Study Committee C6

Active Distribution Systems and Distributed Energy Resources

Paper 10806

A Testbed-based Approach for the Resilience Assessment of Multi-Microgrids

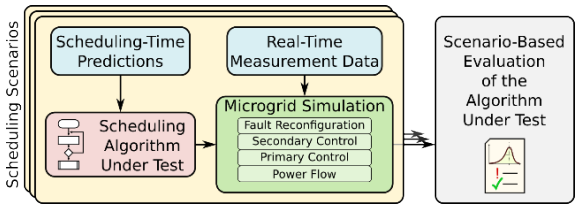
Michael H. SPIEGEL[1], Thomas I. STRASSER[1,2]
[1]AIT Austrian Institute of Technology, [2]Technische Universität Wien (TU Wien), [1,2]Austria
[1]michael.spiegel@ait.ac.at, [2]thomas.i.strasser@ieee.org

Motivation

- Detailed **assessment** of multi-microgrid **scheduling algorithms**
- Overcome limited **number of scenarios** and failures
- Missing **low-level controls** and real-time fault mitigation techniques in **long-term assessments**
- Provide **software design** to enable scalable multi-microgrid assessments
- Efficient integration into **engineering workflows**

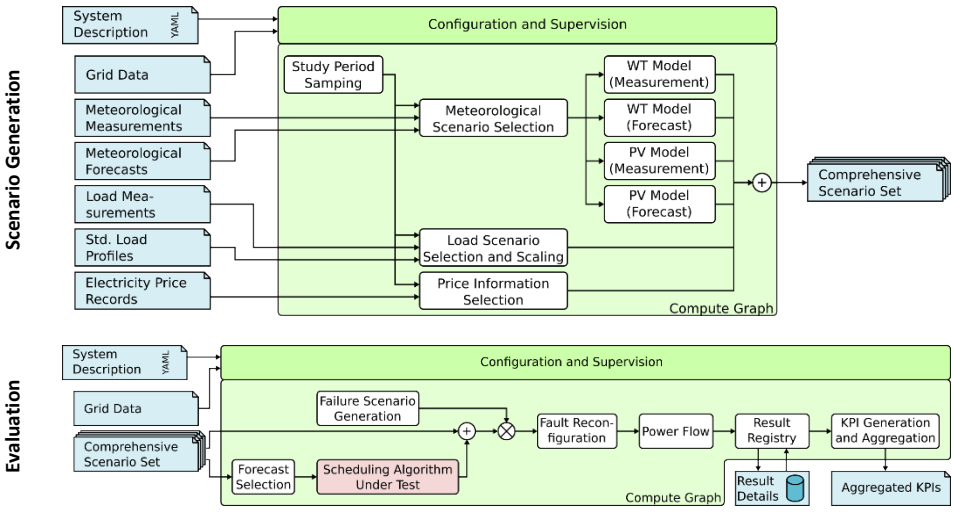
Simulation-Based Assessment Method

- Verify schedules by **independent simulations**
- Series of **extended power-flow** computations
- Include **steady-state** response of **devices** and **low-level controls**
- **Extensive scenario set** (>300.000 scenarios)



Testbed Software Architecture Based on Compute Graphs

- Two main steps: Scenario generation and evaluation
- Program flow described by **input-output relations** (compute graph)
- Vectorized formulation to **enable parallelization** and reduce complexity
- Dynamic scheduling to multiple, **distributed workers** by the Dask framework



Study Committee C6
Active Distribution Systems and Distributed Energy Resources
Paper 10806

A Testbed-based Approach for the Resilience Assessment of Multi-Microgrids

continued

Extended Configuration-Based User Interface

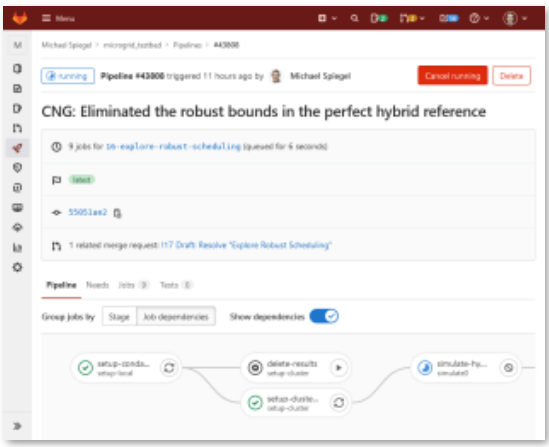
- **Text-based** (YAML) **syntax** for efficient version control
- Include mechanism to **structure** complex **configurations**
- Dynamic reference to environment variables for simplified **test automation**

```yaml
# The seed used for all randomized search operations
search seed: !env "SEARCH_SEED"

# The failure scenarios that are considered in scheduling
failure scenarios:
  _include: "../failure_scenarios/worst-scheduling-case.yaml"

# The power flow configuration
power flow:
  _include: "../power_flow/full_low_level_control.yaml"
  max iterations: 5
```

Configuration Syntax

Integrated Development and Simulation Workflow

- Scripting-friendly software interface
- Tight integration into **software development platforms**
- **Fully automated assessment** procedures
- Precise **link** between software **versions** and test **results**



GITLab Integration

Advanced Debugging Facilities

- **Monitoring** and **debugging** facilities by Dask
- Progress monitoring
- Performance tracing
- Detailed simulation outputs per scenario

Progress View

Study Committee C6

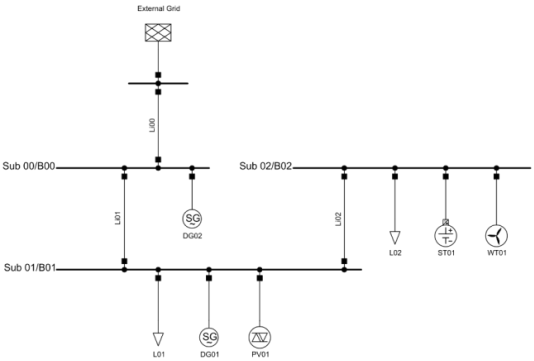
Active Distribution Systems and Distributed Energy Resources

Paper 10806

A Testbed-based Approach for the Resilience Assessment of Multi-Microgrids

continued

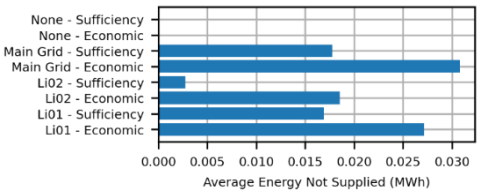
Demonstration Based on Exemplary (Multi-)Microgrid



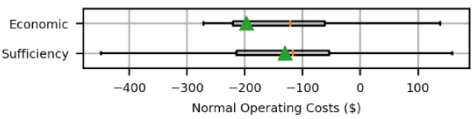
- Simplified (multi-)microgrid to **demonstrate engineering and validation** process
- Inclusion of most essential assets
- Dynamic load and generation profiles
- Integration in state-of-the-art development platform (GITLab)
- Two algorithms assessed
  - Purely **economic scheduling**
  - Sufficiency-based **resilience constraints**

Exemplary Evaluation Results

- Operating costs and resilience metrics covering a **broad spectrum** of **operating conditions**
- **Comparison** of scheduling approaches on a **common ground**



- Reduced unsupplied energy by sufficiency-based resilience constraints (42% at main-grid faults)
- 34% revenue reduction by resilience constraints



Reflection on the Architecture

- **Avoidance** of **global states** supports parallelization
- Additional development overhead of distributed computing is well justified for **extensive workloads**
- Frequent assessment of code changes by **rapid** and **automatized test execution**
- Tight integration into the development toolchain by text-based inputs

Conclusion

- **Comprehensive assessment** of multi-microgrid scheduling approaches **demonstrated**
- **Scalability** pushed by testbed architecture
- **Engineering efficiency** pushed by integrated development workflow