Parallel large-scale optimization and Internet of Things for cyber physical power systems

R. Franke, M. Schroschk, W. Braun, B. Bachmann, M. Johannes, C. Grindler
Agenda

Motivation of digital revolution
PARADOM project
Parallel Optimization
ABB OPTIMAX PowerFit
ABB Ability™ demo
Conclusions
Digital technologies are driving new innovation in industrial markets

Media is focused on B2C but the “killer app” is in B2B

- Virtual/ augmented reality
- Software-defined machines
- Machine learning
- Time-sensitive networking
- Big data
- Inexpensive computing
- Cloud computing
- Cybersecurity
- Connectivity
- Blockchain
- Cloud computing
- 5G
Industrial markets primed to adopt digital technologies

Computing + connectivity + cloud + analytics set to unlock value

Digital S-Curve

Level of digitalization vs. Time

- ABB end-market
- Other industries
PARallel Algorithmic Differentiation in OpenModelica for real-time simulation and optimization (Mai 2016 – April 2019)

**Task:**
- Model complexity
- Optimization horizon
- Optimization step size
- Real-time
- ...

**Customer requirements**

**PARADOM approach:**
1. Parallel Optimization (Multiple Shooting)
2. Parallel Algorithmic Differentiation
3. Parallel solution of equation systems

**This presentation**

Direct Multiple Shooting  Solution of equation systems

Application to digital power systems
Parallel Optimization with control vector parameterization (direct multiple shooting)

Idea:

- Describe control trajectory with control parameters $q^k$
- Introduce initial states of each interval as optimization variables $s_x^k$
- Parallel solution of initial value and sensitivity problems for each interval
- Treat junction conditions between intervals as optimization constraints
Application to planning of power production (dynamic optimal power flow problems)

- Treat power transmission system with nodes and admittances
- Equation based formulation in Modelica
- Translation to FMU with OpenModelica
- Investigation of different modi for model formulation, in particular:
  - **Modus II**
    Optimizer tunes generator nodes (eliminate load and internal nodes – cf. ODE mode)
  - **Modus IV**
    Tune generator and load nodes (eliminate internal nodes – cf. DAE mode)

\[
\begin{pmatrix}
I_1 \\
\vdots \\
I_N \\
\end{pmatrix} =
\begin{pmatrix}
G_{11} & G_{12} & \cdots & G_{1N} \\
G_{21} & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
G_{N1} & \cdots & \cdots & G_{NN} \\
\end{pmatrix}
\begin{pmatrix}
U_1 \\
\vdots \\
U_N \\
\end{pmatrix}
\]

Clemens Grindler: Untersuchung paralleler Methoden der Optimalen Steuerung im Kontext der Leistungsflussoptimierung für Energieübertragungsnetze, Masterarbeit, KiT, 2017
Spy matrices of model Jacobians

**Modus II**
- Sparse structure
- Many small equation systems

**Modus IV**
- Dense structure
- Few large equation system

Jacobian Model 14, Modus 2

Jacobian Model 14, Modus 4
Results for different sparse solvers for equation systems

- Comparison of equation system solvers PARDISO, SuperLU, UMFPACK und BKP
- Model with 14 nodes: 24 resp. 8 tuned variables
- PARDISO shows best performance for Modus II, BKP for Modus IV

Sequential MS, CPU times and speedup for up to 12 CPUs, Model 14
Results for different sparse solvers for equation systems

- Comparison of equation system solvers PARDISO, SuperLU, UMFPACK and BKP
- Model with 89 nodes: 92 resp. 22 tuned variables
- PARDISO shows best overall performance and good speedup for sparse Modus II
Results parallel multiple shooting

- Parallel multiple shooting increases speedup
- Speedup of up to 4 in Modus IV with 9 CPUs

parallel MS, CPU times and speedup for up to 12 CPUs, Model 14
Results parallel multiple shooting

- Parallel multiple shooting increases speedup
- Speedup of up to 9 with 12 CPUs in Modus IV (because more work is done inside parallel FMUs)

Parallel MS, CPU times and speedups with 12 CPUs, Model 89
Results for relation of #CPUs and #Intervals

- Left plot: CPU time increases if #Intervals exceeds #CPUs
- Right plot: substantial speedup starting from 61 CPUs when having 60 intervals
ABB OPTIMAX PowerFit
General System Architecture

- OPTIMAX® PowerFit
  Central control and optimization

- Real-time interfaces

- Interfaces

- TSO Grid Services
  EEX EPEX OTC

- Forecasts Delivery Commitments

- Plant Schedules Plant data

- Energy Management / Trader
  - Energy Management
  - Plant data

- Availability Diagnosis

- Accounting Reporting
  - Availability
  - Diagnosis

- Technical Units
  - Turbine
  - Boiler
  - Condenser

- GPRS / UMTS / LTE

- Firewall / Router / VPN

- Firewalls

- Real-time interfaces

- Optimal schedules
- Unit commitment
- Balancing group optimization
- Balancing power calls
- Combined heat and power optimization
- Multi-Energy site optimization
- Demand response
- Autonomous operation of distribution grids

©ABB
February 5, 2018 | Slide 14
Leveraging the ABB Ability™ platform

Secure digital solutions on-premise, in the cloud, and in an ecosystem
Example: OPTIMAX PowerFit
(red by ABB; blue by Microsoft)
OPTIMAX PowerFit in Azure (with new Cockpit) – Cloud

PowerFit Demo application in Azure
PowerBI for PowerFit – Intercloud

New Microsoft dashboard software

Accessing Azure storage written by PowerFit Demo application
Conclusions

Cloud computing, connectivity and inexpensive computing resources are three drivers of digital revolution.

Exploitation in ABB Ability OPTIMAX PowerFit for management of renewable power considering grid limits.

Parallel optimization with control vector parameterization (multiple shooting):
- Solve model and sensitivity equations in parallel for each time interval.
- Need to find best compromise for treatment of equation systems in model vs by solver (cf. ODE vs reduced DAE vs full DAE).

OpenModelica provides modeling and translation technology:
- Generate FMUs for different modi (degree of elimination of nodes in grid model).
- Well suited for research projects; still limited for commercial use, despite of huge progress during last years (efficiency of modeling workflow for complex models, lacking encryption of model libraries).

PARADOM using OpenModelica and HQP:
- Successful implementation of parallel optimization with multiple FMU instances.
- New OpenModelica feature: algorithmic differentiation of FMUs (fmi2GetDirectionalDerivative).