The Journal of Open Source Software

Easy SimAuto (ESA): A Python Package that Simplifies Interacting with PowerWorld Simulator

Brandon L. Thayer[*1, 2], **Zeyu Mao**[1], **Yijing Liu**[1], **Katherine Davis**[1], and **Thomas Overbye**[1]

1 Texas A&M University **2** Pacific Northwest National Laboratory

DOI: 10.21105/joss.02289

Software

- Review
- Repository
- Archive

Editor: Daniel S. Katz

Reviewers:

- @robinroche
- @tonyelhabr

Submitted: 20 May 2020
Published: 17 June 2020

Summary

The electric power system is an essential cornerstone of modern society, enabling everything from the internet to refrigeration. Due to a variety of forces including climate change, changing economics, and the digital computer revolution, the electric grid is undergoing a period of major change. In order to overcome current and upcoming challenges in the electric power system, such as integrating renewable resources into a system that was not designed for intermittent power sources, researchers and industry practitioners must simulate the electric grid, its component devices, and its operation.

PowerWorld Simulator is a commercial power systems simulation tool that contains a suite of modeling and simulation features including power flow simulation, contingency analysis, transient stability simulation, and more (PowerWorld Corporation, 2020). The Simulator Automation Server (SimAuto) add-on for PowerWorld provides an application programming interface (API) that operates in-memory, allowing users to rapidly configure, run, and obtain results for simulations. PowerWorld and SimAuto are commonly used throughout the research community as well as in industry.

SimAuto was designed to be flexible enough to work with most available programming languages. However, the combination of this flexibility and the in-memory nature of SimAuto communication can make using SimAuto challenging, requiring error-checking, data type conversions, data parsing, low-level interactions with Windows Component Object Model (COM) objects, and more.

Easy SimAuto (ESA) is a Python package that significantly simplifies interfacing with PowerWorld Simulator (Thayer et al., 2020). ESA wraps all available SimAuto functions; provides high-level helper functions to streamline workflows, and provide additional functionality not provided by SimAuto; performs automatic error checking, data type conversions, and data parsing; is easily installable via Python's package installer (pip); has 100% testing coverage; and is fully documented. Similar packages have been created in the past, but lack functions, tests, documentation, and other useful features ESA provides (Boyd, 2018), (Roche, Natarajan, Bhattacharyya, & Suryanarayanan, 2012). Most SimAuto users tend to write their own one-off functions and boilerplate code for interfacing with SimAuto. ESA eliminates this redundancy and abstracts away all the low-level SimAuto interactions so that users can focus on performing higher-level tasks such as automating tasks, configuring simulations, and analyzing results.

ESA helps to meet the needs of both power system researchers and practitioners. As the design and operation of the electric grid becomes more complex, researchers and developers need the ability to incorporate their programs, algorithms, control schemes, etc. into power system simulations. ESA enables its users to fully leverage, extend, and automate the large depth of

*The first two authors of this paper contributed equally to this software.

functionality and tools built into PowerWorld Simulator: procedures that may have previously been performed via a sequence of manual tasks in Simulator's graphical user interface (GUI) can be rapidly built into Python scripts which can be stored in version control and run with a single click. Since ESA uses data types common to data science and scientific computing (e.g., Pandas DataFrames and Numpy Arrays), it is well suited to both academic research and task automation in industry. Due to ESA's use of these common Python data types and libraries, ESA provides a much-needed bridge between power system simulation and machine learning libraries.

ESA has already been utilized in several research projects past and present:

- In (Thayer, 2020a), (Thayer & Overbye, 2020), ESA was used to create a standardized reinforcement learning environment for power system voltage control. This environment was then used to carry out deep reinforcement learning (DRL) experiments in which the algorithm attempts to learn how to best control grid voltages under a diverse set of grid conditions (Thayer, 2020b).
- In (Li, Yeo, Wert, & Overbye, 2020), ESA was leveraged to create and simulate different electric grid scenarios where load, renewable generation levels, generation capacities, scheduled outages, and unit commitment were all varied. The resulting scenarios were used in the Grid Optimization (GO) competition hosted by the U.S. Department of Energy (DOE).
- Geomagnetic disturbances (GMDs) affect the magnetic and electric field of the earth, inducing dc voltage sources superimposed on transmission lines. In (Martinez, Garcia, Klauber, & Overbye, 2020)[1], a planning-based GMD mitigation strategy was developed for large power systems. ESA is leveraged to programmatically place GIC blocking devices in test systems per the proposed algorithm, thus minimizing the effects of GMDs on the power grid.
- ESA is used by an ongoing research project entitled "Real Time Monitoring Applications for the Power Grid under Geomagnetic Disturbances (GMD)," where recently, a real-world GMD monitoring system consisting of six magnetometers was deployed in Texas. The resulting magnetic field measurements are coupled with ground conductivity models to calculate real-time electric fields. These can then be fed to a grid model of Texas using ESA to enable calculation of real-time geomagnetically induced currents (GICs) for monitoring and visualization.
- ESA is used by an ongoing research project entitled "Cyber Physical Resilient Energy Systems (CYPRES)". In this project, ESA is leveraged to automatically map the communication system (like DNP3 outstation and data points) to the power system model.
- ESA is used by an ongoing research project entitled "Generalized Contingency Analysis Based on Graph Theory Concepts and Line Outage Distribution Factors (LODF)." In this project, ESA is leveraged to extract the topology of the power system model and obtain the LODF matrix.

Acknowledgements

ESA was developed by researchers at Texas A&M University. Funding was provided by the Texas A&M Engineering Experiment Station's Smart Grid Center and the U.S. Department of Energy (DOE) under award DE-OE0000895.

The authors of ESA would like to also thank our fellow researchers at Texas A&M who have provided essential feedback during ESA's creation.

[1] accepted, to be published after the delayed conference takes place

References

Boyd, J. M. (2018). pypowerworld. *GitHub repository*. https://github.com/jessemarkboyd/pypowerworld; GitHub.

Li, H., Yeo, J. H., Wert, J. L., & Overbye, T. J. (2020). Steady-state scenario development for synthetic transmission systems. In *2020 IEEE Texas Power and Energy Conference (TPEC)* (pp. 1–6). doi:10.1109/TPEC48276.2020.9042493

Martinez, A., Garcia, K., Klauber, C., & Overbye, T. J. (2020). Undergraduate research on design considerations for a GMD mitigation systems. In *2020 Kansas Power and Energy Conference (KPEC)* (pp. 1–6).

PowerWorld Corporation. (2020). PowerWorld Simulator. https://www.powerworld.com/.

Roche, R., Natarajan, S., Bhattacharyya, A., & Suryanarayanan, S. (2012). A framework for co-simulation of AI tools with power systems analysis software. In *2012 23rd International Workshop on Database and Expert Systems Applications* (pp. 350–354). doi:10.1109/DEXA.2012.9

Thayer, B. (2020a). gym-powerworld. *GitHub repository*. https://github.com/blthayer/gym-powerworld; GitHub.

Thayer, B. (2020b). drl-powerworld. *GitHub repository*. https://github.com/blthayer/drl-powerworld; GitHub.

Thayer, B. L., & Overbye, T. J. (2020). Deep Reinforcement Learning for Electric Transmission Voltage Control. *arXiv e-prints*, arXiv:2006.06728. Retrieved from http://arxiv.org/abs/2006.06728

Thayer, B., Mao, Z., & Liu, Y. (2020). Easy simAuto (ESA). *GitHub repository*. https://github.com/mzy2240/ESA; GitHub.