Constrained Model Identification Using Open-Equation Nonlinear Optimization

Sarah Nikbakhsh, John D. Hedengren, Jackson Udy
Brigham Young University

Mark Darby
CMiD Solutions

Outline

- Overview of Linear Model Identification
- Proposed Method to Include Constraints in Linear Model ID
- Case Study 1: Arduino Temperature Control
- Case Study 2: Reservoir Enhanced Oil Recovery
- Conclusions and Recommendations

General Model Identification Form

- $A(q)y(q) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}e(k)$
  - $A(q) = 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}$
  - $B(q) = b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b}$
  - $C(q) = 1 + c_1 q^{-1} + \cdots + c_{n_c} q^{-n_c}$
  - $D(q) = 1 + d_1 q^{-1} + \cdots + d_{n_d} q^{-n_d}$
  - $F(q) = 1 + f_1 q^{-1} + \cdots + f_{n_f} q^{-n_f}$

FIR: $A = F = C = D = 1$

ARX: $F = C = D = 1$

OE (output error): $A = C = D = 1$

ARMAX: $F = D = 1$

BJ (Box-Jenkins): $A = 1$

Model Identification in Industry

- Applications: Identifying linear models for:
  - Model Predictive Control (MPC)
  - Determining - or optimizing - tuning parameters for PID loops

- Linear problems (in the parameters)
  - FIR, ARX, Subspace

- Process model knowledge (gains, gain ratios, deadtime, etc.)
  - Historically imposed after ID step
  - Recent advances: Ability to impose linear constraints

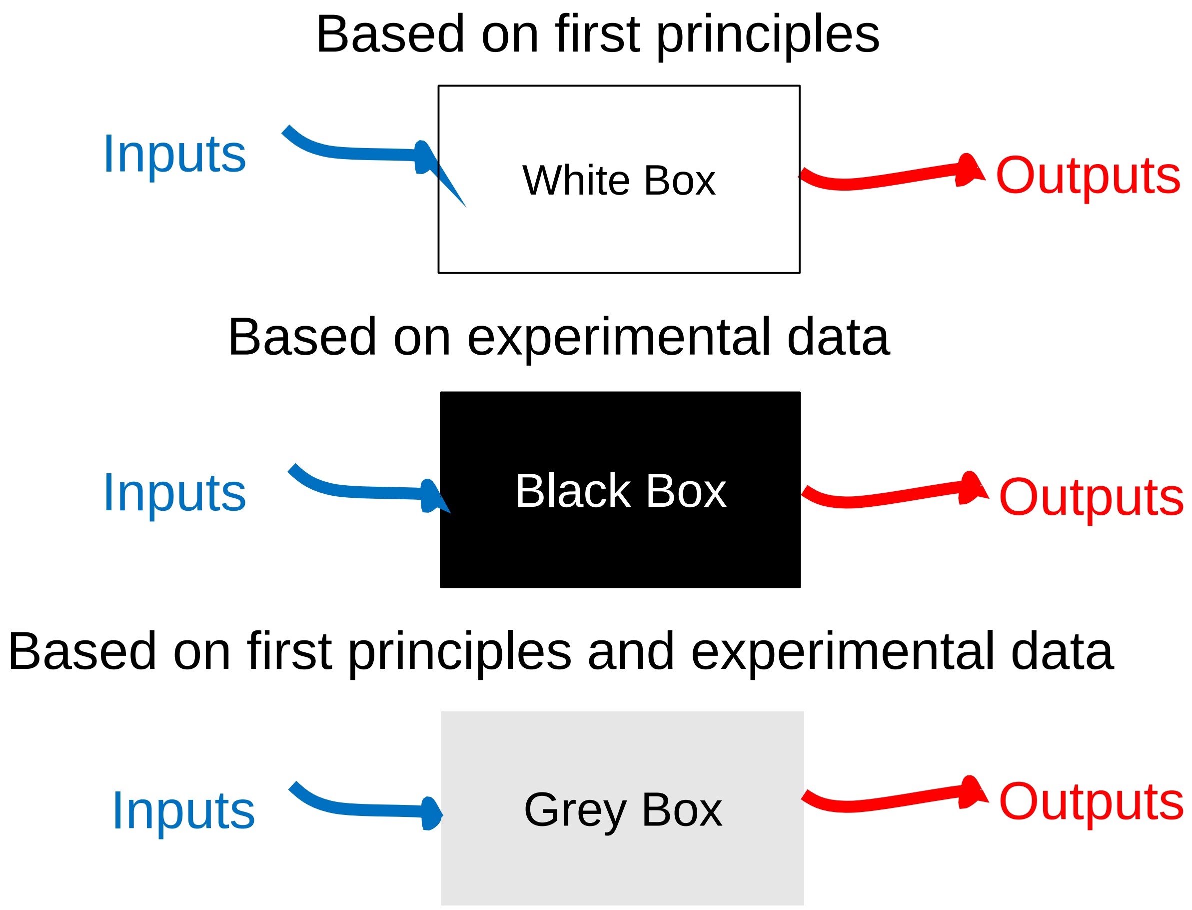
Motivation for Adding Constraints

- Less testing/data required to fit model
- More accurate models → Better performing controllers
- But – Why limit to linear constraints?

Why not allow general – nonlinear constraints and use latest solver techniques?

Model Identification Improvements

- Increase problem formulation flexibility with nonlinear-open equation solvers
- Require less data than unconstrained ID to achieve comparable quality models
- Enforce constraints and known relationships during the identification (gain ratios, RGA, stability, etc.), not as a post-processing step



Nonlinear Optimization Method

- Estimation
  - ℓ_1-norm Objective
  - Process Data in Batch or Real-time
- Solvers and Models
  - Large-scale (100,000+ variables)
  - Mixed Integer Nonlinear Programming
  - Differential Algebraic Equations
  - IPOPT, APOPT, BPOPT Solvers
- APMonitor Optimization Suite
  - MATLAB
  - Python
  - Julia

$$\min_{d} \Phi = w_m^T(e_U + e_L) + w_p^T(c_U + c_L)$$

$$\text{s.t.} \quad 0 = f\left(\frac{\partial x}{\partial t}, x, u, p, d\right)$$

$$0 = g(y, x, u, d)$$

$$0 \leq h(x, u, d)$$

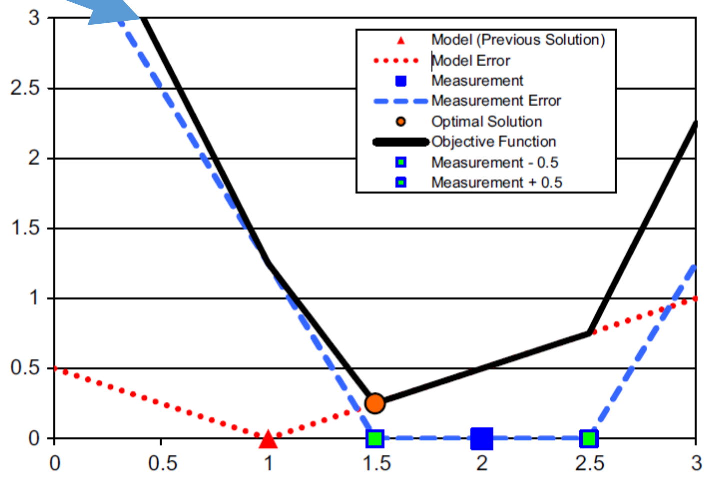
$$e_U \geq y - y_U$$

$$e_L \geq y_L - y$$

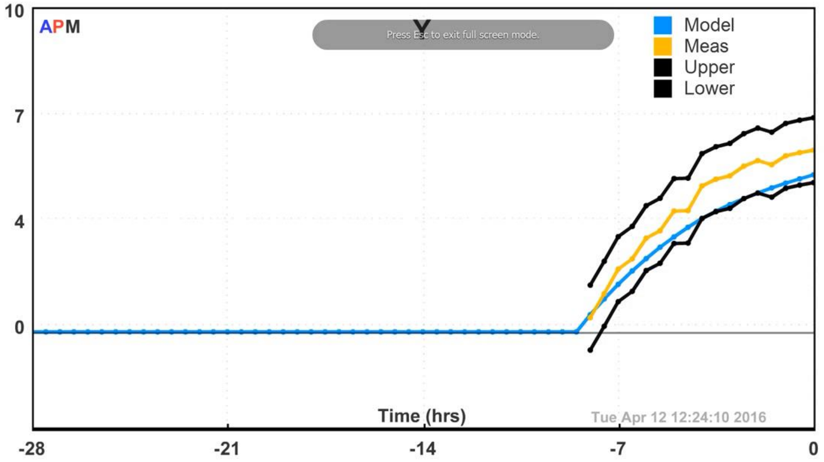
$$c_U \geq y - \hat{y}$$

$$c_L \geq \hat{y} - y$$

$$e_U, e_L, c_U, c_L \geq 0$$



Moving Horizon Estimation (MHE) with Dead-Band

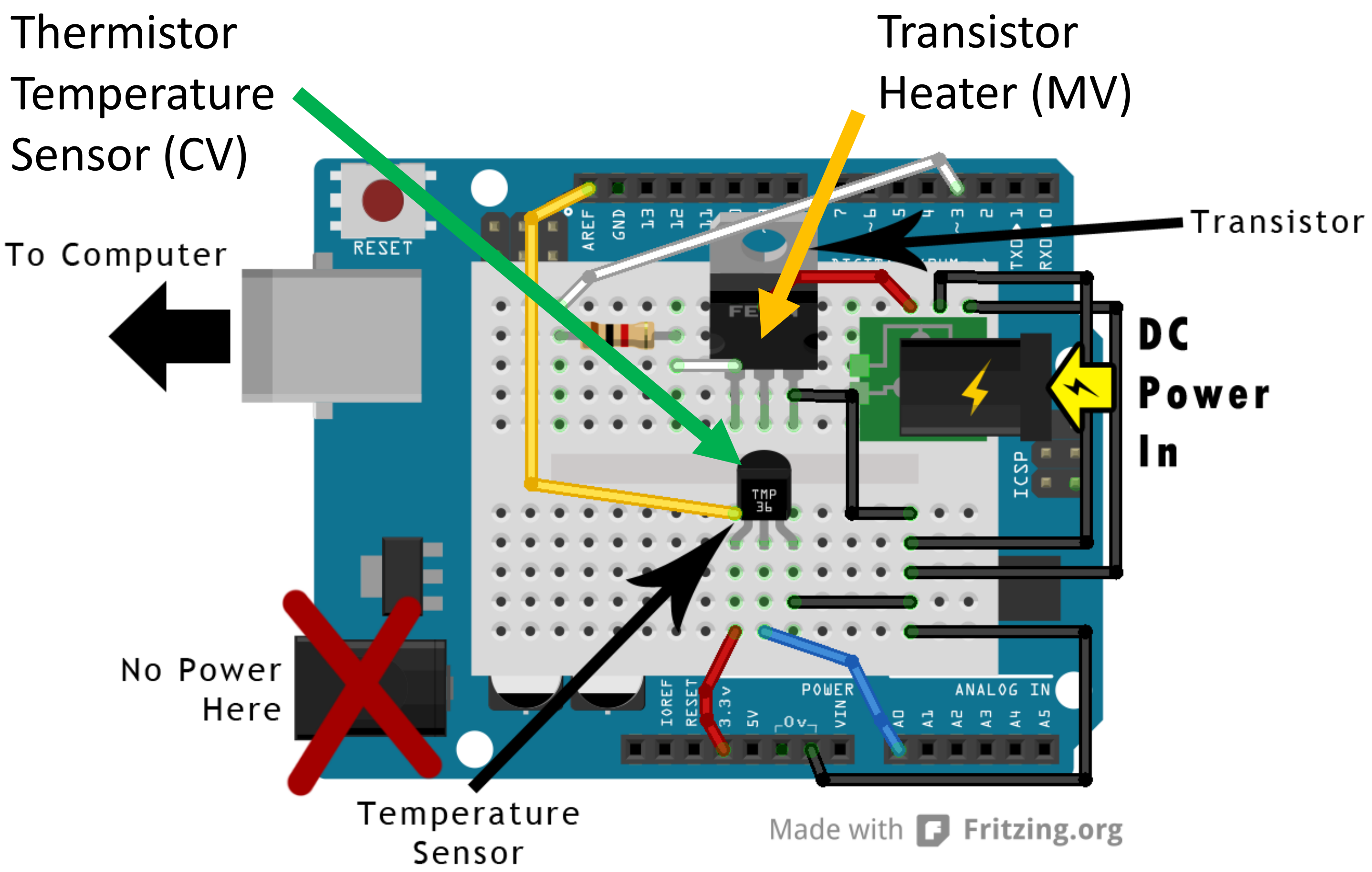


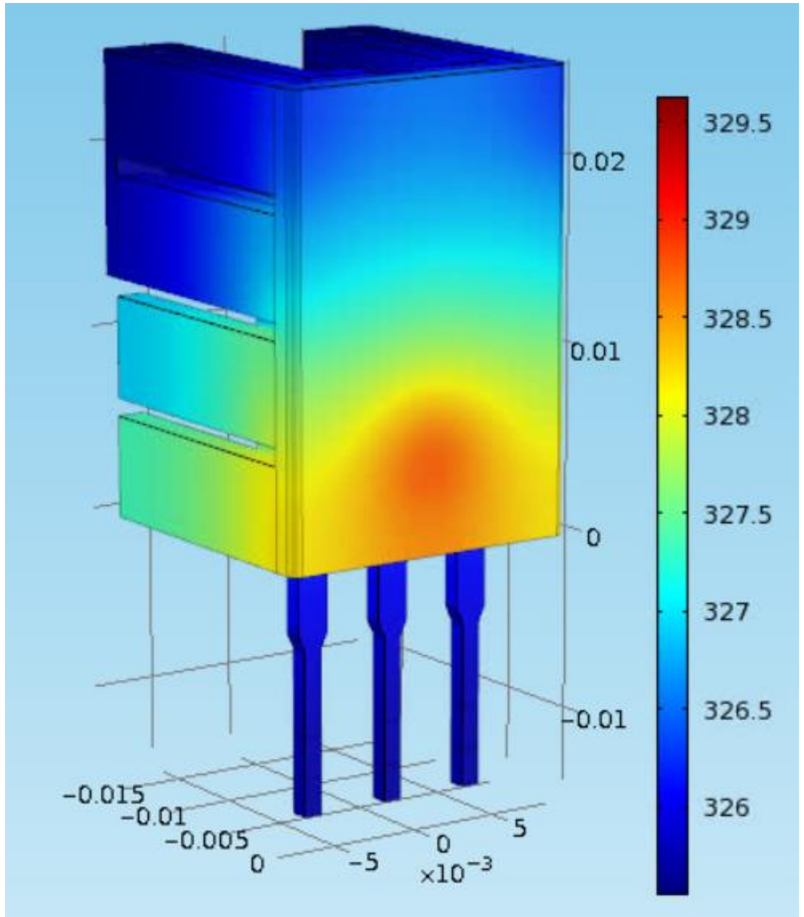
Case Study: Arduino Temperature Control

Objectives

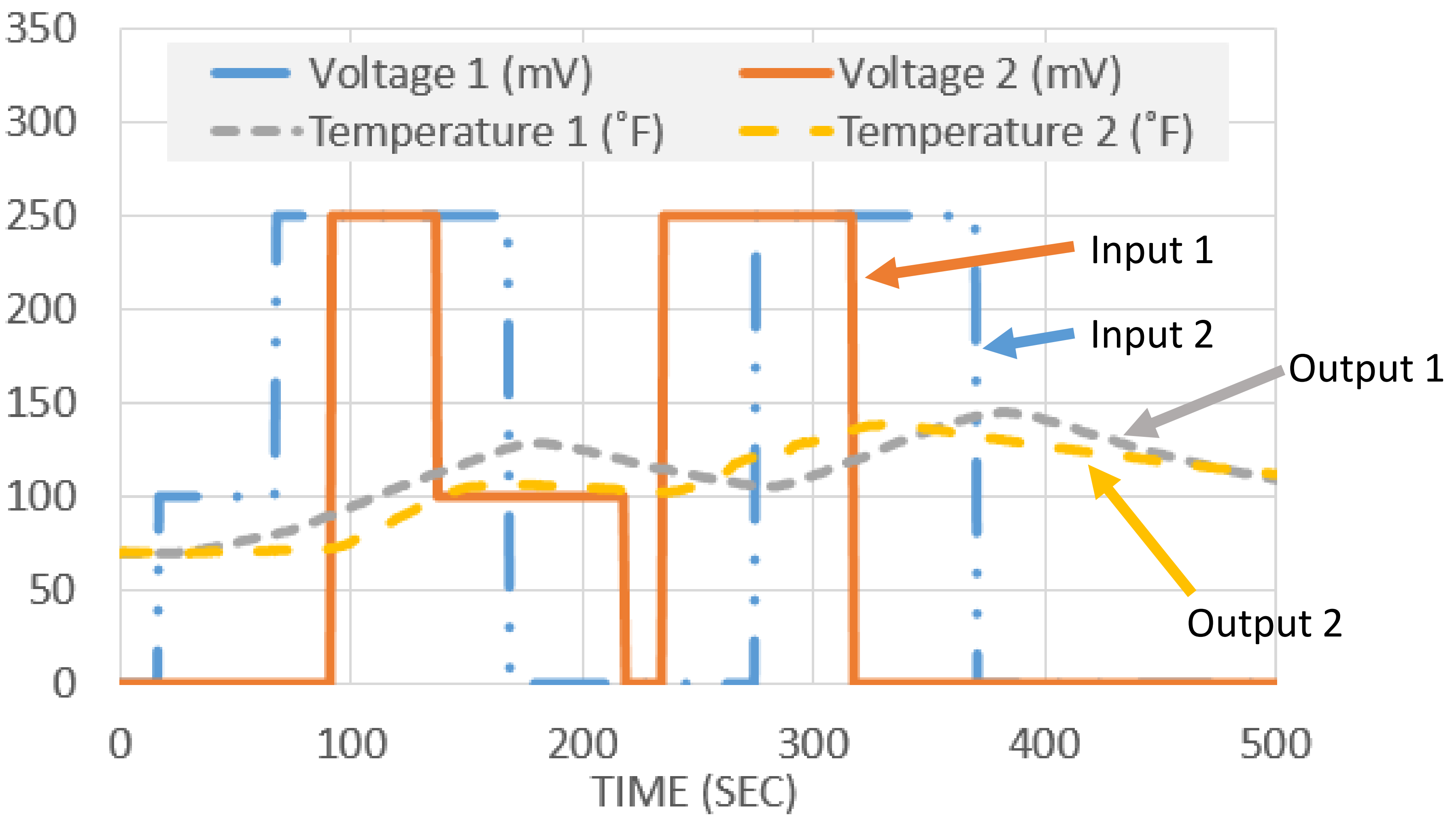
- Demonstrate linear ID with simple 2x2 MIMO system
- Determine effect of constraints with limited data
- Identify models with microcontroller (Arduino)
  - Used at BYU as a Process Control Lab
  - Future: MHE / MPC for Internet of Things (IoT) Applications

Zoned Temperature Control Test (Arduino)

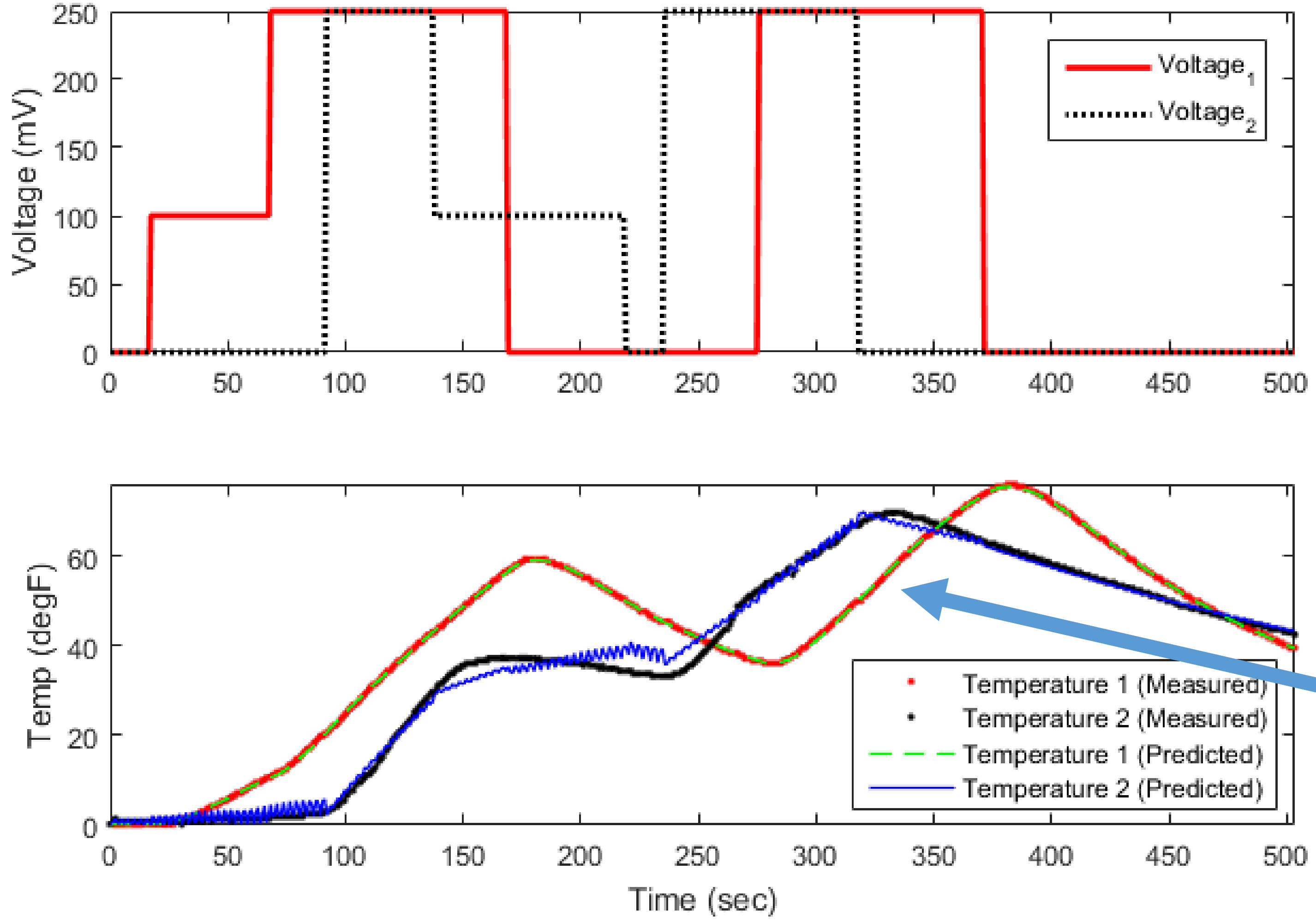


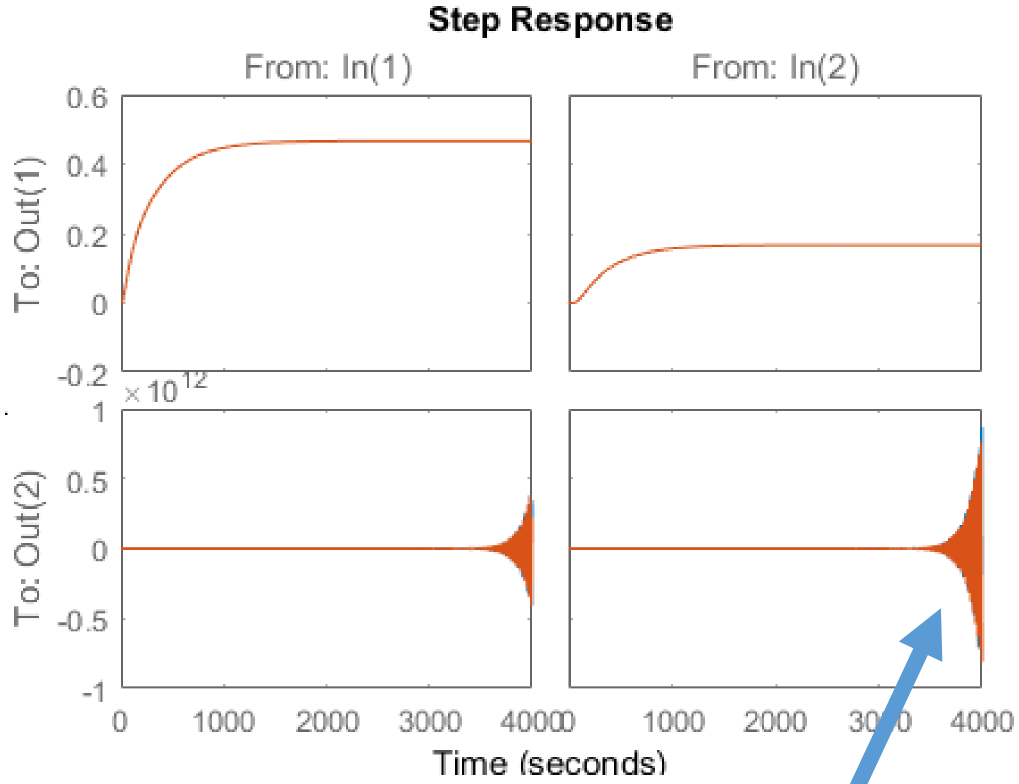


Limited Data for MIMO Identification



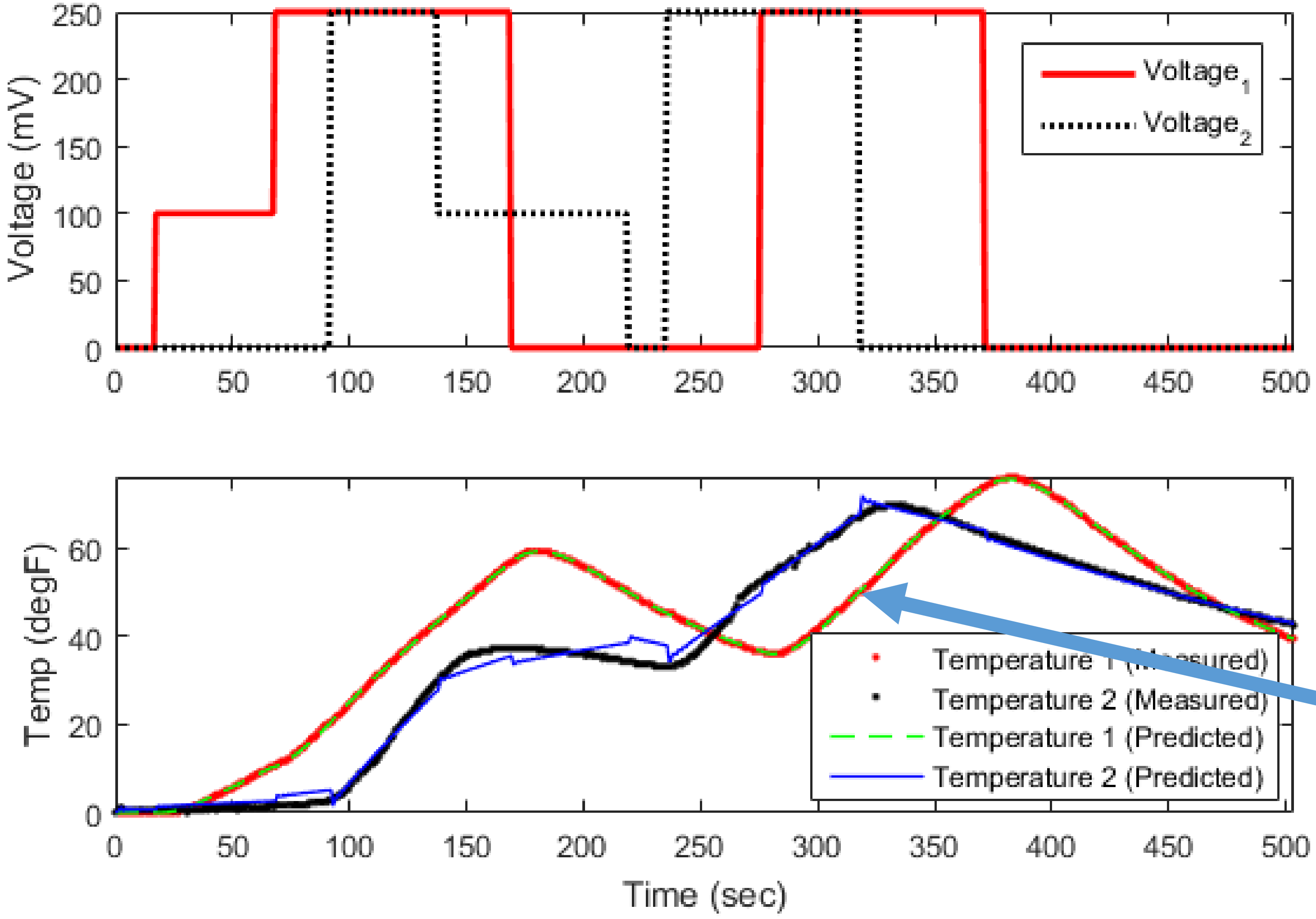
Identification Results, Unconstrained

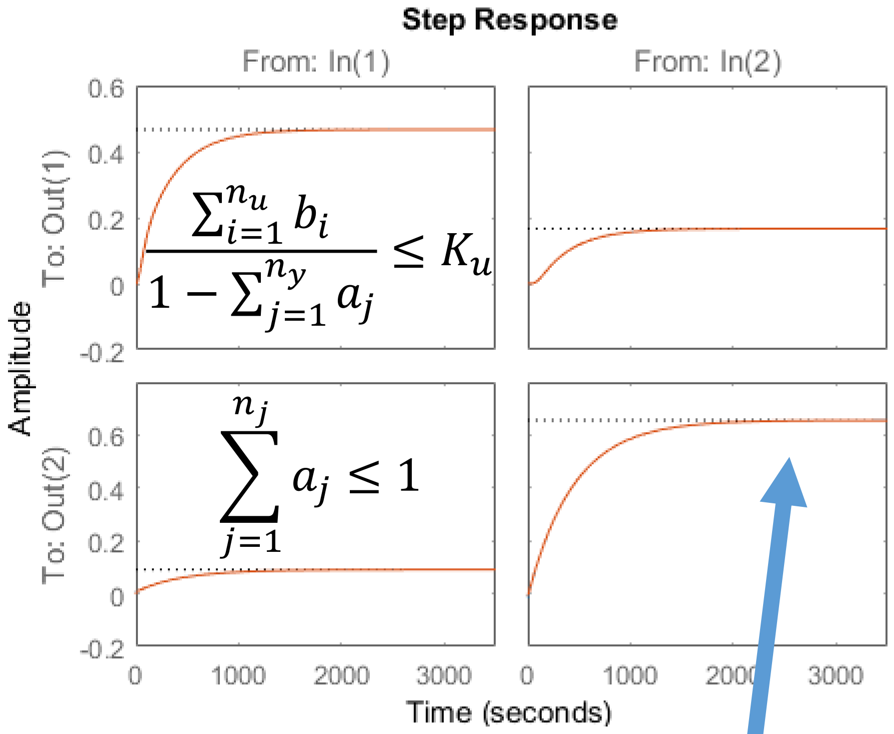




Excellent Fit, but
Unstable Open Loop Step Response

Identification Results, Constrained





Excellent Fit, and
Stable Open Loop Step Response

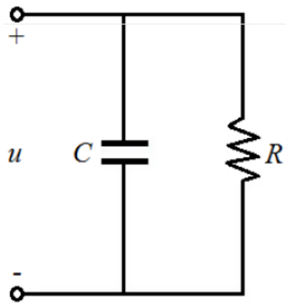
Case Study: Enhanced Oil Recovery

Objectives

- Demonstrate linear ID with high-fidelity simulators
- Investigate scaling of methods to larger scale systems
- Combine linear and nonlinear identification

Injection-Production Modeling, Water-flood Reservoir

Analogous Systems:

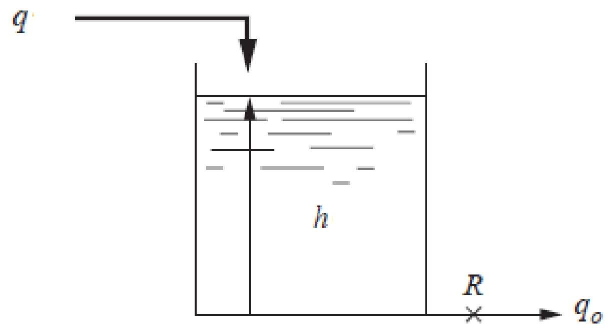


RC circuit

$$I = \frac{\Delta u}{R}$$

Porous media

$$q = \frac{\Delta P}{R}$$



Level tank

$$q_o = \frac{h}{R}$$

CRMIP: capacitance-resistance model (mass balance for each injector-producer pair)

m injection rates

Reservoir systems with *m* injectors and *n* producers

n production rates

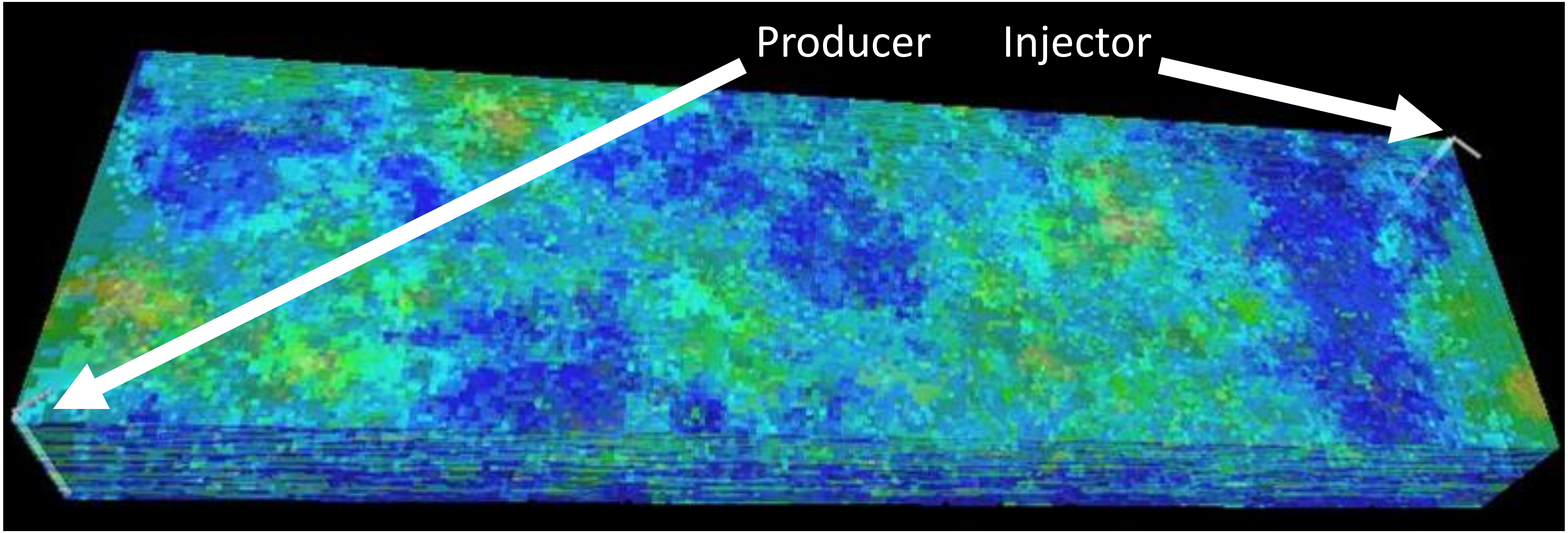
n Bottom Hole Pressure changes

$$q_{ij}(t) + \tau_{ij}\frac{dq_{ij}(t)}{dt} = f_{ij}I_i(t) - J_{ij}\tau_{ij}\frac{d\,BHP_j(t)}{dt}$$

$$i = 1 \cdots m$$
$$j = 1 \cdots n$$

EOR Injection Optimization

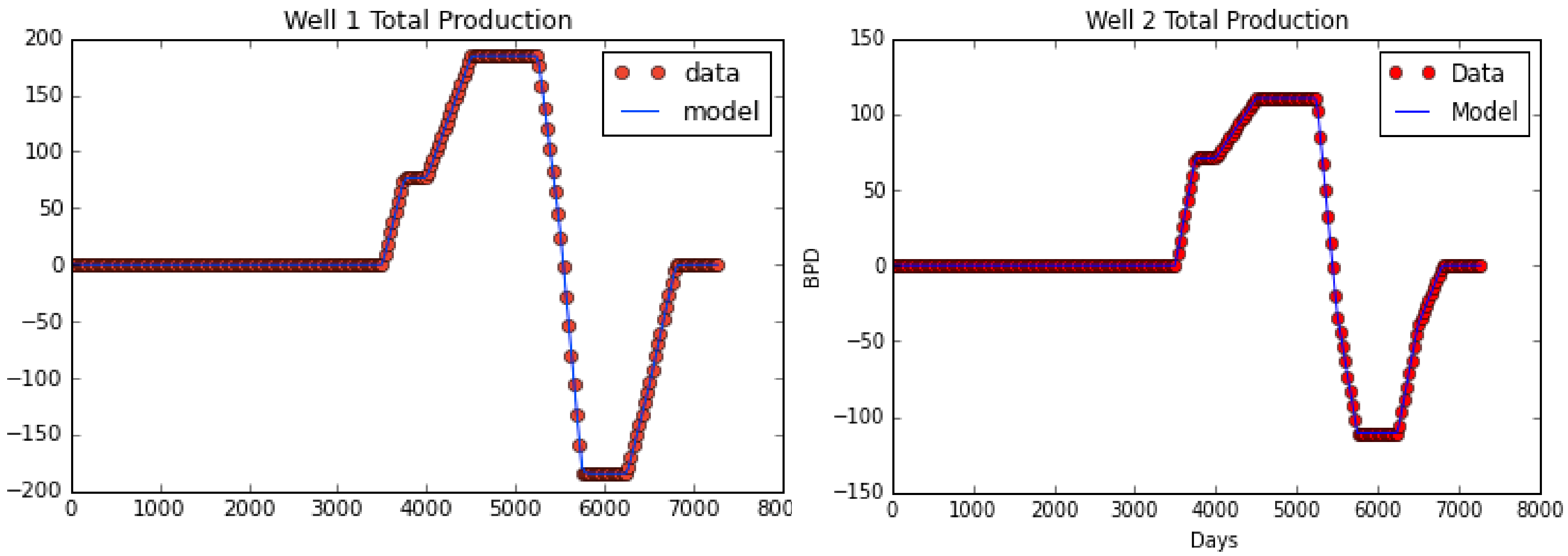
- SPE10 Benchmark, 10,000,000 Cells, CMG Simulator
- Benchmark for comparing upgridding and upscaling approaches
- Ability to predict performance of a waterflood



3 Studies

- 2 injector, 2 producer field
  - Linear and non-linear estimation with constraints
- 4 injector 4 producer field
  - Linear estimation with constraints
- 8 injector 8 producer field
  - Linear estimation with constraints

Results Small 2x2 Field Linear Estimation



Constraints

$$\sum_{i=1}^{n_y} K_i \leq 1$$

$$\tau_i \geq 0$$

$$K = \frac{\sum_{i=1}^{n_u} b_i}{1 - \sum_{j=1}^{n_y} a_j}$$

Solve Time	Constrained	Unconstrained
APOPT	8.3 seconds	0.7 seconds
IPOPT	2.5 seconds	1.3 seconds
BPOPT	2.7 seconds	1.9 seconds

Model Size	2x2
State Variables	1328
Equations	1320
Estimated Parameters	8
Slack Variables	132

Fractional Flow Model (Nonlinear)

- Determine Oil-Water Ratio.
- Estimate a and b parameters
- Nonlinear estimation
- Constraint, b > 0
  - Ensures water-oil ratio increases as injection increases.

$$qo_j(t) = \frac{1}{1 + a_j CWI_j^{b_j}} q_j$$

Total liquid produced at well

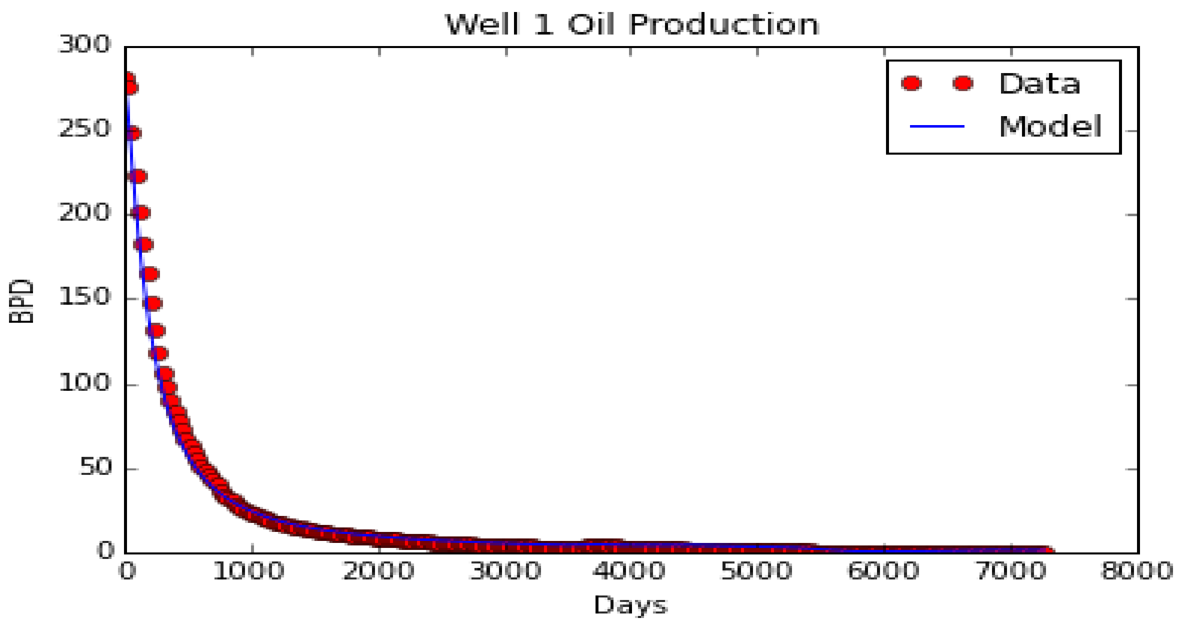
Oil produced at well

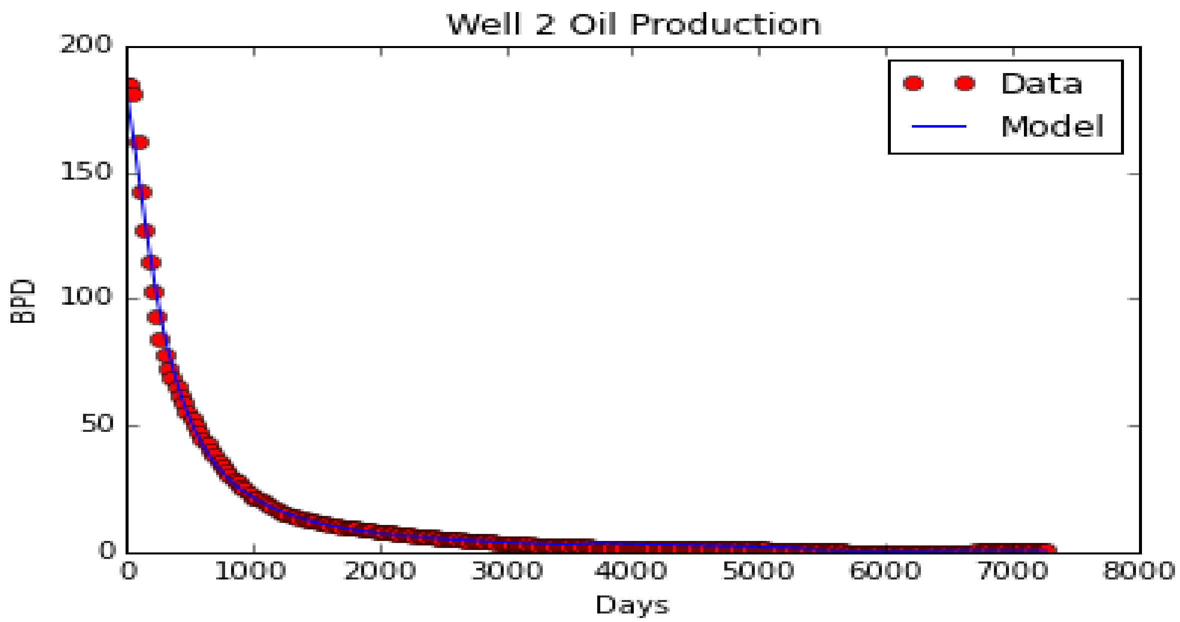
Unknown Parameter

Cumulative Water Injected

Unknown Parameter

Fractional Flow Results (Nonlinear)



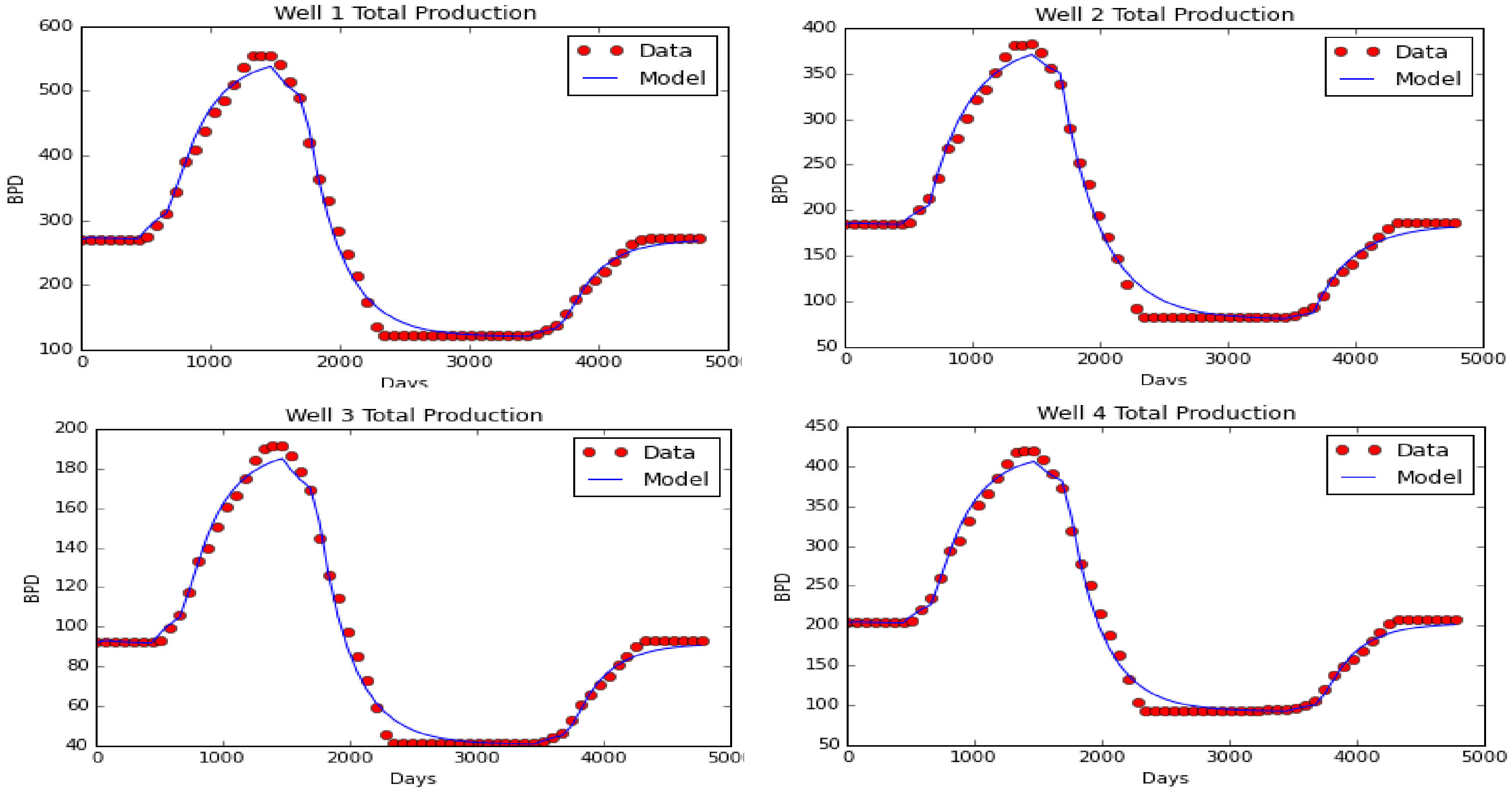


Solve Time	
APOPT	34.9 seconds
IPOPT	4.5 seconds
BPOPT	1.6 seconds

Model Size	2x2
State Variables	6692
Equations	6694
Estimated Parameters	4
Slack Variables	0

- Good convergence with both solvers
- IPOPT significantly faster than APOPT solver.

4 Injector 4 Producer Field



Constraints

$$\sum_{i=1}^{n_y} K_i \leq 1$$

$$\tau_i \geq 0$$

$$K = \frac{\sum_{i=1}^{n_u} b_i}{1 - \sum_{j=1}^{n_y} a_j}$$

Solve Time	Constrained	Unconstrained
APOPT	34.9 seconds	19.3 seconds
IPOPT	4.5 seconds	4.18 seconds
BPOPT	7.0 seconds	6.8 seconds

Model Size	2x2
State Variables	3672
Equations	3640
Estimated Parameters	48
Slack Variables	260

4 Injector 4 Producer Estimation Results

Gains	Producer 1	Producer 2	Producer 3	Producer 4	Total
Injector 1	0.222	0.118	0.073	0.128	**.541**
Injector 2	0.078	0.374	0.005	0.085	**.542**
Injector 3	0.310	0.278	0.121	0.292	**1**
Injector 4	0.501	0.001	0.178	0.316	**.996**

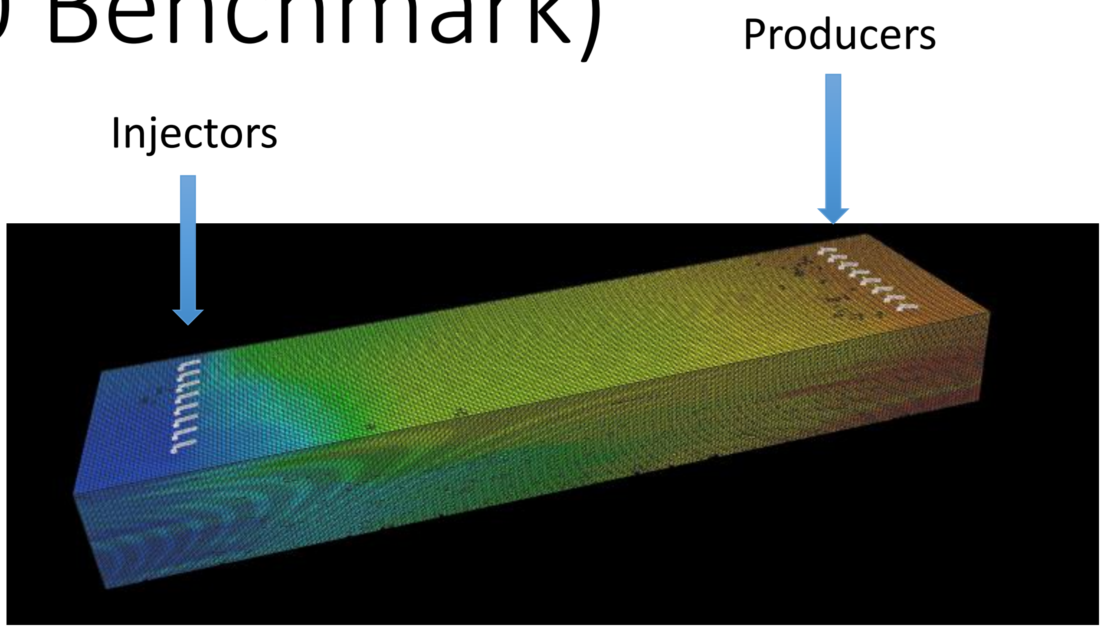
- Injector 1 and 2 have significant losses
- Injector 3 and 4 have high gains (most effective)

Time Constants	Producer 1	Producer 2	Producer 3	Producer 4
Injector 1	294.3	300	291.4	291.1
Injector 2	202.8	256.6	155.4	199.7
Injector 3	250.3	294.7	248.5	250.1
Injector 4	210.2	0.001	205.3	198.2

- Time-constants are typically 155-300 days

Large Scale System (SPE10 Benchmark)

- 8 injector, 8 producer field
- Data simulated with CMG Reservoir Simulator
  - 12 hour simulation time on laptop
- Reduced Order Model Parameter Estimation
  - Constrained estimation
  - IPOPT significantly faster than APOPT solver for large fields

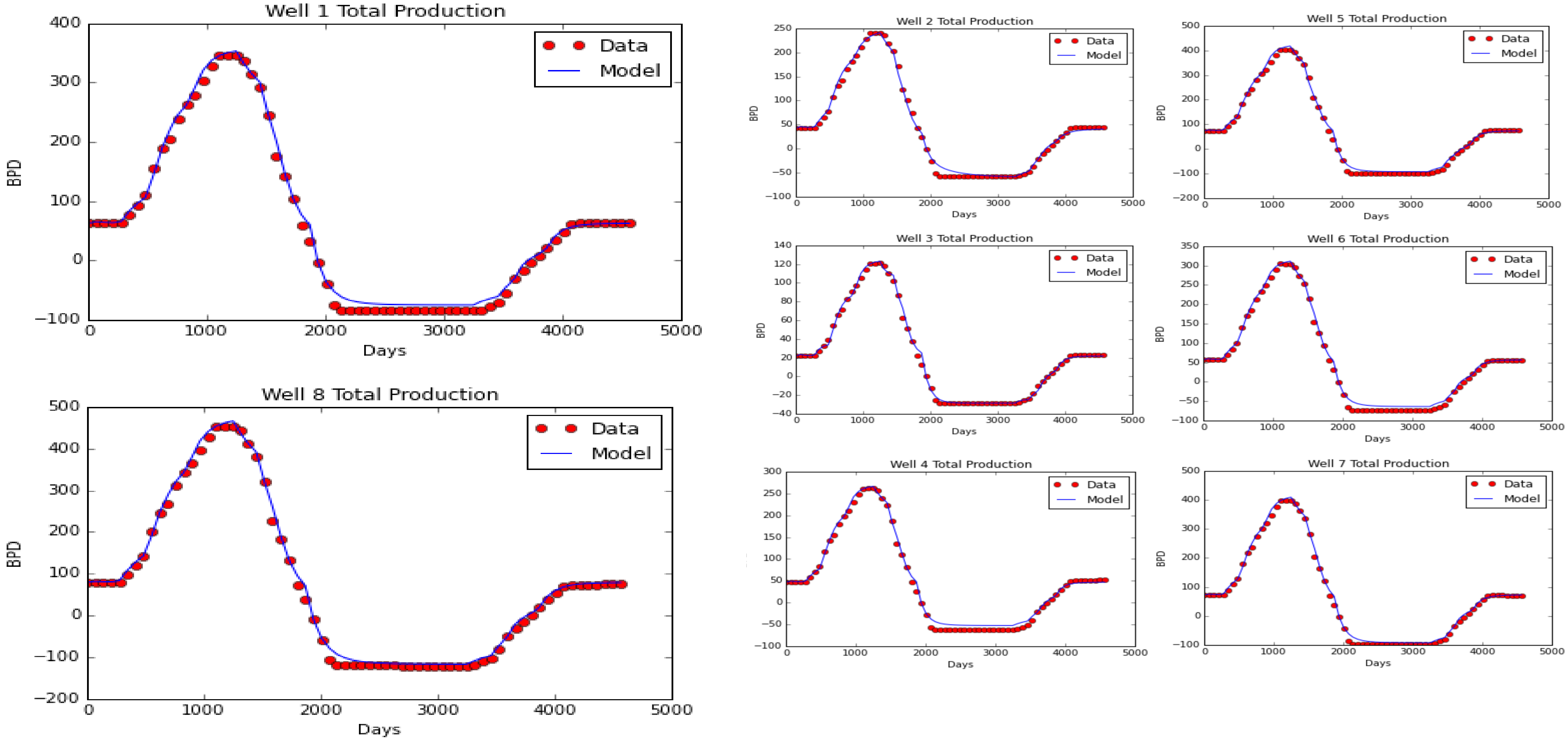


Solve Time	Constrained	Non-constrained
APOPT	32 seconds	56 seconds
IPOPT	34 seconds	4.8 seconds
BPOPT	Did not Converge	Did not converge

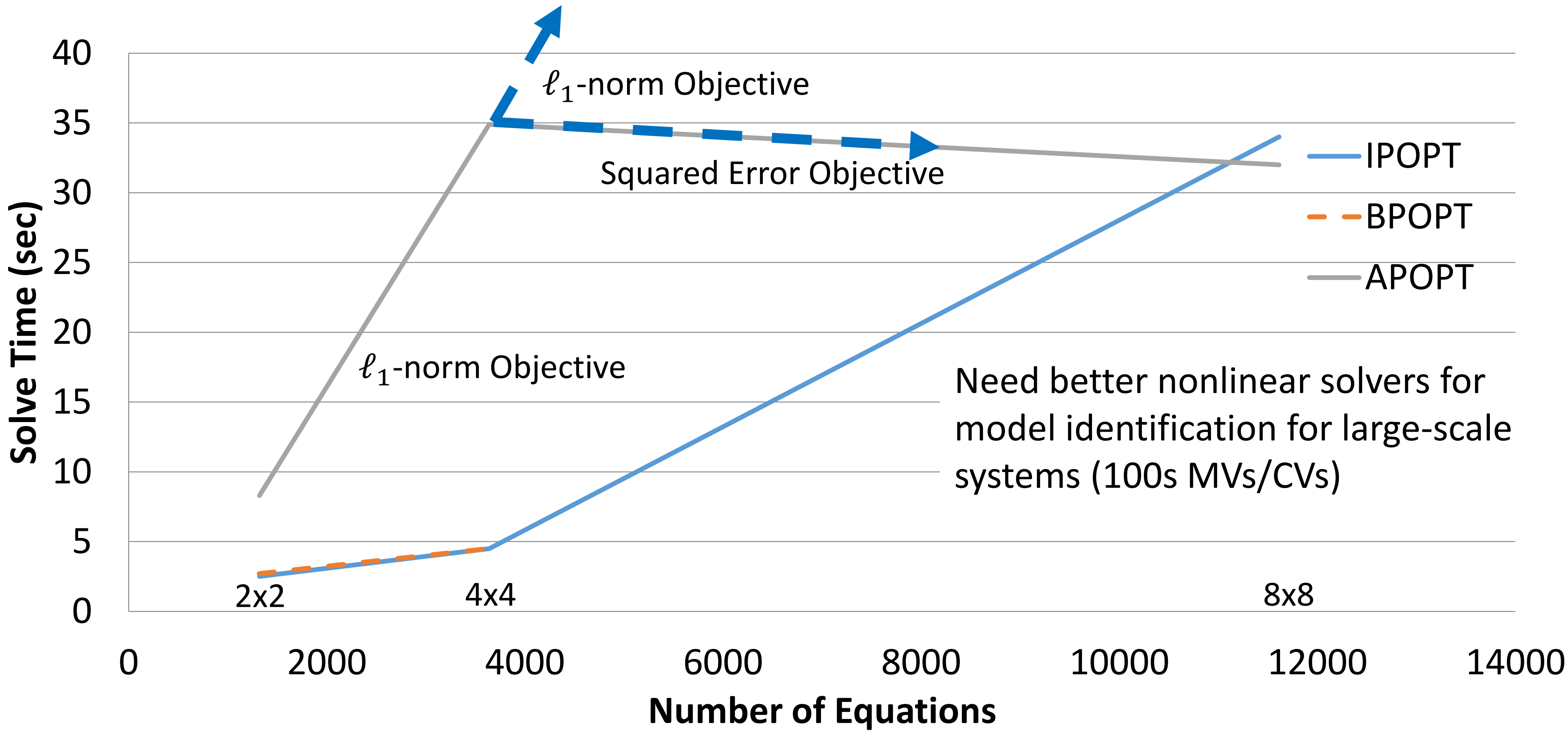
* Squared Error Objective Function

Model Size	2x2
State Variables	11744
Equations	11616
Estimated Parameters	128
Slack Variables	528

8 Injector 8 Producer Results



Solver Computation Time



- Largest field estimated with squared error objective function (reduced problem size)
- BPOPT did not converge on largest problem, comparable to IPOPT on 2x2 and 4x4 estimation problems

Current & Future Work

- Evaluate improvements: Accuracy and reduction in data
  - Use of normal operating data?

- Impose additional constraints – such as
  - Relative gain array (RGA) based on steady-state gains
  - Known geometries and relationships
    - Vessel-hold ups
    - Valve-flow relationships
    - Simplified process models (distillation, HEX, etc.)

Additional Slides

Gains	Producer 1	Producer 2	Producer 3	Producer 4	Producer 5	Producer 6	Producer 7	Producer 8	Total
Injector 1	0.00	0.266	0.035	0.141	0.00	0.00	0.244	.315	**1.00**
Injector 2	0.120	0.162	0.001	0.001	0.155	0.072	0.132	0.360	**1.00**
Injector 3	0.107	0.00	0.072	0.191	0.117	0.098	0.186	0.228	**1.00**
Injector 4	0.168	0.00	0.135	0.162	0.188	0.137	0.156	0.055	**1.00**
Injector 5	0.180	0.00	0.22	0.061	0.191	0.162	0.170	0.131	**0.917**
Injector 6	0.215	0.001	0.000	0.000	0.273	0.210	0.000	0.000	**0.698**
Injector 7	0.130	0.000	0.081	0.106	0.170	0.094	0.128	0.125	**.704**
Injector 8	0.139	0.000	0.031	0.095	0.149	0.137	0.188	0.261	**1.00**