SoftwareX 32 (2025) 102384

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

WEC-Grid: A software tool for integrating wave energy converter models into power system simulations

Alexander Barajas-Ritchie *, Eduardo Cotilla-Sanchez

School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA

ARTICLE INFO

Keywords:
Wave energy converter
Power systems
Software
Integration
Renewable energy
Grid simulation
Open source

ABSTRACT

`WEC-Grid` is an open-source Python framework for integrating wave energy converter (WEC) models into power system simulations. It bridges `WEC-Sim` with established electrical simulators (PSS®E, `PyPSA`) through a unified API, a standardized data model, and an SQLite-backed store. The framework provides modular case handling, WEC-to-grid coupling, and a quasi steady-state simulation loop that harmonizes results across modelers. Core functionalities include reproducible data persistence, cross-modeler comparison, and visualization tools for buses, generators, and WEC farms. By lowering the barrier to marine energy integration studies, `WEC-Grid` supports academic research, teaching, and preliminary industry assessments of grid stability and renewable penetration. Documentation, Jupyter notebooks, and an executable capsule accompany the release, ensuring accessibility and reproducibility. Together, these features position `WEC-Grid` as a flexible platform for exploring the interactions of emerging WEC technologies with existing and future power systems.

Code metadata

Current code version	v1.0.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-25-00384
Permanent link to Reproducible Capsule	https://mybinder.org/v2/gh/acep-uaf/WEC-Grid/v1.0.0
Legal Code License	MIT License
Code versioning system used	git
Software code languages, tools, and services used	Python, Matlab, Jupyter Notebook
Compilation requirements, operating environments & dependencies	https://github.com/acep-uaf/WEC-Grid/blob/main/pyproject.toml
If available Link to developer documentation/manual	https://acep-uaf.github.io/WEC-Grid/
Support email for questions	barajale@oregonstate.edu

1. Motivation and significance

The integration of renewable energy sources, including wind, solar, and wave energy, into power grids is crucial for sustainable energy transitions. However, the intermittent nature and operational variability of these sources pose significant challenges to grid stability, reliability, and power quality. Among these, wave energy's power potential stands out due to its predictable nature, influenced by wind patterns across large ocean areas [1,2]. Consequently, wave energy converters (WECs) present substantial potential, especially for coastal regions and remote communities traditionally reliant on expensive diesel generation [3].

WEC technologies remain in early development stages (Technology Readiness Levels 1–3), limiting current commercial viability [4]. While significant research has supported wind and solar integration into power grids [5–7], the unique impacts of WECs on grid stability are relevantly unexplored [8]. Crucially, WEC integration involves complex hydrodynamic–electrical interactions, which are inadequately captured separately by current power system simulation tools and marine hydrodynamic simulation tools. This modeling gap significantly hampers collaboration between marine energy and power system communities and delays wave energy deployment.

---

* Corresponding author.
E-mail addresses: barajale@oregonstate.edu (A. Barajas-Ritchie), ecs@oregonstate.edu (E. Cotilla-Sanchez).

https://doi.org/10.1016/j.softx.2025.102384
Received 6 June 2025; Received in revised form 9 September 2025; Accepted 24 September 2025
Available online 6 October 2025
2352-7110/Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Table 1
Comparison of selected simulation frameworks relevant to `WEC-Grid`.

Tool	Time domain	Domain focus	Open source	Primary use case
Simscape/Simulink	Dynamic/hybrid	Multi-domain modeling (electrical + mechanical)	No	Component-level design, control co-simulation
OPAL-RT (RT-LAB)	Real-time, dynamic	HIL testing, embedded systems, real-time simulation	No	Controller validation, real-time execution
EMTP/PSCAD	Electromagnetic transients	Faults, switching events, protection	No	Protection studies, transient analysis
PowerFactory	Steady-state and dynamic	Grid planning, protection, EMS simulation	No	Utility-grade planning, EMS + dynamic grid studies
`pandapower`	Steady-state only	Python-based grid planning and OPF	Yes	Academic and industrial power flow studies, automation
`MATPOWER`	Steady-state only	Lightweight OPF and power flow	Yes	Teaching, benchmark testing, optimization research
`PyPSA`	Steady-state and dynamic	Python-based energy system modeling	Yes	Sector-coupled energy system studies, flexibility analysis
`WEC-Grid`	quasi steady-state	WEC power grid coupling with time series and modeler abstraction	Yes	Open-source marine energy integration, flexible power system modeler layer, SQLite time-series database

Addressing this gap requires a unified modeling approach that accurately represents interactions between hydrodynamic behaviors and electrical power systems. Current industry standard power flow and stability tools focus on traditional generation and renewable integration, yet they lack standardized, integrated WEC models. Additionally, power system stability studies (e.g., N-1 contingency analysis) critical for renewable integration become increasingly challenging with intermittent generation sources like WECs. Large-scale planning platforms such as `Switch 2.0` [9] demonstrate the value of modular, open-source toolchains for high-renewable scenarios, but these do not capture the hydrodynamic–electrical coupling unique to wave energy.

1.1. Related works

Existing simulation platforms for renewable integration include commercial tools such as Simscape/Simulink, OPAL-RT, EMTP-RV/PSCAD, and PowerFactory, as well as open-source frameworks like `pandapower` and `MATPOWER`. These tools each have strong capabilities but also limitations for wave energy converter (WEC) studies.

`Simscape` and `OPAL-RT` emphasize component-level and real-time hardware-in-the-loop simulation, while EMTP-RV and PSCAD focus on electromagnetic transients (EMT) for protection and fault analysis. Such approaches are excellent for high-resolution model verification and validation but computationally heavy for planning-focused quasi steady-state workflows. PowerFactory combines EMT and dynamic studies but is proprietary and less flexible for academic use. In contrast, `pandapower` and `MATPOWER` are lightweight and open-source, useful for single-snapshot studies, but lack support for hydrodynamic coupling or high-resolution time-series management.

`WEC-Grid` fills this gap by directly coupling hydrodynamic WEC models with both commercial (PSS$^{®}$E) and open-source (`PyPSA`) power system modelers. Its SQLite-based data management enables efficient quasi steady-state time-series simulation in a modular, extensible, and open-source framework designed specifically for marine energy integration.

Table 1 summarizes key distinctions between `WEC-Grid` and existing simulation platforms.

1.2. `WEC-grid` : addressing the modeling gap

`WEC-Grid` addresses these modeling challenges by integrating WEC hydrodynamic models (from `WEC-Sim`) with established power system analysis tools (e.g., PSS$^{®}$E, `PyPSA`). This open-source framework enables researchers to perform preliminary quasi steady-state analysis of WEC-to-grid interactions, capturing critical power flow and voltage stability insights efficiently.

This first `WEC-Grid` version emphasizes computational simplicity while accurately capturing essential system interactions, laying the groundwork for future enhancements such as dynamic simulations. Future iterations will incorporate transient and frequency-domain modeling using both commercial and open-source dynamic libraries (e.g., Siemens PSS$^{®}$E modules, PowerDynamics.jl [10], PowerFactory).

1.3. Scientific contributions and future research

`WEC-Grid` uniquely bridges hydrodynamic modeling and electrical system simulation, facilitating new research on wave energy integration under diverse ocean and grid conditions. The software supports both academic exploration and practical industry evaluations, providing insights into deployment feasibility, site-specific optimization, and grid stability implications.

2. Software description

`WEC-Grid` is an open-source Python framework for coupling wave energy converter (WEC) hydrodynamic models with power system studies. It connects `WEC-Sim` to established electrical simulators (PSS$^{®}$E, `PyPSA`) through a unified API, a standardized data model, and an SQLite-backed store to support reproducible, quasi steady-state analyses across tools.

2.1. Software architecture

Design overview. At the core is an `Engine` that coordinates time management, power-system modelers, WEC integrations, and persistence/-plotting. A modular *bridge* design decouples domain-specific simulators from shared services, so new modelers or WEC pipelines can be added without changing the Engine.

Bridge pattern and modeler abstraction. The framework implements a classical bridge design pattern: an abstract `PowerSystemModeler` interface defines the core methods for loading cases, advancing time steps, solving power flow, and recording results. Concrete implementations such as `PSSEModeler` and `PyPSAModeler` realize this interface for their respective simulators. This separation decouples simulator-specific logic from the `Engine`, ensuring that new modelers (e.g., `pandapower`, PowerFactory) can be introduced without

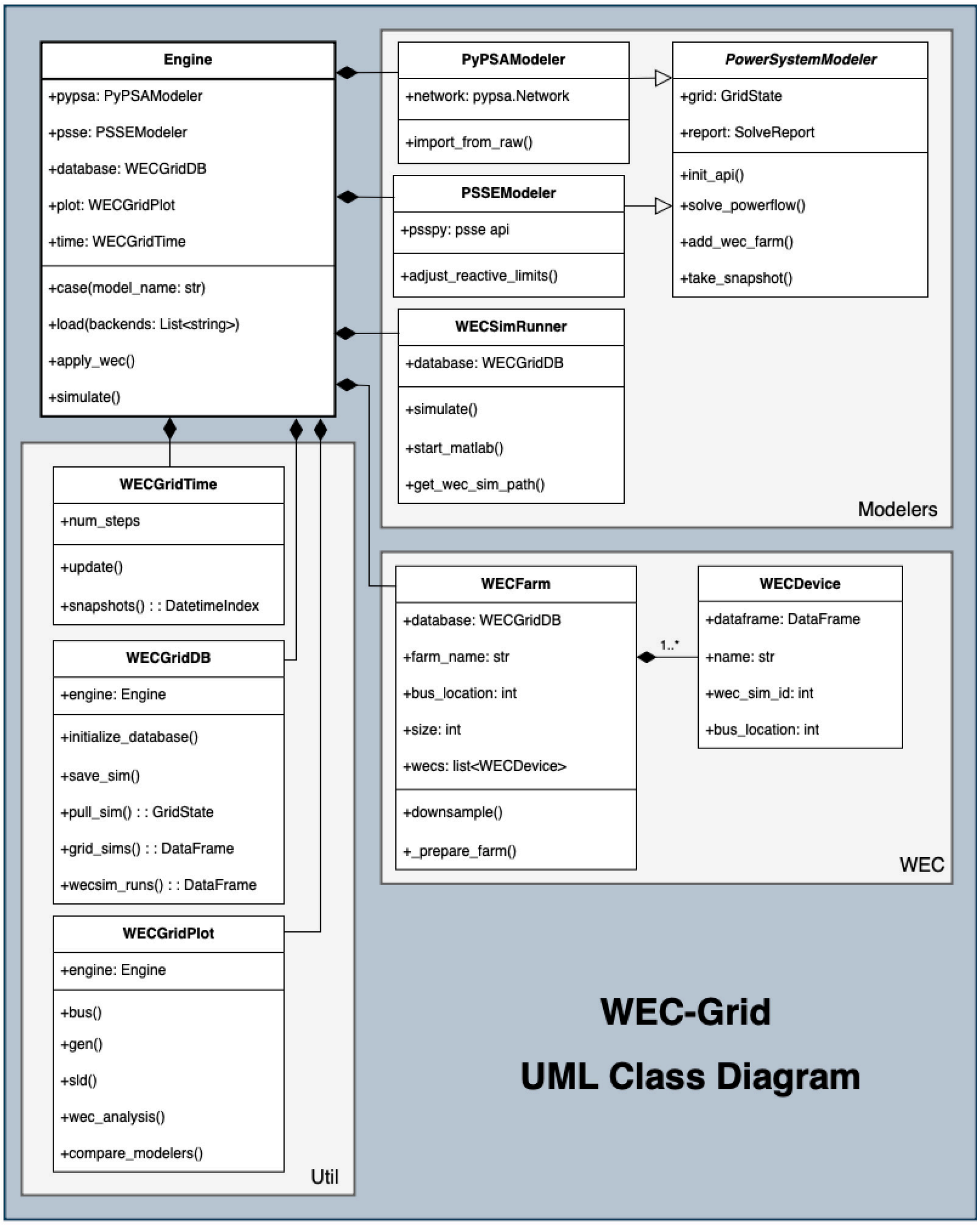

Fig. 1. WEC-Grid architecture overview: Engine, modelers (`PSSEModeler` , `PyPSAModeler`), WEC layer (`WECFarm` , `WECDevice` , `WECSimRunner`), and shared utilities (`WECGridTime` , `WECGridDB` , `WECGridPlot`).

altering the Engine or shared services. In Figs. 1–2, `PSSEModeler` and `PyPSAModeler` are shown as concrete implementations of this abstract bridge. This representation follows recent best practices in software architecture description, where abstract interfaces are explicitly separated from concrete implementations [11].

Layers and responsibilities.

- **Engine.** Initializes a grid case, configures one or more power-system modelers, applies WEC farms, advances the time loop, and exposes persistence and plotting interfaces.
- **Modelers.** Adapters that wrap electrical simulators with a common interface:
  - `PSSEModeler` for PSS$^{®}$E case I/O, power flow, and snapshot capture.
  - `PyPSAModeler` for building/solving PyPSA networks from RAW input with aligned snapshots.
- **WEC layer.** `WECDevice` and `WECFarm` convert WEC-Sim outputs to per-unit power, aggregate devices, and map injections to buses/generators/lines. A helper runner executes WEC-Sim and exports results to the database.
- **Utilities.** `WECGridTime` (timeline/snapshots), `WECGridDB` (SQLite schema; inputs, states, and results), and `WECGridPlot` (time series, single-line diagram, and cross-modeler comparison).

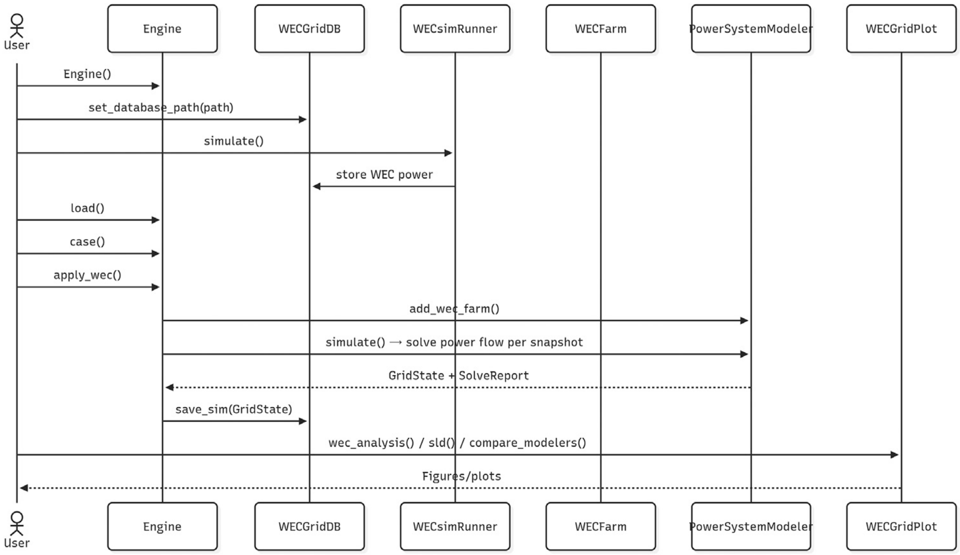

Fig. 2. Engine interaction sequence: initialize modelers, add WEC farms, run the quasi steady-state loop (update injections → solve power flow → snapshot), save results and generate plots.

Data model. All simulators write to/read from a shared schema:

- `GridState` : harmonized bus/generator/line/load snapshots and time series (per-unit on system base).
- `SolveReport` : per-step metadata (iterations, timings, convergence) to support diagnostics and reproducibility.

As shown in Figs. 1–2, the Engine loads a case, initializes modelers, integrates WEC-Sim power, and advances a quasi steady-state loop with persistence and plotting.

2.2. Software functionalities

- **Unified case handling.** Load RAW cases, set base MVA, and initialize one or multiple modelers with aligned snapshots.
- **WEC coupling.** Import WEC-Sim power time series, downsample/convert to per-unit, and attach farms to target buses and generators.
- **quasi steady-state simulations** Advance a configurable timeline; at each step update WEC injections and solve power flow across all enabled modelers.
- **Persistence for reproducibility.** Store inputs, intermediate states, and results in `WECGridDB` ; reconstruct `GridState` by ID for re-analysis and sharing.
- **Visualization and comparison.** Plot bus/gen/line time series and single-line diagrams; compare modelers (PSS®E vs. PyPSA) on aligned snapshots with quantitative error metrics.
- **Configuration.** Environment/user settings for database path (WECGRID_DB_PATH) and WEC-Sim location (WECGRID_WECSIM_PATH); optional helpers for PSS®E pathing.

Minimal example.

```python
import wecgrid

eng = wecgrid.Engine()
eng.case("IEEE_39_bus.RAW")
eng.load(["pypsa"])  # or ["psse"] or both

eng.apply_wec(
    farm_name="RM3-FARM",
    wec_sim_id=1,         # ID of WEC-Sim run in the DB
    bus_location=40,      # where WEC farm injects
    connecting_bus=39     # tie to existing network
)

eng.simulate()              # advance time loop and solve PF
# results and plots available via eng.pypsa.report / eng.plot
```

Listing 1 Minimal WEC-Grid workflow: load case, enable modeler, attach one farm, run quasi steady-state

2.3. Database schema

Simulation inputs, states, and outputs are persisted in an SQLite schema that harmonizes results across all supported modelers. Core tables store grid simulations, WEC simulations, and integration metadata, while per-modeler results (bus, generator, line, and load tables) and high-resolution WEC power series enable efficient queries and reproducible analyses. This structured design supports cross-modeler comparison and post-processing without rerunning simulations (Fig. 3). Similar efforts in open-source data generation pipelines, such as *ModelicaGridData* [12], highlight the growing importance of standardized, reproducible simulation frameworks for large-scale power system analysis.

2.4. Performance and scaling

Benchmarking across IEEE 14–300 bus systems confirms that WEC-Grid exhibits predictable, near-linear scaling. Total runtime grows linearly with system size for both PSS®E (~0.50 s/bus) and PyPSA (~0.64 s/bus), while memory requirements remain modest (~0.0002 GB/bus). CPU (overall) utilization exceeded 98% in all cases, indicating efficient use of compute resources.

Integrating WEC farms adds only limited overhead: PSS®E incurs a one-time cost of about 30% when WEC functionality is enabled, while PyPSA increases smoothly by 3%–6% for one to three farms. These results demonstrate that WEC-Grid scales well with both grid size and number of WEC farms, supporting studies from small test systems to realistic planning-scale networks (Fig. 4).

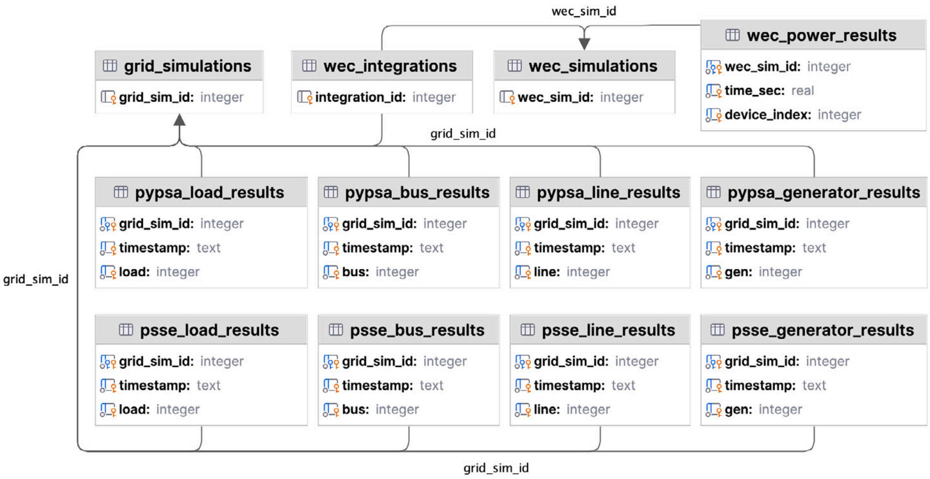

Fig. 3. Database schema linking grid simulations, WEC simulations, and integration results. Foreign keys connect grid runs to WEC time-series outputs for reproducible analysis and comparison.

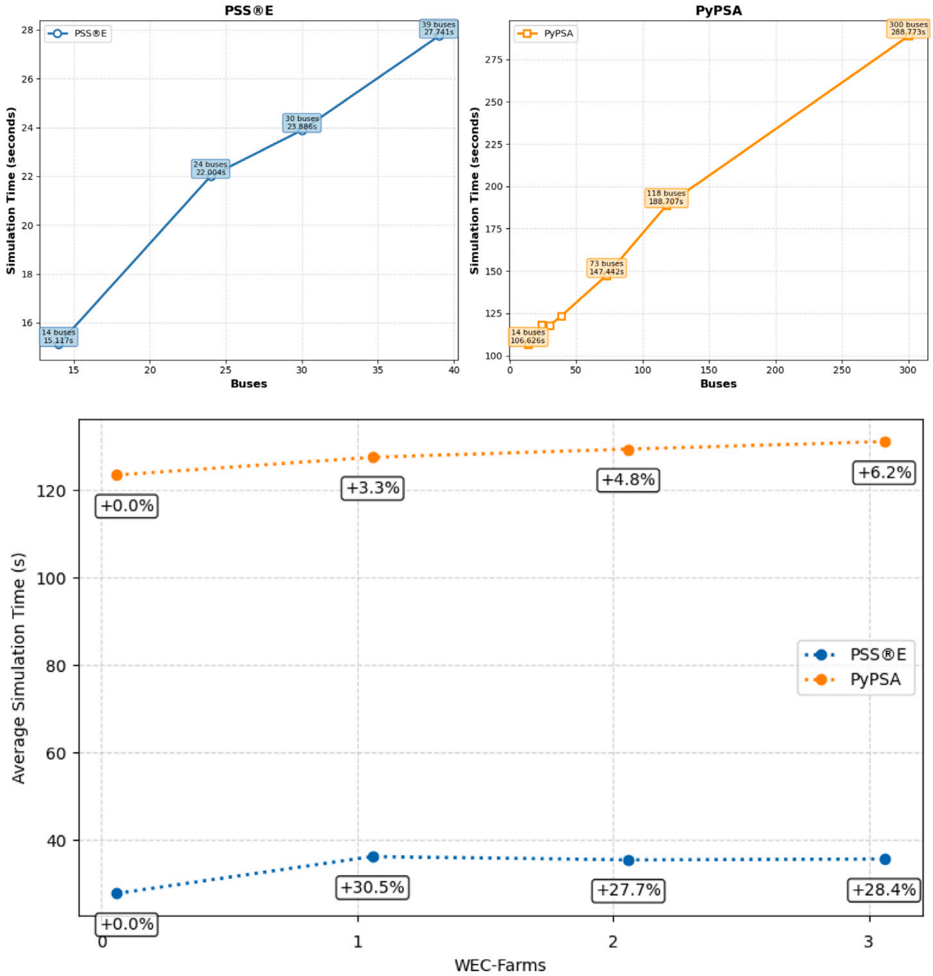

Fig. 4. Performance scaling of `WEC-Grid`. (Top) Runtime scaling with grid size for PSS®E and `PyPSA`, showing predictable near-linear growth across IEEE 14–300 bus systems. (Bottom) Overhead of WEC-farm integration on the IEEE-39 bus system. PSS®E shows a fixed overhead of ∼30%, while `PyPSA` increases smoothly by 3%–6% as farms are added.

3. Illustrative examples

We demonstrate the use of `WEC-Grid` on the IEEE 39-bus (New England) test system, a widely adopted benchmark for power system studies [13]. Three synthetic WEC farms are integrated: two RM3-based and one LUPA-based, each connected at dedicated buses to isolate their effects. A daily load curve with morning and evening peaks was applied to all buses, creating realistic demand conditions for quasi steady-state simulation. The workflow mirrors Example 3 in the project

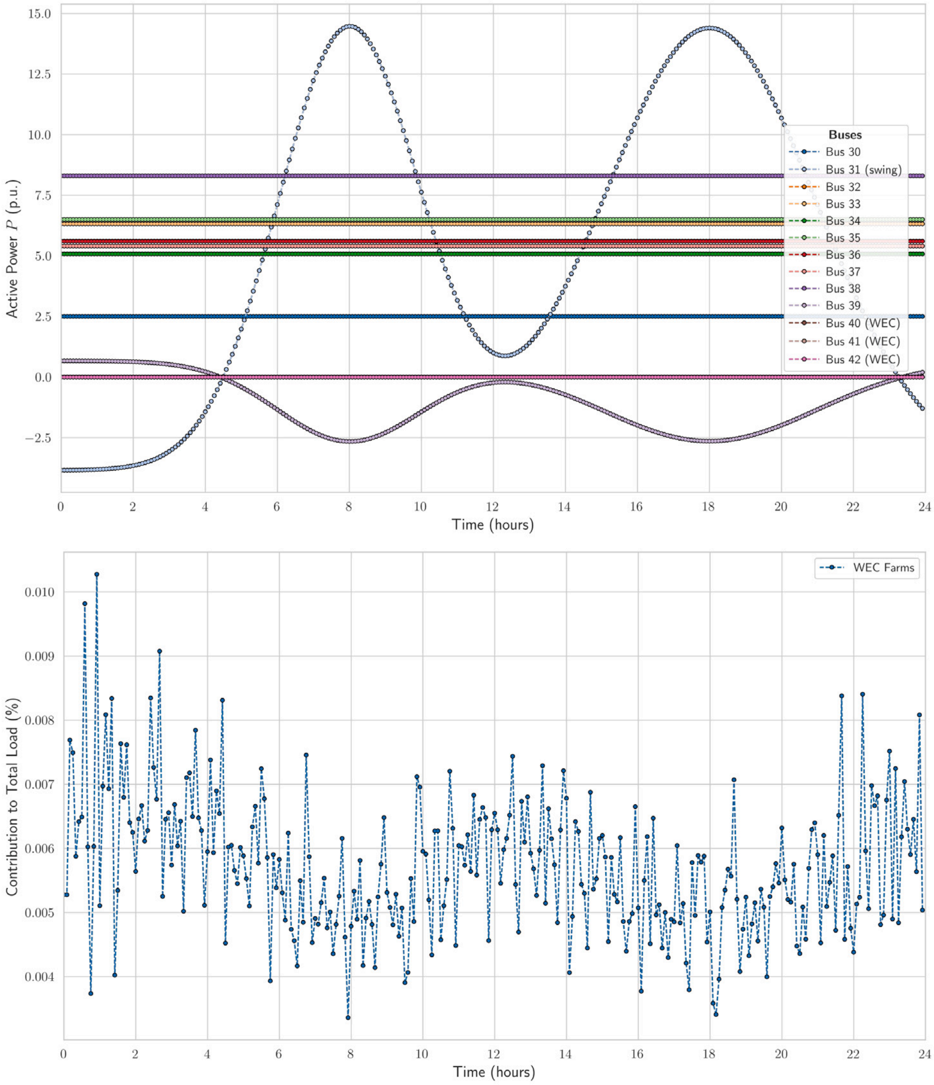

Fig. 5. Illustrative simulation results on the IEEE 39-bus system with three integrated WEC farms. (Top) Active power trajectories from `PyPSA` across selected buses, including three WEC buses (40–42). Daily demand peaks are visible, while WEC injections follow device-specific variability. (Bottom) Aggregate WEC contribution relative to total system load, remaining below 1% but reflecting realistic time-varying behavior driven by hydrodynamic models and the imposed load curve.

documentation and shows how grid cases, modelers, and WEC farms are combined.

Fig. 5 shows the aggregate contribution of all WEC farms to the total system load. Although the penetration is small (below 1%), the time-varying injections reflect realistic device-level variability and align with the synthetic load curve.

Cross-modeler consistency was evaluated by comparing active power and bus voltage angles across PSS$^{®}$E and `PyPSA`. At Generator 11 (WEC farm), the two modelers showed near-perfect agreement with RMSE $\sim 5\times10^{-11}$ p.u. (Fig. 6). At Bus 40, voltage angle trajectories overlapped closely (RMSE 1.48°, correlation $R = 1.0$), as shown in Fig. 6. These diagnostics confirm consistent modeler performance despite differences in implementation.

These examples illustrate that `WEC-Grid` provides a reproducible workflow for WEC-grid integration studies. Load curves, device-level injections, and modeler diagnostics can be combined seamlessly to explore both system-level and cross-modeler performance. Complete notebooks and data sets are provided in the online documentation and GitHub repository.

4. Impact

`WEC-Grid` provides a modular, reusable, and open-source software platform that lowers technical barriers for wave energy integration studies. Its standardized API refers to the function call parity enforced across all supported power system modelers through the abstract base class, ensuring a uniform user experience rather than implying adherence to an external industry-wide API. This design removes the need for custom scripting or manual data handling, while promoting reproducibility and consistent workflows across modelers.

4.1. Research

`WEC-Grid` allows researchers to conduct quasi steady-state simulations directly incorporating validated or experimental `WEC-Sim` data, eliminating the dependence on simplified dispatch curves or static data sets. By integrating realistic wave inputs such as parametric profiles or buoy measurements, researchers can investigate the impact of WEC farm size, placement, and design on grid performance, variability mitigation, and hybrid renewable configurations.

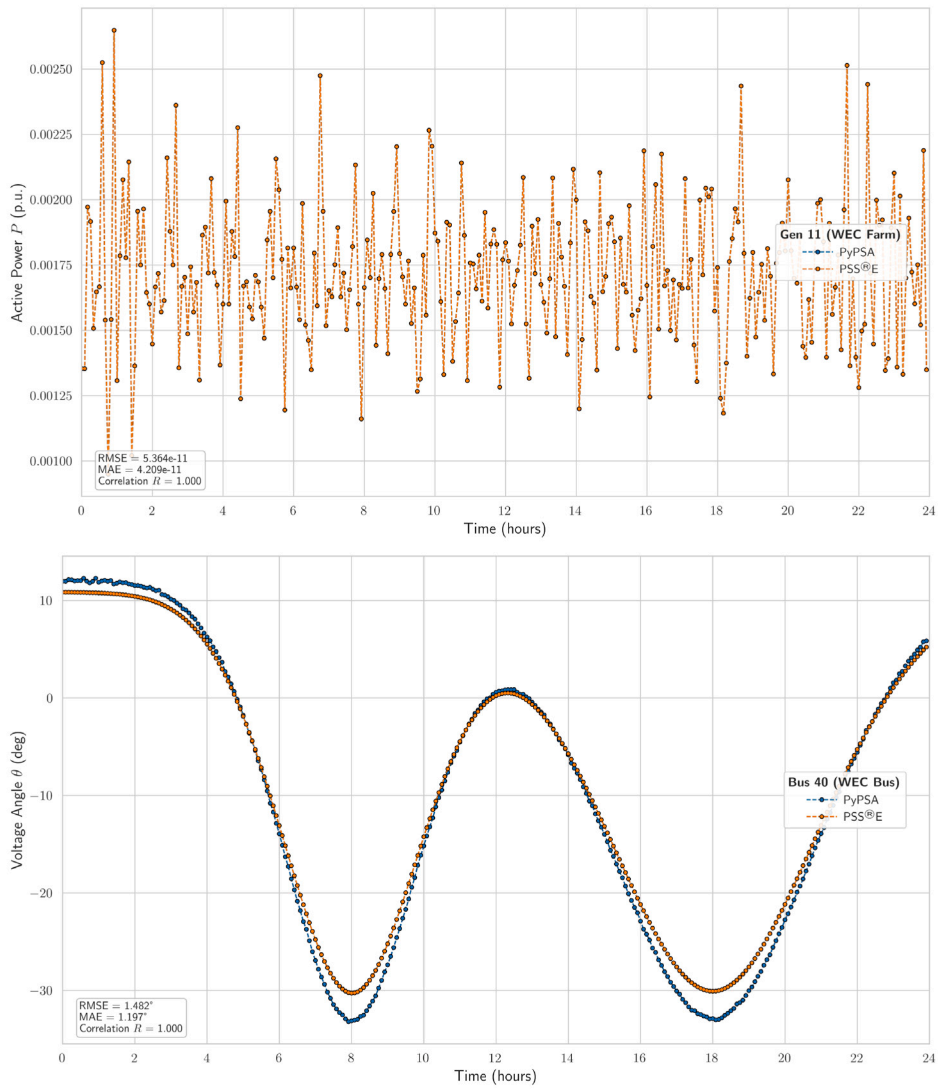

Fig. 6. Cross-modeler comparison between PSS®E and `PyPSA`. (Top) Active power at Generator 11 (WEC farm), showing nearly identical trajectories (RMSE $< 10^{-10}$ p.u.). (Bottom) Voltage angle at Bus 40, with strong agreement (RMSE 1.48°, correlation $R = 1.0$). Together, these results confirm reproducibility across solvers despite different backend implementations.

The open-source licensing, comprehensive documentation, and example-driven Jupyter notebooks enhance accessibility, making `WEC-Grid` suitable for interdisciplinary academic research and education. It supports practical, hands-on exploration of marine and power system interactions at the graduate level, promoting a deeper understanding of renewable energy integration challenges.

4.2. Industry

`WEC-Grid` supports industry stakeholders by facilitating realistic WEC integration studies within utility-standard workflows. Its direct compatibility with proprietary tools like PSS®E and open-source alternatives ensures smooth integration into existing planning and analysis environments. This enables grid operators, planners, and developers to efficiently assess WEC project feasibility, grid stability, and interconnection impacts without extensive workflow modifications.

The built-in database, simulation caching, and standardized metadata management streamline iterative project analyses, reducing the overhead of repeated scenario evaluations. By aligning with standardized device models such as RM3, `WEC-Grid` meets the evaluation needs of regulatory bodies, national laboratories, and early-stage research and development initiatives. Future capabilities, including dynamic simulation and contingency testing, will further enhance its relevance and applicability to operational and reliability studies.

4.3. Emerging research questions

Beyond immediate applications, `WEC-Grid` also opens several avenues for new research. Examples include:

- How probabilistic methods (e.g., Monte Carlo simulations) can quantify the variability and uncertainty of WEC-Grid integration scenarios.
- How simulations at different temporal resolutions (quasi steady-state vs. dynamic) compare in capturing short-term transients versus long-term stability effects.
- How hybrid renewable energy systems — combining WECs with wind, solar, and storage — can be jointly modeled to study complementary variability and optimal dispatch strategies.
- How standardized databases can enable reproducible benchmarking across modelers, sites, and device types.

These research questions illustrate the broader scientific opportunities that `WEC-Grid` enables and position it as a foundation for advancing marine energy integration studies.

4.4. Future directions

Although not implemented in the current release, the architecture of `WEC-Grid` opens pathways toward broader functionality. Near-term extensions could include controller integration with external platforms (e.g., Simulink/Speedgoat), real-time or hardware-in-the-loop co-simulation, and tighter coupling with dynamic grid models. Similar to recent advances in emulation–simulation frameworks for smart grids such as *ESPSGrid* [14], these directions highlight opportunities to combine reproducible simulation with experimental validation, bridging research and deployment needs. These directions remain outside the present scope but highlight the extensibility of the framework and its potential for future collaborative development.

5. Conclusions

`WEC-Grid` bridges a critical gap in renewable energy research by integrating hydrodynamic WEC models with established power system modelers. Its modular, open-source architecture supports quasi steady-state simulations with standardized workflows, consistent data management, and reproducible results. The illustrative case studies demonstrate the framework's ability to couple multiple WEC farms with benchmark grid models, confirming both feasibility and cross-modeler interoperability. As interest in marine renewables continues to grow, `WEC-Grid` provides a foundational tool for rigorous cross-domain studies, supporting both academic innovation and early-stage industry applications.

5.1. Future work

Building on this foundation, our future efforts will focus on extending `WEC-Grid` to support dynamic grid simulation and the integration of hybrid renewable systems such as wind, solar, and storage. These capabilities will enable comprehensive multi-renewable studies under realistic operating conditions, further enhancing the tool's value for planning and stability assessment. Direct validation against experimental datasets from facilities such as PacWave and the Hinsdale Wave Lab also represents a key long-term goal to ensure fidelity and build confidence in coupled WEC–Grid analyses.

Taken together with the broader impact pathways outlined above, these developments position `WEC-Grid` as both a practical tool for present-day studies and a foundation for future cross-domain innovations.

CRediT authorship contribution statement

Alexander Barajas-Ritchie: Writing – review & editing, Writing – original draft, Software. **Eduardo Cotilla-Sanchez:** Writing – review & editing, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported in part by the U.S. Department of Energy Office of Science, Office of Energy Efficiency and Renewable Energy, United States, Water Power Technology Office under grant number #DE-EE0009445 and also supported in part by Pacific Northwest National Laboratory scholarship number 934371896.

References

[1] Jouanne ARv, Brekken TKA, Sanchez EC, Amon E, Yokochi A. Advancing wave energy converter array-to-grid transmission systems. In: IEEE international conference on industrial technology. IEEE; 2015, p. 2476–80. http://dx.doi.org/10.1109/ICIT.2015.7125462.

[2] Parkinson SC, Dragoon K, Reikard G, García-Medina G, Özkan-Haller HT, Brekken TKA. Integrating ocean wave energy at large-scales: A study of the US Pacific Northwest. Renew Energy 2015;76:551–9. http://dx.doi.org/10.1016/j.renene.2014.11.038, URL https://www.sciencedirect.com/science/article/pii/S0960148114007538.

[3] Johnson B, Gibson NL, Cotilla-Sanchez E. A coupled Karhunen–Loève and anisotropic sparse grid interpolation method for the probabilistic load flow problem. Electr Power Syst Res 2021;193:107044. http://dx.doi.org/10.1016/j.epsr.2021.107044, URL https://www.sciencedirect.com/science/article/pii/S0378779621000249.

[4] Mendoza N, Mathai T, Forbush D, Boren B, Weber J, Roberts J, Chartrand C, Fingersh L, Gunawan B, Peplinski W, Preus R, Roberts O. Developing technology performance level assessments for early-stage wave energy converter technologies. Renew Energy 2021.

[5] Ameur A, Loudiyi K, Aggour M. Steady state and dynamic analysis of renewable energy integration into the grid using PSS/E software. Energy Procedia 2017;141:119–25. http://dx.doi.org/10.1016/j.egypro.2017.11.023, URL https://www.sciencedirect.com/science/article/pii/S1876610217354292.

[6] Attya AB, Anaya-Lara O, Ledesma P, Svendsen HG. Fulfilment of grid code obligations by large offshore wind farms clusters connected via HVDC corridors. Energy Procedia 2016;94:20–8. http://dx.doi.org/10.1016/j.egypro.2016.09.183, URL https://www.sciencedirect.com/science/article/pii/S1876610216308220.

[7] Njoka GM, Mogaka L, Wangai A. Impact of variable renewable energy sources on the power system frequency stability and system inertia. Energy Rep 2024;12:4983–97. http://dx.doi.org/10.1016/j.egyr.2024.10.057, URL https://www.sciencedirect.com/science/article/pii/S2352484724007133.

[8] Armstrong S, Cotilla-Sanchez E, Kovaltchouk T. Assessing the impact of the grid-connected Pacific marine energy center wave farm. IEEE J Emerg Sel Top Power Electron 2015;3(4):1011–20. http://dx.doi.org/10.1109/JESTPE.2015.2429577, URL https://ieeexplore.ieee.org/document/7101221/.

[9] Johnston J, Henríquez R, Maluenda B, Fripp M. Switch 2.0: A modern platform for planning high-renewable power systems. SoftwareX 2019;10:100251. http://dx.doi.org/10.1016/j.softx.2019.100251, URL arXiv:1804.05481, [math].

[10] Plietzsch A, Kogler R, Auer S, Merino J, Gil-de Muro A, Liße J, Vogel C, Hellmann F. PowerDynamics.jl—An experimentally validated open-source package for the dynamical analysis of power grids. SoftwareX 2022;17:100861. http://dx.doi.org/10.1016/j.softx.2021.100861, URL https://linkinghub.elsevier.com/retrieve/pii/S2352711021001345.

[11] Górski T. Smarts: A java package for smart contract test suite generation and execution. SoftwareX 2024;26:101698. http://dx.doi.org/10.1016/j.softx.2024.101698, URL https://linkinghub.elsevier.com/retrieve/pii/S2352711024000694.

[12] Dorado-Rojas SA, Fachini F, Bogodorova T, Laera G, De Castro Fernandes M, Vanfretti L. ModelicaGridData: Massive power system simulation data generation and labeling tool using modelica and python. SoftwareX 2023;21:101258. http://dx.doi.org/10.1016/j.softx.2022.101258, URL https://linkinghub.elsevier.com/retrieve/pii/S2352711022001765.

[13] Pai MA. Energy function analysis for power system stability. In: The kluwer international series in engineering and computer science, power electronics and power systems, Boston, MA: Springer; 1989, http://dx.doi.org/10.1007/978-1-4613-1635-0.

[14] Sande J, Wu Zhuo F, Alonso M, Pastrana S. ESPSGrid: Emulation-simulation for power smart grids. SoftwareX 2025;31:102307. http://dx.doi.org/10.1016/j.softx.2025.102307, URL https://linkinghub.elsevier.com/retrieve/pii/S2352711025002730.