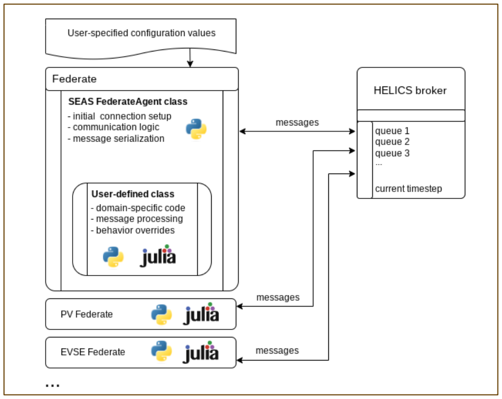
SEAS Communication Engine: An Extensible, Flexible Wrapper for Co-Simulation Agents

Sam Helman[1,2], Kinshuk Panda[1], Deepthi Vaidhynathan[1], Bernard Knueven[1], Devon Sigler[1], Wesley Jones[1], Jennifer King[1]

[1]National Renewable Energy Laboratory (NREL), [2]University of Minnesota



User-specified configuration values

Federate

SEAS FederateAgent class
- initial connection setup
- communication logic
- message serialization

User-defined class
- domain-specific code
- message processing
- behavior overrides

julia

PV Federate julia

EVSE Federate julia

...

messages

messages

messages

HELICS broker

queue 1
queue 2
queue 3
...

current timestep

Fig. 1: SEAS Communication Engine set up for simulating a distribution grid with photovoltaics (PV) and electric vehicle supply equipment (EVSE). Domain-specific Julia code is loaded for the simulated resource's lifecycle.

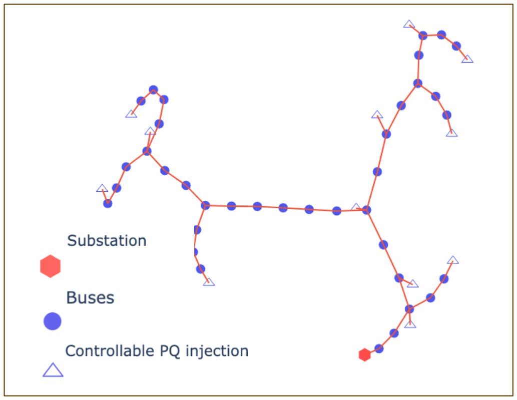


Fig. 2: SmartDS feeder[4] used to demonstrate the Realtime PV signal tracking controller.

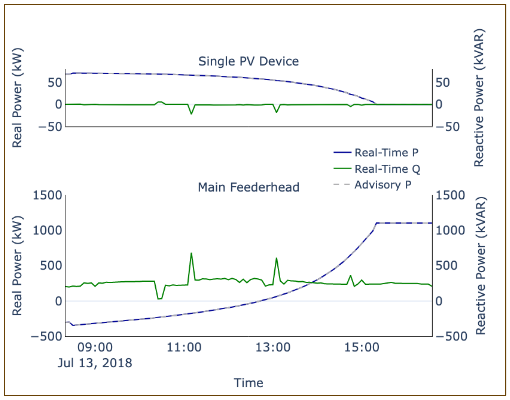


Fig. 3: Simulation results for tracking a set point for a distribution feeder head and a single PV device where the underlying communication between the various devices is handled by the SEAS Communication Engine.

The SEAS (Simulation and Emulation of Advanced Systems) Communication Engine is a flexible software library designed to enable large-scale co-simulation with simulated and emulated hardware for power systems.

Motivation

- Make it easier for power systems domain scientists to run advanced co-simulation experiments on local and HPC systems.
- Provide user-friendly high-level abstraction for HELICS that handles setup, low-level bookkeeping, and connection logic, enabling the researcher to focus on their domain-specific logic.
- Improve portability of code between different co-simulation experiments by using a common underlying engine and messaging system.

Use Cases

- Simulation of different control schemes for balancing power within distribution areas.
- Integration of different high-fidelity domain-specific simulation engines into larger experiments.
- Building portable, self-contained agents representing distributed energy resources that can be reused across simulations.

Engine Description

- Python package built on HELICS that uses a queue-based messaging system to coordinate between federates and broker for co-simulating grid-interactive models.
- Flexible, in that common logic and functionality can be inherited and adapted for a particular use case.
- Underlying models can be black-box and only need Python wrappers to interface.
- Automates network connection to a broker and the instantiation of message queues.
- Provides a simplified API for asynchronous communication between federates.
- Implements automated logic for time synchronization within co-simulation experiments.
- Adds simplified message-type annotation to prevent erroneous communication between federates and brokers.
- Includes examples demonstrating basic python usage as well as how to wrap domain-specific Julia code.

Outcome

- See **https://nrel.github.io/SEAS** for the full software package, included documentation. See the paper for a full description of validation and testing.

Future Work / Extensions

- Act as a management and load balancing layer for a group of HELICS federates, rather than just a single federate.
- Improve support for HELICS federates in different computing paradigms, e.g., the cloud, to connect using our API.
- Stronger data validation and structure; implement a full message-passing protocol on top of HELICS.

References

1. B. M. Kelley, P. Top, S. G. Smith, C. S. Woodward, and L. Min, “A federated simulation toolkit for electric power grid and communication network co-simulation,” in 2015 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), 2015, pp. 1–6.
2. D. P. Chassin, J. C. Fuller, and N. Djilali, “Gridlab-d: An agent-based simulation framework for smart grids,” 2014. Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 2020, pp. 1–5.
3. B. Palmintier, D. Krishnamurthy, P. Top, S. Smith, J. Daily, and J. Fuller, “Design of the helics high-performance transmission-distribution-communication-market co-simulation framework,” in 2017 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), 2017, pp.1–6.
4. San Roman, F. de Cuadra, N. Gensollen, T. Elgindy, and P. Duenas, “SMART-DS synthetic electrical network data opendss models for sfo, gso, and aus,” 12 2020. [Online]. Available: https://data.openei.org/submissions/2981

This work was authored by the National Renewable Energy Laboratory (NREL), operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. This research was supported by the Energy Systems Integration Facility-Facilities and Infrastructure funding and the Transformational Laboratory Directed Research and Development program at NREL. A portion of this research was performed using computational resources sponsored by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

The 2024 Conference on Innovative Smart Grid Technologies, North America
Washington D.C.
02/19/2024-02/22/2024
NREL/PO-2C00-88607