Economic Assessment of Nuclear Hybrid Energy Systems: Optimization using RAVEN

Aaron S. Epiney, Andrea Alfonsi, Cristian Rabiti, Jun Chen

Idaho National Laboratory (INL), 2525 Fremont Av. Idaho Falls, ID, 83415
aaron.epiney@inl.gov, andrea.alfonsi@inl.gov, cristian.rabiti@inl.gov, jun.chen@inl.gov

INTRODUCTION

The energy market supply is under pressure to achieve competing goals, such as minimizing the cost of electricity and greenhouse gas emissions while achieving grid resilience and reliability and meeting regulatory requirements. To study different scenarios for possible evolutions of the energy market that assume a certain renewable penetration and the possible introduction of *hybrid nuclear reactors*, the HYBRID modeling and simulation project has been initiated by INL [1]. The term hybrid nuclear reactor here refers to nuclear power plants that not only produce electricity, but can also sell heat to an industrial process, like a hydrogen generation or water desalination plant.

The RAVEN (Risk Analysis Virtual ENvironment) code developed at INL [2] together with the modeling language Modelica [3] (the Dymola compiler is used) have been selected as the main modeling tools for the HYBRID modeling and simulation project. This summary outlines the challenge, presents an overview of the developments in the RAVEN code needed to solve it and shows an illustrative example calculation.

The HYBRID modeling and simulation project is a multi laboratory effort, involving research teams from three different U.S. national laboratories, led by the Idaho National Laboratory (INL) and supported by the Argonne National Laboratory (ANL) and the Oak Ridge National Laboratory (ORNL).

BACKGROUND

One of the goals of the HYBRID modeling and simulation project is to assess the economic viability of hybrid systems in a market that contains renewable energy sources like wind. The Nuclear Hybrid Energy System (NHES) would include a nuclear reactor that not only generates electricity, but also produces by-products utilizing excess heat/electricity, like hydrogen or desalinated water. The idea is that the possibility of selling non-electric energy provides cushion to the volatility introduced by the renewable energy sources [4].

The problem to solve is to find the optimal configuration of an NHES that will minimize the cost of electricity, while accounting for defined constraints on the capability of the NHES to meet demand. These constraints have a fundamental role in enabling the economic evaluation framework by monetizing the ability of the NHES being analyzed to better cope with electricity demand volatility. The introduction of such constraints leads to the calculation of an effective cost of electricity that differs from the levelized cost of electricity (LCOE) since the cost of electricity is computed *a posteriori* to account for the effective utilization of each subsystem.

The system that is studied is modular and made of an assembly of components. For example, a system could contain a nuclear reactor, a gas turbine, a battery, a by-product production subsystem and, possibly, renewables. This system could correspond to the size of a balance area, but in theory any size of system is imaginable. The system is modeled in the 'Modelica' language.

To assess the economics of the system, an optimization procedure will be performed to obtain a set of parameters, which defines the configuration of the NHES to find the minimal cost of electricity. Fig. 1 shows a diagram of the software framework for the NHES modeling and optimization. As one can see, the statistics and optimization code RAVEN is used as a driver for the whole problem. RAVEN is running the optimization, i.e. RAVEN changes the input parameters in the system model, provides the needed time histories for demand, wind, etc., runs the Dymola system model, collects output from Dymola and assesses the next optimization step. As mentioned, the figure of merit for the optimization is the cost of electricity. The Dymola output is used in a simple cash flow analysis that will reveal the cost of electricity.

The optimization routine seeks various combinations of input variables (according to a defined optimization algorithm) to find the minimum cost of electricity while the system is requested to cope with random synthetic time histories of electricity demand and renewable supply. The analysis does not rely on bidding on marginal cost for every hour, but tries to find the minimal cost to produce a certain amount of electricity with a given time profile, e.g. for a representative week or year. This means that the number of parameters to optimize is high. In addition to the mean demand and the component capacities, the utilization factor for every hour for every component is also an optimization parameter. The number of optimization parameters for a one-month optimization with a three-component system would be 2164:

1 for the mean demand.
1 for the renewable supply (this defines the renewable capacity).
2 for the nominal capacities of the 2 components additional to the renewable in the system.
3*720 for the number of utilization factors (3 components, 720 hours/month).

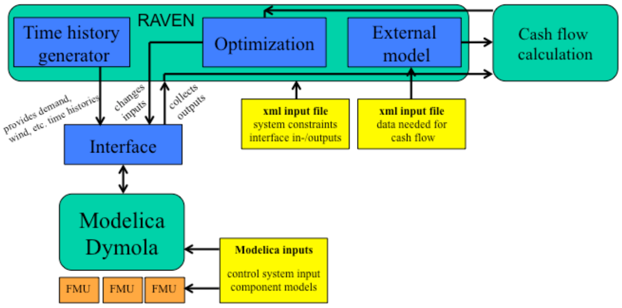

Fig. 1. HYBRID system modeling and optimization framework.

NEEDED RAVEN CAPABILITIES

As one can see from Fig. 1, for the HYBRID modeling and simulation problem, different capabilities inside RAVEN are needed. The following subsections will detail the different capabilities and associated RAVEN developments.

"Ensemble model"

As one can see from Fig. 1, multiple "models" are involved in the solution of the problem, e.g. the synthetic time history generation, the code interface to the Dymola code, the cash flow calculation as well as a few pre- and postprocessors to convert and process the data that are not shown in the figure. A capability in RAVEN has been developed that is able to manage the data flow between these models called "Ensemble Model" [5]. The user can input the needed inputs and provided outputs for each "model" and RAVEN will connect the models based on their Input/Output relations and decide in which order they have to be executed. The "Ensemble Model" framework in RAVEN can deal with different complexities, from simple sequential execution of the involved models, to non-linear systems that need iteration of a model or a set of models.

Alias System

The different "models" involved in an "Ensemble Model" can have different variable names for the same quantity. For example, the nominal capacity of the renewable energy may be called "cap_ren" in Dymola, "renewable_capacity" in the cash flow model and "C_ren" in the RAVEN input. An alias system has been created inside RAVEN that allows to alias any model input or output variable to a RAVEN variable. In the above example, in the RAVEN input C_ren will be aliased to cap_ren in the Dymola code interface model and C_ren will be aliased to renewable_capacity in the external model interface for the cash flow model.

Synthetic time history generation

Another new RAVEN capability is the synthetic time history generator. This capability is needed to generate synthetic scenarios for renewable generations and grid loads. The generated time series are prepared to statistically conform to the actual measurement but possess different temporal profiles. In particular, a combined model with Fourier series and autoregressive moving average (ARMA) [6] is utilized to de-trend the yearly measurements and to characterize the autocorrelation of the residues. In RAVEN, the model can be trained with a database of any number of data points. The trained model is then able to generate synthetic time series. The synthetic data generation consists of generating independent white noise for each time step, utilizing the ARMA model to compute residues for each time step, and finally adding the Fourier series representing seasonal trends. The user can ask the model to produce synthetic data for any representative time period, e.g. a week or a month as well as a desired time discretization, e.g. hourly. By using multiple synthetic histories, one can claim to do an optimization for a given scenario taking into account the stochastic nature of the problem, instead of optimizing for just one historic data set (which is equivalent of optimizing the problem assuming perfect knowledge of the future). Fig. 2 and Table 1 show an example of synthetic electricity demand generation.

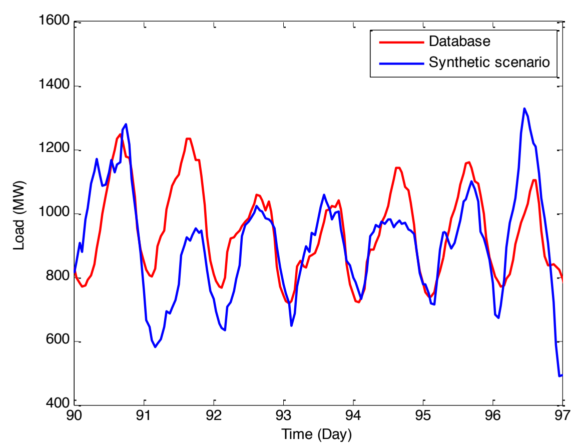

Fig. 2. Synthetic load scenario and the actual database for selected 7 days.

Code interface to Modelica/Dymola

In order for RAVEN to communicate with Dymola, a code interface has been developed. When a Dymola model, e.g., the NHES model, is implemented, platform dependent C-code and the corresponding executable are generated for simulation. After the executable is generated, it may be run multiple times (with Dymola license). Furthermore, separate text files containing model parameters and initial conditions are also generated as part of the build process. The RAVEN Dymola interface modifies input parameters by changing copies of these files. RAVEN can then run the Dymola executable with these changed files. Dymola generates a .mat file containing the requested outputs. The Dymola code interface can read this output file and extract the variables needed by RAVEN.

Table I. Comparison between Synthetic and Actual Data

Statistics	Database	Synthetic
Mean (load)	1102.3	1103.4
Standard deviation (load)	222.2	223.8
Mean (step to step diff.)	0	0
(step to step difference)	48.4	54.2

Cash flow analysis

As mentioned, the optimization of the NHES tries to minimize the cost of electricity. In order to compute the cost of electricity, a cash flow analysis is performed. The cash flow analysis is used to determine the cost of the electricity that would make the Net Present Value (NPV) equal to zero, enforcing therefore a fair economical profit for the NHES in its whole. The cash flow analysis is implemented in an "external model" written in Python in RAVEN. The "Ensemble Model" framework in RAVEN has access to this model.

Stochastic optimization

The NHES optimization problem described above is stochastic, i.e. every time RAVEN generates a set of input data for Dymola, the synthetic time history generator will provide a different history. For example, if RAVEN asks for the same mean electricity demand twice, the temporal profiles will be different. To optimize such stochastic problems, the stochastic optimization algorithm Simultaneous Perturbation Stochastic Approximation (SPSA) [7] has ben implemented in RAVEN. Conventional gradient-based algorithms assume that the gradient can be evaluated for every point of the function to be optimized (loss function), which may be difficult for high dimensional problems and stochastic problems. The SPAS algorithm does not have this requirement, but estimates the gradient of the loss function and follows this gradient to the functions minimum. Since the Dymola model of the NHES and the associated cash flow model consists of hundreds of optimization variables and require minutes or even hours to perform one evaluation, SPSA is chosen as the optimization engine in this project. The user can also input constraint functions for the input space (see Fig. 3), i.e. upper bound and lower bound for each optimization parameter as well, as well as more complex constraints like f(a,b)<c. Constraints on the outputs can be handled implicitly by adding a penalty to the loss function.

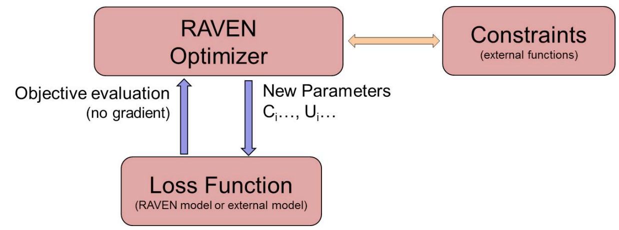

Fig. 3. RAVEN optimization workflow.

PROOF OF INFRASTRUCTURE FUNCTIONING

To test the new developed models and RAVEN capabilities in an integrated way, an example has been calculated. The goal of this example is to demonstrate the correct data flow between the RAVEN external models, in particular:

- The "synthetic time series"
- A simple system model (to be replaced with Dymola model)
- The cash flow model
- The optimization

The "Ensemble Model" and data flow of the example is shown in Fig. 4. First, RAVEN samples the mean demand and the installed capacities for all components except the nuclear reactor, which is supposed to have a fixed capacity. RAVEN then passes one sample of the mean demand and the installed wind capacity to the "Synthetic time series generation" model. This model generates time series for the demand and available wind capacity. Values of the demand and the available wind for every hour of a year are passed back to RAVEN. RAVEN passes these data together with the sampled installed capacities for all components to the "Simple system model". This external model computes the electricity produced, fuel consumed and CO_2 produced for each component in order to satisfy the demand curve. In this example, the "simple system model" contains dispatch rules for the different components and the utilization factors (as mentioned in the Background above) are not part of the optimization parameters. The production, fuel consumption and CO_2 production are then passed back to RAVEN. RAVEN passes these data together with the sampled installed capacities for all components to the cash flow model. This model computes and passes back to RAVEN the cost of electricity for this configuration of the system.

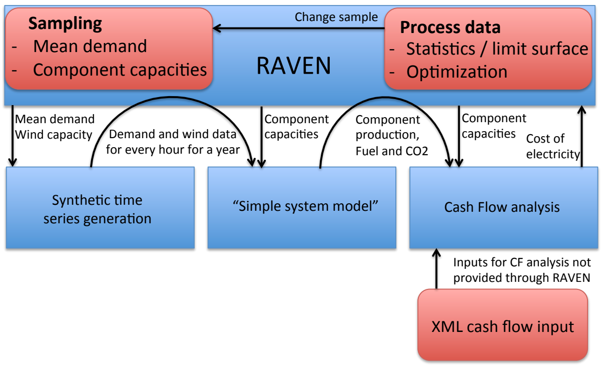

Fig. 4. Data flow for example calculation.

It is worth mentioning that with this methodology, the optimization does not know exactly the demand of the future, even though the optimization is done over the whole year. Since a new synthetic time history is generated for each sample passed by RAVEN, the optimization can be considered for a given mean demand only, not for a given (known) demand history.

The simple example includes a nuclear reactor, wind farm, gas turbine and battery storage. RAVEN has been asked to sample the four variables mean demand and renewable, gas turbine and battery storage capacity between 100MW to 1,000MW while the reactor capacity is constant at 300MW. The optimizer has computed 250 iterations of the optimization using a carefully chosen set of optimizer parameters. Fig. 5 shows the optimization path for the three dimensions mean demand, renewable and gas turbine capacity while Fig. 6 indicates how these dimensions converge. It can be seen that the algorithm converges to a minimum cost of electricity of \$0.016/kW. It should be mentioned, these numbers are just to illustrate the proper working of the integrated example with the optimizer and do not reflect a real minimum cost of electricity for a NHES.

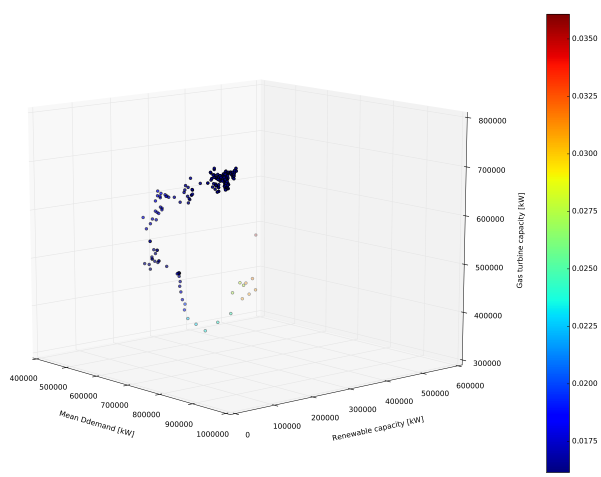

Fig. 5. Optimizer path for cost of electricity.

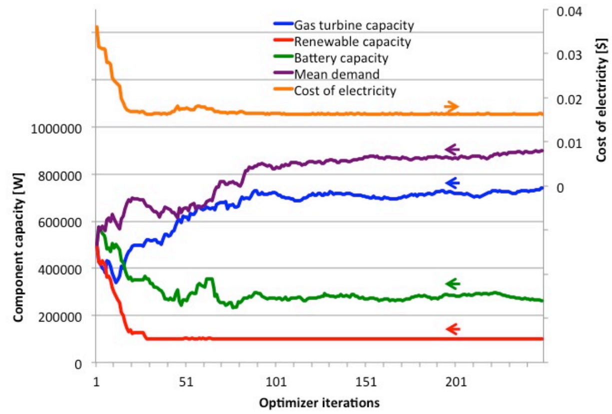

Fig. 6. Iteration of optimization variables and cost of electricity.

REFERENCES

1. C. RABITI et al., "Strategy and Gaps for Modeling, Simulation, and Control of Hybrid Systems," INL/EXT-15-34877, INL, April 2015.

2. C. RABITI et al., "RAVEN user manual", INL/EXT-15-34123, Revision 5, February 2016.

3. H. ELMQVIST et al., "Modelica—The Next Generation Modeling Language an International Design Effort," proceedings of the 1st World Congress on System Simulation (WCSS'97), Singapore, September 1–3, 1997.

4. A. EPINEY et al., "Status on the Development of a Modeling and Simulation Framework for the Economic Assessment of Nuclear Hybrid Energy Systems (FY 16)", INL/EXT-16-39832, INL, September 2016.

5. A. ALFONSI et al., "RAVEN Facing the Problem of assembling Multiple Models to Speed up the Uncertainty Quantification and Probabilistic Risk Assessment Analyses", proceedings of the 13th international conference on Probabilistic Safety Assessment and Management (PSAM 13), Seoul, Korea, 2-7 October, 2016.

6. J. CHEN et al.," Synthetic wind speed scenarios generation for probabilistic analysis of hybrid energy systems", *Energy*, 2016, http://dx.doi.org/10.1016/j.energy.2016.11.103

7. J. C. SPALL, "An Overview of the Simultaneous Perturbation Method for Efficient Optimization," Johns Hopkins APL Technical Digest, **19**(4), pp. 482–492 (1998).