Verified Parameter Identification for Solid Oxide Fuel Cells

Ekaterina Auer, Stefan Kiel, Andreas Rauh

University of Duisburg-Essen, University of Rostock

June 14, 2012

Modeling, Simulation and Control of Solid Oxide Fuel Cells

SOFCs: devices converting chemical energy in electricity

+ high efficiency, flexibility wrt. fuel
– high operating temperature

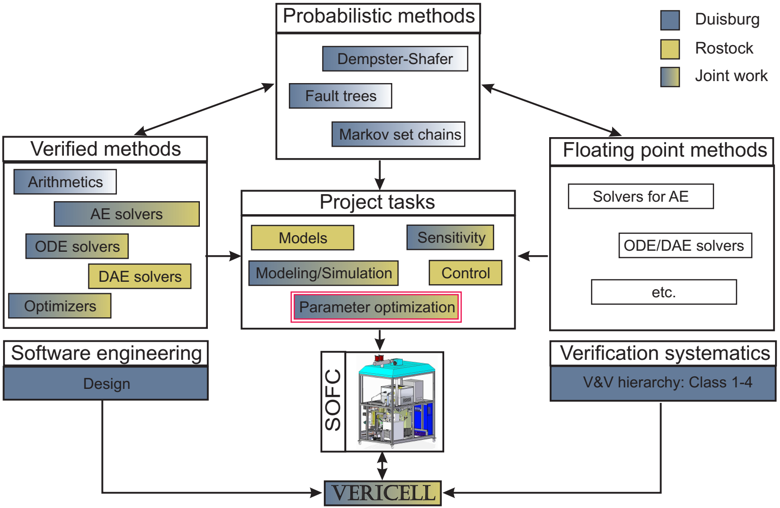
Techniques: Linear controllers (+ stationary NL characteristics)

– High complexity of models
– Intended for stationary operating points

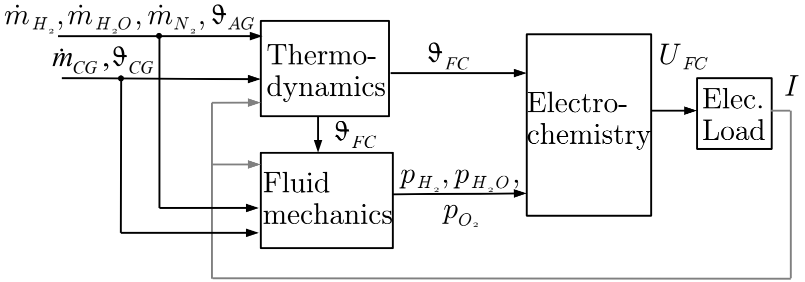
Our goals:

- Models better suitable for control
- Verified methods for robustness
- Modeling/simulation/control in VeriCell

Project VerIPC-SOFC



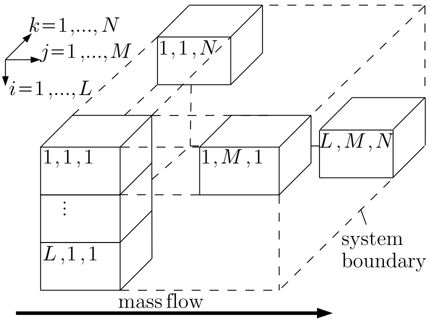
A Conceptual Model for SOFCs



A. Rauh, T. Doetschel, E. Auer, and H. Aschemann. *Interval methods for control-oriented modeling of the thermal behavior of high-temperature fuel cell stacks.* In In Proc. of SysID 2012, 2012. Accepted.

Control-Oriented ODE-Based SOFC Models: Thermodynamical Subsystem

PDEs $\longrightarrow$ ODEs



Models $(L \times M \times N)$

- $1 \times 1 \times 1$
- $3 \times 1 \times 1$
- $3 \times 3 \times 1$

ODEs for the $1 \times 1 \times 1$ Model

$$
\begin{aligned}
\dot{\vartheta}_{FC} &= \dot{m}_{H_2} \cdot \left(p_{\Delta H,2} \cdot \vartheta_{FC}^2 + p_{\Delta H,1} \cdot \vartheta_{FC} + p_{\Delta H,0}\right) + 6 \cdot p_A \cdot (\vartheta_A - \vartheta_{FC}) + (\vartheta_{AG} - \vartheta_{FC}) \\
&\cdot \quad (\dot{m}_{H_2} \cdot (p_{H_2,2} \cdot \vartheta_{FC}^2 + p_{H_2,1} \cdot \vartheta_{FC} + p_{H_2,0}) \\
&+ \quad \dot{m}_{H_2O} \cdot (p_{H_2O,2} \cdot \vartheta_{FC}^2 + p_{H_2O,1} \cdot \vartheta_{FC} + p_{H_2O,0}) \\
&+ \quad \dot{m}_{N_2} \cdot (p_{N_2,A,2} \cdot \vartheta_{FC}^2 + p_{N_2,A,1} \cdot \vartheta_{FC} + p_{N_2,A,0})) + I_{FC} \cdot p_{el} - \dot{m}_A \cdot (\vartheta_{FC} - \vartheta_{CG}) \\
&\cdot \quad (77 \cdot p_{N_2,C,0}/100 + 11 \cdot p_{O_2,0}/50 + 77 \cdot p_{N_2,C,1} \cdot \vartheta_{FC}/100 \\
&+ \quad 11 \cdot p_{O_2,1} \cdot \vartheta_{FC}/50 + 77 \cdot p_{N_2,C,2} \cdot \vartheta_{FC}^2/100 + 11 \cdot p_{O_2,2} \cdot \vartheta_{FC}^2/50) \\
&= f(\vartheta_{FC}, p)
\end{aligned}
$$

- Solution: $\vartheta_{FC}(t)$
- Initial condition: $\vartheta_{FC}(0) = 299.7053$ K
- No analytical solution
- Time-dependent inputs $\dot{m}_i$, $\vartheta_{AG,CG,A}$, I_{FC}

ODEs for the $1 \times 1 \times 1$ Model

$$\begin{aligned}
\dot{\vartheta}_{FC} &= \dot{m}_{H_2} \cdot \left(p_{\Delta H,2} \cdot \vartheta_{FC}^2 + p_{\Delta H,1} \cdot \vartheta_{FC} + p_{\Delta H,0}\right) + 6 \cdot p_A \cdot (\vartheta_A - \vartheta_{FC}) + (\vartheta_{AG} - \vartheta_{FC}) \\
&\cdot \quad (\dot{m}_{H_2} \cdot (p_{H_2,2} \cdot \vartheta_{FC}^2 + p_{H_2,1} \cdot \vartheta_{FC} + p_{H_2,0}) \\
&+ \quad \dot{m}_{H_2O} \cdot (p_{H_2O,2} \cdot \vartheta_{FC}^2 + p_{H_2O,1} \cdot \vartheta_{FC} + p_{H_2O,0}) \\
&+ \quad \dot{m}_{N_2} \cdot (p_{N_2,A,2} \cdot \vartheta_{FC}^2 + p_{N_2,A,1} \cdot \vartheta_{FC} + p_{N_2,A,0})) + I_{FC}^2 \cdot p_{el} - \dot{m}_A \cdot (\vartheta_{FC} - \vartheta_{CG}) \\
&\cdot \quad (77 \cdot p_{N_2,C,0}/100 + 11 \cdot p_{O_2,0}/50 + 77 \cdot p_{N_2,C,1} \cdot \vartheta_{FC}/100 \\
&+ \quad 11 \cdot p_{O_2,1} \cdot \vartheta_{FC}/50 + 77 \cdot p_{N_2,C,2} \cdot \vartheta_{FC}^2/100 + 11 \cdot p_{O_2,2} \cdot \vartheta_{FC}^2/50) \\
&= f(\vartheta_{FC}, p)
\end{aligned}$$

Parameters of interest or their approximations (in red)

- heat capacities c
- the heat enthalpy ΔH
- thermal resistances

Parameter Identification for SOFC Systems

General goal: Parameterize the models in a robust and accurate way

Parameters: $p = [p_{H_2,0}\ p_{H_2O,0}\ p_{N_2,A,0}\ p_{N_2,C,0}\ p_{O_2,0}\ p_{\Delta H,0}]$

The rest: replaced by floating point approximations

Reference: a set of measurements* $\{y_m(t_k)\}$ over 19963 seconds

Task: $$\Phi(p) = \sum_{k=1}^{T} \sum_{j=1}^{dim} \left(y_j(t_k,p) - y_{j,m}(t_k)\right)^2 \to \min$$

* obtained using the test rig in Rostock

Parameter Identification for the $1 \times 1 \times 1$ Model

$$\min_{p \in p_0} \Phi(p) = \min_{p \in p_0} \sum_{k=1}^{T} \left(y_k(p) - y_m(t_k)\right)^2$$

$p \rightarrow$ The parameters to identify

$t_{k+1} - t_k \rightarrow$ The step size of 1 second

$p_0 \rightarrow$ The search interval of the width 2

$y_k(p) \rightarrow$ The simulated temperature ϑ_{FC} at time t_k

$y_m(t_k) \rightarrow$ The measured temperature at time $t_k = k$

$T \rightarrow$ The number of measurements (19963)

How to Obtain Reliable $y_k(p)$?

The Euler method

- $\mathbf{y}_k := \mathbf{y}_{k-1} + h \cdot f(\mathbf{y}_{k-1}, \mathbf{p})$
- – "Verified approximation" (rounding)
- – Overestimation
- + Easy derivatives (AD)
- + *Easily portable to the GPU*

Verified IVP solvers

- $y(t_k, p) \in \mathbf{y}_k$
- VNODE-LP, ValEncIA-IVP, etc.
- + Verifies the whole model
- – Derivatives require solving an extra ODE
- – High computational effort

Currently, we use the Euler method only.

Problem At Hand

- A bound constraint problem ($\mathbf{p}_0 \in \mathbf{IR}^6, \mathrm{w}(\mathbf{p}_0) = 2.0$)
- A condition for the consistency
- The initial vector $\mathbf{p}_0$ is derived by floating-point methods ± 1

Difficulties

- Objective function is computationally expensive
- Calculating derivatives is slow (even with AD libraries)
- Considerable overestimation

Highly flexible software is necessary!

Consistent States

Consistent parameter vectors

A state vector **p** is consistent if $\forall t \in \{0, ..., T\}$:

$$\mathbf{y}_t(\mathbf{p}) \subseteq y_m(t) + \Delta\ \mathbf{y}_m$$

with the worst-case measurement error $\Delta\ \mathbf{y}_m = [-15, 15]$ holds.

Inconsistent parameter vectors

A state vector **p** is inconsistent if $\exists t \in \{0, ..., T\}$:

$$\mathbf{y}_t(\mathbf{p}) \cap (y_m(t) + \Delta\ \mathbf{y}_m) = \emptyset$$

Basic Strategy in Rostock

Branch-and-bound in the direction μ where the sensitivity measure

$$\Delta J^{<l>} = \sum_{\nu=1}^{T} \left[\frac{\partial \mathbf{y}^{<l>}(t_\nu)}{\partial \mathbf{p}^{<l>}} \cdot \frac{\mathrm{w}(\mathbf{p}^{<l>})^2}{\mathrm{w}(\mathbf{p}^{<0>})} \right]$$

is maximized; discard the inconsistent states.

Features:

- derivatives necessary
- no parallelization $\rightarrow$ slow

Strategy in Duisburg

Branch-and-bound based on the Hansen method*

Basic pattern

1. `p`$\leftarrow \mathcal{L}$
2. `Discard p if it is infeasible`
3. `Discard p if` $\underline{\Phi(\mathrm{p})} > \overline{D}$
4. `Contract p`
5. `Update` $\overline{D}$
6. `Add p to` $\mathcal{L}_{\text{final}}$ `if termination criteria are satisfied`
7. `Split p and add new boxes to` $\mathcal{L}$

Features

- flexible thanks to UniVerMeC
- derivative-free, if desired
- parallelized $\rightarrow$ fast

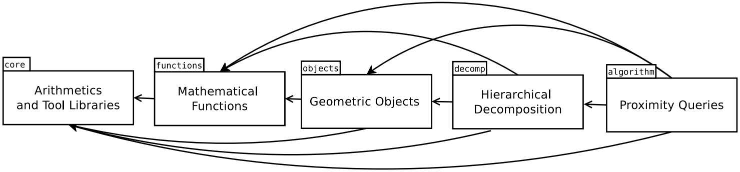
Termination criteria

- w($\mathbf{p}$)$\leq \epsilon_p, \epsilon_p > 0$
- w$\left(\Phi(\mathbf{p})\right) \leq \epsilon_\Phi, \epsilon_\Phi > 0$

E. Hansen and G. W. Walster. *Global Optimization Using Interval Analysis.* Marcel Dekker, New York, 2004.

UniVerMeC

Unified Framework for Verified GeoMetric Computations



core *Adapter for underlying arithmetic libraries*

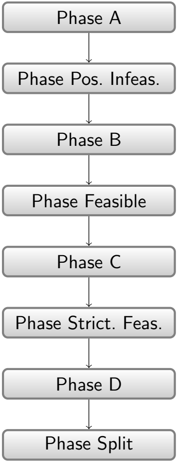
functions *Uniform representation for functions*

objects Implicit surfaces, CSG models, polyhedrons

decomp Spatial decomposition, *Multisection schemes*

algorithms Distance computation, *Global optimization*, ...

A Configurable Optimization Algorithm in `UniVerMeC`



Phase A
Use the midpoint test

Phase Feasible
Update the upper bound

Phase D
Linearize and prune using the consistency constraint

Phase Split
Calculate bound on $\Phi(\mathbf{p})$
Check for (in)consistent states

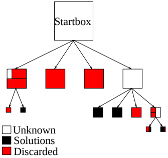
Parallelization: Multicore CPU

Characteristics

A lot of local work

$\rightarrow$ objective function evaluation, midpoint test

Workload sharing



The B&B tree is unbalanced

No a priori subproblems

Take the available box from $\mathcal{L}$

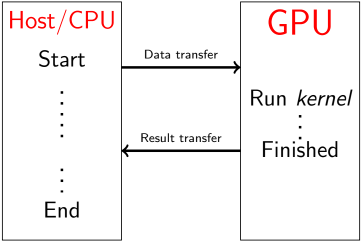
Parallelization (OpenMP)

$\mathcal{L}$ is shared between all threads

$\rightarrow$ for shared-memory only, a bottle neck

Parallelization: the GPU

GPUs are highly specialized programmable units for rendering



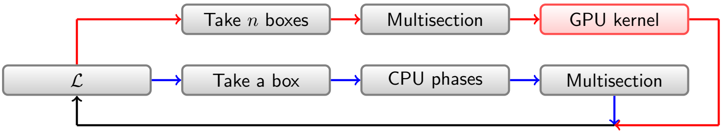
Features:

- the CPU starts the GPU part
- Data transfer is expensive (long latency)
- the CPU can proceed while the GPU computes

$\rightarrow$ Parts not well suited for the GPU are executed on the CPU

$\rightarrow$ Perform as many computations as possible on the transferred data

Integration into the Optimization Algorithm



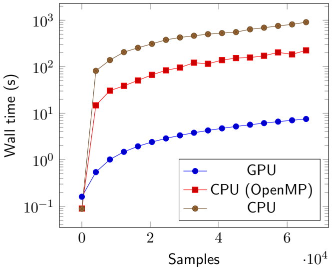
Our approach:

- Run the GPU and the CPU in parallel
- Store the working list $\mathcal{L}$ in the host memory
- One CPU thread feeds the GPU with data

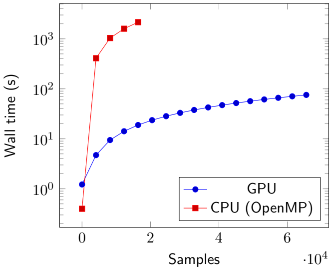
  $\rightarrow$ currently, only bounds on Φ are derived using the GPU
- The other CPU threads work normally

Benchmarks for Computing the Objective Function

$1 \times 1 \times 1$ Model



$3 \times 3 \times 1$ Model



Reference system

Intel Core i7-860 (2.8 GHz)
gcc 4.7 on Linux
GeForce GTX 560 Ti, CUDA 4.2

Results

$1 \times 1 \times 1$: GPU $30\times$ faster

$3 \times 3 \times 1$: GPU $114\times$ faster

Results for the $1 \times 1 \times 1$ Model (1)

As of now, we can only discard non-optimal boxed in a verified way.

No verified minimum, only candidates!

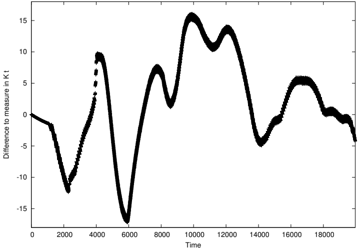
The candidate with the lowest e is chosen as the solution, where

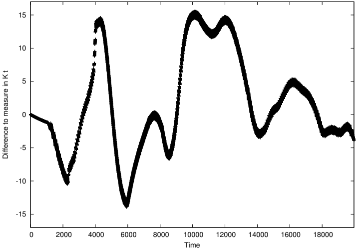
$$e = \sqrt{\frac{\sum_{k=1}^{T} \left(y_k - y_m(t_k) + f(y_k, \mathrm{mid}(\mathbf{p}))\right)^2}{T}}$$

is similar to the root mean square error measure (practice-motivated).

Results for the $1 \times 1 \times 1$ Model (2)

Difference between measured and simulated temperature





Version	e	Wall time
GPU	7.42944	$\approx 2m$ (135s)
CPU (D)	7.68	$\approx 42m$ (2491s)
CPU (R)	11.84	$\approx 11h$
FP	5.17	$\approx 6h$

Conclusions

Results:

- Interval branch and bound optimization algorithms are well suited for parallelization
- Usage of GPU caused a speedup of 18 (against the parallel CPU version)
- The runtime for the parameter identification could be significantly decreased

Future work:

- Apply the algorithm to the $3 \times 3 \times 1$ model
- The Euler method $\rightarrow$ verified IVPS
- Better use of GPU memory hierarchies