Machine Learning Tools Set for Natural Gas Fuel Cell System Design

ECS Transactions, 103 (1) 2283-2292 (2021)
10.1149/10301.2283ecst ©The Electrochemical Society

Machine Learning Tools Set for Natural Gas Fuel Cell System Design

D. Wang[a], J. Bao[a], Z. Xu[a], B. Koeppel[a], O. A. Marina[a], A. Noring[b], M. Zamarripa-Perez[b], A. Iyengar[b], E. Eggleton[c], D. T. Schwartz[c], A. Burgard[b], and D. Miller[b]

[a] Pacific Northwest National Laboratory
[b] National Energy Technology Laboratory
[c] University of Washington

This study is focusing on leveraging the system design tools set for the next-generation solid oxide fuel cell (SOFC) based natural gas fuel cell (NGFC) system. Conventionally, system design and optimization of NGFC systems rely heavily on traditional reduced order model (ROM) techniques and designers' experience level. For overcoming the technical barriers of system design, multiple multi-physics models and machine learning (ML) tools have been utilized to automate the conceptual design process and enhance the reliability of solutions for the NGFC system. The proposed tools set includes a physics-informed ML tool for automated ROM construction that leverages advances in deep neural networks to significantly reduce ROM prediction error for the NGFC power island compared to traditional approaches. The constructed physics-informed ML ROM can be used in system design, and optimization tools set Institute for the Design of Advanced Energy Systems (IDAES) Process Systems Engineering (PSE) framework. The tools set also provides a user-friendly graphic user interface built within Jupyter Notebooks, and the complete tools set is open-source public available.

Introduction

Nowadays, there is increasing demand for clean and renewable energy sources. Solid oxide fuel cells (SOFCs), which produce energy through the electrochemical reaction, are considered a promising candidate as electricity providers due to their high efficiency and low emission of greenhouse gas (1). Solid oxide fuel cells usually operate at high temperatures (above 600 °C); the high temperature allows the fuel cells to take a wide variety of fuels, both renewable and non-renewable (1). This is one reason why SOFCs are becoming more attractive as an alternative energy source, along with overall high thermal efficiency, long-term performance stability and relatively low operation cost. Among various types of fuel cells, natural gas fuel cells (NGFC) are considered a potential source of power for household use due to the high-efficiency conversion of methane into heat and energy.

Machine learning technology is rapidly developing and used in many different areas, such as searching results, filtering spams, system recognition (3). It is also being utilized in the fields of healthcare, security, and robotics through deep learning (3). Machine learning can also better understand customers' preferences and estimate costs for the

manufacturing industries, as it can handle large amounts of data and learn to make high-accuracy predictions (4). Moreover, machine learning is widely used in software development. For the software developing process, finding faults in the software at an early stage is extremely helpful for testing activities and minimizing development costs (4). Although many simulation and modeling works have been undertaken to investigate fuel cell health and try to improve system performance, such as reducing the operating temperature of the NGFC to make it a viable portable energy source (2), we found that rare efforts have been made to apply machine learning methods to system design and optimization of the NGFCs.

In this study, we propose to leverage the conceptual design tools with the Institute for Design of Advanced Energy Systems (IDAES) Process Systems Engineering (PSE) framework to develop an optimized, next-generation solid oxide fuel cell (SOFC) based natural gas fuel cell (NGFC) system. First, deep neural networks (DNN) and physics-informed DNN are introduced for automated reduced-order models (ROMs) construction to significantly reduce ROM prediction errors compared to traditional approaches (from around 10% to 5%). Then, importing the DNN or Physics-informed DNN constructed ROM into the IDAES-PSE framework will significantly reduce the dependence of the conceptual design process on designers' expertise and time consumption.

Machine Learning for cell and stack power island ROM construction

An example of the natural gas fuel cell system is shown in Figure 1. It includes an auto-thermal natural gas reformer, a SOFC stack power island, an oxy-combustor, etc. The auto-thermal reformer provides a prescribed degree of external reformation of the pipeline natural gas with CH_4 as the primary component to modulate the amount of endothermic on-cell reforming. The power island or SOFC stack produces energy and heat through electrochemical reactions. In previous works, PNNL has developed a 2D SOFC-MP (solid-oxide fuel cell multi-physics) software to simulate the operating conditions in planar symmetric SOFC stacks (5, 6). The oxy-combustor recovers unutilized anode-exhaust chemical energy. As can be seen in Figure 1, an anode gas recirculation loop is included to maintain a suitable oxygen-carbon ratio (O/C) to prevent carbon formation in addition to provide the steam required in the reformation process. More details of the NGFC system and the SOFC-MP solver can be found in (2, 5, 6).

Numerical simulation of power island is not computationally efficient for the design and optimization of SOFC systems, so reduced-order models (ROM) are developed as an efficient alternative to the numerical simulation model of the SOFC stack. The model inputs are listed in Table I. Based on the sampled average current density, fuel temperature, etc., numerical simulations with the SOFC-MP solver are conducted. Various model outputs which are considered essential for the system design are listed in Table I as well.

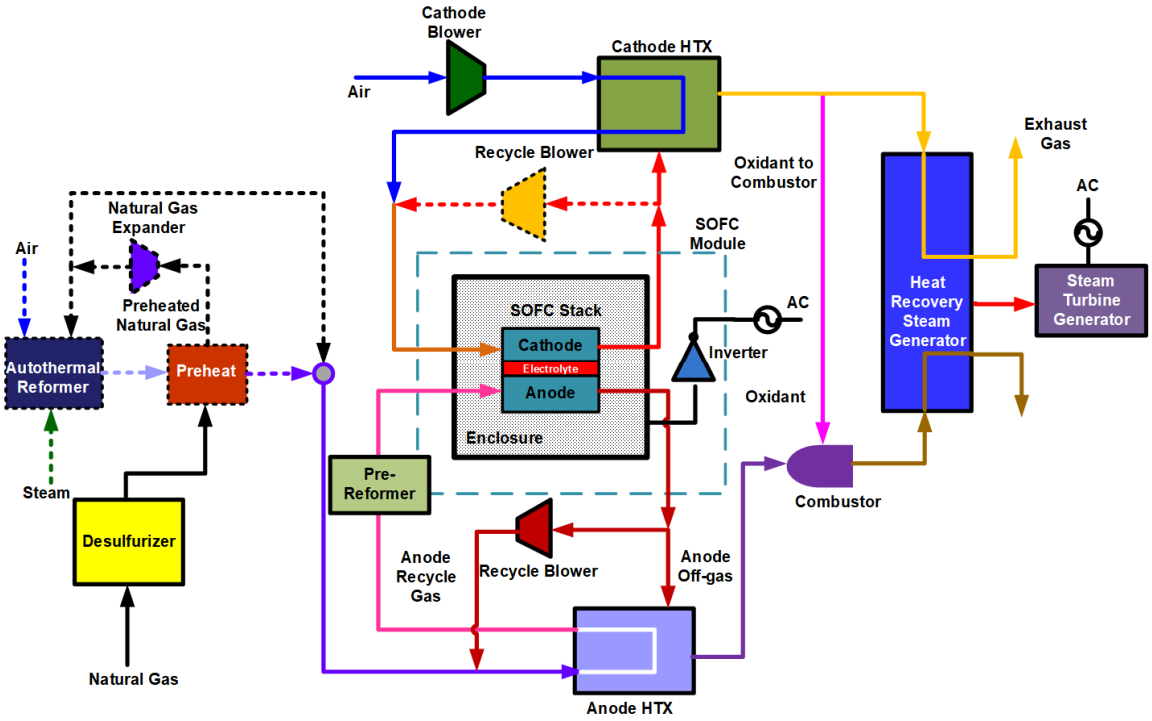

Figure 1. Schematic chart of NGFC system.

Table I. Model inputs and outputs.

Input parameters	Output parameters
Average Current Density	Stack Voltage
Fuel Temperature	Avg Cell Temperature
Internal Reforming	Outlet Fuel Temperature
Oxidant Temperature	Outlet Air Temperature
Oxidant Recirculation	Delta Air Temperature
Oxygen to Carbon Ratio	Air Heat Exchanger Effectiveness
Stack Fuel Utilization	Fuel Utilization
Stack Oxidant Utilization	…
Pressure	

Approach Based on Deep Neural Networks (DNN)

Traditionally, the kriging algorithm can be used to build ROMs for SOFC stack performance prediction (7). In this study, the proposed system design tool imports Deep Neural Networks for ROM construction. DNN is a machine learning method based on Artificial Neural Networks (ANN), which is based on the human nervous system. It usually includes one input layer, one output layer and multiple hidden layers. Figure 2 shows the sketch of the deep neural networks in this study. The DNN is utilized to develop the regression model that builds the relationship between the model inputs and outputs. Correspondingly in this study, it can construct reduced order models from the SOFC-MP solver inputs and outputs. The numbers of hidden layers and neural nodes in each layer are carefully adjusted to achieve a reasonable convergence rate and high prediction accuracy of the validation set. The red box in the plot gives an example of the structure of an individual neural node. In an individual node, the inputs multiply the weights, add the bias, and then be modified by the activation function. In this study, a

rectifier function is used as the activation function, and the TensorFlow package is utilized for constructing and training the DNN (8).

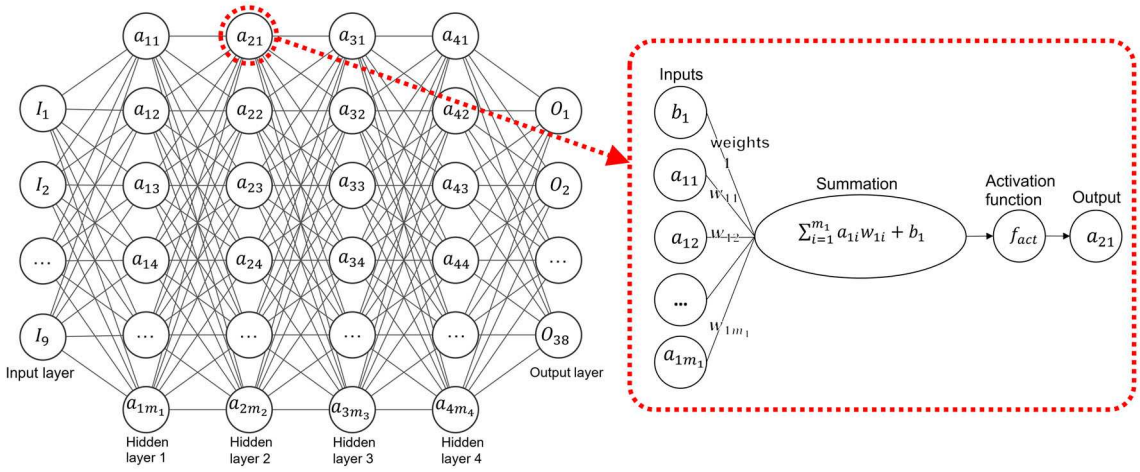

Figure 2. Sketch of the Deep Neural Networks used for learning the relationships between NGFC inputs and output parameters.

Deep neural networks can both make a regression (continuous outputs) and classification (discrete outputs). DNN-based classifiers were developed in this study to classify the physically operational conditions for nine categories of NGFC power islands. For each NGFC system configuration, 20,000 training datasets are employed, 80% for training purposes and the rest for testing purposes. The accuracies of each trained classifier can be seen in Figure 3 as grey bars. As can be seen, for all nine categories of NGFC systems, the classification prediction error rates are less than 8%.

DNN-based ROMs were developed as well for the same nine categories of NGFC stacks. More than 8,000 cases are employed as the training dataset for each configuration, while 2,000 cases are for testing purposes. And for each NGFC system, the ROM prediction accuracy is quantified by the averaged root-mean-square-error (RMSE) of all the ROM output parameters. As shown in Figure 3 as orange bars, the performance of ROM predictions is impressive, within 5% for all nine categories of NGFCs.

$$RMSE = \sqrt{\sum_{i=1}^{N}\left(y_{i,pred} - y_{i,true}\right)^2 \Big/ N} \tag{1}$$

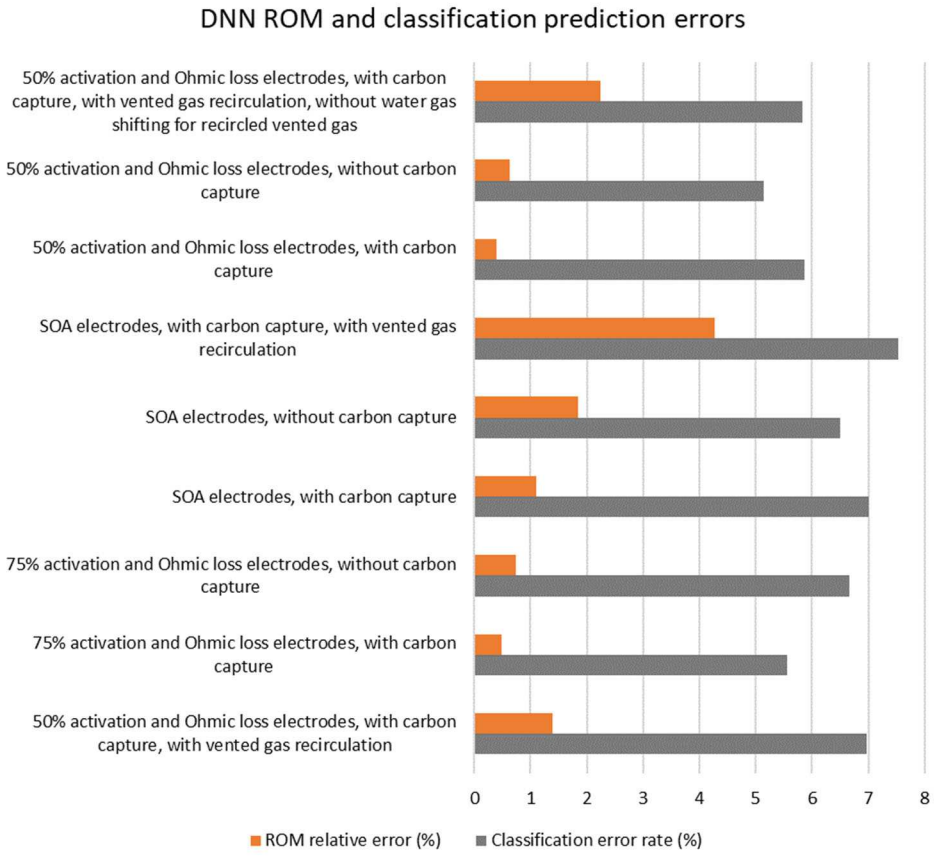

Figure 3. The accuracies of DNN-based classifiers and ROMs.

Approach Based on Physics-informed DNN

To further increase the classifier/ROM accuracy, the physics-informed DNN (Phy-DNN) method is developed by coupling the previously developed DNN-based classifier/ROM with a simple, fast-forward physics model. The fast forward physics model is a simple mass balance model (MBM). It can estimate whether the NGFC is physically operational and provide the approximations of the power island inlet and outlet species composition based on the requested current. Due to the methane reforming in the NGFC stack, this fast-forward physics model cannot accurately determine the species factions. Still, it can estimate these values almost instantly, which can improve the DNN regression model accuracy.

Figure 4 shows the algorithm of the physics-informed DNN-based classifier. The step-by-step illustration is: 1. Build the DNN-based ROM, attain ROM predictions; 2. Get the fast forward physics model approximations; 3. Combine the ROM predictions and physics model approximation with the original training dataset, construct the classifier with the above DNN method. Utilizing this Phy-DNN algorithm, classifiers were built for the nine categories of NGFC power islands. The classification error rates are shown in Figure 5 as red bars, along with the classification rates of the DNN-based classifiers shown as grey bars. As can be seen in the plot, for all nine NGFC systems, the Phy-DNN based classifiers further reduce the classification error rate of each individual NGFC to less than 5%, respectively.

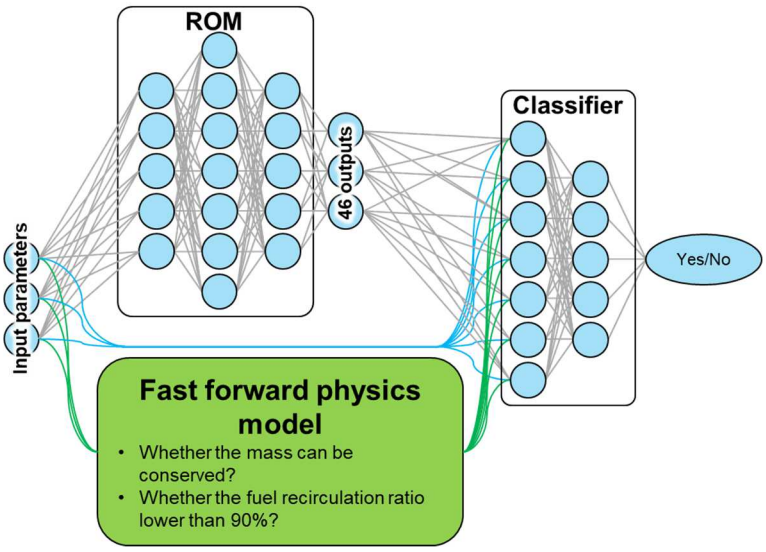

Figure 4. Sketch of the Physics-informed DNN-based classifier.

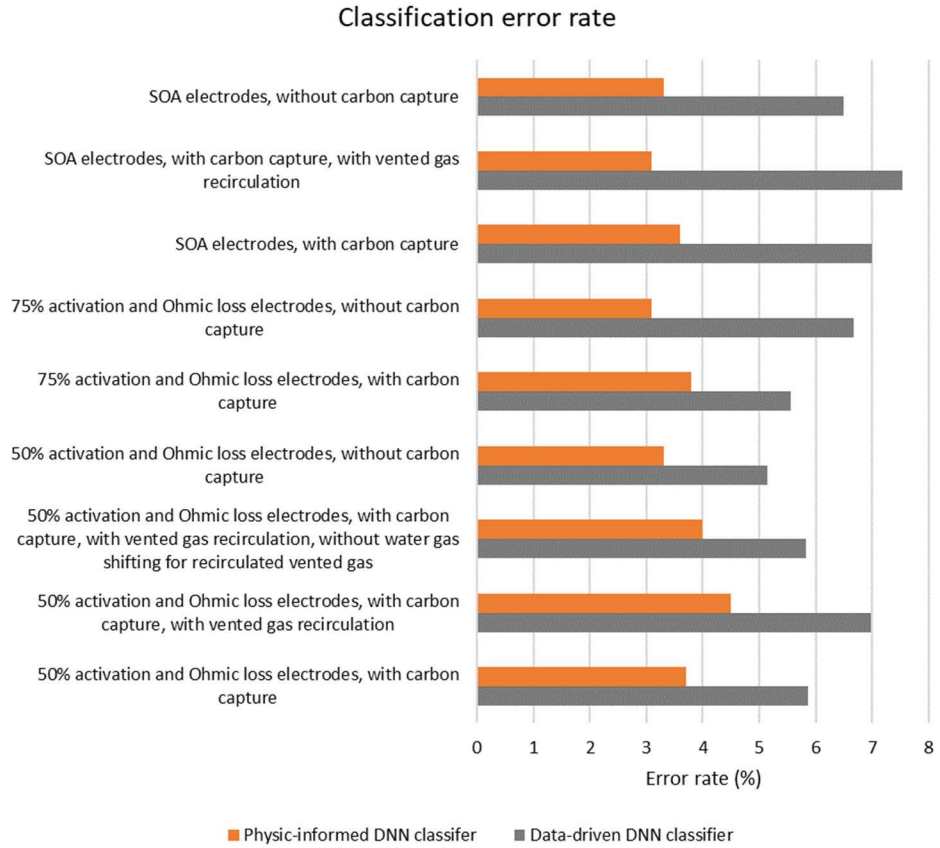

Figure 5. Comparison of the accuracies between DNN-based and Physics-informed DNN-based classifiers.

The same methodology is also implemented to build an enhanced fidelity ROM for NGFCs. Figure 6 shows the algorithm of the Phy-DNN based ROM. The step-by-step illustration is: 1. Attain estimations of inlet and outlet species composition with the mass balance model; 2. Calculate the differences between the MBM estimations and the actual values; 3. Train the differences to construct the ROM. To be noted, this Phy-DNN based ROM can only be applied to the output parameters relating to power island inlet and outlet species composition. By doing this, we expect the Phy-DNN ROM predictions closer to the actual values compared to the traditional methodologies. Figure 7 provides straightforward comparisons of the ROM prediction rates among the traditional kriging-based, the DNN-based and the Phy-DNN-based methods. As shown in the plots, the Phy-DNN ROM prediction error rates are considerably lower than the others, especially for the small training dataset sizes.

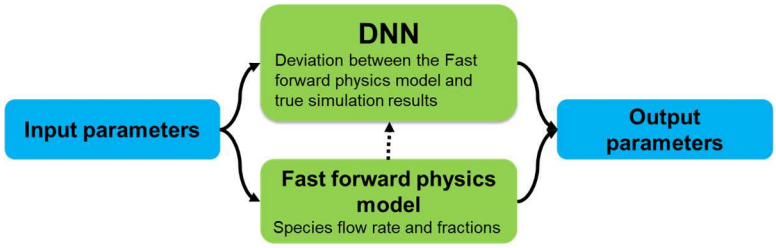

Figure 6. Sketch of the Physics-informed DNN-based ROM.

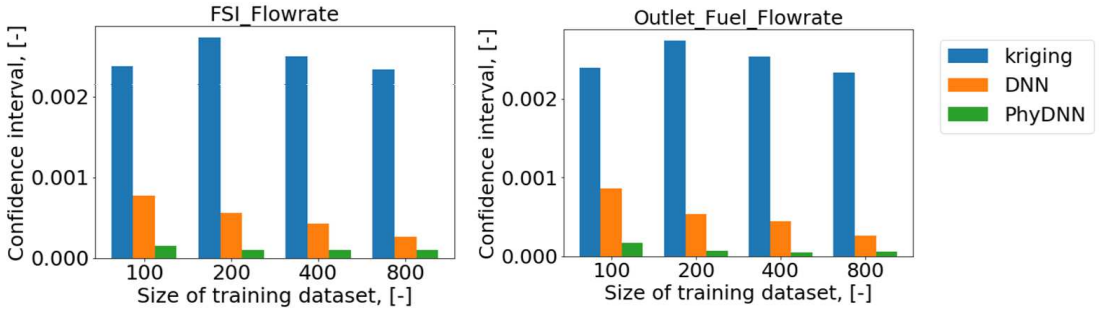

Figure 7. Comparison of the accuracies among different ROMs.

Automated ROM Construction Tool/GUI

A graphic user interface (GUI) for automated ROM construction has been developed and open-source released on Github. Figure 8 shows snapshots of the notebook. With enabling the Jupyter Notebook extension, "Collapsible Headings", all details and cells are collapsed into the titles of the four main steps, as shown in Figure 8(a). Please note that the extension "Collapsible Headings" is unnecessary for the proposed tools; it only shows the notebook in a relatively cleaner style.

Step 1 involves importing the packages, and the actual ROM construction takes place Steps 2 to 4: "Prepare cases," "Run Simulations," and "Build ROMs and Predict." In Step 2, the user can prepare either SOFC stack-only cases or the NGFC/IGFC stack cases with nearby supporting units and pipelines. After the simulation cases are ready, the user can choose to run simulations on a local desktop computer or the remote high-

performance computing (HPC) cluster in Step 3. There are detailed instructions and explanations for the steps in the notebook, and the user can easily follow the steps to complete the simulations and collecting the result data. Once all the result data are collected, the user can use the Kriging-based, the DNN-based and/or the Phy-DNN-based methods to train ROM models from the simulation results in Step 4, as shown in Figure 8 (d). The confidence interval and data visualization are also in this step. Finally, the constructed ROM will be stored in the files that can be loaded by IDAES system modeling.

This ROM GUI and the Jupyter Notebook are designed for general applications. Theoretically, with certain modifications, preparing cases and running simulations can be compatible with other physics-based solvers. The requirement is that the solver should use a text input file for controlling simulations. The current "Step 4 Build ROMs and Predict" step can already be used with data from other simulations if the data are in the appropriate format.

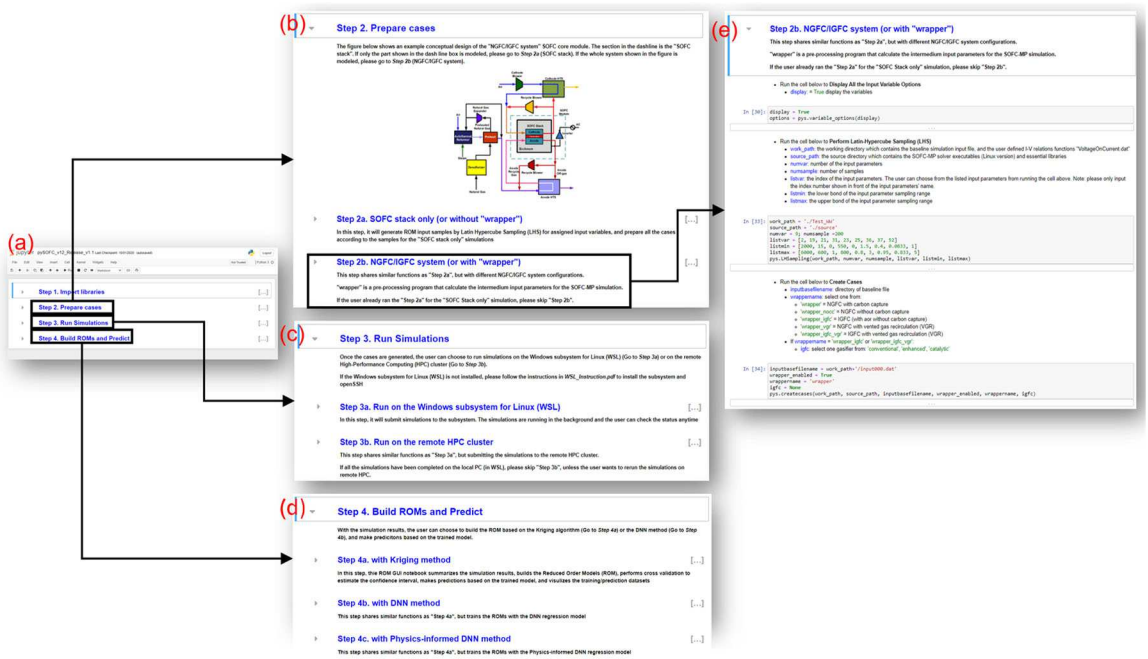

Figure 8. Screenshots of the ROM construction Graphic User Interface.

System Design and Superstructure Optimization

The superstructure of the NGFC system is built within the IDAES-PSE framework, and it incorporates an auto-thermal natural gas reformer, a SOFC power island, an exhaust combustor, etc. The above constructed ROM is linked to the IDAES system modeling framework. It can calculate the SOFC electrical performance as a function of system parameters at a low cost. As the IDAES system modeling optimization is a gradient-based optimization framework, a relatively light-weighted and high-accuracy ROM can significantly benefit the system optimization in the IDAES framework for computational efficiency and convergence.

For this example cell, the active cell area is 550 cm^2, and the single cell power output is ranging from 80 to 300 w. Table II shows preliminary results of the superstructure optimization of the NGFC system. The first row is the base case, and the following rows are the optimal solutions for a single variable (highlighted in the table), respectively. The Interior Point Optimizer (IPOPT) is utilized here to find the maximum system efficiency. As can be seen, for each variable, the optimizer converged to an optimal solution with higher efficiency than the base case.

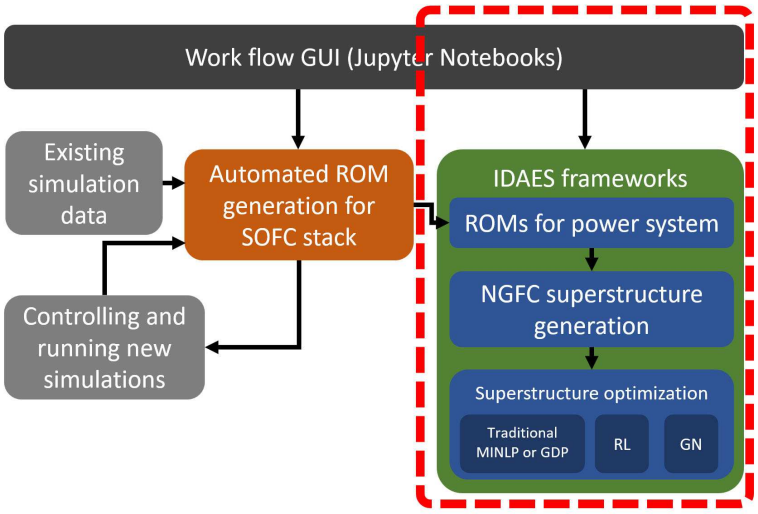

Figure 9. ROM imported into NGFC superstructure for system design optimization.

Table II. Single-variable Optimization of the NGFC System

Optimization Target: Efficiency	Fuel Temperature	Fuel Utilization	Internal Reformation Percentage	Oxidant Recirculation Fraction
60.10%	348.33 C	80%	60%	50%
60.19%	329.68 C	80%	60%	50%
60.62%	348.33 C	82.70%	60%	50%
61.48%	348.33 C	80%	70.30%	50%
61.00%	348.33 C	80%	60%	71.30%

Conclusions

To leverage the conceptual design tools with the IDAES-PSE framework to optimize next-generation SOFC-based NGFC system, DNN and Physics-informed DNN are introduced for automated reduced-order model construction. Comparing to traditional ROM construction methods, the machine learning methods significantly reduce computational cost and increase the ROM prediction accuracy: getting classification rates and ROM prediction error rates within 5% for different NGFC system configurations.

An automated ROM construction GUI is developed in this study as well. It can prepare cases and run simulations with physics-based solvers, and train ROM models from the simulation results or third-party datasets in the appropriate format. The ROM GUI has been publicly released on Github:

https://github.com/NGFC-Lib/NGFC-Lib/tree/master/jupyter/Python_ROM_GUI

Later, the DNN-based or Physics-informed DNN-based ROMs are imported to the superstructure of the NGFC system built within the IDAES-PSE framework. It will significantly benefit the system design and optimization of NGFCs.

Acknowledgments

This research was supported by the United States Department of Energy (DOE) Advanced Research Projects Agency-Energy (ARPA-E) Design Intelligence Fostering Formidable Energy Reduction and Enabling Novel Totally Impactful Advanced Technology Enhancements (DIFFERENTIATE) program.

References

1. Andersson, M., J. Yuan, and B. Sundén, SOFC modeling considering hydrogen and carbon monoxide as electrochemical reactants. Journal of Power Sources, 2013. 232: p. 42-54.
2. Choudhury, A., H. Chandra, and A. Arora, Application of solid oxide fuel cell technology for power generation—A review. Renewable and Sustainable Energy Reviews, 2013. 20: p. 430-442.
3. Lee, J.H., J. Shin, and M.J. Realff, Machine learning: Overview of the recent progresses and implications for the process systems engineering field. Computers & Chemical Engineering, 2018. 114: p. 111-121.
4. Gondra, I., Applying machine learning to software fault-proneness prediction. Journal of Systems and Software, 2008. 81(2): p. 186-195.
5. Koeppel BJ, C Lai, and MA Khaleel. 2011. SOFC-MP 2D User Manual. PNNL-20433, Pacific Northwest National Laboratory, Richland, Washington.
6. Lai K, BJ Koeppel, K-S Choi, KP Recknagle, X Sun, LA Chick, V Korolev, and M Khaleel. 2011. “A quasi-two-dimensional electrochemistry modeling tool for planar solid oxide fuel cell stacks.” Journal of Power Sources 196 (2011):3204–3222.
7. Zhang, X., Current status of stationary fuel cells for coal power generation. Clean Energy, 2018. 2(2): p. 126-139.
8. Abadi, M., et al., TensorFlow: Large-scale machine learning on heterogeneous systems. 2015.