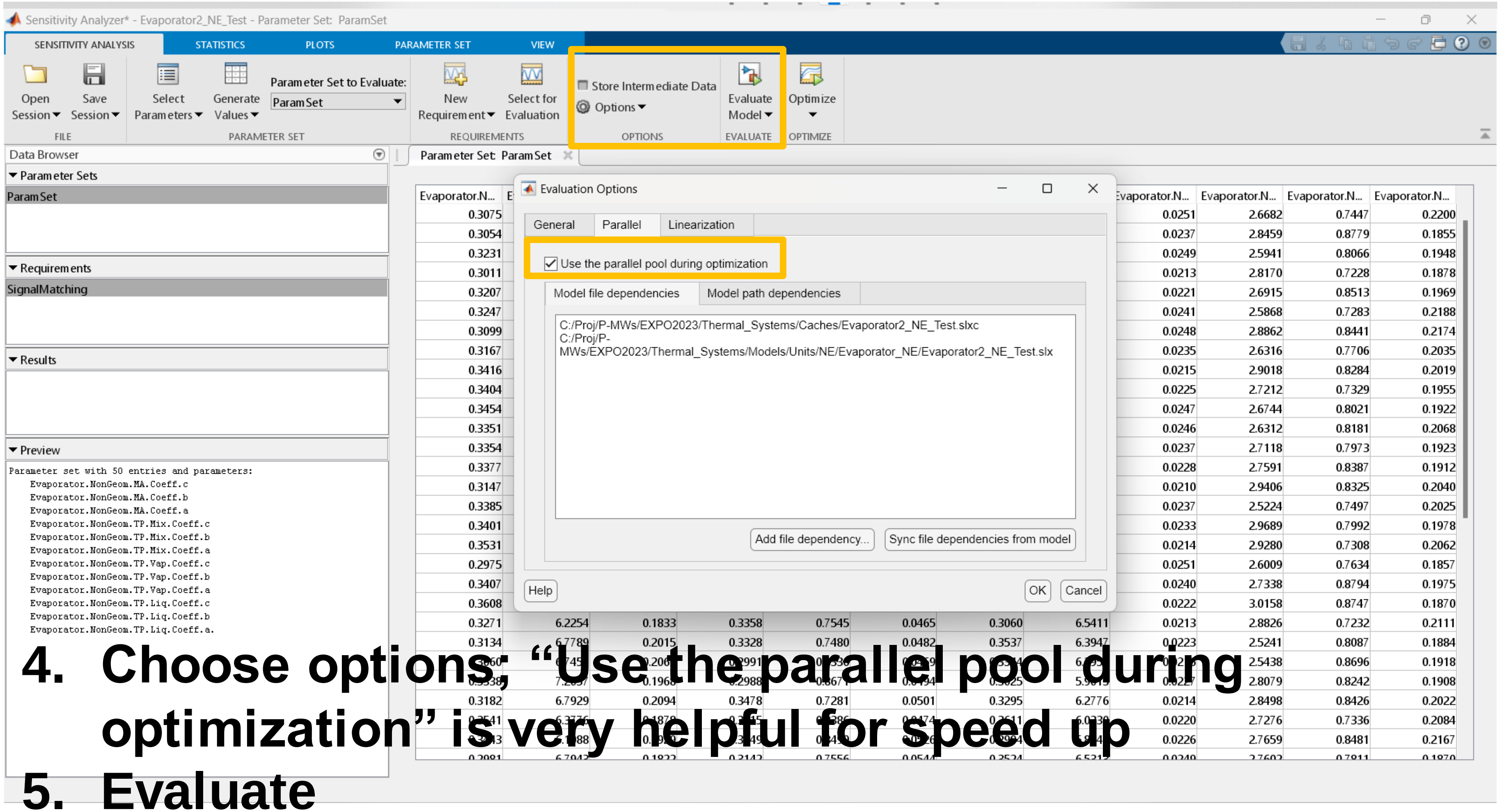
Sensitivity Analyzer



3. Generate parameter values to be used in exploration

Parameter estimation with experiment data

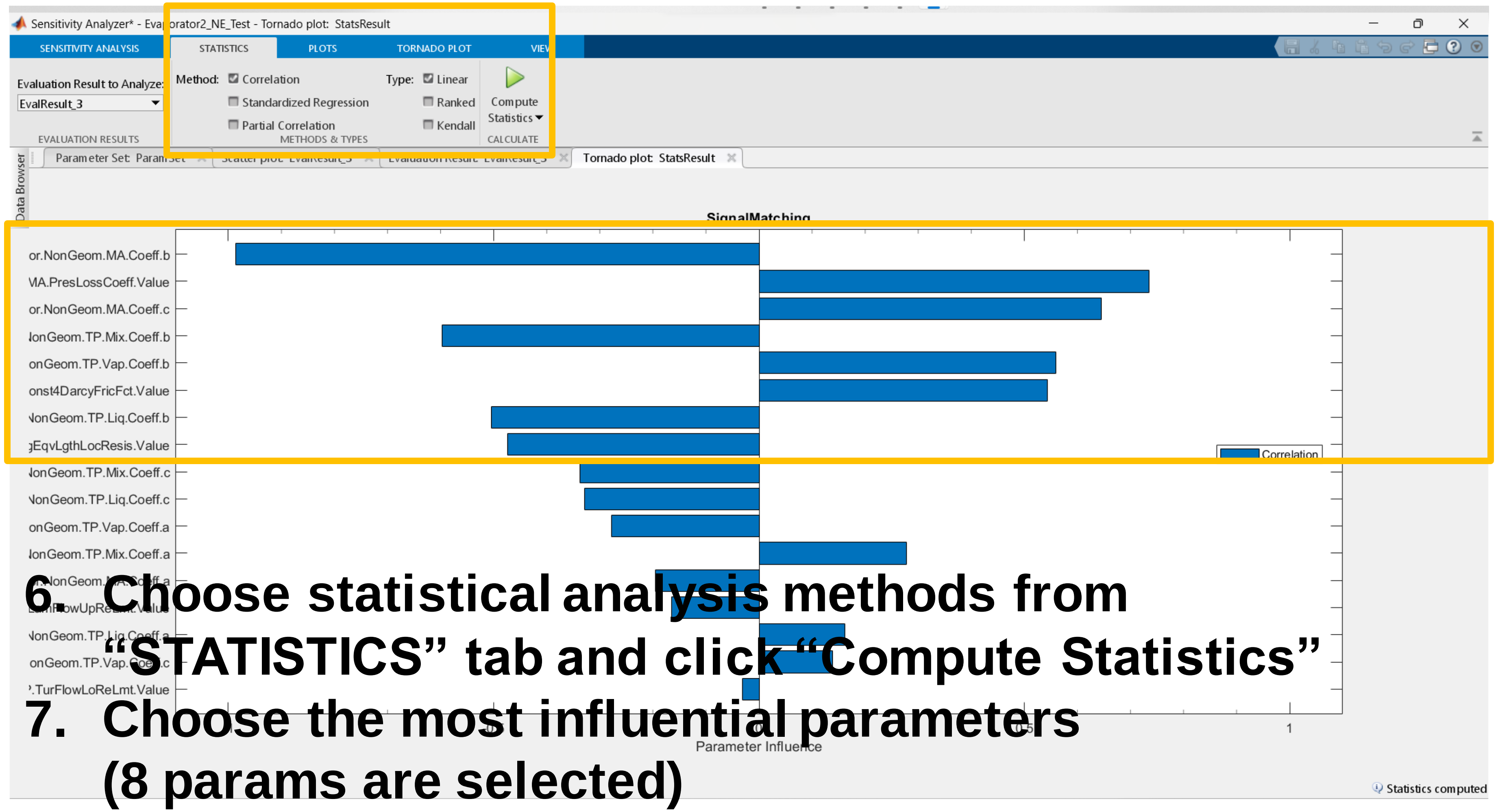
Sensitivity Analyzer



4. **Choose options: "Use the parallel pool during optimization" is very helpful for speed up**
5. **Evaluate**

Parameter estimation with experiment data

Sensitivity Analyzer



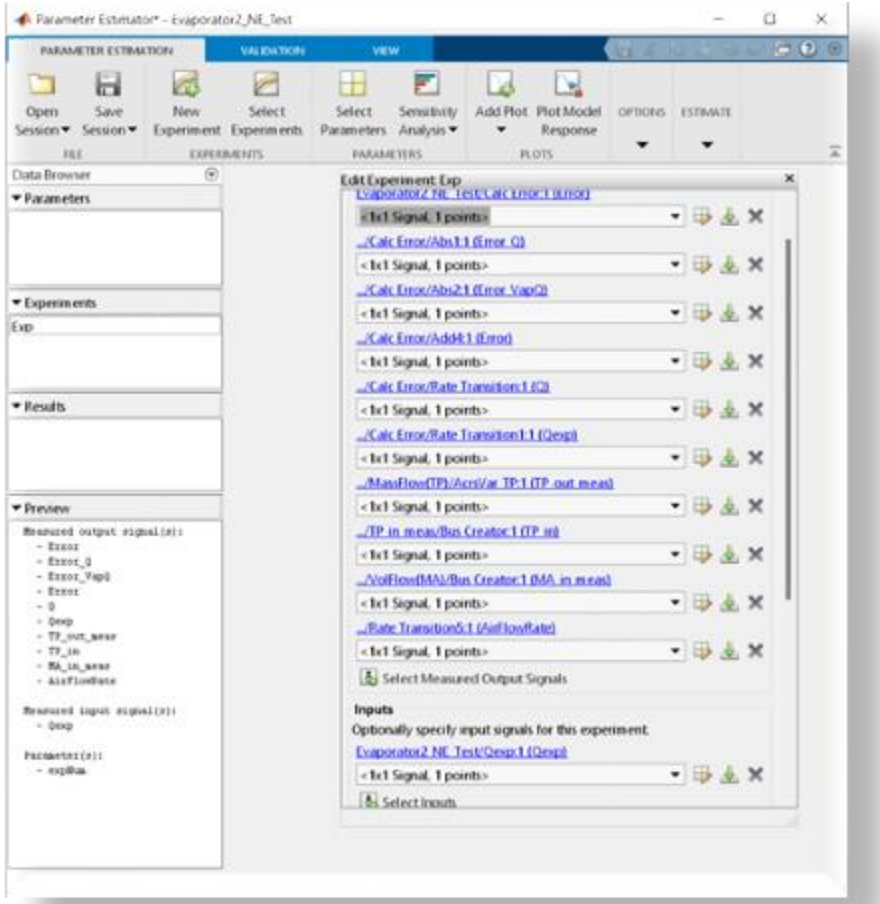
6. Choose statistical analysis methods from "STATISTICS" tab and click "Compute Statistics"
7. Choose the most influential parameters (8 params are selected)

Parameter estimation with experiment data

Parameter Estimator

- Easy to use
  - MATLAB App based convenient GUI
  - Auto-code generation

- Limitations
  - Local minimum problem
    Most solvers of SDO are not global optimization algorithm (except “patternsearch”)
  - Linear / nonlinear constraints are not applicable



Parameter Estimation App

Parameter estimation with experiment data
Global Optimization Toolbox

- Derivative-free algorithms
  - To break away local minima, they use heuristic / random / stochastic algorithms
  - "GlobalSearch", "MultiStart" and "surrogateopt" use derivatives,
  - But they have their own mechanism to overcome local minima problem

- Global Optimization Toolbox

ga / particleswarm	**GlobalSearch / MultiStart**	**patternsearch / simulannealbnd**	**surrogateopt**	**gamultiobj / paretosearch**
Population-based (genes, particles) heuristic algorithm	Multiple-starting points with derivative algorithm	Single-starting points with random / stochastic exploration algorithm	Uses approximation function(surrogate)	Applicable for multi-objective optimization problems

Parameter estimation with experiment data

Global Optimization Toolbox – GA

- Genetic Algorithm (GA)
  - It is a general tool that can be used extensively in various optimization problems
  - The function “ga” of Global Optimization Toolbox has the most options that can reflect various kinds of constrains such as “Liner”, “Non-linear”, “Integer constraint”

- Terms & Concepts
  - To use the “ga” function, users should know terms basic concepts of genetic algorithm
  - The terms were taken from the principles of genetics and natural selection

Parameter estimation with experiment data

Global Optimization Toolbox – GA

- Fitness function
  – Equivalent to objective function of optimization problems

- Individual
  – An object with a gene that has a value of a variable to be optimized

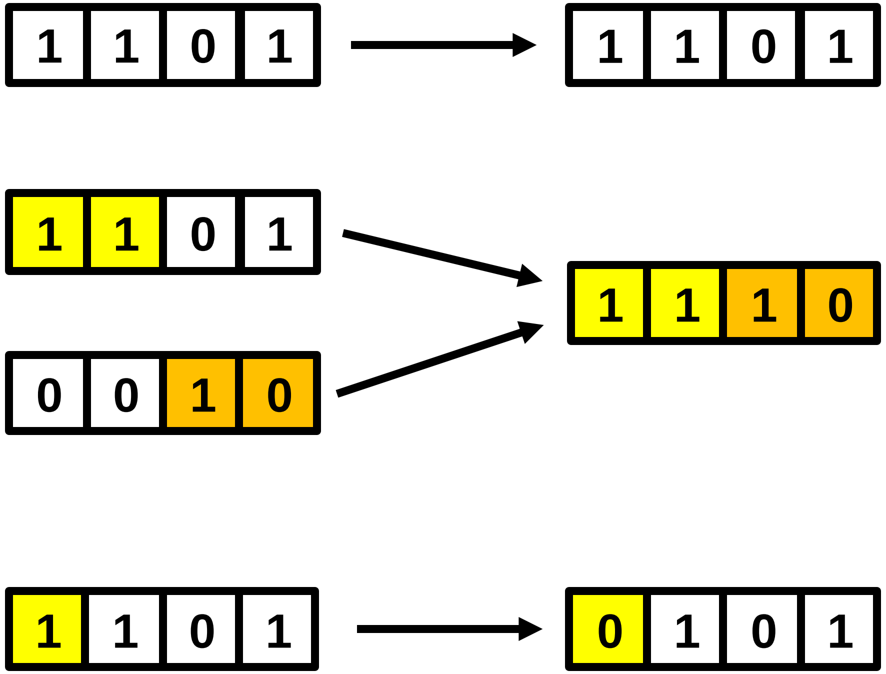
- Population
  – A set of individuals that are evolved together

- Generation
  – A stage of GA, individuals in a generation are evaluated using fitness function and best individuals are selected for reproduction

Parameter estimation with experiment data

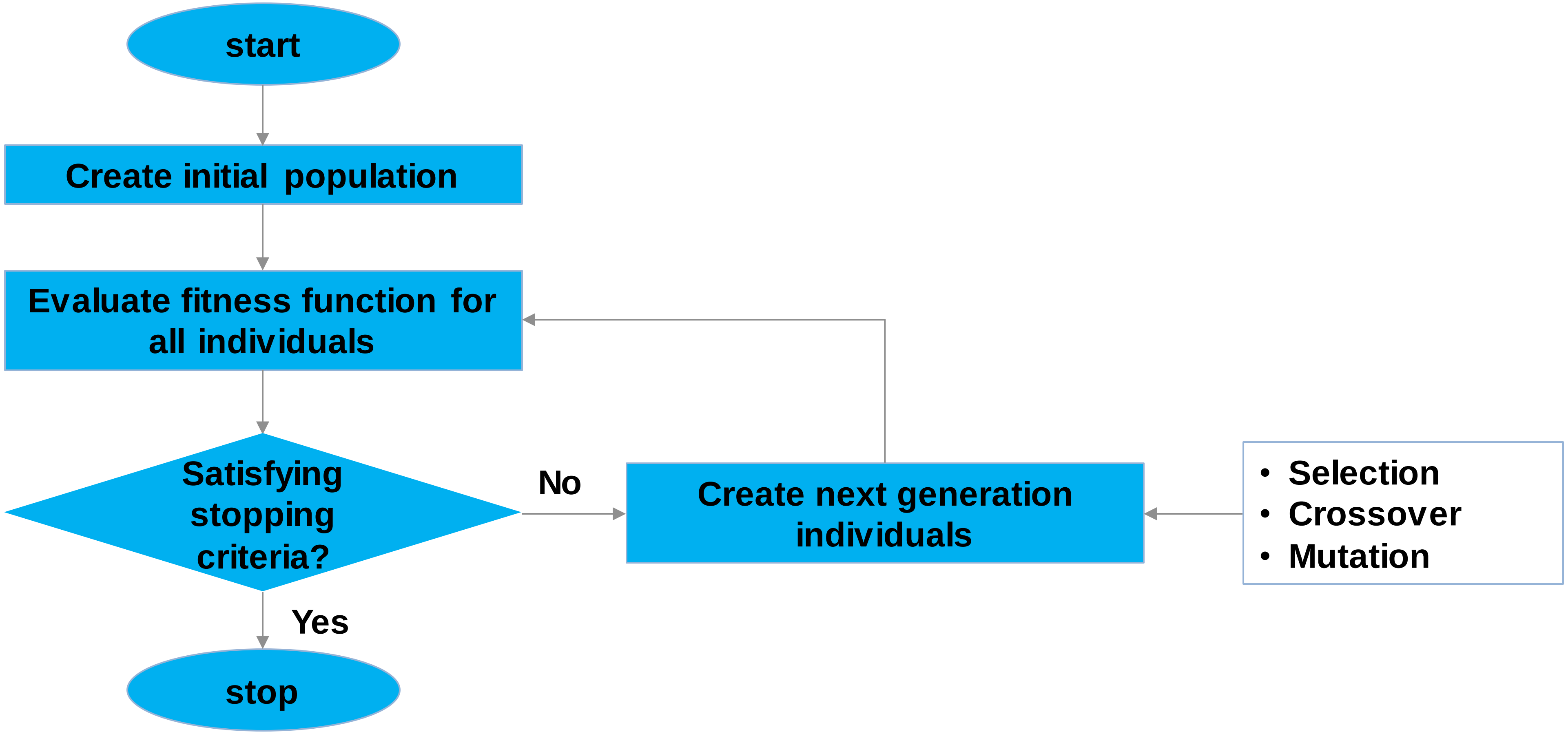
Global Optimization Toolbox – GA

- Selection
  - Elite individuals are selected and survive in the next generation
- Crossover
  - Genes of individuals are combined with genes of other individuals for the next generation
- Mutation
  - An unexpected change in gene



Parameter estimation with experiment data

Global Optimization Toolbox – GA



Parameter estimation with experiment data

Evaporator modeling

- Experiment data

Vdot_air (CMS)	T_{air}_in (°C)	Hd_in_abs (kg/kg)	T_{air}_out (°C)	Q_air (kW)	Mdot_ref (kg/s)	T_{ref}_in (°C)	T_{ref}_out (°C)	P_{ref}_in (bar)	P_{ref}_out (bar)	x_ref
0.0500	27	0.0200	5.0	3.00	0.03000	3.00	5.00	2.50	2.25	0.300

  - Refrigerant absorbs heat from moisture air as much as experiment data
  - Refrigerant phase should be converted from mixture to vapor

- Parameters to be tuned
  - 12 Colburn equation coefficients
  - 5 pressure loss parameters

Parameter estimation with experiment data

Evaporator modeling

- Objective function

Create simulation input data object

Update with optimization vars

Set vars with updated vars for simulation

```
%% Create Simulink simulation input data type "in"
in = Simulink.SimulationInput(mdl);
%% Loop for selected experiments
for expNum = expSet
    % Reference experiment result update
    [Qexp] = createQexp(expNum,tini,ts,tfin,Q_air);
    % For initial condition settting
    [Evaporator] = initEvaporator2(expNum,Evaporator);
    % I/O condition setting
    [Rsrv_TP_in,Rsrv_TP_out,Rsrv_MA_in,Rsrv_MA_out] = ...
        createReservoirs_Evaporator2_NE(expNum,Evaporator);

    % Insert update workspace variables with updated data infomation
    in = in.setVariable("expNum",expNum);
    in = in.setVariable("Qexp",Qexp);
    in = in.setVariable("Evaporator",Evaporator);
    in = in.setVariable("Rsrv_TP_in",Rsrv_TP_in);
    in = in.setVariable("Rsrv_TP_out",Rsrv_TP_out);
    in = in.setVariable("Rsrv_MA_in",Rsrv_MA_in);
    in = in.setVariable("Rsrv_MA_out",Rsrv_MA_out);
```

Parameter estimation with experiment data

Evaporator modeling

- Objective function

Simulate with simulation input object

Evaluate object function with Simulink simulation results

```
% Simulation
ErrorMessage = [];
try
    % Simulink "run" with "in"
    out = sim(in);
catch
    % For the case of simulation error
    ErrorMessage = "Invalid parameters!"
end
```

```
% Evaluate objective function
if isempty(ErrorMessage) && isempty(out.ErrorMessage)
    % Account only steady-state error with scale factor like "0.9"
    idx = find(out.logsout.get('Error').Values.Time > tfin*0.9);
    data = out.logsout.get('Error').Values.Data(idx(1):end);

    % Accumulate average error
    Ftmp = Ftmp + mean(data);
else
    % Penalty for simulaton error case
    Ftmp = 100;
end
```

Parameter estimation with experiment data

Evaporator modeling

- Option setting for optimization

- Set population size
- Set elite size

```
%%
optset.nvars = length(optset.x0);
% ga, gamultiobj, GlobalSearch, particleswarm, MultiStart, surrogateopt
optset.PopulationSize = 4*optset.nvars;
if optset.PopulationSize > 40
    optset.PopulationSize = 20;
end
optset.EliteCount = ceil(0.10*optset.PopulationSize);    % ga

% Check conditions
assert(optset.PopulationSize > 1);
```

Parameter estimation with experiment data

Evaporator modeling

- Option setting for optimization

```
%% Stopping Criterion
% 1. Tolerances
% FunctionTolerance is a lower bound on the change in the value of the obje
optset.FunctionTolerance = 1e-3;
% ConstraintTolerance is an upper bound on the magnitude of any constraint
optset.ConstraintTolerance = 1e-3;
% OptimalityTolerance is a tolerance for the first-order optimality measure
% First-order optimality is a measure of how close a point x is to optimal
optset.OptimalityTolerance = 1e-3;          % GlobalSearch, MultiStart
% Tolerance on function values for considering solutions equal, specified a
% Solvers consider two solutions identical if they are within XTolerance re
% and have objective function values within FunctionTolerance relative diff
optset.XTolerance = 1e-3;                   % GlobalSearch, MultiStart
% When the current mesh size is less than the value of MeshTolerance, the a
optset.MeshTolerance = 1e-3;                % patternsearch, paretosearch
optset.StepTolerance = 1e-3;                % patternsearch

% 2. Iterations and Function Counts
optset.MaxIterations = 100;
optset.MaxFunctionEvaluations = optset.MaxIterations*(optset.nvars + 1);

% 3. Others
optset.MaxTime = 1*60*60;         % [sec]
optset.MaxGenerations = 20;    % ga, gamultiobj
% The algorithm stops when the average relative change in the fitness funct
% over MaxStallGenerations is less than Function tolerance.
optset.MaxStallGenerations = optset.MaxGenerations*0.8;   % ga, gamultiobj
optset.FitnessLimit = 0.15;      % ga, (Lower limt of objective function va
optset.ObjectiveLimit = optset.FitnessLimit;            % surrogate
```

- There are various stopping criterion for optimization solvers
- For "ga" solver
  - Max generation
  - Max stall generation
  - Max time
  - Fitness limit

  are the main stopping criterion

Parameter estimation with experiment data

Evaporator modeling

- Constraints
  - Linear inequality / equality
  - Nonlinear
  - Integer
- Example (Linear inequality)
  - $-x(1) + x(2) \leq -1$
  - $-x(1) + x(2) \leq 5$
  - $Aineq = \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix}$
  - $bineq = \begin{bmatrix} -1 \\ 5 \end{bmatrix}$

```
%% Constraints
optset.Aineq = [];
optset.bineq = [];
optset.Aeq = [];
optset.beq = [];
optset.nonlcon = [];
optset.intcon = [];
```

Parameter estimation with experiment data

Evaporator modeling

- There are “ga” solver its own options such as
  - Initial population creation
  - Plot function during optimization
  - Hybrid function that runs after “ga” optimization to find more delicate solution

```
%% ga
optset.CreationFcn = 'gacreationuniform';
% PlotFcn options
% 'gaplotbestf' plots the best score value and mean score versus generation
% 'gaplotbestindiv' plots the vector entries of the individual with the bes
% 'gaplotexpectation'  plots the expected number of children versus the raw
% 'gaplotrange' plots the minimum, maximum, and mean score values in each g
optset.ga.PlotFcn = {@gaplotbestf,@gaplotbestindiv};
optset.ga.OutputFcn = {@outputfcn_ga};
optset.ga.hybridsolver = 'fmincon';
optset.ga.hybridopts = optimoptions(...
    optset.ga.hybridsolver,...
    'Algorithm','sqp',...
    'ConstraintTolerance',optset.ConstraintTolerance,...
    'Display','iter',...
    'MaxFunctionEvaluations',ceil(min(100,optset.MaxFunctionEvaluations/10)
    'MaxIterations',ceil(min(3,optset.MaxIterations/10)),...
    'OptimalityTolerance',optset.OptimalityTolerance,...
    'PlotFcn','optimplotfval' ...
    );
```

Parameter estimation with experiment data

Evaporator modeling

- Run “ga”
  - **x = ga(fun, nvars, A, b, Aeq, beq, lb, ub, nonlcon, intcon, options)**
  - fun: objective function handle
  - nvars: number of variables
  - A, b: linear inequality constraint matrix and vector
  - Aeq, beq: linear equality constraint matrix and vector
  - lb, ub: lower and upper bounds
  - nonlcon: non-linear constraint
  - intcon: integer constraint
  - options: other options such as max time, max iteration, max generation, etc.

Find minimum of function using genetic algorithm - MATLAB ga (mathworks.com)

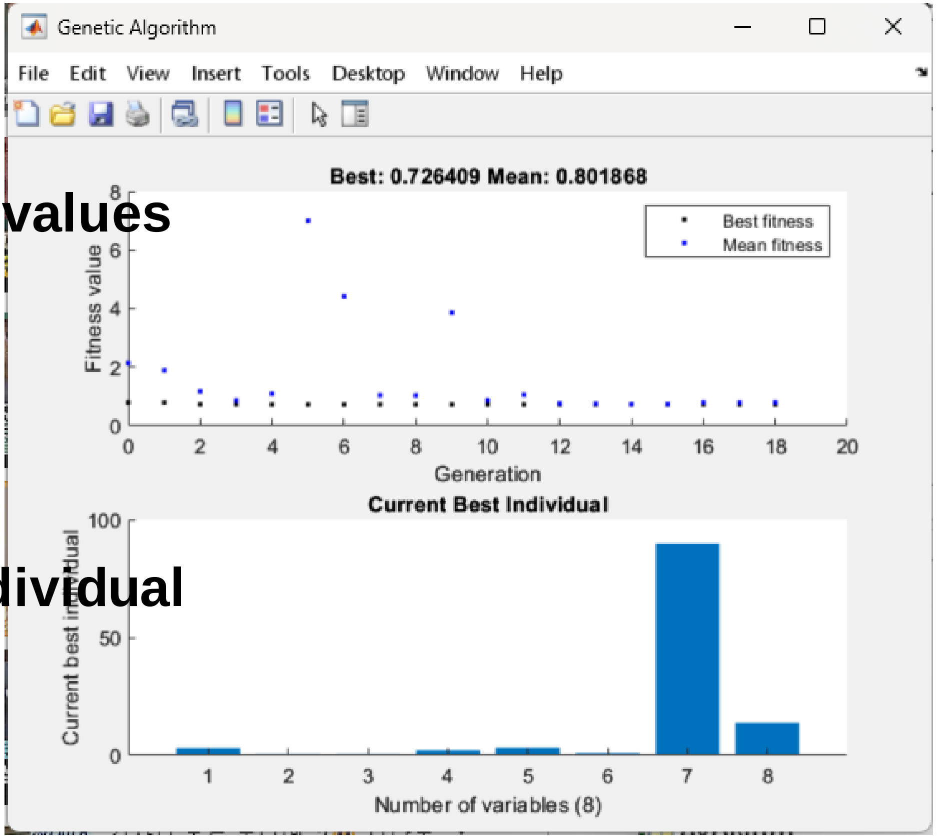
Parameter estimation with experiment data

Evaporator modeling

- "ga" optimization

Fitness values

Best individual



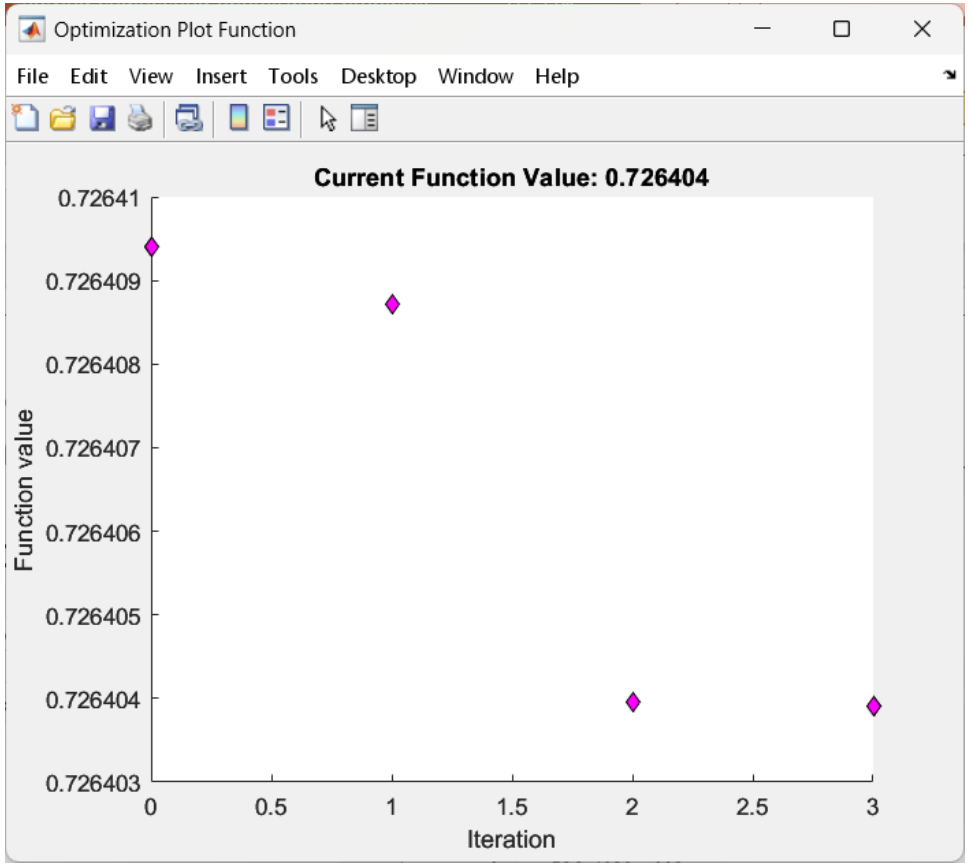
Other params except the estimated 8 params are remain at default values

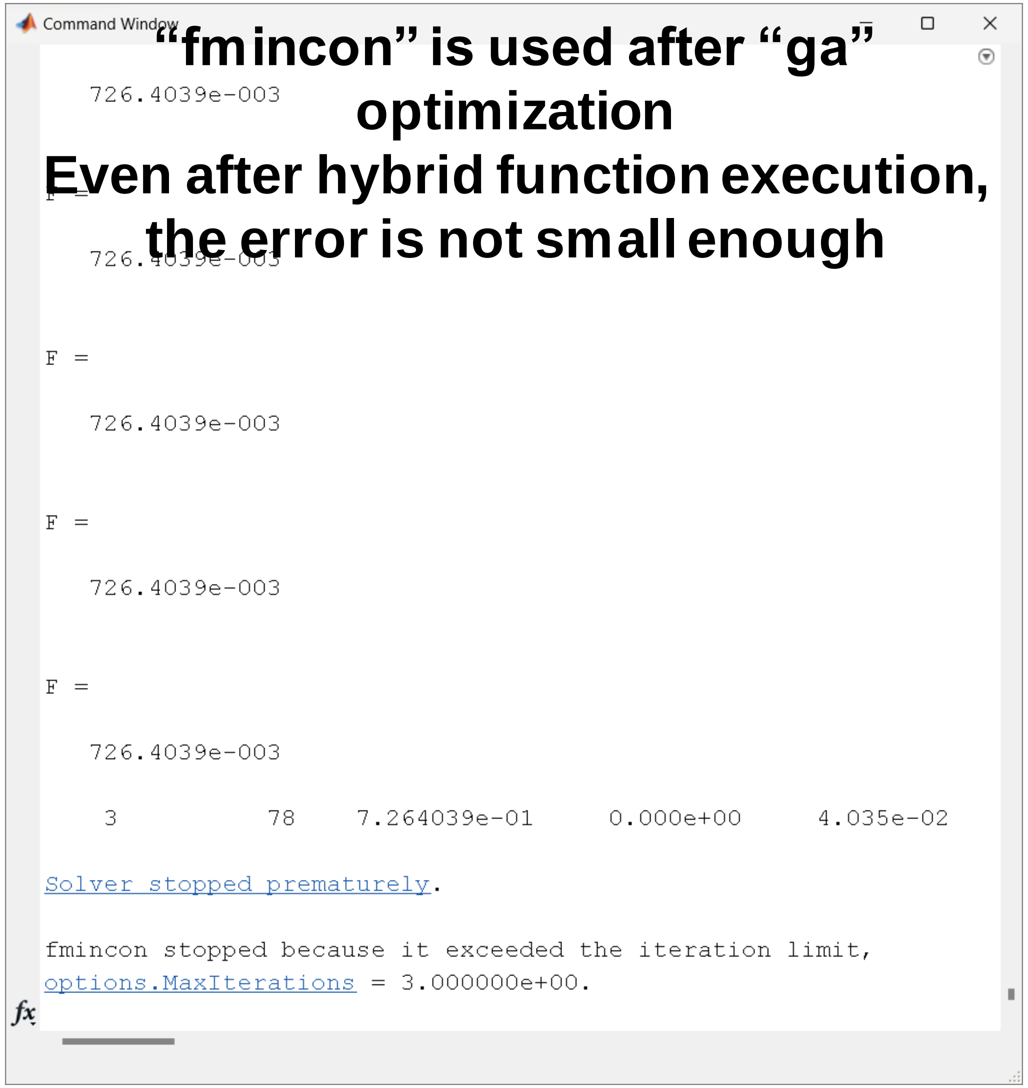
```
Command Window

Solver stopped prematurely.

fmincon stopped because it exceeded the iteration limit,
options.MaxIterations = 3.000000e+00.

FMINCON terminated.

info =

  struct with fields:

      StartTime: 14-May-2023 17:15:47
      StartDate: 14-May-2023
          model: 'Evaporator2_NE_Test'
         solver: "ga"
        problem: [1×1 struct]
        options: [1×1 optim.options.GaOptions]
         optset: [1×1 struct]
           fval: 726.4039e-003
       exitflag: 1.0000e+000
         output: [1×1 struct]
     population: [32×8 double]
         scores: [32×1 double]
    exitmessage: "Without nonlinear constraints↵ — Average cumu
           xopt: [3.1019e+000 601.9430e-003 601.8701e-003 2.1019e
       StopTime: 14-May-2023 17:52:11
       TimeCost: 00:36:24

fx >>
```

Parameter estimation with experiment data

Evaporator modeling

- Hybrid function optimization





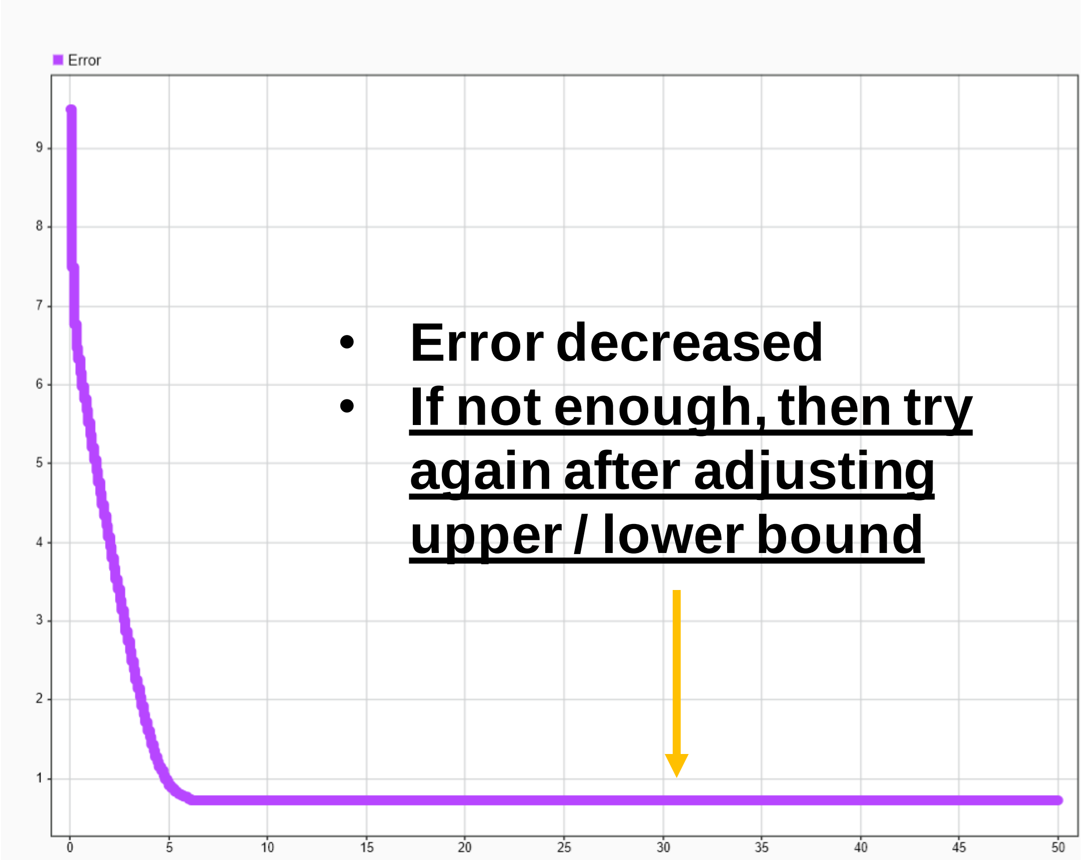
“fmincon” is used after “ga” optimization

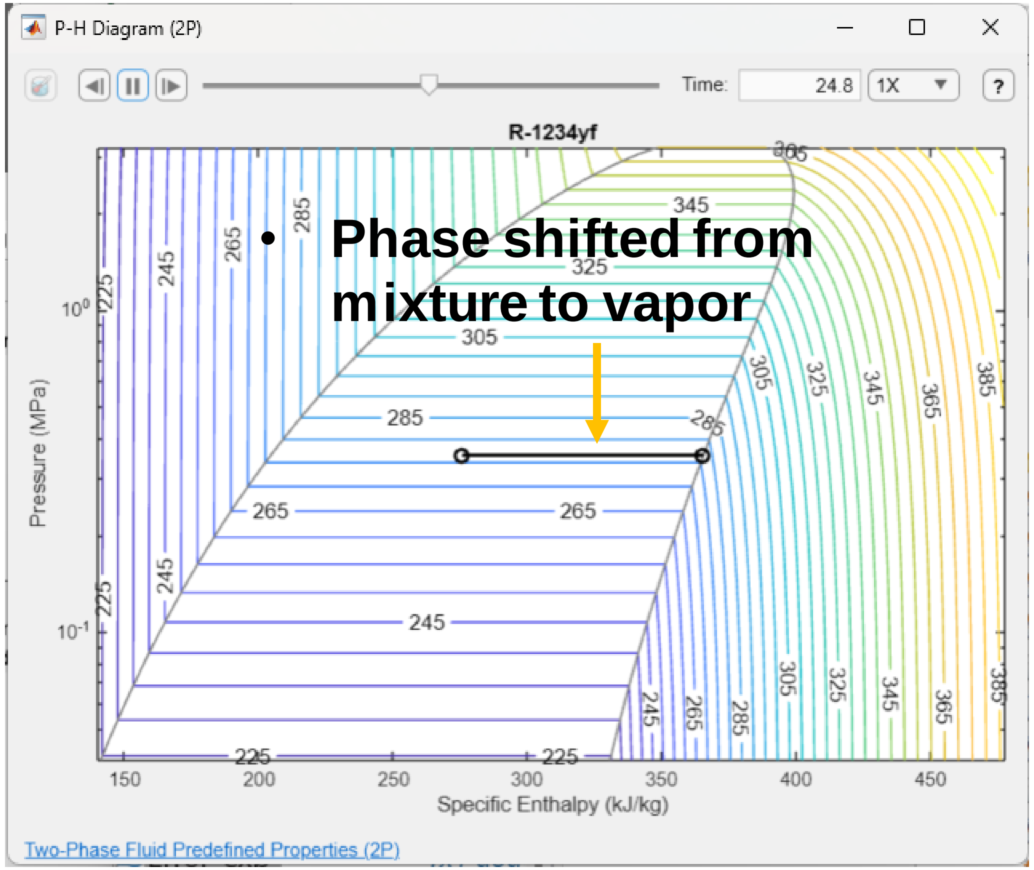
Even after hybrid function execution, the error is not small enough

Parameter estimation with experiment data

Evaporator modeling

- Results





Optimal calibration with experiment data

Problem statement

- **Goal:**
  To introduce an optimization workflow using "**Model-Based Calibration Toolbox"** in look-up table modeling
  - Look-up table calibration example for motor current control
  - How to model statistical modeling with experiment data
  - How to calibrated look-up table for optimized performance with the fitted model

Optimal calibration with experiment data

Look-up tables for flux-based motor controller

- Calibration Table Generation Workflow Steps

Workflow Steps	Description
Collect and Post Process Motor Data	**Required data:** • D/Q-axis current: I_d, I_q • Motor torque and speed • D/Q-axis flux: λ_d, λ_q • Allowed flux: λ_{max}
Model Motor Data (MBC Model Fitting)	**Use a point-by-point model to fit data:** • Import data • Filter and group data • Fit model
Generate Calibration (MBC Optimization)	**Calibrate and optimize the data using objectives and constraints:** • Create functions (Objective, Constraint) • Create LUT (Lookup Tables) from model • Optimization • Fill and export LUT

Optimal calibration with experiment data

Look-up tables for flux-based motor controller

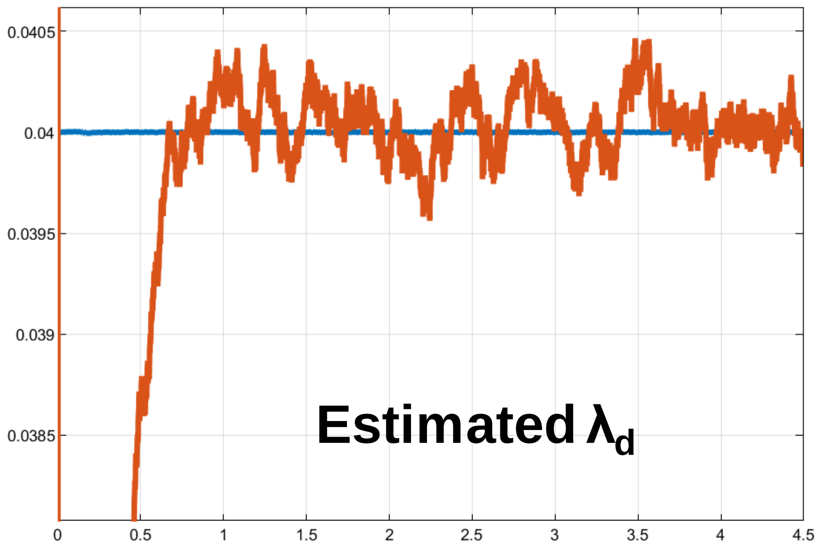
- Test motor and process data
  - PMSM equations
    - $V_d = R_s i_d + \frac{d\lambda_d}{dt} - p\omega_m \lambda_d$
    - $V_q = R_s i_q + \frac{d\lambda_q}{dt} + p\omega_m \lambda_q$

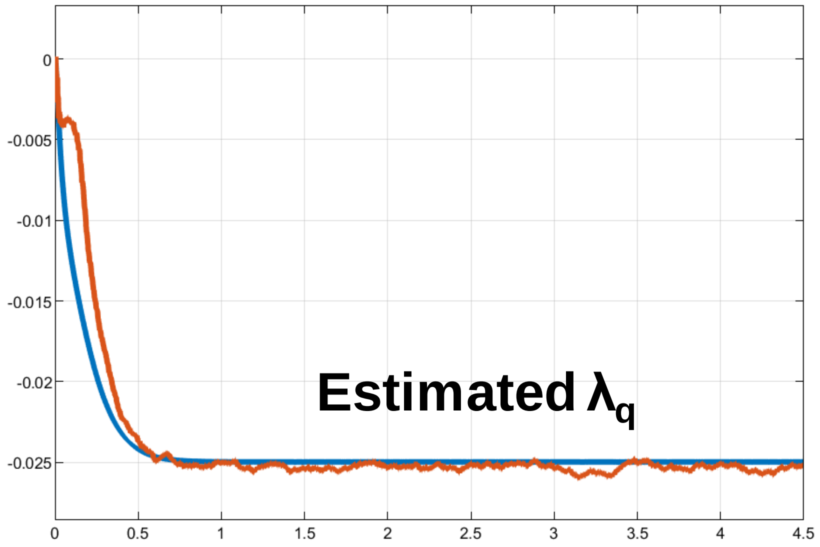
    for steady-state, the 1st and 2nd equations become

    - $\lambda_d = \frac{(R_s i_d - V_d)}{p\omega_m}, \lambda_q = \frac{(V_d - R_s i_q)}{p\omega_m}$: d/q flux linkage
    - $\lambda_{total} = \sqrt{\lambda_d^2 + \lambda_q^2}$
    - $\lambda_{max} = \frac{V_{dc}}{\sqrt{3} p\omega_m}$: **allowed flux @given speed**

    or calculation with L_d and L_q

    - $\lambda_d = L_d i_d + \lambda_m$
    - $\lambda_q = L_q i_q$

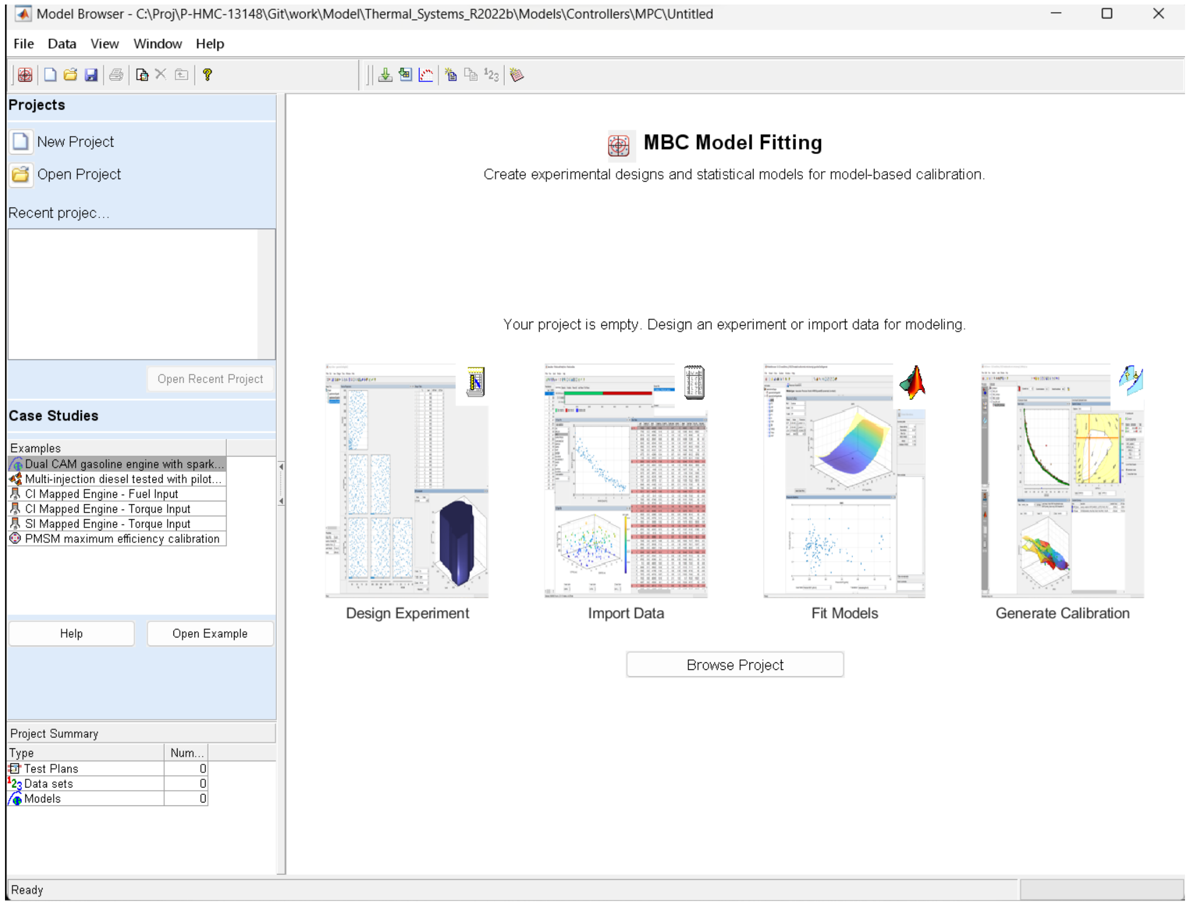




Optimal calibration with experiment data

Look-up tables for flux-based motor controller

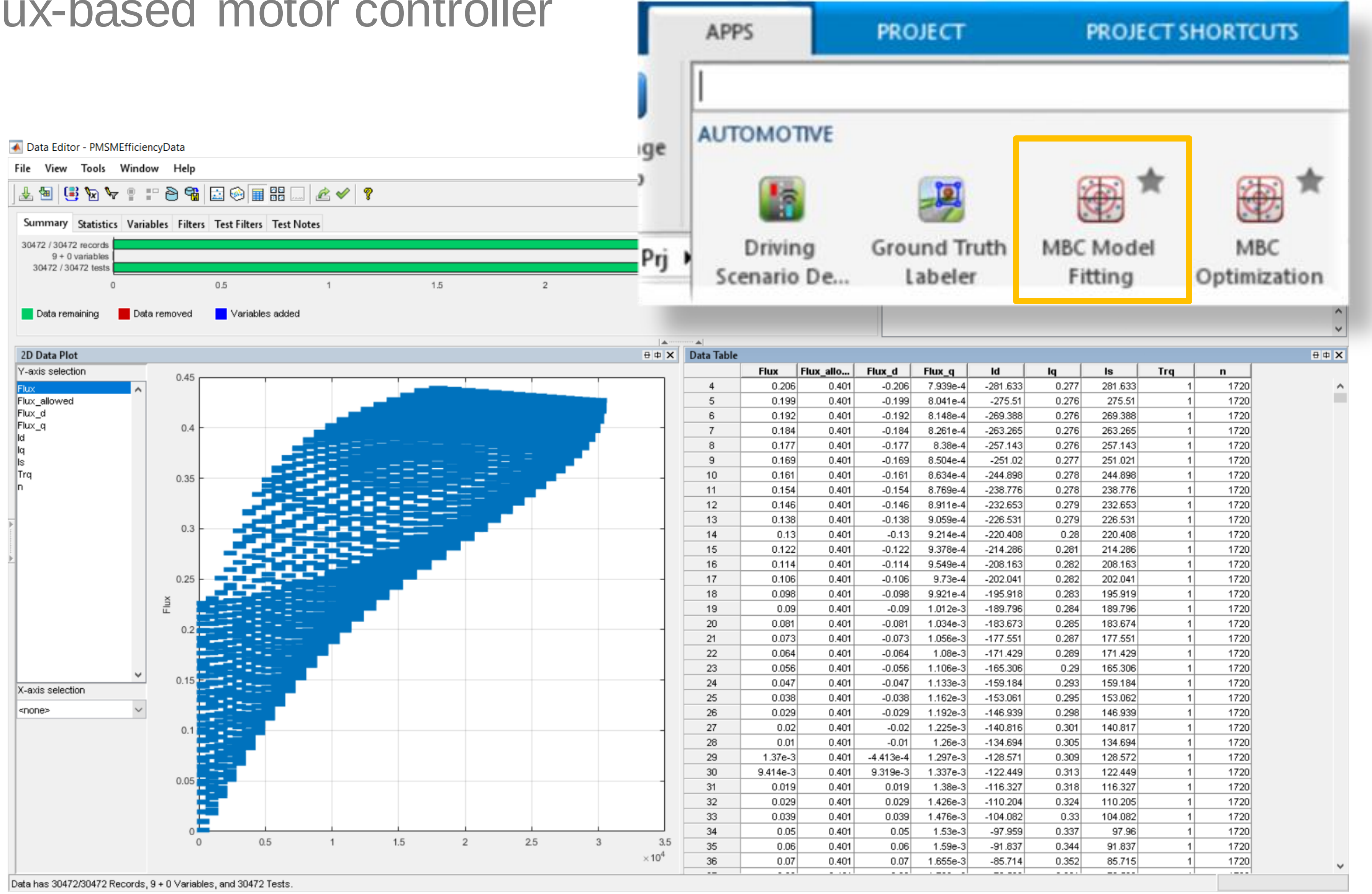
- Model-Based Calibration Toolbox
  - Is specialized for design of experiments, fitting statistical models, and generating calibrations and lookup tables of complex nonlinear systems
  - Users can automate the model fitting and calibration process by using the toolbox apps or MATLAB functions



Optimal calibration with experiment data

Look-up tables for flux-based motor controller

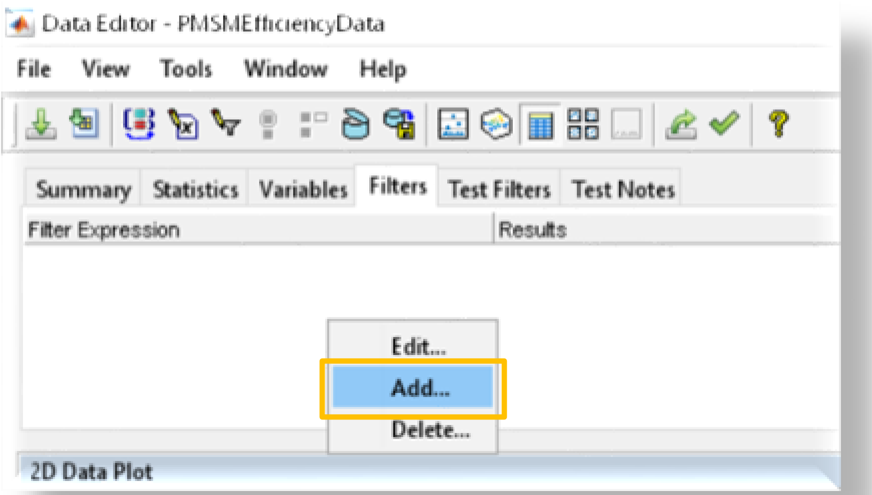
- Import data
  - Flux total
  - Flux allowed
  - Flux d-axis
  - Flux q-axis
  - Current d-axis
  - Current q-axis
  - Torque
  - Speed

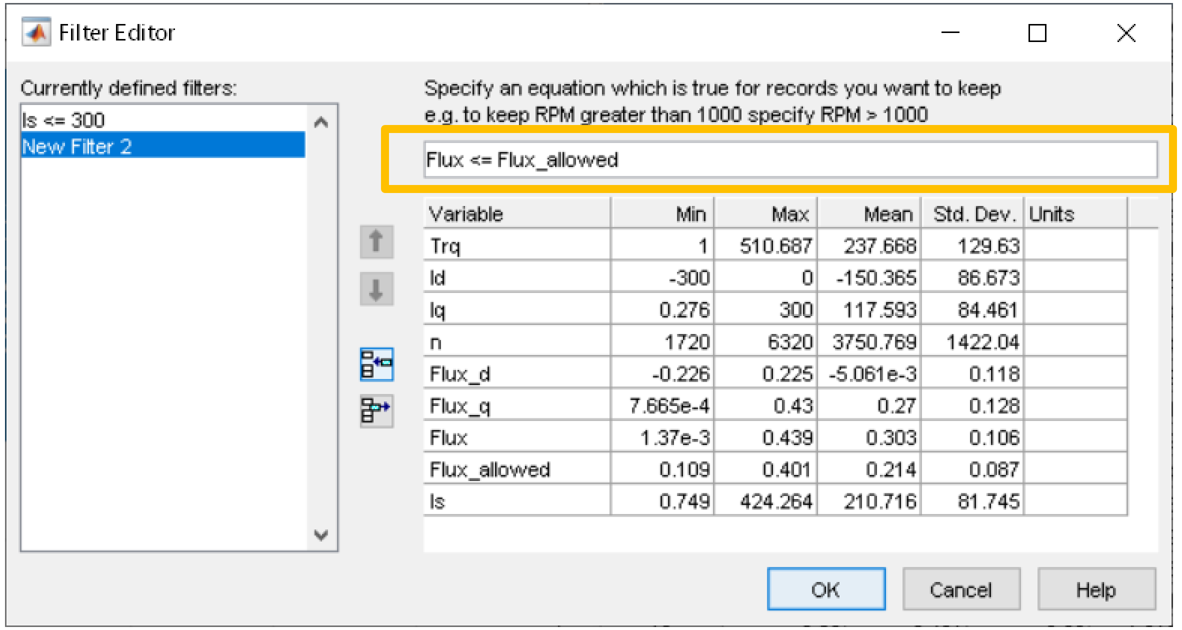


Optimal calibration with experiment data

Look-up tables for flux-based motor controller

- Filter and group data
  - Add filtering conditions

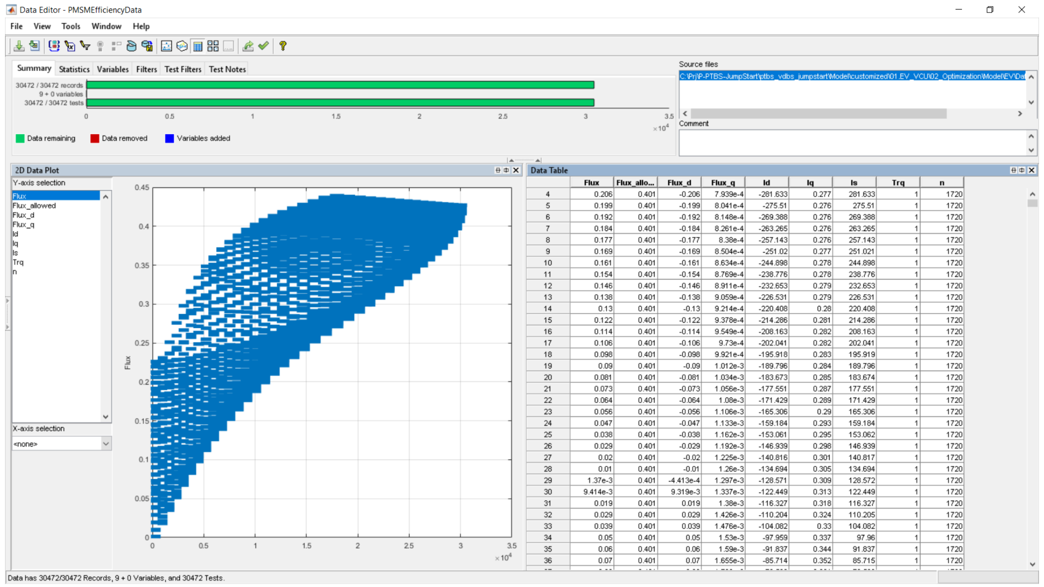




Optimal calibration with experiment data

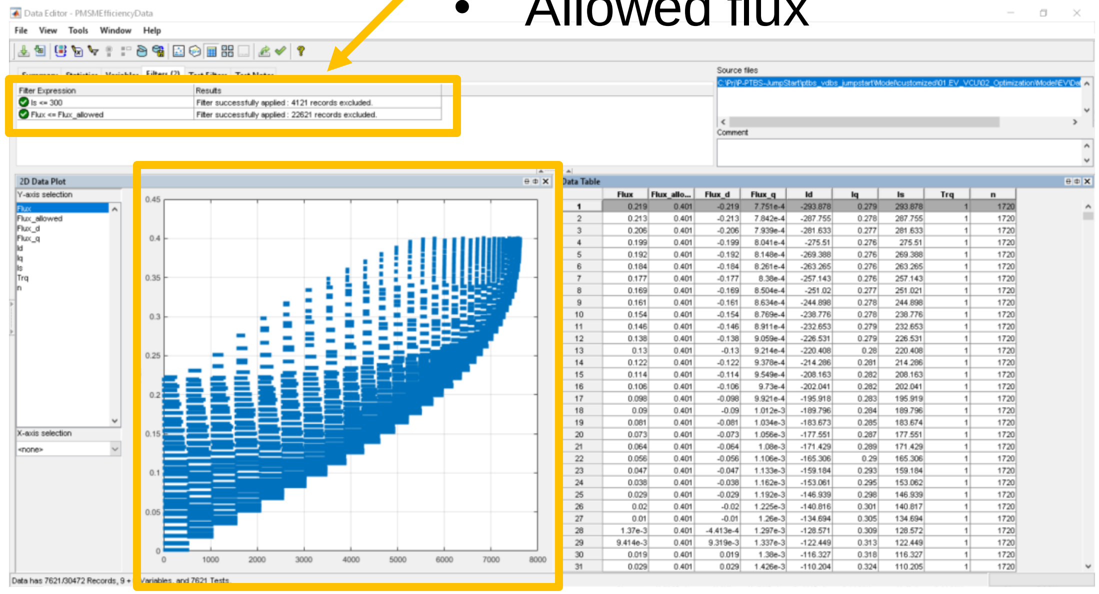
Look-up tables for flux-based motor controller

- Filter and group data



Before

- Max current
- Allowed flux

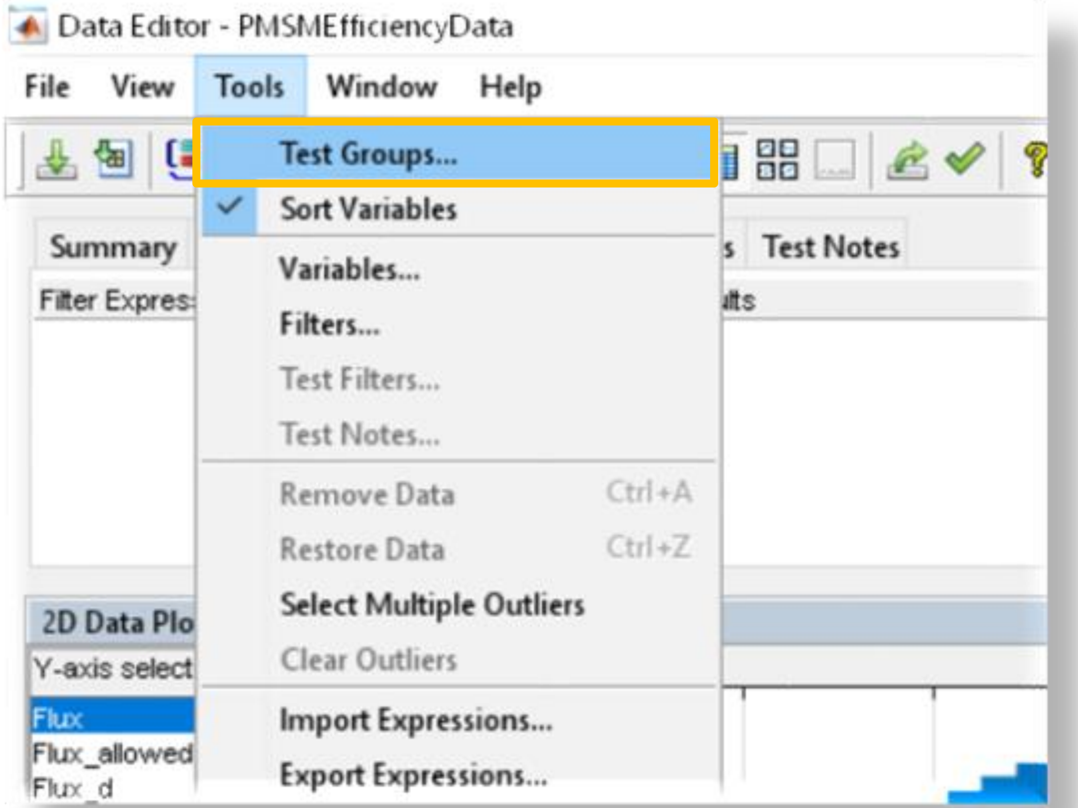


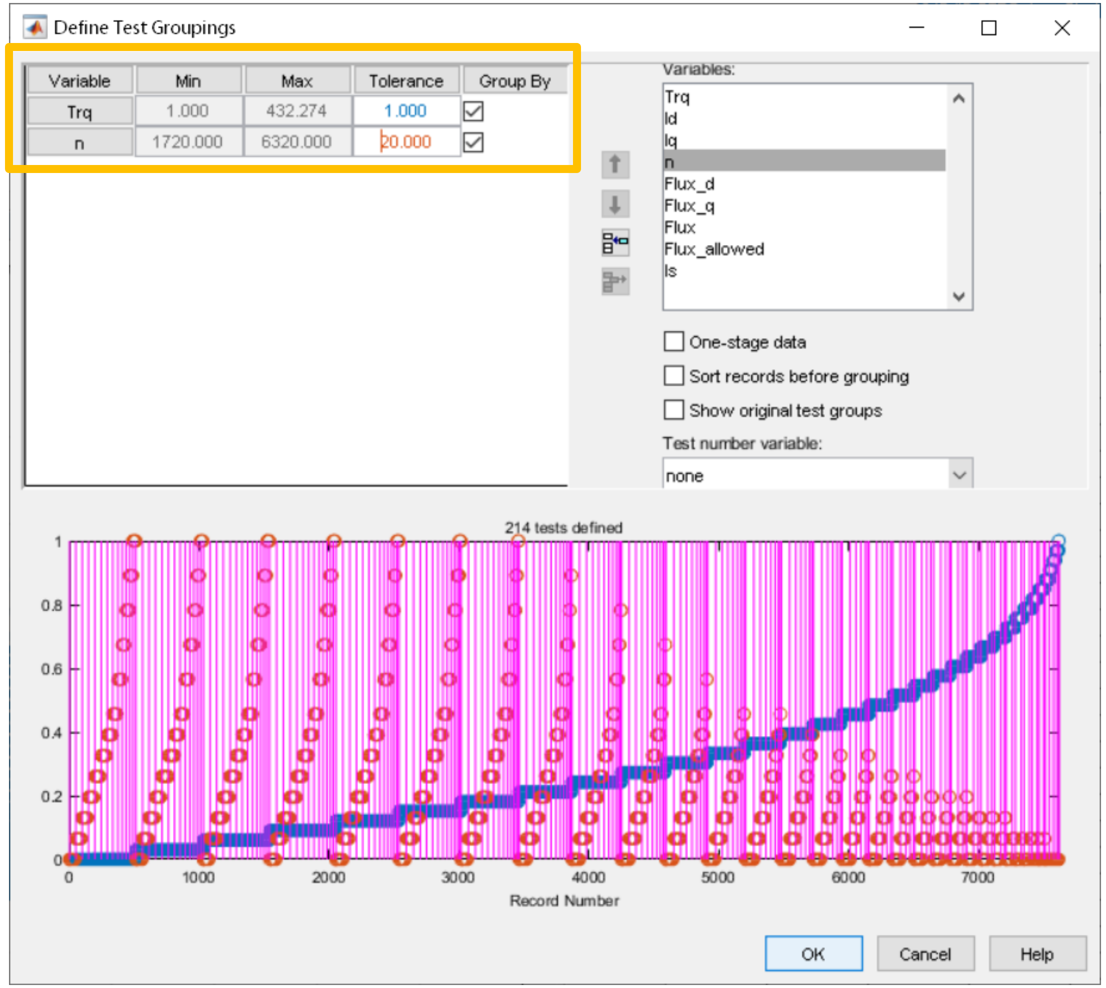
After

Optimal calibration with experiment data

Look-up tables for flux-based motor controller

- Filter and group data
  - By torque & speed

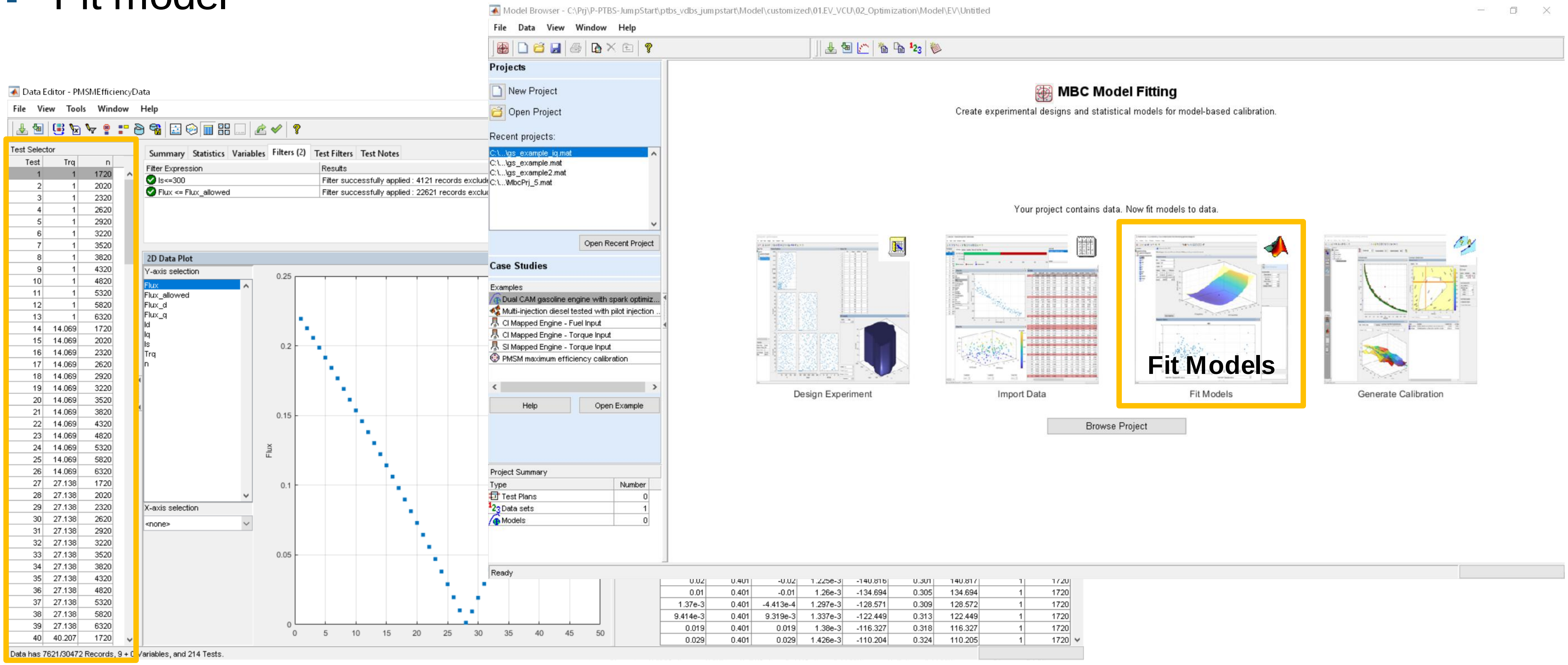




Optimal calibration with experiment data

Look-up tables for flux-based motor controller

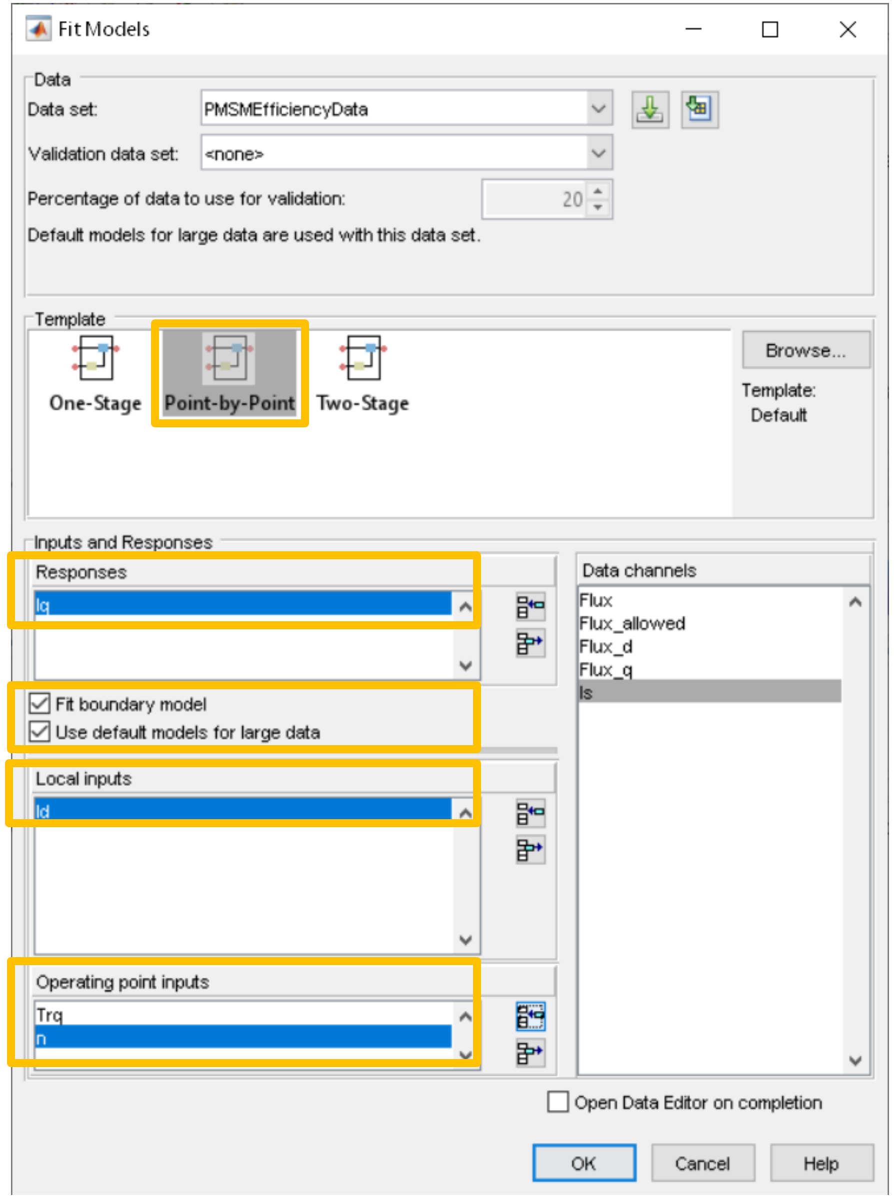
- Fit model



Optimal calibration with experiment data

Look-up tables for flux-based motor controller

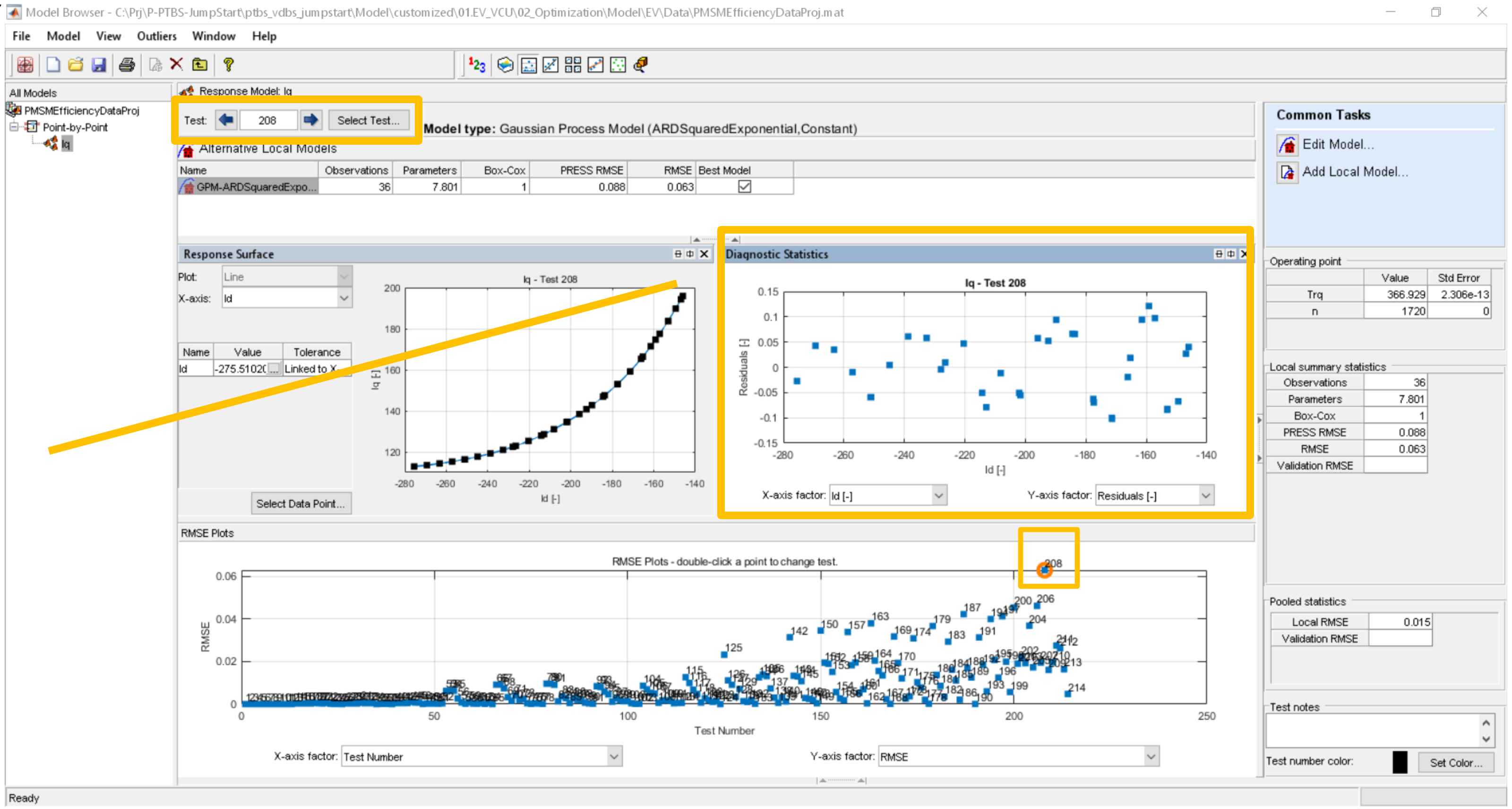
- Fit model
  - Point-by-Point:
    Allows to build a model at each operating point
  - Responses / Local inputs
    - Set Iq as the response of the model
    - Set Id as the local inputs
  - Select modeling options
    - Fit boundary model:
      Describes the limits of the operating envelope
    - Use default models for large data:
      Default model is GPM (Gaussian Process Model)
  - Operating point inputs
    - Torque
    - Motor rotating speed



Optimal calibration with experiment data

Look-up tables for flux-based motor controller

- Fit model
  - Test 208 shows a little large RMSE
  - You can move to test 208 by using “Select Test”
    - Shows a little large distribution
    - If there are some outliers, they can be removed by “Outliers”



Optimal calibration with experiment data

Look-up tables for flux-based motor controller

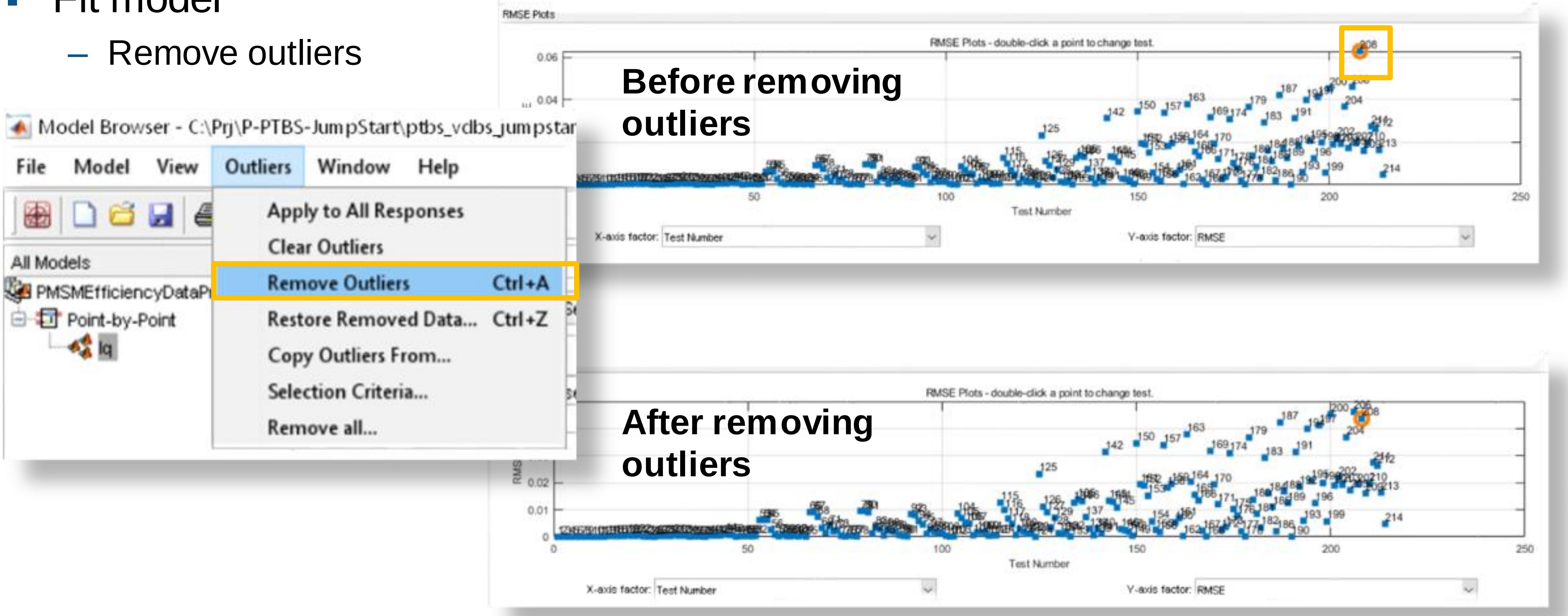
- Fit model
  - Remove outliers

Selected outliers in the small circles

Optimal calibration with experiment data

Look-up tables for flux-based motor controller

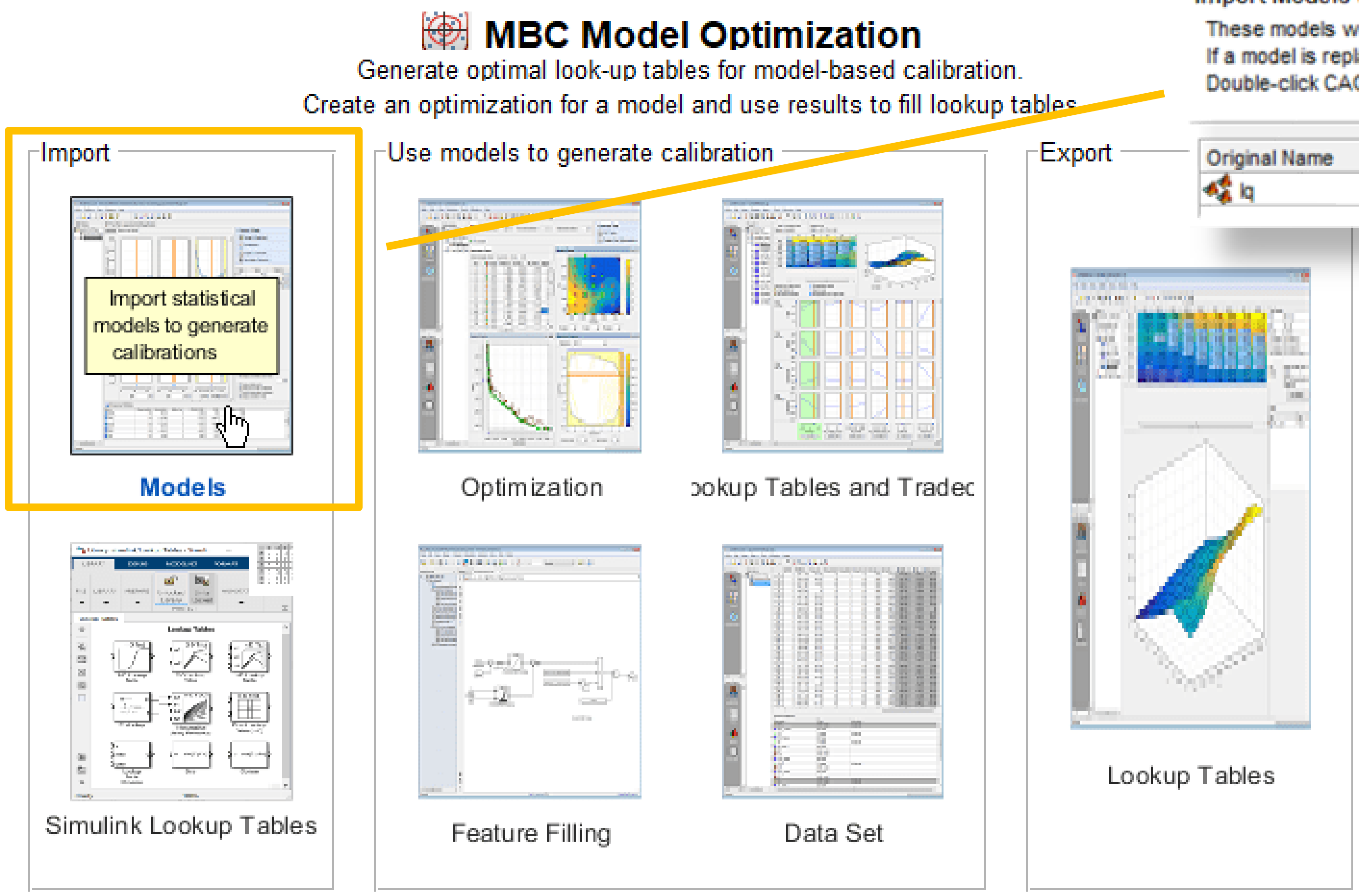
- Fit model
  - Remove outliers

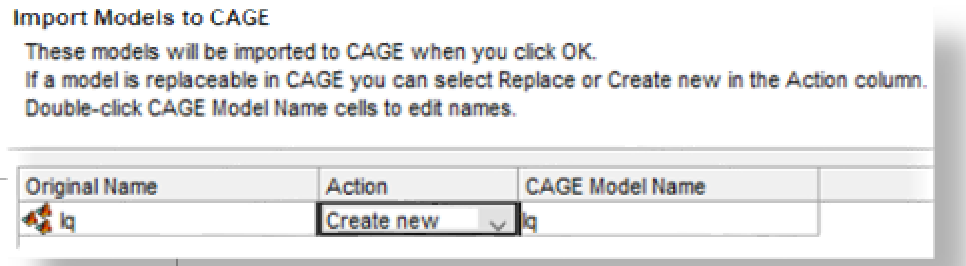


Optimal calibration with experiment data

Look-up tables for flux-based motor controller

- Create functions



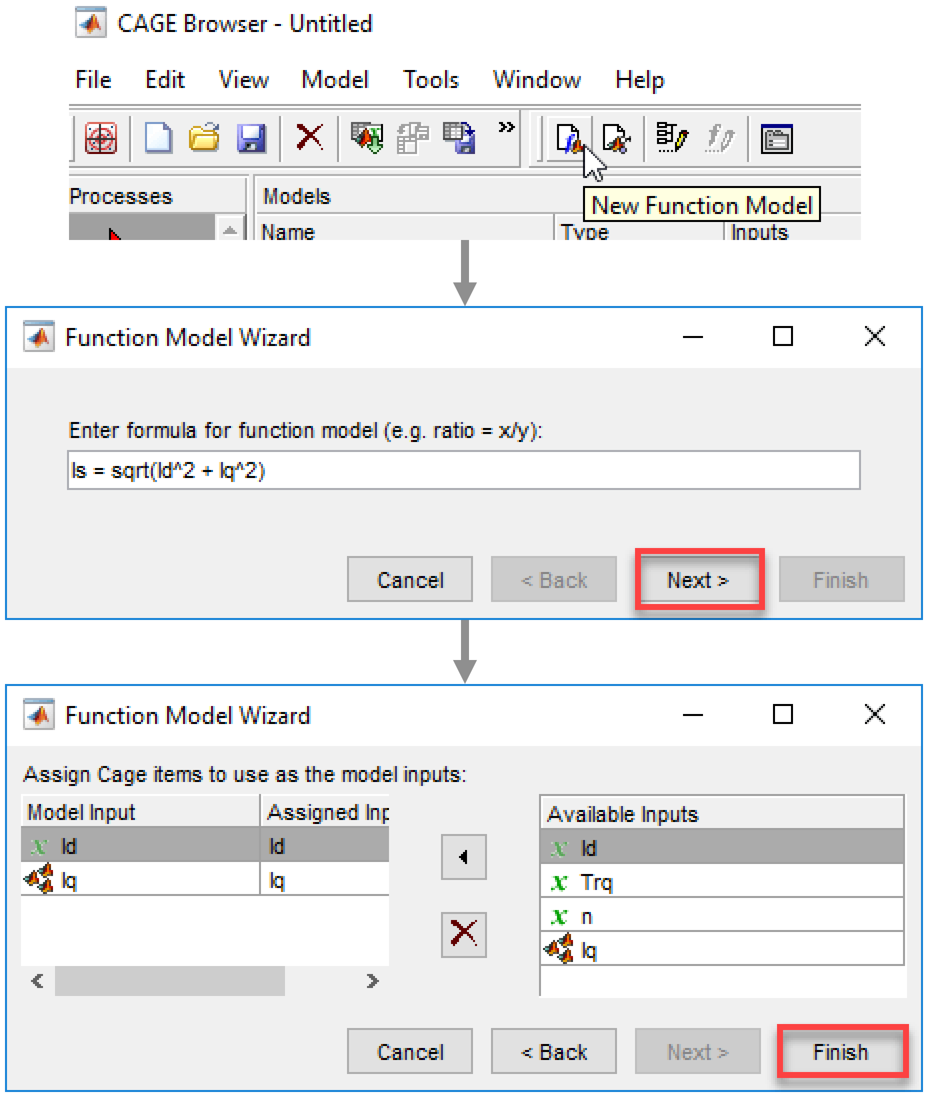


Import model

Optimal calibration with experiment data

Look-up tables for flux-based motor controller

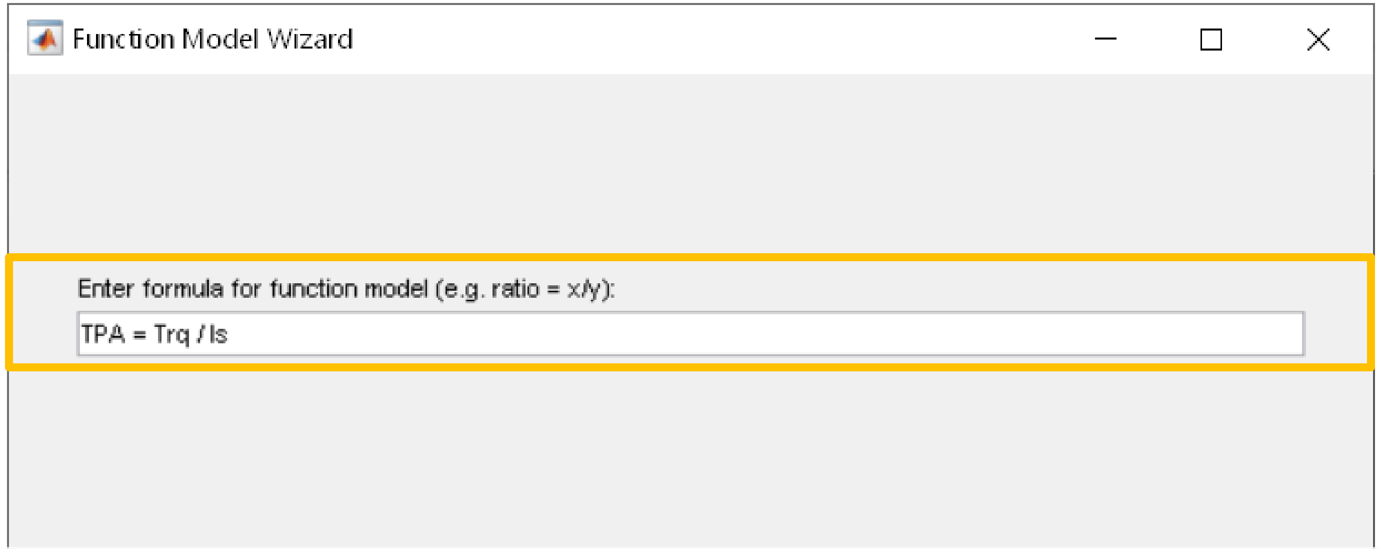
- Create functions
  - Current magnitude, Is
  - Is ≤ 300 A, the constraint



Optimal calibration with experiment data

Look-up tables for flux-based motor controller

- Create functions
  - TPA = Trq / Is
  - TPA(Torque Per Ampere) is the objective function to be maximized by optimization



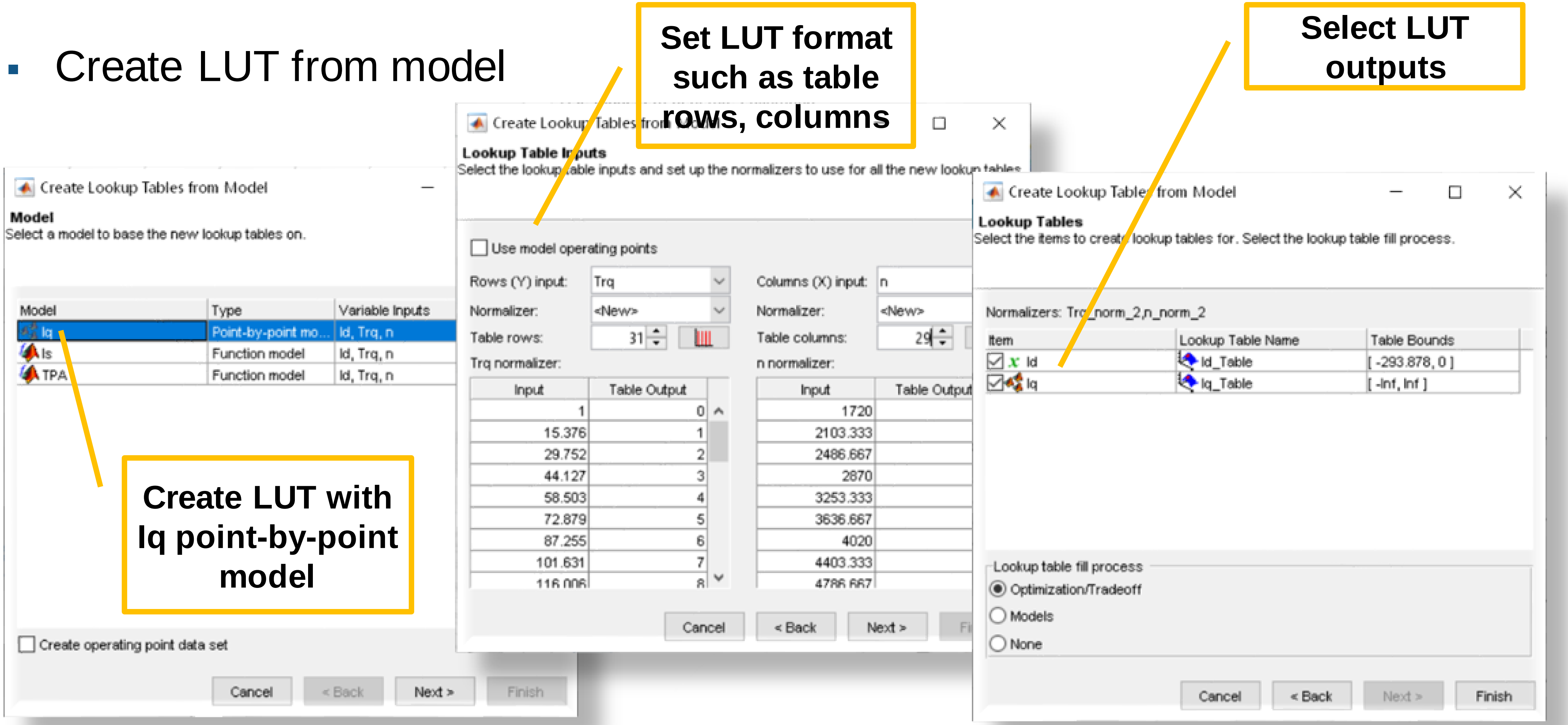
Models

Name	Type	Inputs	Lower Output Limit	Upper Output Limit	Description
Iq	Point-by-point ...	Id, Trq, n	-Inf	Inf	Created by tchoi on 30-Mar-2022.
Is	Function model	Id, Iq	-Inf	Inf	sqrt(Id^2 + Iq^2)
TPA	Function model	Is, Trq	-Inf	Inf	Trq / Is

Optimal calibration with experiment data

Look-up tables for flux-based motor controller

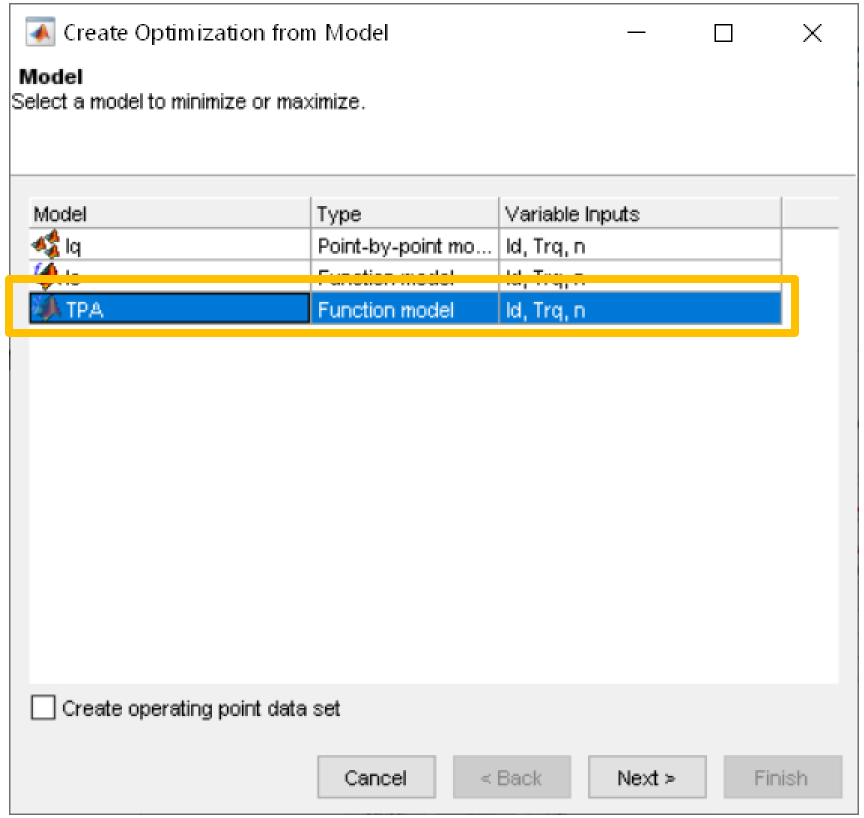
- Create LUT from model

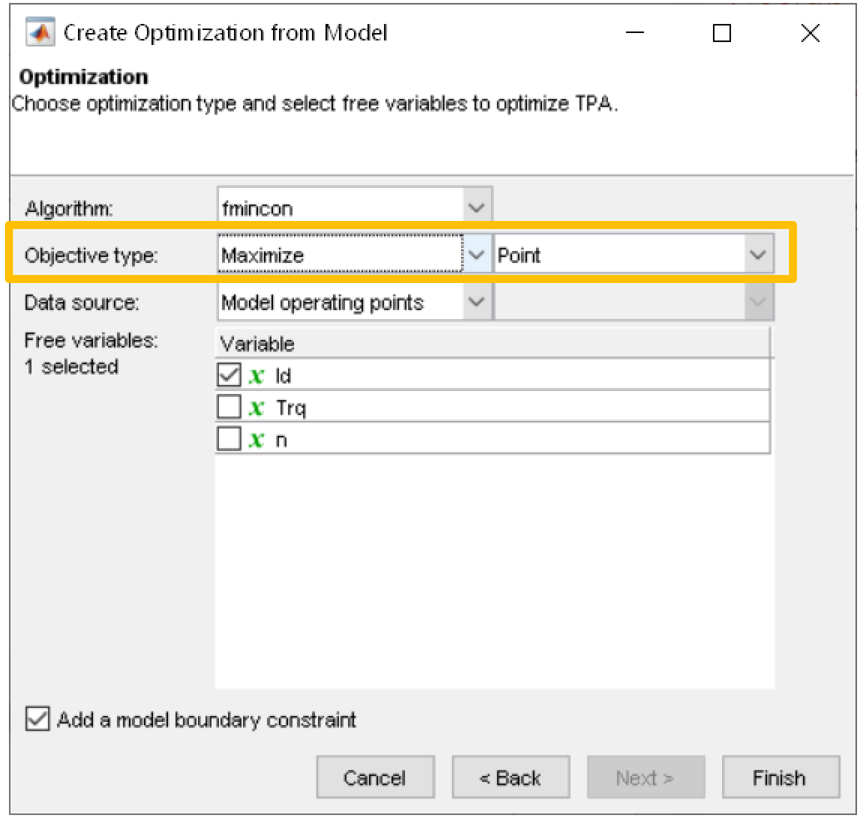


Optimal calibration with experiment data

Look-up tables for flux-based motor controller

- Optimization
  - Objective function: TPA, maximized

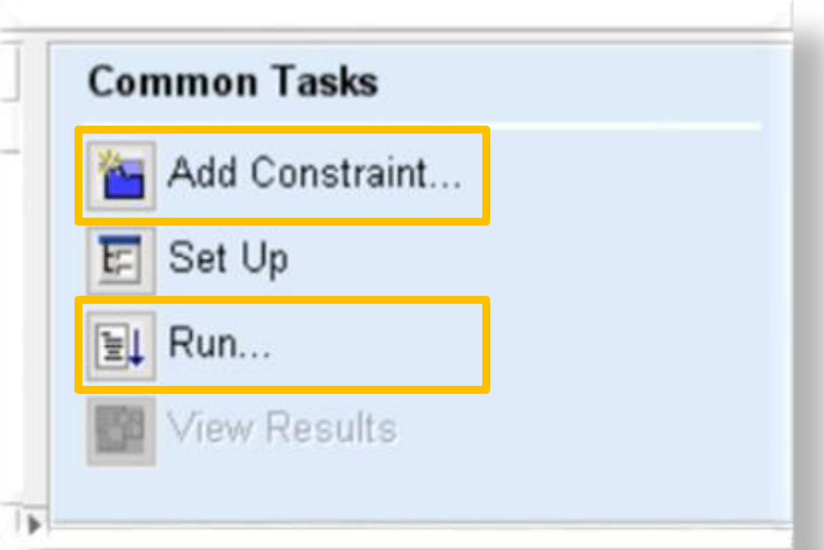


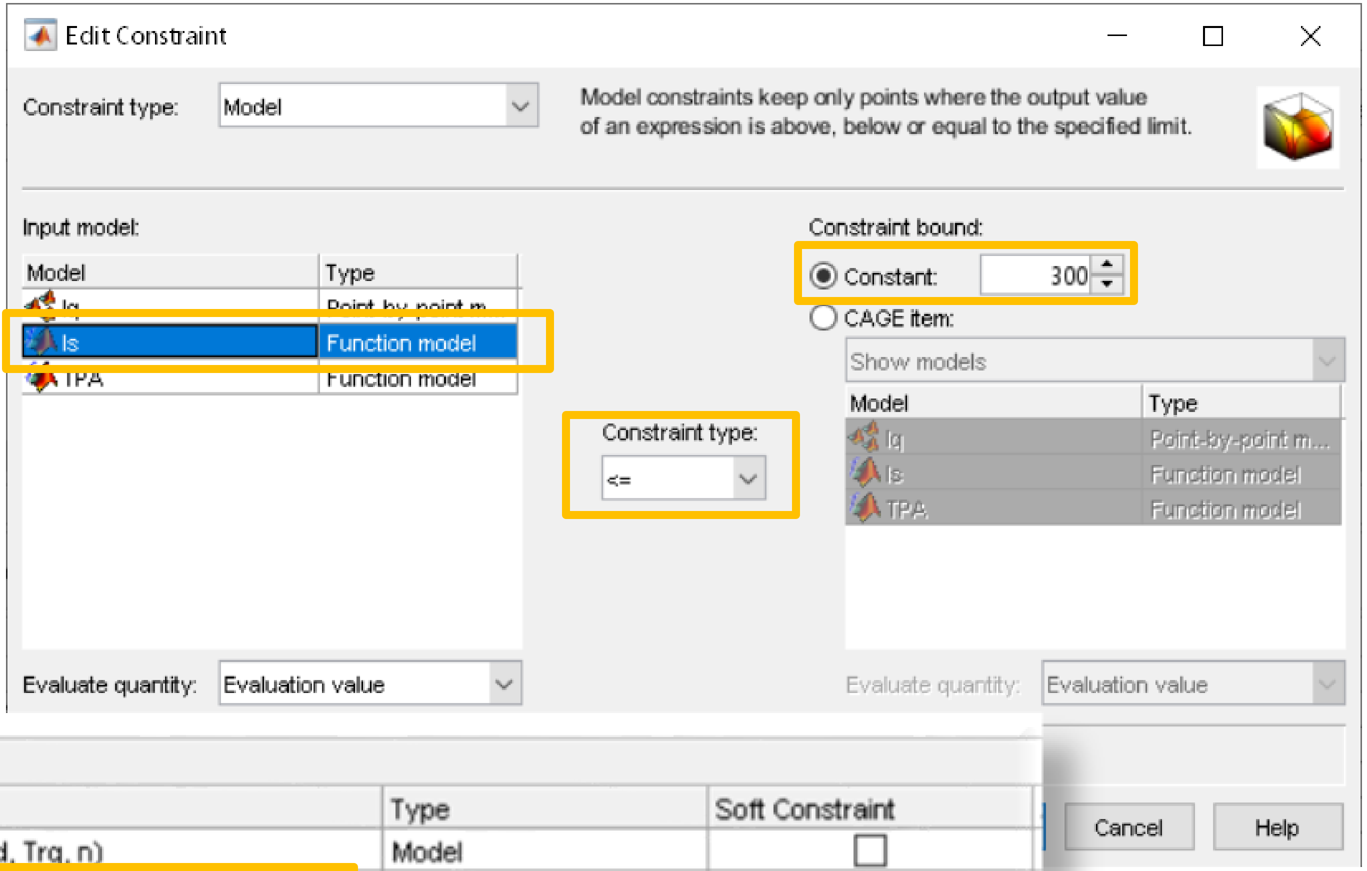


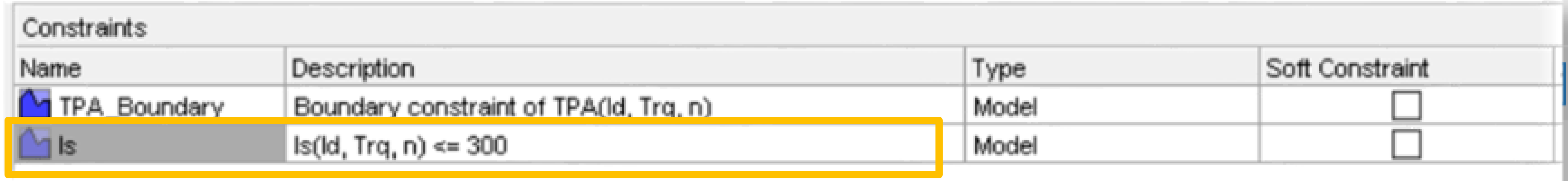
Optimal calibration with experiment data

Look-up tables for flux-based motor controller

- Optimization
  - Add Constraints & Run



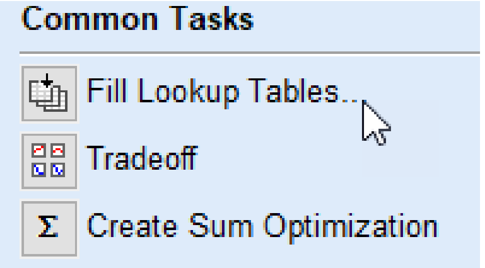


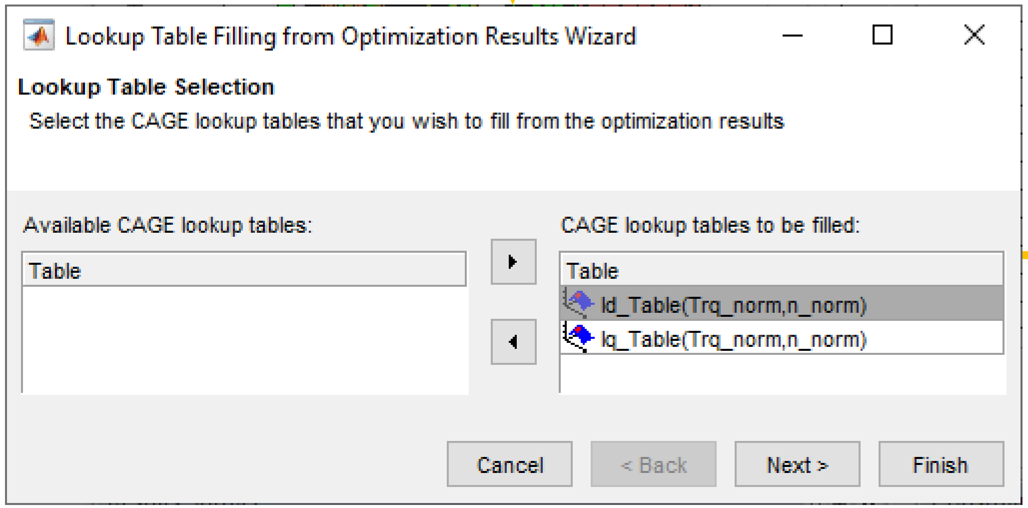


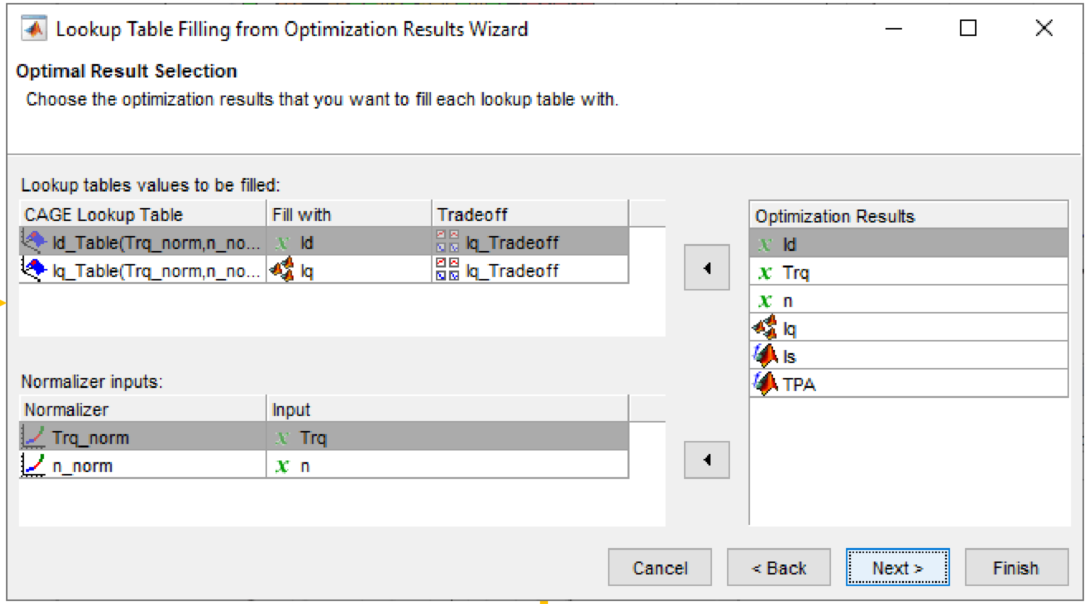
Optimal calibration with experiment data

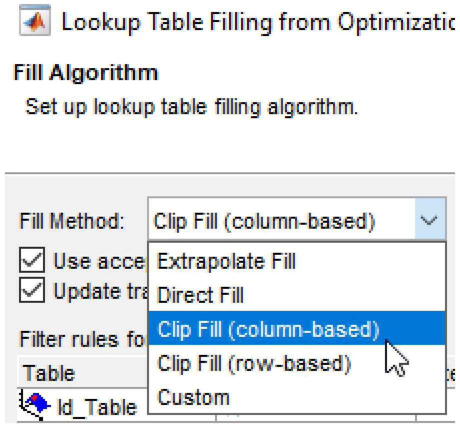
Look-up tables for flux-based motor controller

- Fill and export LUT





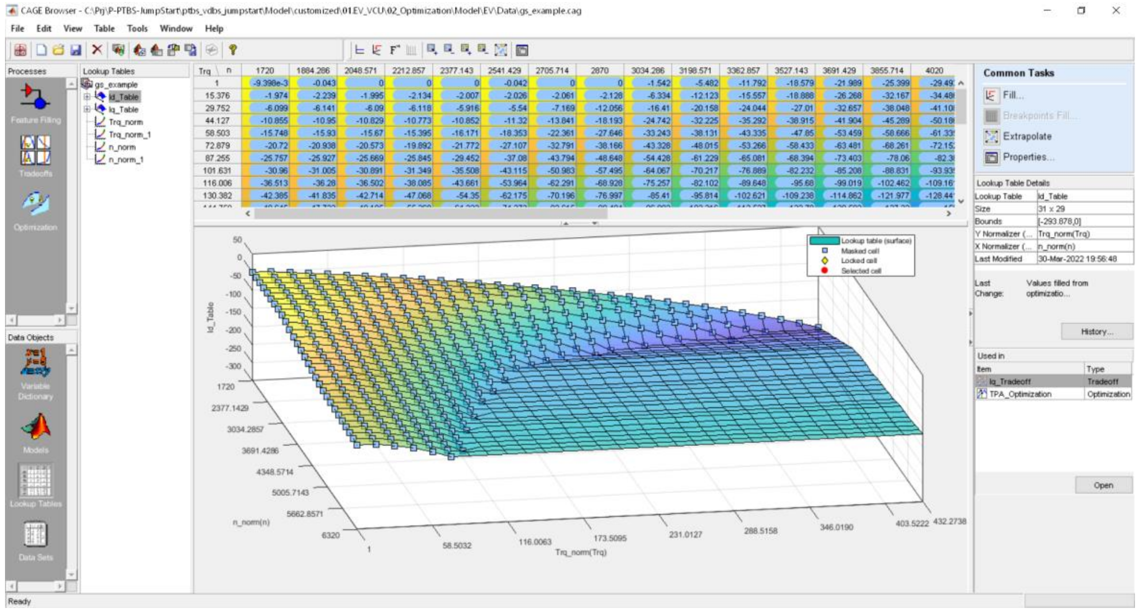


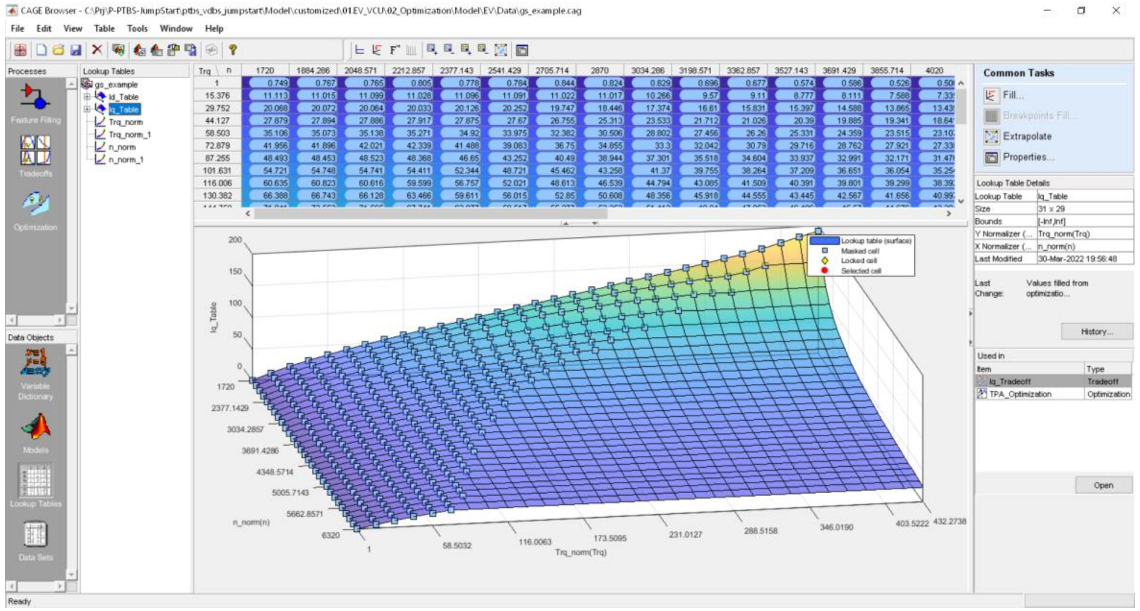


Optimal calibration with experiment data

Look-up tables for flux-based motor controller

- Fill and export LUT
  - Examine LUT

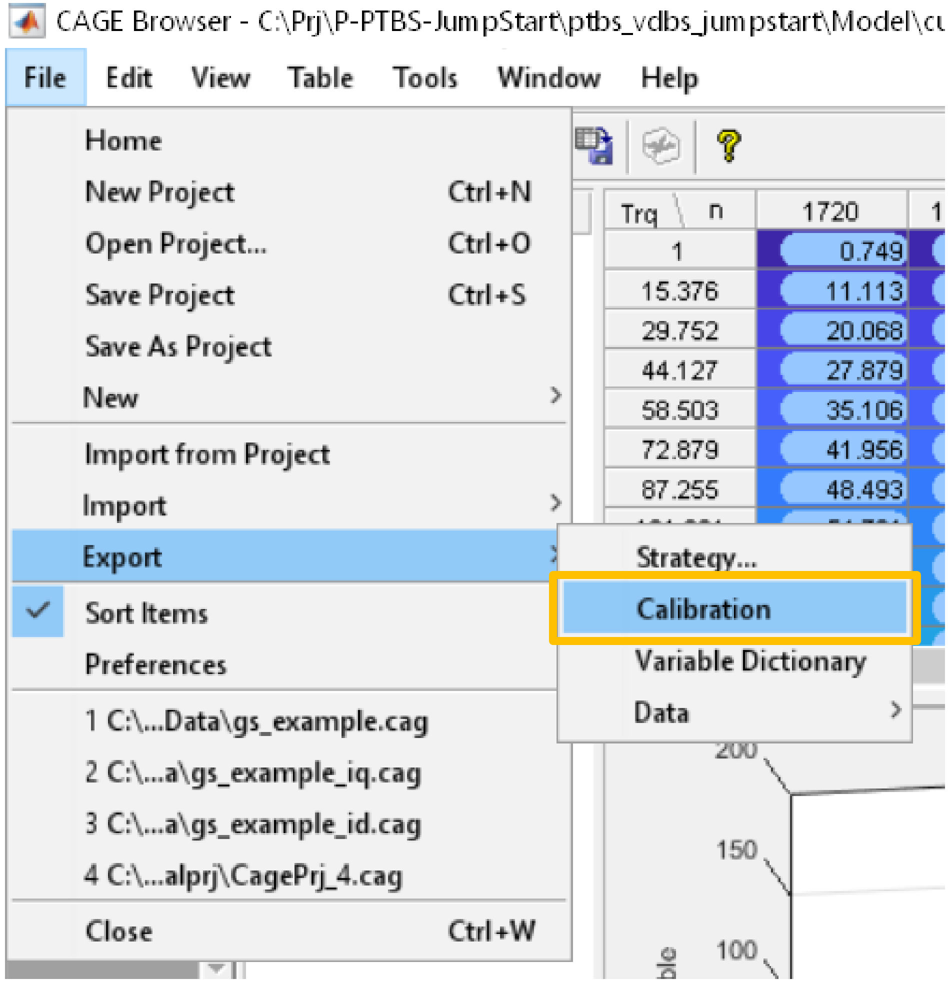


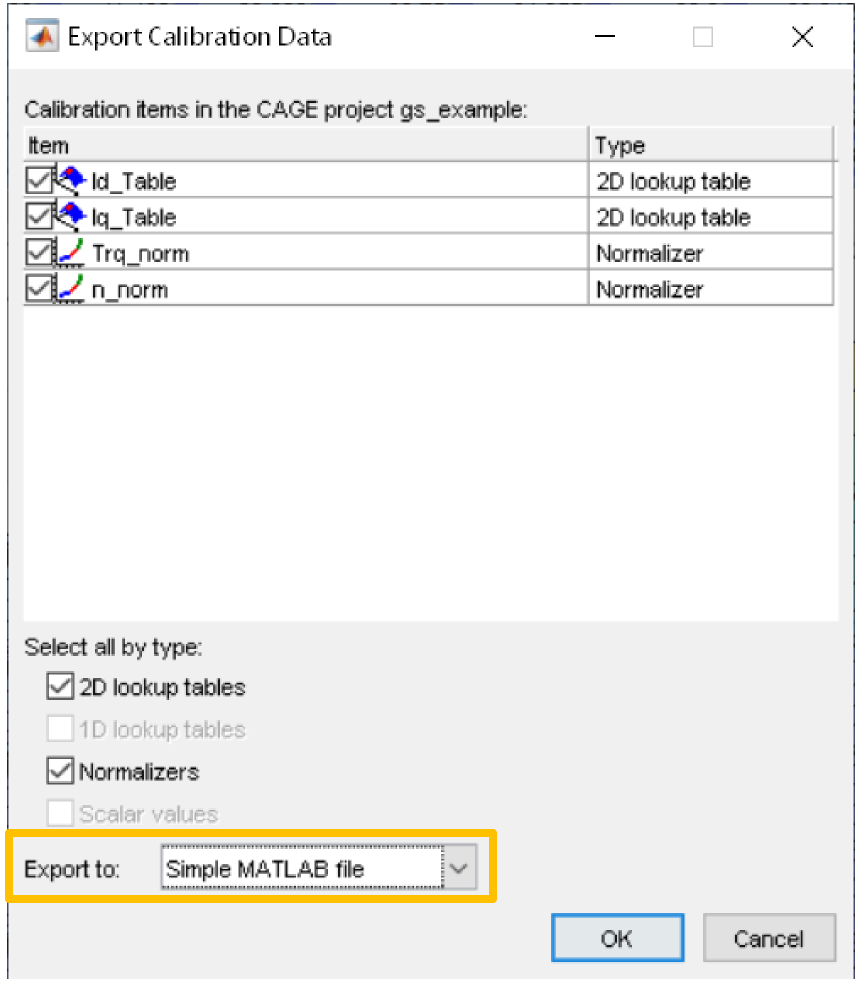


Optimal calibration with experiment data

Look-up tables for flux-based motor controller

- Fill and export LUT





Conclusions

- Has shown **how to apply "Global Optimization Toolbox" to parameter estimation** for physical modeling
  - **The optimization workflow** "Sensitivity Analyzer" and "Global Optimization Toolbox" combined has shown how to **make parameter estimation problem much more efficient**
- Has shown **how to create optimized lookup table with "Model-Based Toolbox"**
  - From experiment data import / preprocessing / fitting model / table optimization
- **These workflows can be modified easily to be applicable to other kinds of optimization problems** beyond the example shown above

MATLAB EXPO

Thank you

MathWorks®