OMG: A Scalable and Flexible Simulation and Testing Environment Toolbox for Intelligent Microgrid Control

Stefan Heid[1], **Daniel Weber**[2], **Henrik Bode**[2], **Eyke Hüllermeier**[1], and **Oliver Wallscheid**[2]

1 Chair of Intelligent Systems and Machine Learning, Paderborn University, Paderborn, Germany **2** Chair of Power Electronics and Electrical Drives, Paderborn University, Paderborn, Germany

DOI: 10.21105/joss.02435

Software

- Review
- Repository
- Archive

Editor: David Hagan

Reviewers:

- @katyhuff
- @gonsie

Submitted: 26 May 2020
Published: 05 October 2020

Summary

The OpenModelica Microgrid Gym (OMG) toolbox provides a transient simulation framework for local energy grids based on power electronic converters. OpenModelica is used as the backend, allowing users to set up arbitrary electric grid designs via its well-known graphical user interface in a plug-and-play fashion (Fritzson et al., 2018). Simulations can be configured using a python interface, making it easy to integrate software modules for the realization and testing of closed control loops. In addition, the OpenAI Gym interface is provided to connect data-driven reinforcement learning algorithms for investigating intelligent microgrid control approaches (Brockman et al., 2016).

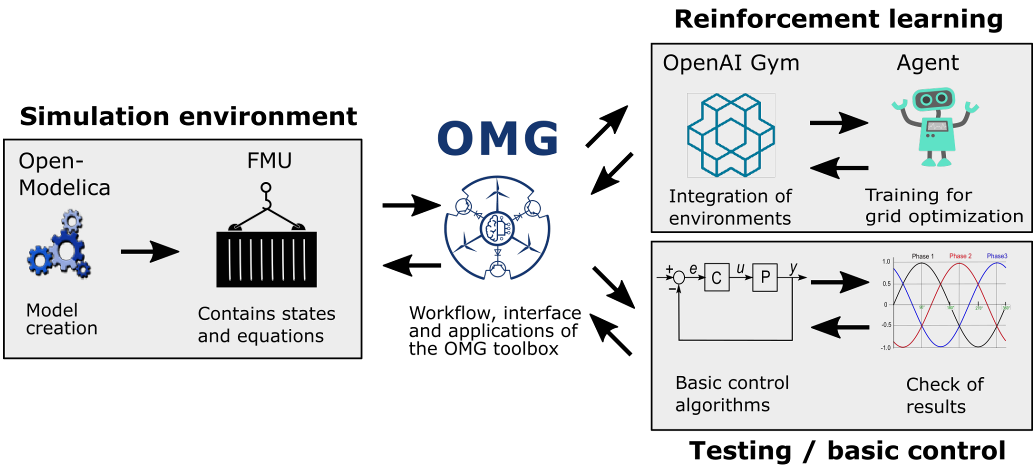

Fig. 1: Overview of the interconnections between the different parts of the OMG toolbox. The OpenModelica and OpenAIGym logos are the property of their respective owners.

Background on microgrids and their control

Micro- and smart-grids (MSG) play an important role for integrating renewable energy sources in conventional electricity grids and for providing power supply in remote areas (Lund, Østergaard, Connolly, & Mathiesen, 2017). Due to their high efficiency and flexibility, power electronic converters are largely used to drive modern MSG. Power electronics describes the application of solid-state electronics to the control and conversion of electric power, which is largely performed with semiconductor switching devices such as diodes or power transistors. This includes energy conversion in terms of voltage and current amplitude, frequency and phase angle, as well as the number of phases between two or more electrical energy systems to be connected.

Controlling MSGs is a challenging task due to the high requirements on energy availability, safety, and voltage quality. This is particularly demanding due to the wide range of different MSG topologies depending on their field of application like industrial campuses, residential areas or remote off-grid electrification (Kroposki et al., 2008). This results in high demand for comprehensive testing of new control concepts during their development phase and comparisons with the state of the art to ensure their feasibility. This applies in particular to data-driven control approaches such as reinforcement learning (RL), the stability and operating behavior of which cannot be evaluated a priori (Garcıa & Fernández, 2015).

State of field

`OMG` is a Python-based package for the modeling and simulation of microgrids based on power electronic energy conversion. The OpenModelica (Fritzson et al., 2018) library enables the user to define their microgrid (i.e. a local electricity grid containing arbitrary sources, storages and loads) in a flexible and scalable way or to use certain predefined example grids. Due to the component-oriented modeling framework based on OpenModelica, dynamic processes on small time scales are in focus, which allows for accurate control and test investigations during transients and steady-state. This is an essential difference to already available open-source solutions for the simulation of electrical energy networks, which, in contrast, generally depict large-scale transmission networks with abstracted models in the (quasi)-stationary state (e.g. PyPSA (Brown, Hörsch, & Schlachtberger, 2018) or Pandapower (Thurner et al., 2018)). In addition to the pure modeling and simulation of microgrids, basic building blocks for setting up a hierarchical control framework on the inner and primary level (Guerrero, Chandorkar, Lee, & Loh, 2013) are provided with `OMG`.

Interfaces for control and reinforcement learning

The API is designed to provide a user-friendly interface to connect a modeled microgrid (the simulation environment) with a wide range of control methods such as classical linear feedback control or model predictive control techniques (cf. Fig. 1). Moreover, the standardized OpenAI Gym interface (Brockman et al., 2016) is also available for training data-driven control approaches like RL. This enables users who want to integrate contemporary open-source Python-based RL toolboxes such as `Stable Baselines3` (Raffin et al., 2019), `TF-Agents` (Guadarrama et al., 2018) or `keras-rl` (Plappert, 2016). Many auxiliary functionalities for the essential operation of microgrids are shipped with OMG such as coordinate transformations for basic controller classes, monitoring wrappers, and phase-locked loops for frequency and phase angle extraction. Following this structure, nearly every control approach, including data-driven RL, can be implemented and tested with `OMG` in a relatively short amount of time. To highlight the challenges of data-driven control approaches in safety-critical environments, application examples using safe Bayesian optimization (Berkenkamp, 2020) for automated controller design are provided in the toolbox.

Intended use and targeted audience

`OMG` is designed to be used by students, academics, and industrial researchers in the field of control and energy engineering and data science. The primary objective of the toolbox is to facilitate entry for new users into the modeling, control, and testing of microgrids and to provide a platform on which different control methods (including RL) can be compared under defined conditions (benchmarks).

Features

The `OMG` toolbox provides the following key features:

- A library for the scalable and flexible design of local electricity grids in OpenModelica. Users can select between a wide range of different grid components and connect them in a plug-and-play approach.
- Dynamic simulation of local electricity grids on component level including single and multi-phase systems as well as AC and DC operation.
- Easy exchange of models between computing platforms and simulation of the models by using the FMI 2.0 standard (Modelica Association, 2020) with C++ code inside and PyFMI (Modelon AB, 2020) for access in Python. Appropriate numeric solvers for the underlying system of ordinary differential equations can be easily chosen within the usual Python packages (e.g. SciPy) due to the usage of co-simulation.
- Calculation, evaluation and monitoring of every single time step covering states, action and auxiliary quantities provides an interface for manual or automated inspection. The latter is particularly useful for the automatic training of data-driven control approaches such as reinforcement learning.
- Large variety of predefined and parameterizable controllers (droop, voltage, current in multi- and singlephase) available, easy implementation of user-defined control structures possible.
- Monitoring tools to follow the live performance of the RL agent and to map the overall grid behaviour depending of each selected parameter set
- Interesting use cases applying safe data-driven learning to highlight the requirement of safety in a delicate control environment are available.

Examples

Detailed examples are shown in the OMG whitepaper (https://arxiv.org/pdf/2005.04869.pdf) including the implementation and evaluation of a safe Bayesian controller (Berkenkamp, 2020). The SafeOpt learning algorithm is applied to an automatic controller tuning problem with safety-relevant state constraints in different microgrid topologies (e.g. different number of inverters, load characteristics). Furthermore, the provided evaluation tools enable users to compare the performance of different RL algorithms against each other and against manually tuned inverters.

Availability and installation

`OMG` is supported and tested on Linux and Windows. Mac users are asked to run this toolbox on a Linux VM. The package should be installed in a conda environment. `PyFMI` can be installed via `conda install -c conda-forge pyfmi`, the `OMG` package by `pip` Python package manager using `pip install openmodelica_microgrid_gym` command. The source code, guide and datasets are available on the GitHub repository (https://github.com/upb-lea/openmodelica-microgrid-gym).

Individual contributions of the authors

Following are shown the main fields of each individual contributor of OMG:

- S. Heid: Main software architecture, software module integration, unit tests
- D. Weber: Application examples, control-related auxiliary features (e.g. basic expert controllers), unit tests
- H. Bode: Design of the specific OpenModelica library created for OMG, grid modelling and simulation, data transfer between OpenModelica and Python, unit tests
- O. Wallscheid: Concept design and idea generation, testing and technical feedback, administrative project management
- E. Hüllermeier: Administrative project management, concept-oriented feedback

Acknowledgements

The authors kindly acknowledge the valuable contributions and advice regarding grid and controller design by Jarren Lange. The authors would also like to acknowledge the funding and support of this work by the Paderborn University research grant.

References

Berkenkamp, F. (2020). SafeOpt: Safe Bayesian Optimization. Retrieved from https://github.com/befelix/SafeOpt

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI gym. Retrieved from https://arxiv.org/abs/1606.01540

Brown, T., Hörsch, J., & Schlachtberger, D. (2018). PyPSA: Python for Power System Analysis. *Journal of Open Research Software*, *6*(1). doi:10.5334/jors.188

Fritzson, P., Pop, A., Asghar, A., Bachmann, B., Braun, W., Braun, R., Buffoni, L., et al. (2018). The OpenModelica Integrated Modeling, Simulation and Optimization Environment. In. doi:10.3384/ecp18154206

Garcıa, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, *16*(1), 1437–1480.

Guadarrama, S., Korattikara, A., Ramirez, O., Castro, P., Holly, E., Fishman, S., Wang, K., et al. (2018). TF-Agents: A library for reinforcement learning in tensorflow. *GitHub repository*. https://github.com/tensorflow/agents. Retrieved from https://github.com/tensorflow/agents

Guerrero, J. M., Chandorkar, M., Lee, T., & Loh, P. C. (2013). Advanced Control Architectures for Intelligent Microgrids, Part I: Decentralized and Hierarchical Control. *IEEE Transactions on Industrial Electronics*, *60*(4), 1254–1262. doi:10.1109/TIE.2012.2194969

Kroposki, B., Lasseter, R., Ise, T., Morozumi, S., Papathanassiou, S., & Hatziargyriou, N. (2008). Making Microgrids Work. *IEEE Power and Energy Magazine*, *6*(3), 40–53. doi:10.1109/MPE.2008.918718

Lund, H., Østergaard, P. A., Connolly, D., & Mathiesen, B. V. (2017). Smart Energy and Smart Energy Systems. *Energy*, *137*, 556–565. doi:10.1016/j.energy.2017.05.123

Modelica Association. (2020). Functional Mock-up Interface. Retrieved from https://fmi-standard.org/

Modelon AB. (2020). PyFMI. Retrieved from https://github.com/modelon-community/PyFMI

Plappert, M. (2016). Keras-rl. *GitHub repository*. https://github.com/keras-rl/keras-rl; GitHub. Retrieved from https://github.com/keras-rl/keras-rl

Raffin, A., Hill, A., Ernestus, M., Gleave, A., Kanervisto, A., & Dormann, N. (2019). Stable baselines3. *GitHub repository*. https://github.com/DLR-RM/stable-baselines3; GitHub.

Thurner, L., Scheidler, A., Schäfer, F., Menke, J., Dollichon, J., Meier, F., Meinecke, S., et al. (2018). Pandapower: An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems. *IEEE Transactions on Power Systems*, *33*(6), 6510–6521. doi:10.1109/TPWRS.2018.2829021