Digital Chemical Engineering 5 (2022) 100050

Contents lists available at ScienceDirect

Digital Chemical Engineering

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

Best practices and methods

Digitalization of an experimental electrochemical reactor via the smart manufacturing innovation platform

Berkay Çıtmacı[a], Junwei Luo[a], Joon Baek Jang[a], Prakashan Korambath[b], Carlos G. Morales-Guio[a,**], James F. Davis[a,b], Panagiotis D. Christofides[a,c,*]

[a] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA
[b] Office of Advanced Research Computing, University of California, Los Angeles, CA 90095-1592, USA
[c] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA

ARTICLE INFO

Keywords:
Electrochemical reactor
CO_2 reduction
Smart manufacturing innovation platform
Data-driven modeling
Machine learning
Gas chromatography
Data connectivity

ABSTRACT

The exponential increase in data produced over the last two decades has revolutionized the way we collect, store, process, analyze, model, and interpret information to improve profitability. Manufacturing is no exception. However, Smart Manufacturing, the digital practice, organization, workforce, and infrastructure transformation for collection and deployment of data and models at scale and at all levels of manufacturing, is a complex, costly, and labor-intensive journey that is still seeing slow adoption. The Clean Energy Smart Manufacturing Innovation Institute (CESMII), a national Manufacturing USA public-private partnership sponsored by the Department of Energy, is addressing this scaled use of data and modeling in manufacturing. CESMII has focused on how to collect and use operating data for numerous applications that improve productivity, precision, and performance of manufacturing operations from factory floor to supply chain using process simulation, predictive analytics, monitoring and control, and real-time optimization. Because contextualized data are key, CESMII has developed the Smart Manufacturing Innovation Platform (SMIP) to lower the barriers to the data that are needed to accelerate data-based model building, improve data visualization, and more quickly gain insights. Reusable, standards-based ways of doing data collection, ingestion, and contextualization are particularly important for scaling access and use of data. The SMIP uses a standards-based definition and construct for reusable information models called an SM Profile. When an SM Profile is used in conjunction with the SMIP, the SMIP ensures the availability of contextualized, operational data for model building. The present work demonstrates Smart Manufacturing and the application of the SMIP for building several data-centered models for the operation and control of an experimental electrochemical reactor that reduces carbon dioxide (CO_2) gas to valuable liquid and gas chemicals, such as alcohols, olefins, and syngas. We describe how the SMIP plays a central role in more effective model building and we demonstrate how the electochemical reactor can be controlled and optimized for the desired products. Use of the SMIP involves the transmission of real-time sensor measurements to a cloud resource so that the operating data are available to all model building experts. The data collection and transmission process is fully automated to greatly reduce the need for manual manipulation of the data. Data-driven machine learning models are used for advanced real-time state estimation, real-time optimization, and model-based feedback control for the reactor. The application models are implemented as a system to monitor the data flow and control the electrochemical reactor with a single visualization interface. SM Profiles are used to demonstrate reusability of the information models for the reactor and the instrumentation. The application packages, algorithms, and user interfaces developed are cast as Docker images in a library to facilitate reusability of the application models.

1. Introduction

The 21st century has witnessed a rapid increase in the amount of data produced in industrial sectors as a result of many more sensors and devices, i.e., IoT (the Internet of Things), and a much greater ability to collect and manage data. The use of operational data to improve manufacturing operations with advanced sensors, controls, platforms, and modeling at much greater scale and integration is described as Smart Manufacturing in the U. S. and Industry 4.0 in Europe (e.g., Christofides et al., 2007; Kang et al., 2016; Yang et al., 2020). Smart Manufacturing

* Corresponding author at: Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA.
** Co-corresponding author. Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA.
E-mail addresses: moralesguio@ucla.edu (C.G. Morales-Guio), pdc@seas.ucla.edu (P.D. Christofides).

https://doi.org/10.1016/j.dche.2022.100050
Received 31 July 2022; Received in revised form 21 August 2022; Accepted 21 August 2022

is the digital transformation that results from embedding digital technology into nearly every aspect of all manufacturing operations and practices. Application objectives encompass increasing productivity, product quality, and performance, ensuring safety, and reducing energy consumption, greenhouse gas (GHG) emissions (Saudagar et al., 2019) and carbon footprint as integrated Key Perfomance Indicators (KPIs). Smart Manufacturing also involves the seamless integration of advanced sensing with data-centered modeling, modeling that spans first-principles modeling, digital twin, Artificial Intelligence (AI), and machine learning, for simulation, diagnosis, real-time optimization, prediction, advanced control, and data-driven decision-making from factory floor to supply chains. An abundance of the right data allows for the automation of processes and the development of intelligent systems, which are crucial to the management of higher level KPIs and complex objectives, eliminating human error, and using human resources in better ways not only on the factory floor but throughout supply chains and manufacturing ecosystems. Data and modeling also facilitate much broader and more timely data availability, acquisition and selection that can accelerate insights and research breakthroughs.

Digital transformation defined by Smart Manufacturing includes building and sustaining data-centered models, collecting, ingesting and contextualizing the right data, making data reusable, addressing the information technology (IT) infrastructure and scaling the deployment software applications to address the wide diversity of manufacturing operations. Although the potential benefits of Smart Manufacturing are understood, the needed digital transformation across the industry is difficult in practice due to cost and complexities associated with the current infrastructure and business practices and the lack of tools, infrastructure, standards, and skills for sustained data management and model building. The tasks of digital transformation are new and there is a significant need for advanced operator training (Phuyal et al., 2020). The ability to extract more meaningful relations and greater insights involves working with the data in a more integrated fashion with methods for large, more complex data sets. Smart manufacturing defines the use of real-time data to increase the effectiveness of analyses, and it positions the role of data for greater understanding from the beginning of a development process.

Smart Manufacturing also emphasizes the relationships within large data sets in an efficient and timely manner for operational modeling and troubleshooting. This includes embracing the unique opportunities to leverage cloud technologies for storing large amounts of data and making them accessible anywhere and anytime and for shared computational requirements to significantly decrease the time needed for the analysis, optimization, control, and scale-up. However, cloud technologies have their challenges. Various cloud vendors still lock data in respective platforms that do not interface easily. Cloud technologies do offer the needed services to store large data sets and make them accessible anywhere and anytime but cross-platform data transfer is challenging. Cybersecurity vulnerabilities increase when integrating multiple cloud products and/or ingesting data from multiple vendor instrumentation. The data transfer with manufacturing process equipment, sensors, or local processor must be encrypted (HTTPS), and the data stored on the cloud must be protected. A manufacturing framework must address the necessary authorizations and authentications to keep the user data safe.

In recent years, machine learning algorithms have evolved, offering greater opportunity for modeling and identifying hidden patterns or trends in big data sets, but they require working with data in more extensive ways. The spectrum of data-driven modeling options and machine learning algorithms and their span of application are also expanding. Regression methods, such as artificial neural networks (ANN), support vector machines, and gradient boosting, have been successful in modeling process operational data. Wu et al. (2019) explored the use of recurrent neural networks (RNNs) for modeling time series data in chemical processes and demonstrated that RNNs can be effectively used in nonlinear control schemes, such as model predictive control (MPC). Several studies have successfully integrated ML algorithms into chemical engineering processes to develop new robust predictive models for multi-phase flows and reactors. Zhu et al. (2022) have summarized recent ML applications to hydrodynamics, heat and mass transfer, and reactions in single-phase and multi-phase flow systems. Machine learning approaches have also found use in the realm of quantum calculations to predict chemical properties and reactivity. For example, a deep neural network (DNN) was constructed to estimate reaction rate parameters from an extensive partition function database (Komp and Valleau, 2020). The use of transfer learning in a DNN for activation energy estimation was also demonstrated starting from a data set generated from Density Functional Theory (DFT) calculations (Grambow et al., 2020). The use of AI and ML reaches beyond modeling of process data from traditional point sensors. For instance, Haas et al. (2020) developed a convolutional neural network model that is used to detect effervescence in a multi-phase flow with a high-speed camera that collects images of fluid dynamics and counts the number of bubbles. In another study, the pressure drop of a cyclone separator was modeled with a hybrid genetic algorithm, radial basis function neural network (GA-RBFNN), and was subsequently used to optimize separation parameters (Elsayed and Lacor, 2012). Collectively, these studies highlight a wide spectrum of applying ML methods in modern-day processes and the diverse ways in which they can be integrated into Smart Manufacturing practices.

Smart Manufacturing and Industry 4.0 applications have emphasized productivity, precision, and performance. For example, Kumar et al. (2017) developed a Smart Manufacturing approach for managing temperature variations in a steam methane reformer (SMR). The application exploited a reduced-order model (ROM) drawing on large data sets from computational fluid dynamics (CFD) simulations, which were too computationally time-consuming to use in real-time. For real-time application, an optimizer used the ROM to determine the set-points for the fuel flowing to many distributed burners. As conditions in the SMR changed, the ROM was updated periodically and automatically using the CFD simulations. Infrared (IR) cameras were used as advanced thermal imaging sensors to measure the temperature spatially throughout the SMR. Similarly, Ren et al. (2021) designed an Industry 4.0 framework for metal alloy additive manufacturing that optimized productivity and precision by employing micro-scale, meso-scale and part-scale finite element method (FEM) modeling of the manufactured parts. The operational portion of the framework comprised a convolutional neural network (CNN) that was trained for production monitoring and defect detection. It was used to study the process to continuously update additive manufacturing process recipes. Large amounts of image data were stored in the cloud and cloud analytics were used to automatically update the CNN model with incoming manufactured part images. This update process included a strategy for splitting the data into testing and training data sets for automatically updating the CNN model. Korambath et al. (2016) and Botcha et al. (2018) presented how a common platform accelerates and facilitates their research on application model development in discrete parts manufacturing.

One of the promising research areas in sustainable energy production and chemicals manufacturing is electrochemistry. Electrochemical CO_2 reduction is an attractive and emerging process that can produce valuable chemicals such as ethylene, ethanol, acetaldehyde, and syngas. Due to the complexity of the reaction mechanisms, industrial-scale, first-principle models have not yet been developed. Consequently, data-driven models are an attractive alternative for modeling this process and enabling model-based process optimization and control strategies (Mistry et al., 2021). In this research, the objective was to demonstrate that the complex electrochemical reactor could be controlled without the electrochemical reaction being fully understood yet. To proceed, data needed to be generated from an experimental electrochemical cell for which there was already data from many runs at various condi-

tions. The electrochemical cell was also available for additional runs. Having access to raw data at various conditions introduced significant data acquisition and data modeling challenges but the ability to work with the operating data directly provided significant opportunity to understand the reaction better. Smart Manufacturing and AI/ML methods were brought together to study and demonstrate how to control an electrochemical reactor to optimize the desired products.

For our electrochemical reactor, the Smart Manufacturing digitalization process started with the objective of building machine learning models from data and demonstrating that the complex catalytic reaction can be controlled to optimize the conversion of CO_2 to ethylene. In the present study, we focus on the Smart Manufacturing methods and infrastructure used to accelerate the development of the reaction, optimization, and control models. This involved connecting the reactor instrumentation to the Smart Manufacturing Innovation Platform (SMIP) developed by the Clean Energy Smart Manufacturing Innovation Institute (CESMII), the national Manufacturing USA Institute sponsored by the Department of Energy. The SMIP integrates the necessary tools for data collection, connection to historians and standards-based Open Platform Communication (OPC) servers, data storage, correlation extraction, and data contextualization. The SMIP facilitated flexible access to data and supported model evaluation, selection, and building. Digitalization of the experimental system and automation of the data connections, ingestion and contextualization, facilitated communications among the domain, modeling, instrumentation, and IT experts. Having ready access to the contextualized data from the experiments in real-time and historically across experiments accelerated both the modeling development efforts and the implementation of the control models to demonstrate reactor controllability.

The Smart Manufacturing lens also carries the objective of building templates for the finished models to save time and effort in reusing them in similar applications. These reusable templates are called “SM Profiles.” Smart Manufacturing uses industry standards in the SMIP and the Profile construct to simplify the connection, ingestion, and contextualization of the real-time data from the required instrumentation. When a data interface is constructed or a particular data contextualization approach is set, these can be captured and reused. This is especially difficult for legacy equipment and instrumentation and when multiple vendor products are involved. In the last decade, many of the product vendors created proprietary software tools for their equipment, which are often not compatible with each other and/or difficult to interface with external software applications. Legacy software applications themselves have tended to embed data for a particular function essentially trapping it for other uses. Given the widespread nature of these issues, the lack of a standard model or common platform to exchange data has limited the opportunities in terms of exploiting available data modeling options in the advancement of complex manufacturing systems.

This paper presents how the Smart Manufacturing approach has been applied to the digitalization and control of an electrochemical CO_2 reduction reactor and its instrumentation with the SMIP. The rest of this manuscript is organized as follows. In the next section entitled “Smart Manufacturing in Experimental Electrochemical Reactor Setup,” the experimental reactor setup and the key Smart Manufacturing hardware and software setups are described. In the section, entitled “Advanced Sensors,” the real-time point sensor measures are described and the need for an automated gas chromatography (GC) spectra processing algorithm is explained. In the section entitled “The Role of CESMII Smart Manufacturing Innovation Platform (SMIP) in Electrochemical Operation Research,” the SMIP involvement with examples and a novel Laboratory Instrument Engineering Workbench (LabVIEW) interface are presented. In the section entitled “Virtualization,” the advantages of the use and implementation of Docker containers are discussed. Finally, the iterative evaluation process and resulting advantages of using ML and hybrid models are addressed in the section entitled “Electrochemical Reactor Modeling Using SMIP”.

2. Smart manufacturing in experimental electrochemical reactor setup

The primary elements of the Smart Manufacturing setup for UCLA’s experimental electrochemical reactor are shown in Fig. 1. With automated processes and workflows, data collected from the reactor are used to extract insights and build operational machine learning models. The reactor inputs include electrical potential, current, and rotational speed. These are data collected by the potentiostat, which is composed of a sensor for measuring the electrical current and potential, and an actuator for tuning the applied potential. Liquid product concentrations are measured using nuclear magnetic resonance (NMR) after the experiment is completed while gas concentrations are measured on-line with gas chromatography (GC) run at 20 min intervals. For real-time control, we focused on the GC analysis for the controlled variables. The required gas injection and processing of the GC measurements need to be automated and modeled to use the gas product analysis for real-time control purposes. Overall, input and output data are processed to extract relationships among the applied potential, electrode rotation speed and temperature, and the gas production rates and concentrations in building the ML models. There were unique challenges in building models from actual experimental data, however. For example, the limited amount of GC data produced per experiment eliminated the use of some ML modeling approaches, including recurrent neural networks which require large data sets to be effectively trained. There are also important reaction phenomena that needed to be captured in the data sets and subsequently reflected in the data-based model. The most important was being able to model the quick deactivation of the atomically flat catalyst. Rapid deactivation causes a change in selectivity that shifts production away from the desired products. This phenomenon was overcome by enhancing statistical ML methods with kinetic constants, which were calculated by Luo et al. (2022). The cumulative integral of the current was used to establish a correlation between the current and catalyst deactivation, and feature engineering was used by Çıtmacı et al. (2022) to increase the model performance. This was essential for a feedback control strategy that could optimize the products while driving the process to a desired, energy-optimal set-point. Our experimental research setup is shown in Fig. 2.

Fig. 3 shows a generalized perspective of the Smart Manufacturing (SM) building blocks and how the interactions between the building blocks are addressed with SM Profiles. Below is a summary of how the different building blocks described by Davis et al. (2020) map to our modeling objectives and onto our experimental electrochemical reactor setup. Details on the electrochemical reactor modeling, optimization, and control are presented in greater depth in Luo et al. (2022) and Çıtmacı et al. (2022).

With reference to Fig. 3, sensing is a fundamental building block of Smart Manufacturing. Contextualization is the process that converts raw data into an interpreted form, which includes the data type, name, location, purpose, units, processing, and relevant meta information. Collecting and contextualizing the current and applied potential at one second intervals from the potentiostat is a straightforward example of converting data from point sensors directly measuring the quantities of interest. Automating the contextualization of the raw Gas Chromatograph (GC) data for the gas concentrations measurements is much more involved. Specifically, GC gas concentrations are determined by comparing measurements with calibration files using standard gases and known concentrations. The raw GC data are collections of electrical signals from the GC process in which each measurement run takes just over 14 min in our setup. These raw data are contextualized by eliminating irrelevant intervals and classifying them with respect to collusion intensity. The data are passed to a software application developed in this study called the GC code, which automates the data interpretation process relative to known baselines.

The data for the input settings and the output gas concentrations were collected over the time of a reaction across many runs and used

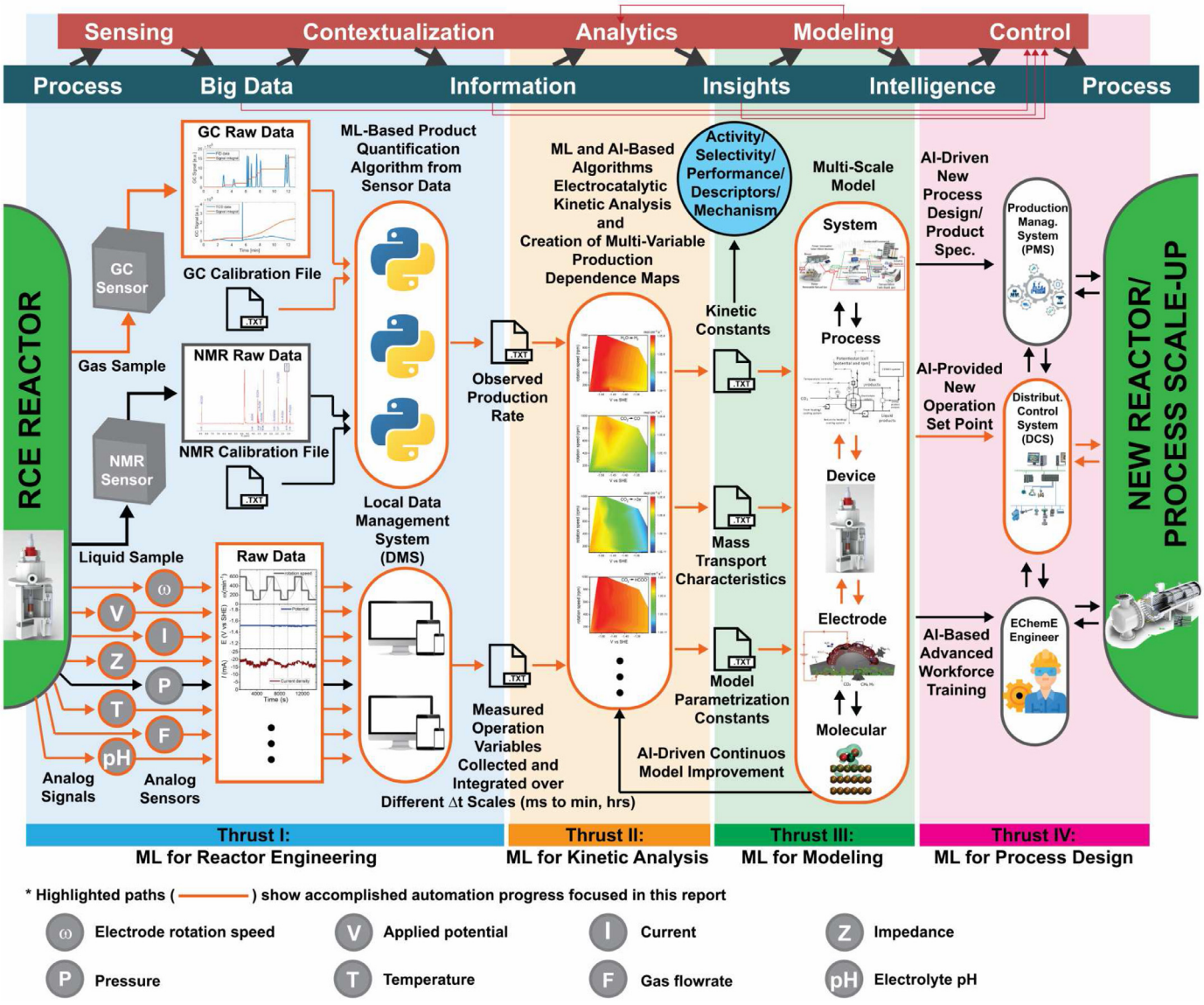


Fig. 1. Data flow and automation strategy for the experimental setup. The tasks achieved in our research are highlighted in orange.

to determine several correlations needed in model building. The cumulative integral of the current was identified as a proxy measurement of catalyst degradation (Çıtmacı et al., 2022) and was used to predict and plan catalyst regeneration procedures. Surface potential is a calculated measurement that uses current, applied potential, and solution resistance. It proved to be a better representation of the electro-catalytic driving force compared to only the applied potential. The surface potential was calculated and recorded with appropriate units for use in the ML model. The wide-ranging input conditions across many runs provided the data used to build a dynamic model that included nonlinear relations between surface potential and current (Çıtmacı et al., 2022).

Steady-state and dynamic ML models were used to generate new insights on the reaction kinetics. First, we demonstrated that the ML models based on GC data can be used to predict reaction rates. This reaction rate model was then inserted into a first-principles gas-phase dynamic mass balance model to estimate the gas-phase ethylene (C_2H_4) concentration (Çıtmacı et al., 2022). The ability to regulate the gas-phase ethylene concentration by manipulating the applied potential was explored using the dynamic model. The fast-decaying catalyst activity introduced additional non-linearity and uncertainty that was accounted for by modifying the feedback controller parameters in real-time similar to classical controller gain scheduling. Moreover, the reaction was found to go through a selectivity shift from the desired ethylene product to the undesirable methane products at potentials that were more negative than a threshold potential. This led to a control strategy in which the target set-point needed to be approached slowly with a small proportional controller gain to delay the selectivity shift. Finally, a computational method was developed to include GC sensor feedback data to correct the gas-phase C_2H_4 estimation.

These reaction insights from the modeling process were incorporated to produce model predictions which in turn were combined with process optimization tools to maximize the process economics and energy savings. The optimization was accomplished using a steady-state neural network model, which determined the most profitable set points for ethylene production. The Interior Point Optimizer tool (IPOPT; see Remark 1 below for more details) was used along with electricity costs and chemical sales prices. Finally, the feedback control of the electrochemical reactor was realized using a hybrid (i.e., combining machine learning with first-principles) model constructed with open-loop data. An ML estimation-based feedback controller was used to control C_2H_4 gas-phase concentration by manipulating the applied potential (Çıtmacı et al., 2022).

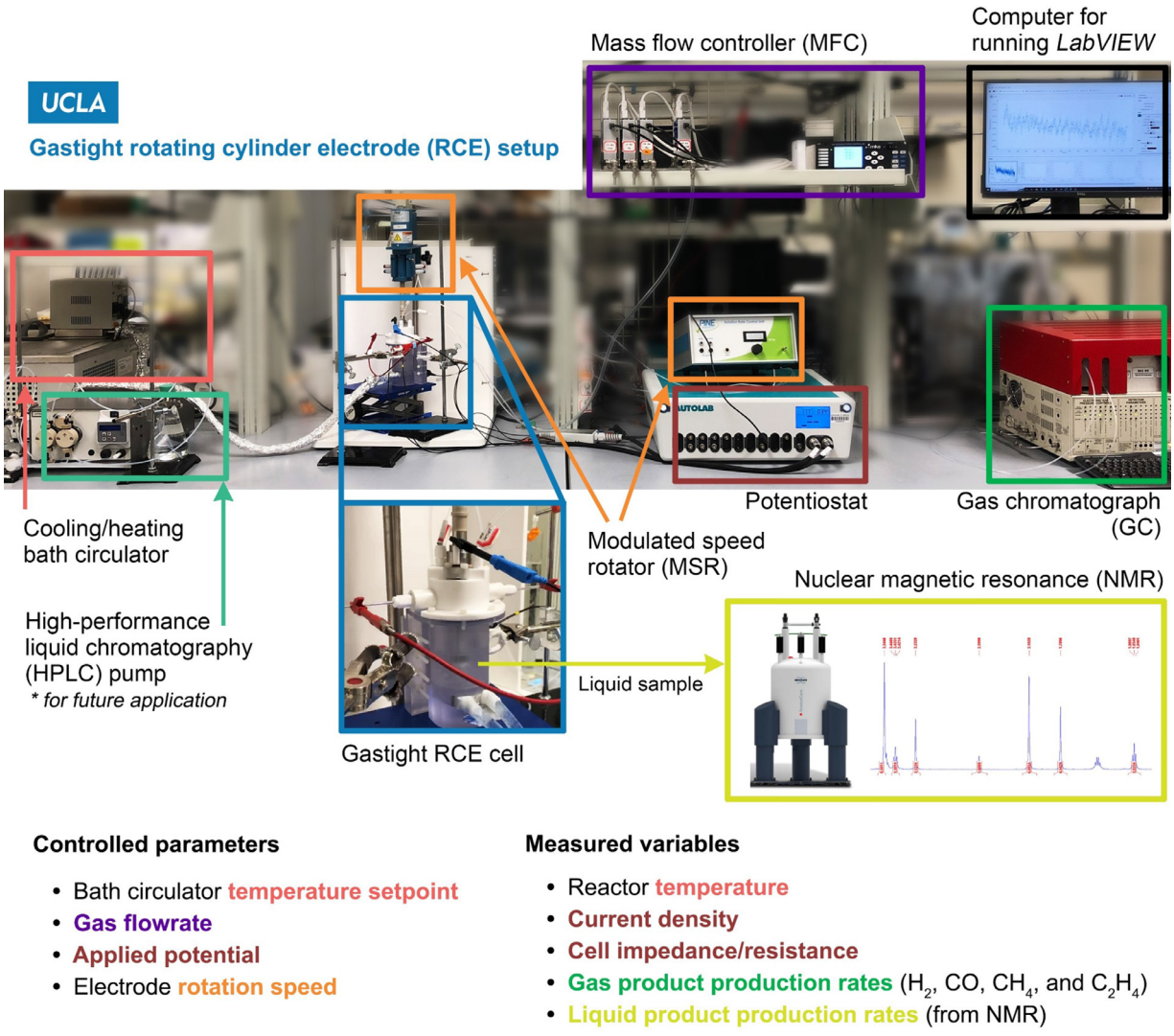


Fig. 2. UCLA gas-tight RCE reactor setup.

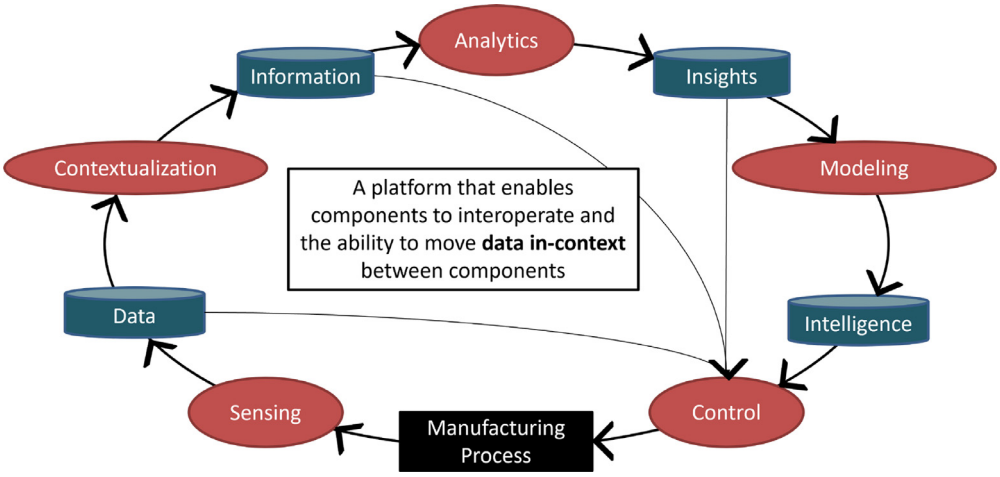


Fig. 3. Smart Manufacturing building blocks.

Remark 1. The development of a first-principles dynamic model involving electrochemical reactions is still in its infancy. Our work demonstrated the value of data-based modeling for further process exploration, understanding, and scale up. It is important to emphasize that working with experimental data presented multiple challenges compared to using simulated data. Connectivity problems, uncontrollable variations in experimental conditions, and shifts that caused anomalies and excursions all needed to be carefully handled during model building. At the same time, working with the experimental data was essential in understanding and modeling important reaction phenomena from a control perspective that had not been previously done. The richness of this study would not have been possible without access to the operating data.

3. Advanced sensors

As process sensors become cheaper and smarter (in the sense of measuring complex process properties), advanced algorithms are needed to translate primary sensor measurements into useful information about the process that can potentially be used for modeling and control purposes. A thermocouple is a good example of a sensor that is frequently used in process industries. The voltage measured by a thermocouple is converted to a temperature unit through an algorithm that is simple and well-understood for data contextualization. On the other hand, the vastly richer image data from the IR cameras employed in the steam methane reformer work previously mentioned (Kumar et al., 2017) was used to map the radial and axial temperature distributions of tubular reforming reactors in a SMR furnace. Cameras were placed around the outside of the furnace. However, due to the sequence and orientation of the cameras, some of the tubes were not fully visible. An algorithm was developed to convert the infrared images into temperature values and interpolate the temperature values for the regions that could not be observed. These direct and modeled measurements were then used to determine the fuel distribution to achieve a much more uniform temperature throughout the furnace.

The automated GC processing algorithm developed in this study is a similar example of smart sensing. As stated, gas chromatography is an analytical technique that separates a sampled gas mixture into its components that are then quantified. A gas chromatography unit contains a long thin column, where the gases travel until they hit a detector. Separation occurs inside the column containing materials that serve as stationary phases while a carrier gas (mobile phase) transfers analytes toward the detectors. The impact and the quantity of gas molecules on the detector are represented with peaks in height and breadth over the time of the measurement. As the concentration of a gas increases, the area of the corresponding peak also increases. The concentrations of each gas species analyzed can be quantified by comparing the area under the peaks to that of a calibration file, which contains reference signals generated using a standard gas consisting of known concentrations of all the gases of interest. The ratio between the area of a peak appearing during electrolysis and the calibration peak gives the concentration of the produced gas when multiplied by the known calibration concentration.

In automating the GC analysis so the results can be used on-line, it is necessary to process the measurements with a baseline correction to obtain accurate results. This is because the areas under the various peaks need to be calculated via numerical integration and the peaks that do not have consistent baselines on the X axis yield misleading areas. Automating this baseline process is therefore a vital part of the on-line approach. Most of the proprietary GC software programs create an automatic baseline. However, in most cases, the baseline fails to bring the base of each peak to the X axis since the algorithms calculate a best fit accounting for the entire measurement rather than calculating individual regions. Automation requires analyzing the baseline for each peak and then arranging the needed baseline for the components of interest. When this peak and baseline process is done manually, it is subject to substantial human error from run to run. The automated approach resulted in a much more consistent interpretation.

Fig. 4 illustrates the PeakSimple GC analysis software interface. One of the products of interest here is ethylene shown with a peak highlighted in yellow. The peak has a baseline higher than the X axis which needs to be corrected manually for an accurate composition calculation.

3.1. Automated GC code

The stability of the black line shown in Fig. 4 indicates that the quality of data coming from the GC detector is high. However, this is not necessarily the case. The example shows how an increase in column temperature and an injection of water from a saturated gas can cause the baseline to drift and the signal curve (black line) to be much higher or inclined with respect to the X axis. This situation with the raw GC sensor data is described in this article as noisy data. To create an autonomous intelligent system, the GC code needs to run reliably without human intervention and must be robust with making corrections to this noisy data. Accordingly, an autonomous GC data contextualization algorithm was developed in Python with the following workflow:

- Start a GC run at the desired times.
- Extract the raw GC data file in American Standard Code for Information Interchange (ASCII) format.
- Baseline the raw GC data.
- Detect when the overall baseline is not accurate enough.
- Calculate various baselines for each peak and recalculate the optimal baseline.
- Calculate the area from raw data if the optimal baseline still has bases above the X axis.
- Calculate the areas from the calibration files and determine measured concentrations using calibration area.
- Send concentration data to a database.

A plot generated by the automated GC code is displayed in Fig. 5. The purple line is the raw data coming from the GC sensor. Compared to the black line in Fig. 4, the raw data is much more inclined and higher. However, the automated GC code can deal with the noisy data and calculate the areas under the curves. The different colors around the peaks represent more peak-specific techniques to improve accuracy, which will be discussed in detail in the following subsection. Fig. 6(a) shows noisy raw data for a H_2 measurement and how the baseline is corrected.

3.2. Automated GC working mechanism

There are four gas products coming from our electrochemical CO_2 reduction reactor that must be measured by GC. The products are CH_4, CO, C_2H_4, which are detected via a Flame Ionization Detector (FID) equipped with a methanizer, and H_2, which is detected via a Thermal Conductivity Detector (TCD). Two separate signal channels for the FID and TCD detectors are needed and there is a plot generated for each of them. Fig. 6 compares manual and automated baseline plots for the TCD channel and the raw H_2 data. TCD senses changes in filament temperature and resistance due to thermal conductivity differences between the analytes and the reference carrier gas, while FID detects ions generated from the pyrolysis of organic analytes for the other gases. When equipped with a methanizer, the FID unit can detect CO and CO_2 with great sensitivity.

Raw data in the form of signal intensities from these detectors are transmitted to the SMIP where the contextualization process occurs. The GC code itself is tuned using GC results from 43 open-loop experiments. This tuning process accounts for extreme cases and creates a hierarchy for peak detection and integration. Baselining is implemented using asymmetric least squares smoothing as proposed by Eilers and Boelens (2005). This method generates the optimal baseline by solving the

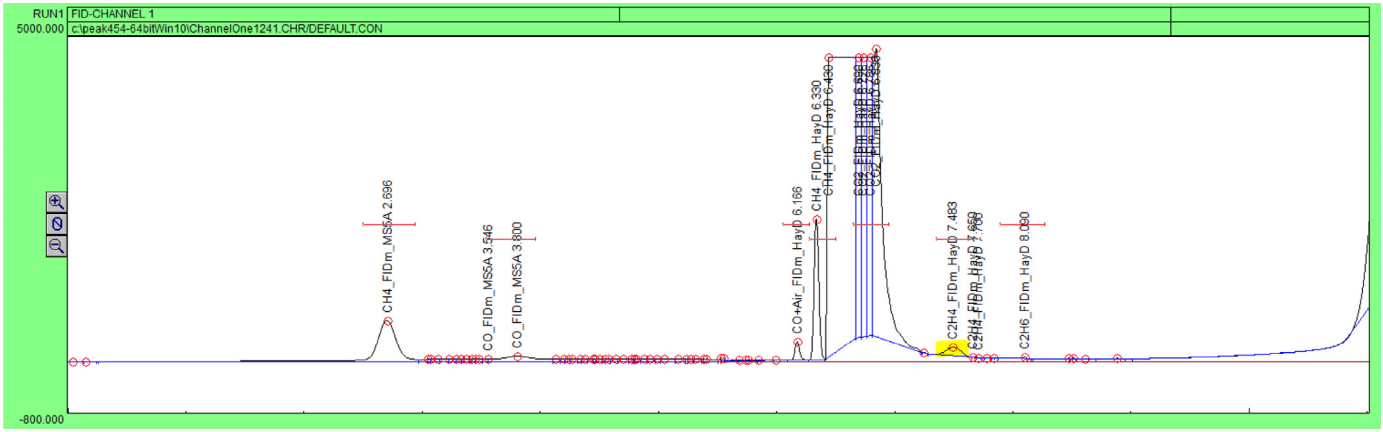


Fig. 4. Manually corrected C_2H_4 peak baseline. The black line is the raw data coming from the detector, the blue line is the baseline, and the red circles are the peaks identified by the software. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

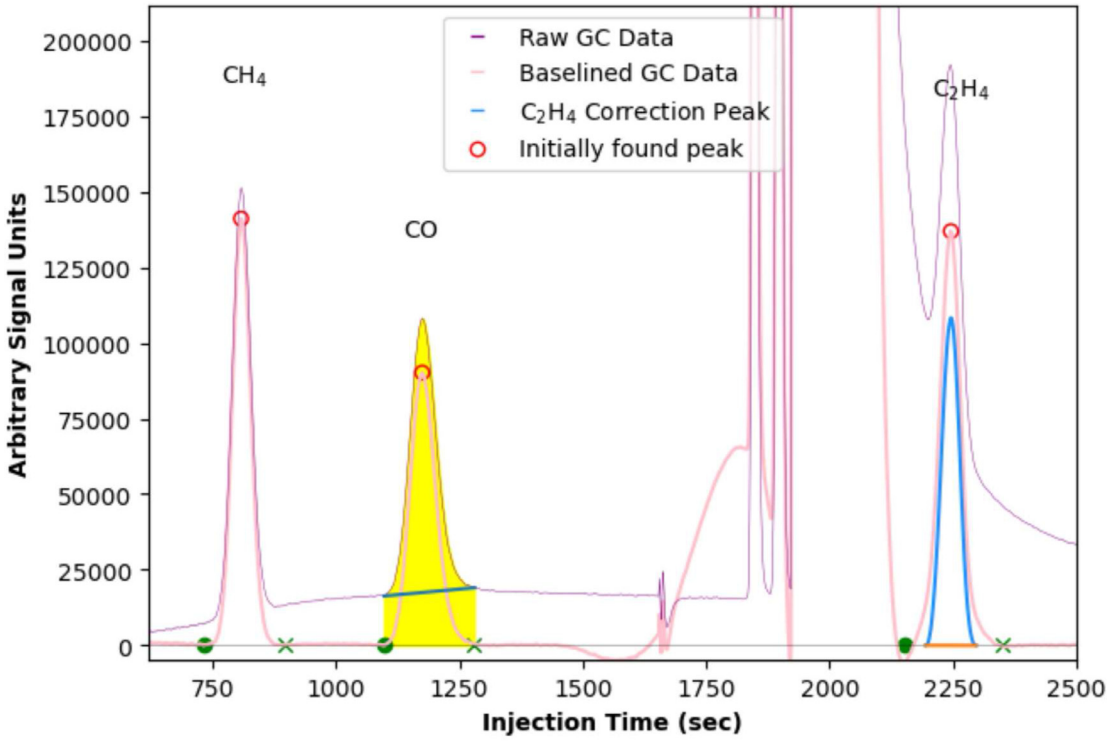


Fig. 5. Automated GC peak and area calculation example.

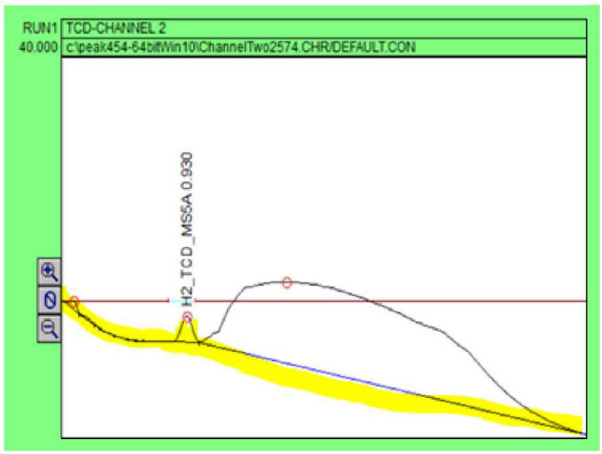
following optimization problem:

$$S = \sum_i w_i \left(y_i - z_i\right)^2 + \lambda \sum_i \left(\Delta^2 z_i\right)^2 \tag{1}$$

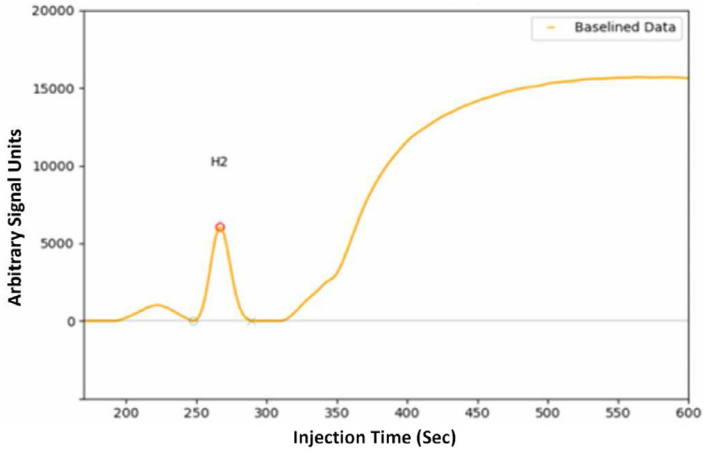
where S is the regression cost function, y_i is the signal to be baselined, z_i is the smooth baseline, and $\Delta^2 z$ is equal to $z_i - 2z_{i-1} + z_{i-2}$. The first summation describes the performance of the fit and the second summation describes the smoothness of the fit. w_i is a factor related to the asymmetry and λ is related to smoothness. In our code, the asymmetry parameter is set to a pre-determined value and λ is varied over a range to find the ideal baseline fit. When S is minimized, the corresponding z is the baseline. This is a function of raw data and λ, which is a factor that is observed to work well in the range of $[10^4, 10^9]$ (Eilers and Boelens, 2005).

The pink baseline shown in Fig. 5 was created with a fixed asymmetry parameter. Various values for λ were tested for the calibration peaks and the best performing lambda value was selected as the default value. However, this process does not always give accurate results for our peaks, especially when the sensor data is noisy. When this happens, the code isolates the peak location and tries various λ values in these particular excluded regions and selects the baseline closer to zero. This additional tuning in the methodology also contributes to the reusability of the code and more generally ensures that the most optimal baseline can be selected for the peaks. The peaks are found with a user-defined function, that checks for the increasing and decreasing sides of the time-series peaks. The peak bases are selected mathematically using the "peak-prominences" function in the Python library. A Python script determines the horizontal line marking when it intersects the sides of the peak. This is repeated at lower and lower positions on a peak. The two lowest signal values determine the base length for a peak (Scipy, 2022).

When the set of peak detection analyses are applied to the methane peak in Fig. 7, the pink baseline is replaced by the lime peak. As shown, the CH_4 raw data is inclined and could be confused for 2 peaks. The GC code initially considers the red circle to be the pinnacle of the peak.



(a) Raw data that needs manual correction shown on GC Software Interface



(b) Automated GC Results shows that data is automatically processed to find accurate peaks

Fig. 6. Automated GC and manually corrected raw data comparison from TCD channel hydrogen data.

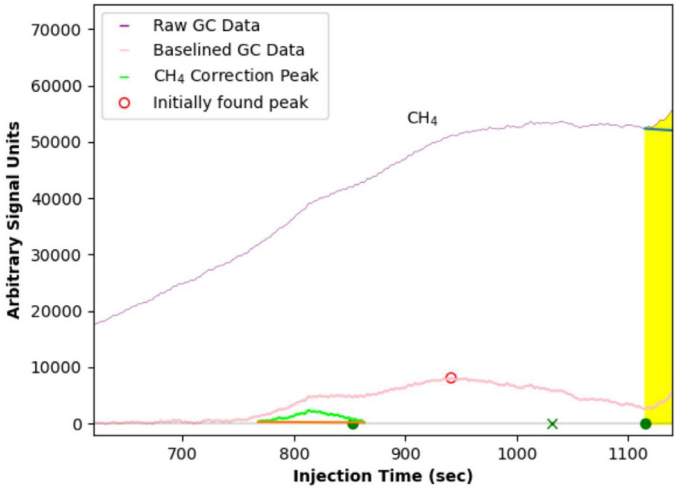


Fig. 7. Methane peak corrected by a supervisor algorithm. The first found peak is shown with the red circle, however, the corrected peak is shown in lime. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

However, the code recognizes this is incorrect. The green marks on the X axis are the initial predictions for the baseline that corresponds to the high values on the pink line. The code, therefore, understands that this peak is questionable. The GC code initiates the additional tuning process by adjusting the baseline to find the best alignment with the level of inclination. Ultimately the code determines an optimal representation of the actual peak for the CH_4 component, which is shown as the lime-colored peak.

It is notable that all remaining peaks have an inherent hierarchy (with respect to the height of each peak) despite the base code being the same. Thus, this code can be adapted for any peak of any gas product if special cases are accurately identified in the peak detection hierarchy. In principle, the peak detection hierarchy does not need to be always applied. For example, in Fig. 5, the CO area is highlighted in yellow indicating that the GC code was able to provide a more robust estimation by simply manipulating the raw data and was able to find the base of the raw data peak to calculate the peak area under the curve that is subtracted by the area under the green line. This direct and simple approach, typically also found in commercial software for GC data analysis, gives better results only for CO compared to the more complex baseline-corrected area calculation needed to also extract accurate data for other gas products. The corrected peaks are shown in lime for CH_4, yellow for CO, and blue for C_2H_4. In our system, it is especially difficult to correctly baseline the C_2H_4 peak, since it appears on a line that overlaps with a CO_2 peak, which is the reactant itself. Indeed, in the majority of cases, the more complex correction algorithm was used in the calculation of the C_2H_4 signal area.

4. The role of the CESMII smart manufacturing innovation platform

4.1. SMIP overview

As the use of data has proven to increase the profitability and efficiency of process operations, more effort has been put into developing model-based software solutions. One interesting framework is the Parametric Optimization and Control (PAROC) Platform developed by Pistikopoulos et al. (2015) for creating high-fidelity models for control and optimization purposes. Simeone et al. (2019) have built a platform service that uses an optimization and compatibility engine for sales support of metal cutting machines based on material and energy consumption options.

The Smart Manufacturing Innovation Platform is a standards-based software platform for connection, ingestion and contextualization of data to be used for building applications. The SMIP uses standardized information models and ensures the availability of contextualized data from machines and process components for broader application. It is a software infrastructure that integrates the information and operational technologies (OT) needed for building and deploying data and model applications in operations. Consistent OT and IT integration are facilitated when data are exchanged and models are shared across operations, factories, and companies (Davis et al., 2020; Edgar and Pistikopoulos, 2018). The accessibility of data and applications from various sources also creates interoperability requirements across vendor software and hardware products. The SMIP is designed for the OT and IT integration while simplifying connection, ingestion, contextualization, reusability, and availability of operating data for operational applications. It makes use of a standardized information model called the SM Profile.

For our electrochemical reactor, LabVIEW hardware is used to interface with the physical reactor and the instrumentation. LabVIEW is interfaced with the SMIP. This configuration of the operational SMIP supports three major services:

1. An edge data management device associated with the physical devices that can collect, contextualize, and transmit the data to the core SMIP services, i.e., LabVIEW.
2. The SMIP core services that receive, store, and make data available.
3. Application models that integrate or interoperate with the SMIP to consume data and support actions for improving manufacturing operations.

The Hypertext Transfer Protocol Secure (HTTPS) transfer of real-time operational data from the reactor to the SMIP core services is performed through GraphQL. GraphQL commands are similar to commands in Structured Query Languages (SQL) except that they are initiated over an HTTPS connection using a web-based Application Programming Interface (API). GraphQL can perform typical CRUD (Create, Read, Update, Delete) queries similar to a Representational State Transfer (REST) command except that the program is constrained to the transmission of necessary data due to the limit of the network bandwidth (Hartig and Pérez, 2018). Surveys have found GraphQL to be superior to REST because of its user-friendly Application Programming Interface (API) to query data (Brito and Valente (2020)). Data storage in the SMIP is provided by a PostgreSQL database in which the data are archived using the time-stamp at which the data is transmitted. Data that require no time-stamps such as model numbers are stored as configuration data in the SM Profiles, which also contain the information model configurations for equipment items in various operation services. SM Profiles can be any abstraction of importance, e.g., a machine, sensor, a Key Performance Indicator, process, etc. Configuration information in SM Profiles about measurements are stored as attributes that include data name, data type (e.g., float, int, string, etc.), and units (e.g., seconds, volts, etc.). The idea is that all users (human or machine) of the data from that same type of equipment can expect to receive the same consistently contextualized data defined by the agreed upon information model captured in the SM profile. The SM Profile is a direct extension of the Open Process Communications Unified Architecture (OPC UA) information model that specifies the interoperability standards for structured data communication among producers and consumers of data.

Since GraphQL commands for changing the data in the database can overwrite the existing data on SMIP, role-based authorization for changing data is enforced. The SMIP has an Integrated Development Environment (IDE) that accommodates Python, PHP, and SQL in developing data-driven models using the data stored in the SMIP. Additionally, the SMIP offers ready-to-use data visualization applications, such as a trend analyzer, to visualize time series data or to compute data correlations. In addition to GraphQL interfacing, the SMIP also offers custom gateway connectors that facilitate the high-speed ingestion of data into the SMIP. Currently, operational databases and historians, such as OSI PI and Wonderware are supported as well as live data sources that are OPC DA (Data Access) and OPC UA (Unified Architecture) compatible.

4.2. Using the SMIP for the CO_2 reduction reactor

The SMIP architecture used in our electrochemical reactor research is shown in Fig. 8. Sensors collect data from the reactor through a LabVIEW edge interface. The data are transmitted securely to the SMIP at 1 s intervals with the GraphQL commands. The time-stamp and the data value together ensure that data are not overwritten. The time-stamp is also essential in selecting data from particular operating time periods including within a single run. GraphQL API commands can be issued in several programming languages such as Python, JavaScript, Curl, etc. We have chosen Python for easy implementation with LabVIEW (see also Section 4.4) and because we have been using the Python IDE locally. The current version of the SMIP allows any authorized user to use the GraphQL query commands to select and download data to a local computer environment to more easily evaluate different modeling approaches as well as different data sets. Once a satisfactory modeling approach is worked out locally, the model can be implemented and run in the SMIP or run locally while interfacing with the SMIP. The choice to download the data for local use or to use the SMIP directly is dependent on the network bandwidth and the computational resources needed. Computation resources on the SMIP are currently limited with no co-processors or GPU to accelerate model building. Our project made use of the additional SMIP tools for monitoring the real-time data flow through trend charts, building process layouts of the reactor system and instrumentation, and displaying data values for system components.

4.3. Reusable profiles for electrochemical reactors

Every day, engineers are building models to improve the efficiency of manufacturing, but the lack of commonly acceptable formats is a major impediment when reusing existing models and applications, reusing contextualized data, sharing or combining data, and collaborating with other model builders. The implementation of the SMIP resolves and uses broad agreements on formats, making it possible for data producers and consumers to easily exchange information in a common format. Through the use of SM Profiles described by Davis et al. (2020), the SMIP uses a common information model for commonly used equipment based on existing agreements on standards (i.e., OPC UA, VDMA, MT Connect, etc.) that producers of data can use to deliver data to consumers to build and run data-driven models. SM profiles can be constructed and made available not only for particular equipment items but also for sensors, actuators, groupings of equipment items, and operational abstractions like KPI calculations. A research project like ours contributes to new SM Profiles that can be consumed by the SMIP users. In our research team, those responsible for the operation of the reactor acted as producers of the data and those responsible for modeling acted as consumers to test the use of the SMIP. The developers of the SMIP observed how the various features worked and were used.

SM Profiles ensure a clear understanding of the expected data structure and its content. They can be created with a Profile designer and placed in an SM Library. Since the SM Library and the SM Profiles are fully compatible with the SMIP, SM Profiles can be selected and used for specific applications. A key objective is much greater standardization and reusability of these information models for similar equipment and operational types. Once an SM Profile is built, consumers, e.g., manufacturers, researchers, etc., can use the previously used profiles by extending them as needed to more easily start data collection and modeling for another application. SM Profiles allow for reusability at different levels of description detail, e.g., equipment types, vendor types, service types, and particular applications by taking advantage of object-oriented programming concepts. The greater the match in specificity, the greater a direct profile match and the fewer extensions that need to be made. SM Profiles are therefore structured drawing upon objective-oriented programming structures for the item or process concept of interest so that extensions can be built at the level of description determined by the match between the Profile and the new application. It is also possible for a Profile to use another Profile. Profiles also do not need to be directly equipment centered. An example of this is the automated GC analysis. As a Profile, the automated GC analytics developed in this research are available for extension to other on-line GC applications. The SMIP makes sharing and implementing operational applications user-friendly and much faster. Like the GC code application in this research, if no SM Profile exists, a new Profile needs to be built, but it now becomes available for a next similar application.

The SM Profile constructed for the Electrochemical Reduction Reactor is shown in Fig. 9. The "CO_2 Reduction Reactor" is the top level profile constructed with the "Gas Chromatograph," "Potentiostat," and "Modeling" as subprofiles. Operational data are collected and transmitted to the SMIP where each raw data item is stored in the context of a corresponding data expectation called an attribute in the profile. Each attribute has the relevant datatype information and the associated measurement units. Each data item is also time-stamped. For example, a

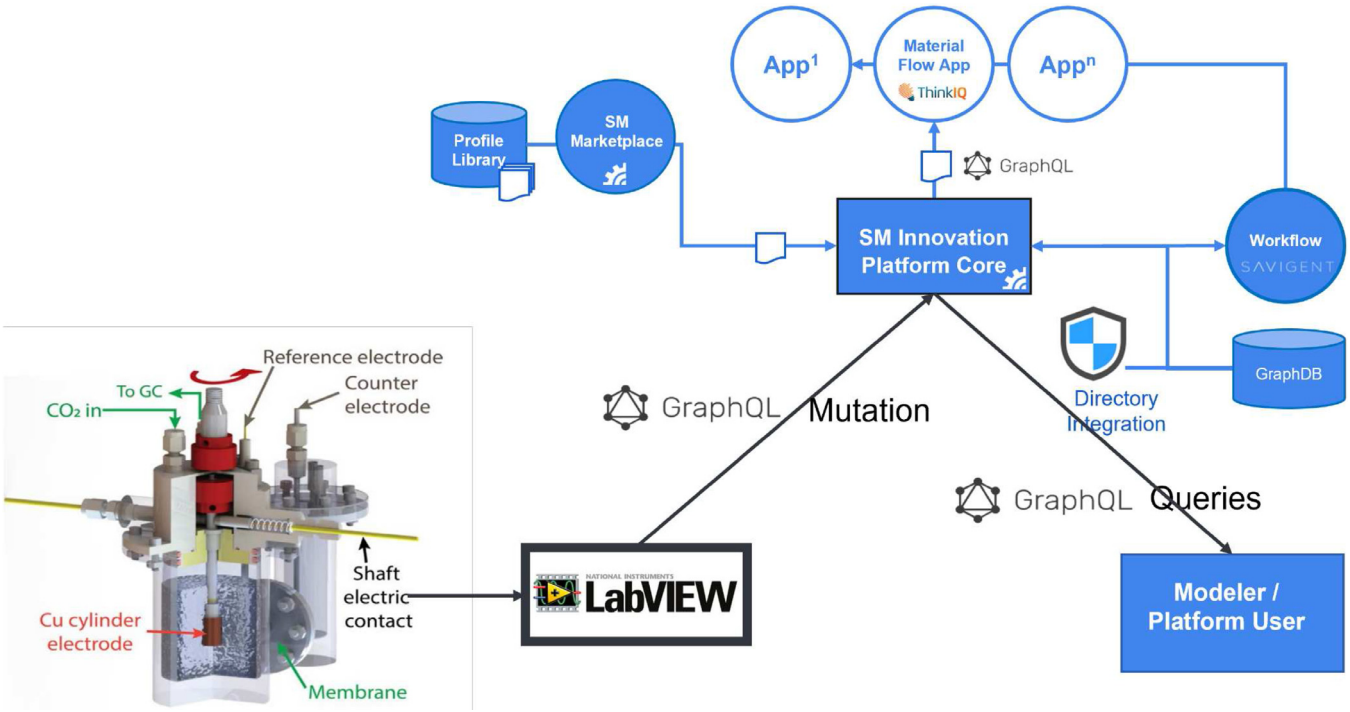


Fig. 8. SMIP architecture.

CO2 Reduction Reactor	UCLA CHE	Electrochemical reactor in Morales-Guio lab	In Work	8 4
Gas Chromatogram	UCLA CHE / CO2 Reduction Reactor	Sensor for quantifying gas products in real-time	In Work	5 1
C2H4	...CHE / CO2 Reduction Reactor / Gas Chromatogram	Ethylene gas concentration in reactor headspace	In Work	27
CH4	...CHE / CO2 Reduction Reactor / Gas Chromatogram	Methane gas concentration in reactor headspace	In Work	27
CO	...CHE / CO2 Reduction Reactor / Gas Chromatogram	Carbon monoxide gas concentration in reactor headsp...	In Work	27
H2	...CHE / CO2 Reduction Reactor / Gas Chromatogram	Hydrogen gas concentration in reactor headspace	In Work	27
Mass Flow Controllers	UCLA CHE / CO2 Reduction Reactor	Controls the mass flowrate of CO2 into the reactor	In Work	1 1
Modelling	UCLA CHE / CO2 Reduction Reactor	Relevant modelling data for steady-state and dynamic ...	In Work	9 3
Database	...CHE / CO2 Reduction Reactor / Modeling	Previously processed data used for modelling	In Work	9
Steady State Database	...CHE / CO2 Reduction Reactor / Modeling	Database for steady-state reactor operation data	In Work	14
Nuclear Magnetic Resonance	UCLA CHE / CO2 Reduction Reactor	Sesonr for quantifying liquid products offline	In Work	1
Potentiostat	UCLA CHE / CO2 Reduction Reactor	Arranges the potential given to the reactor solution	In Work	8 1

Fig. 9. Hierarchical equipment profile interface on SMIP for the electrochemical reactor.

single temperature measurement from a particular sensor at a certain point in time is collected, ingested, and stored as a number (but with expected units for that particular sensor device defined from the Profile) and time-stamped with the time the data item was collected. Every attribute in a Profile is also assigned a tag ID number automatically to define the data storage location. The tag ID and time-stamp are the two required attribute parameters which are needed to store or retrieve the data from the SMIP. Consumers of the data can use GraphQL queries to identify equipment and tag IDs for attributes and access data for any desired time interval.

With respect to reusability, if a new user sets up a similar electrochemical CO_2 reactor system like that used in this research, that user can now start with the SM Profile constructed by us. The Profile specifies the information models for system as sub-equipment, i.e., the potentiostat, the GC, and the rotation unit. If Fourier transform infrared spectroscopy (FTIR) was used instead of the GC to measure the gas products, a standards-based Profile for the FTIR would need to either be found in the SM Library and extended or developed as a first Profile. The new FTIR profile would replace the GC sub-profile while the other two sub-equipment Profiles would remain the same. Data transfer to the platform is facilitated since rebuilding information models and interfaces for two of the components is avoided. What changes is that a new tag ID is generated for the attributes associated with the new FTIR measurement equipment. Any equipment specific data, like vendor name, model number, etc., would also need to be updated.

Another example of reusability is the infrared (IR) camera information model used in Kumar et al. (2017), which was mentioned in Section 3. When the specific IR camera operating data from the particular steam methane reformer in the study are removed, what remains is the SM Profile that can be used in other IR thermo-imaging applications requiring spatial temperature measurements. This is analogous to the automated GC code mentioned above and in Section 3.1. While the GC code in this study was developed to quantify H_2, CO, CH_4, and C_2H_4 gases at ppm levels, this code can be reused when, for example, specific C_2H_4 inputs are replaced with other gas products and there is a need to quantify the peaks. Reusability also extends into the IT function. The GC code processes input data in an ASCII format, making it easy to change to other GC instrumentation data file formats. Similarly, the automated GC code triggers PeakSimple software to start the GC run, which can be adjusted for other GC vendor software.

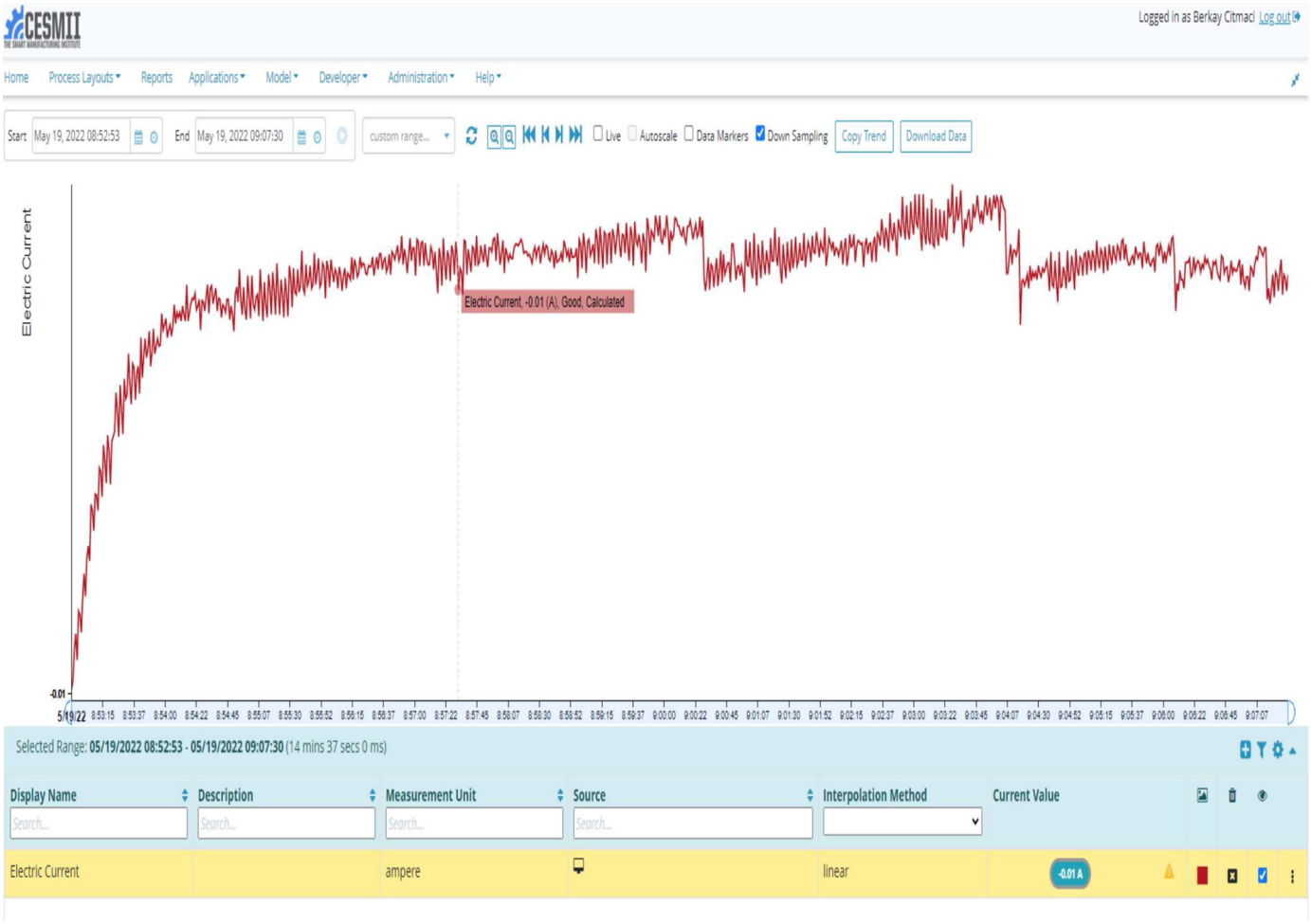


Fig. 10. Trend visualization tool on SMIP demonstrated on real-time electric current data.

Another advantage of SM Profiles and the SMIP is that they offer easy real-time/offline monitoring of the process data with process report and trend analysis. For example, one of the key parameters in the electrochemical reactor operation is the real-time current to the electrode. Fig. 10 displays this with the trend analyzer. It is possible to select multiple attributes associated with an equipment for display at the same time on the same trend analyzer interface.

4.4. Data management

As information technologies are increasingly used for data and modeling applications, the importance of cloud-based cyberinfrastructure, resources, and services has increased. The SMIP is currently hosted as a set of cloud services on the Microsoft Azure cloud platform and uses the PostgreSQL database on Azure to store the data. The SMIP's cloud-hosted database takes advantage of the cybersecurity tools provided by the Azure platform. Additionally, all data transactions are encrypted. It is easy to change (create, update, delete) or query (read) the data in real-time using GraphQL. Sensor data are transferred through automated GraphQL commands from a local machine (edge device) to the SMIP. Data are stored according to the attributes in SM Profiles as described previously. To provide a specific example, the potentiostat has an SM Profile (see Fig. 9) with several measurement attributes of which one is "electric current". Data from the amperage sensor on the potentiostat are collected and transmitted via LabVIEW and stored in the SMIP PostgreSQL database as floating point numbers assigned the Tag ID associated with the "current" attribute and the time-stamp when the data were collected. At any time, the database can be accessed using this Tag ID and the time-stamp can be used to locate data that are being or have been streamed. This data storage structure is more useful than, for example, a sequence of numbers. For this sort of real-time data transfer, a time-stamp reflects the point in time that a data item is collected, which also corresponds to the time at which the data was observed. Authentication for adding data to the database (which is interpreted as changing the data) relies on required security tokens obtained from the SMIP using login credentials.

Traditional databases typically require users to generate data containers or tables before starting collection. Additionally, these data tables need to be configured and connected to specified servers. For example, there would be the need to compose a database schema that defines how the data items are to be structured, and how they relate to one another. This upfront structuring also includes the security keys to store and retrieve the data. For these "structured" data tables to be queried by an external application such as Python, an administrator must make the relevant table settings available. Although this approach has been shown to be useful in many industrial applications, it requires domain specific knowledge to predict the table configurations. The use of the SMIP eliminates the need for this configuration process because the data are not stored in tables, but are stored only with time-stamps and Tag IDs identified with the SM Profile. Data contributors only need to have the SMIP endpoint URL and the security credentials to store data associated with previously created attributes and their Tag IDs. Consumers of the data also just need to have the SMIP endpoint URL and security credentials to download the data using GraphQL queries. Another advantage is that it is easy to store attribute data (i.e., measurements) that have different sampling periods. For example, the electric current

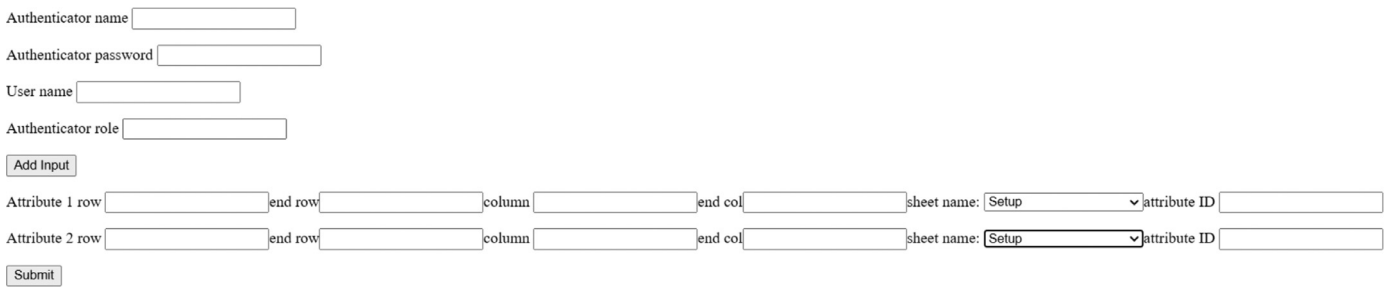


Fig. 11. The interface of data upload tool. The user needs to enter SMIP credentials, the path to the spreadsheet file that will be uploaded to the platform, and the columns/rows within the spreadsheet to upload to the platform.

from the potentiostat is recorded on a per second basis, while the GC measurements are recorded at every 20 min. To aid the user who has no knowledge of Python or GraphQL, we have developed a script and a Django-based web interface to select the relevant columns from data tables, i.e., data captured in a spreadsheet, and transmit them to the platform. The interface is shown in Fig. 11. This tool has been particularly useful in our project for uploading earlier (legacy) experimental data stored in an Excel spreadsheet.

4.5. Process equipment and data connectivity

The ability to display all trending and point data measurements collected on an operation, regardless of source or vendor, is a first step in Smart Manufacturing. This task can be difficult when different sensors and local machines use different connection protocols. This often happens when different vendor products are used. The SMIP addresses this by offering a wide variety of connectors that connect the SMIP to equipment, instrumentation, sensors, and a full range of control and management hardware and software for transferring data via GraphQL. Connectors are like translators, enabling digital communication between, for example, a sensor and the platform. For our experimental reactor, LabVIEW was already in use as the digital environment for digitizing data and displaying results. The SMIP did not have a LabVIEW driver, but LabVIEW had numerous drivers that are compatible with commonly used experimental or industrial equipment. In addition, several software development kits (SDKs) were available for building novel and complex data connections to instrumentation that were compatible with LabVIEW. Lastly, LabVIEW can communicate with external programming scripts such as those written in Python or Matlab. Our decision was to continue with LabVIEW and make use of the software development kits to build a LabVIEW connector to the SMIP. There are plans to add this LabVIEW connector into the SMIP's connector list so it is available to the many experimentalists already using LabVIEW.

In our project, the potentiostat, a Metrohm Autolab Model 302N, was connected to LabVIEW using the Autolab Software Development Kit 1.10 (Autolab, 2013). The Development Kit made it possible to build an interface connection that we could manage instead of using the more restrictive Autolab NOVA software provided by the vendor. Our interface made it possible not only to collect data from the potentiostat but to also change the input for controlling the experiment making it possible to close the control loop. Since the gas chromatograph (GC) did not have a LabVIEW driver, we developed a Python script that automatically triggered the GC measurements at predefined times by opening the PeakSimple software and initiating a run. When the GC run was completed, another script transmitted the raw data to the SMIP to be processed and quantified as described previously. With the contextualized data in the SMIP, the LabVIEW interface could query the data and bring the GC data together with the other data. The LabVIEW interface was also set up to display a plot of the processed GC data and the relevant peaks. The rotation unit also did not have a LabVIEW driver but it was connected via a Compact Reconfigurable Input Output (CompactRIO) system, a National Instruments product that enables engineers to connect input/output modules without drivers. Even though the rotation speed was kept constant throughout the experiments for this paper, the CompactRio can adjust the rotation speed in real time should we decide to use it. Finally, the mass-flow controllers (MFCs) are connected to the LabVIEW interface via a function provided by the vendor for sending priority commands specified in the user manuals. This made it possible to set an MFC to a specific flowrate and either hold or change it. An example portion of the LabVIEW interface is shown in Fig. 12.

The LabVIEW interface can both send control signals and acquire real-time data from the potentiostat and the gas chromatograph, and it can communicate with the platform for real-time data storage and query through GraphQL. LabVIEW has a feedback control feature which is used for the control of the gas-phase ethylene product concentration by manipulating the applied potential. A real-time change of applied potential is made by the potentiostat based on the feedback value calculated by an estimator-based proportional-integral (PI) feedback controller (other control methods can also be used in this framework as discussed in Remark 3 below) on LabVIEW.

For LabVIEW to be able to send data to the SMIP, the LabVIEW script shown in Fig. 13 was developed. The script on the right side is written in JSON to query https://uc.cesmii.net/graphql, the University of California SMIP domain. To execute a query, the JSON script needs the time-stamp, Tag ID, start, and end times. Time-stamps are obtained from LabVIEW in real time and the middle functions convert the time-stamps to the required string format. The LabVIEW code, shown in Fig. 13, uses the "ReplaceTimeSeriesRange" command to write data to the SMIP database through GraphQL. This command replaces the data assigned to the time-stamp within a time interval with a specific start time and end time. If there is no existing data in this range, new data is written to the database without any replacement. The time interval for replacement must contain the assigned time-stamp. In our example code, this time range is defined between the time-stamp (now) and a distant future end time. This way, the start time is always renewed at each request to change the data and the end time is kept the same to prevent any timing conflict. "HTTPS client nodes" are used from the Data Communication - Protocols section in the LabVIEW functions palette to specify the destination link. This is similar to the Python "Requests" library. User authentication into LabVIEW is required before the experiment starts. The "open handle" function defines the SMIP username and password. This script then sends data to the SMIP on a per second basis.

Remark 2. The optimizer uses Interior Point OPTimizer (IPOPT) which is an open-source software package for nonlinear optimization provided by the COIN-OR Foundation (Wächter, 2009; Wächter and Biegler, 2003; 2005). A Python library, Pyipopt, developed by Eric Xu is used to connect python scripts to IPOPT. Instructions for installing IPOPT and PyIPOPT can be found on the official website of IPOPT and in the following Github page (https://github.com/xuy/pyipopt), respectively.

Remark 3. The electrochemical reactor in this study produces multiple products and has multiple process inputs. Thus, it is possible to control

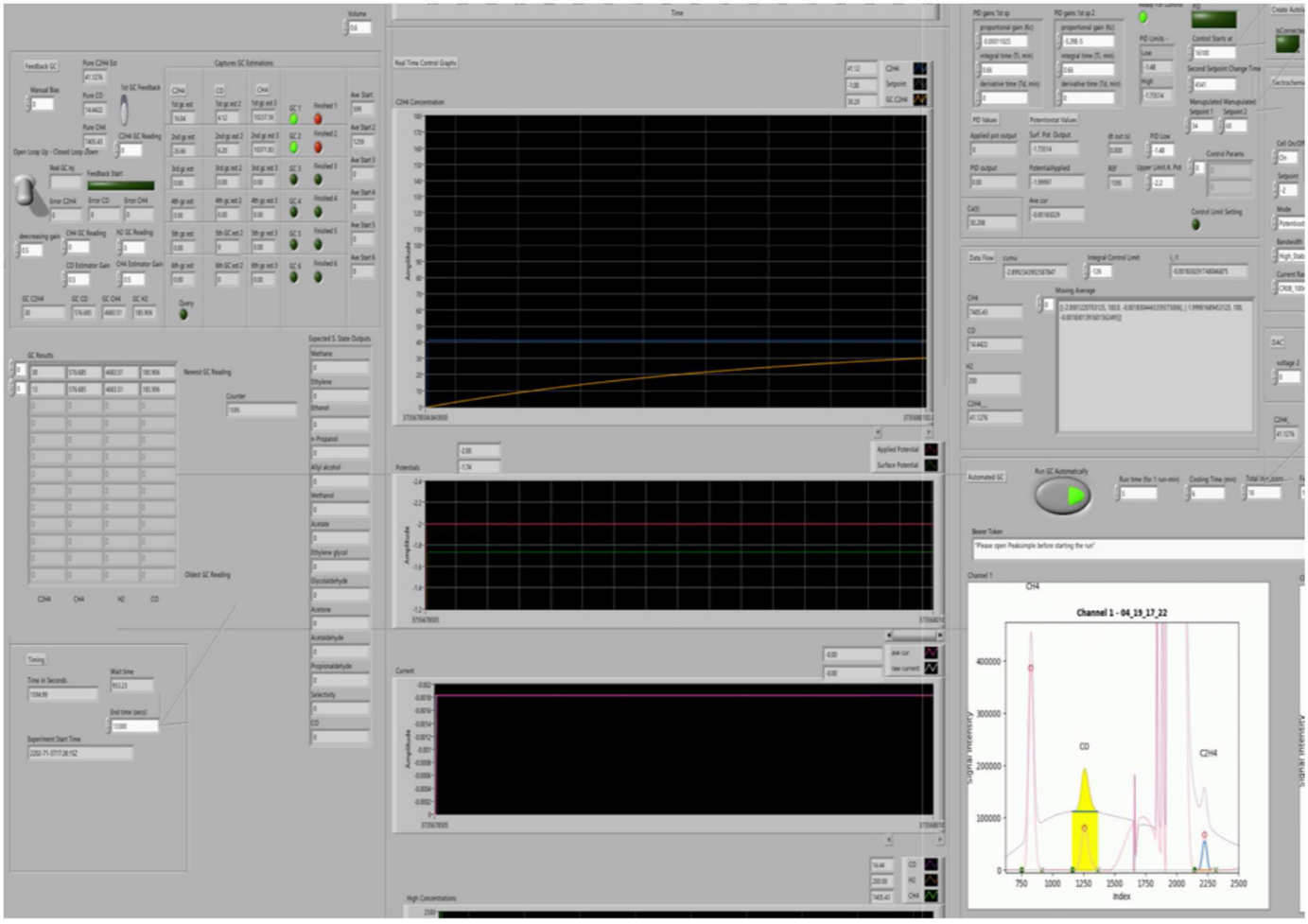


Fig. 12. LabVIEW interface and representative real-time data plots.

this reactor with multivariable control methods, such as MPC. LabVIEW has a "Predictive Control" function palette, which can be used to implement an MPC scheme. Since we use ML-based models and the open-source optimizer IPOPT as the optimizer, the calculation of the control action by the MPC can be done using a Python script within a Docker container (see below, a self-contained executable script). This script can get the data from the platform, make calculations, and send the output signals to the actuators in LabVIEW through the SMIP.

5. Virtualization

We have been emphasizing the advantages of the SMIP, but there are two limitations in its current version. First, the SMIP is not set up to provide enough computational power for large data computations (e.g., training ML models and/or solving complicated optimization problems) particularly those arising with real-time model predictive control with nonlinear models. We needed to download the data from the SMIP and do the computations outside of the SMIP and then return the results to it. Secondly, the client interface to the SMIP needs to be simplified and there needs to be the ability to more easily reuse models and software functionality. This section introduces the application of Docker technology to communicate with the SMIP and address these limitations.

5.1. Docker overview

Docker technology is an open-source application for virtualizing an executable image with all the run-time library, tools, and codes in a container that can be quickly deployed on multiple operating systems. The Docker application is composed by four main components: Docker Client and Server, Docker Images, Docker Registries, and Docker Containers (Rad et al., 2017). When using Docker, users give command lines to the Docker Client, which then converts those commands into a request form and sends it to the Docker Server. The Docker Server can be understood as the background script that is running behind the screen. Usually, the Docker Client and Server are installed on the same machine, but they can also be installed separately. Specifically, a Docker image is a read-only file that contains the OS (e.g., Ubuntu for Linux-based application), libraries (e.g., TensorFlow, Numpy, Pandas for machine learning programs) and tools (e.g., Jupyter Notebook), which can be shared by different containers. Containers can be understood as writable layers built on top of images so that users can make changes and run applications similarly to developing a new program or generating data by running applications built in the images. In short, by using Docker, users can create and run isolated applications with various virtual operating systems on the same machine.

5.2. Docker-SMIP synchronization

Docker makes sense as a way to interface applications with the SMIP because it is analogous to interfacing to an external computer that is programmed accordingly. Docker offers advantages because the execution images are lightweight files that can be easily packed and distributed through standard uploading and downloading processes. Docker containers were therefore ideal for the generalized tools developed in this

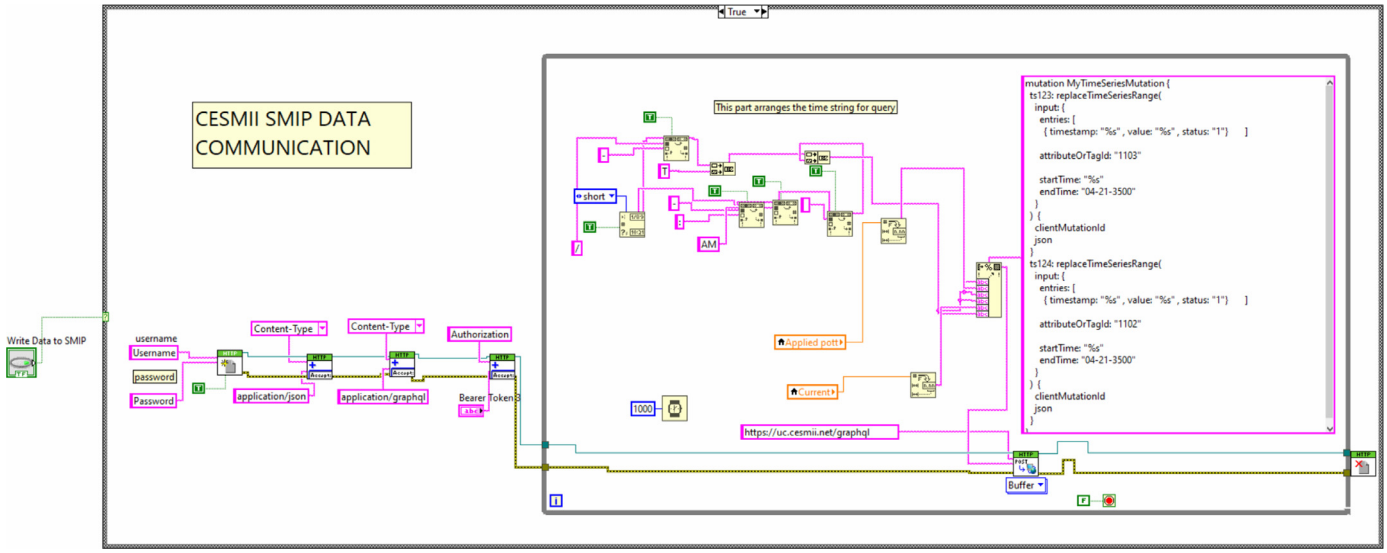


Fig. 13. Data query script on LabVIEW.

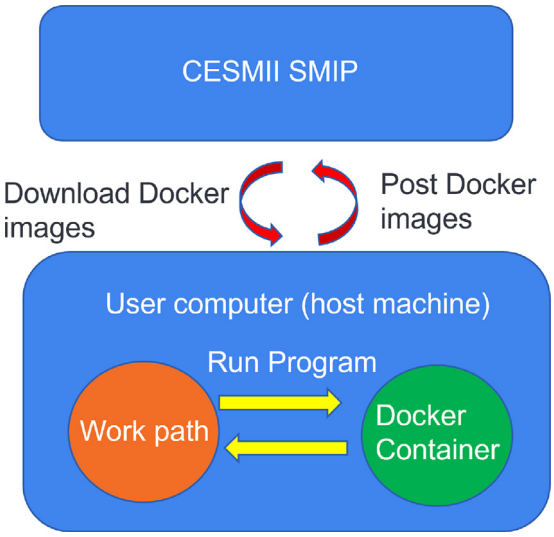


Fig. 14. Docker-SMIP synchronization.

research that required external processing power, while still interfacing with the SMIP. Docker images can be stored on the SMIP, easily downloaded to a local machine and used. When on the local machine, the tools and programs can be modified and saved as Docker images on the SMIP. In this way, Docker application images are to modeling applications as Profiles are to information models from a reusability standpoint. The SMIP, Profiles and Docker applications are SMIP and execution compatible. When executed, the Docker applications whether they are run on the SMIP or an external processor, still use the data on the SMIP - data that are defined by the Profiles. This workflow is illustrated in Fig. 14. In this work, we built Docker application images to simplify and automate the ability to upload existing data from other files and to run a computationally intensive optimizer that used a neural network machine learning model to process experimental data. We note that, in addition to creating tools for the SMIP, Docker can also be used for other management tasks. For example, code and program dependencies can be packed by a researcher into Docker images, which can then be used to reproduce their working environment effectively. Docker images can be easily shared between team members and can be used for version management.

6. Electrochemical reactor modeling using the SMIP

Dynamic models, whether first-principles or data-centered, are central to process control applications. First-principles models are useful in providing insights and describing process behaviors (Yang et al., 2020). However, there are situations where it is difficult to develop first-principles models with sufficient completeness, fidelity, or range for an objective. Examples include phenomena that are difficult to model, such as wear, deterioration, or deactivation, complexities that may not be fully known such as reaction kinetics (like, for example, in this study), or fidelity requirements involving more computational power and/or time to be useful. Data-centered modeling can be a feasible alternative if enough data are available over the range of interest since data-centered models do not generalize well. There are advantages to hybrid models in which data-centered and first-principles modeling are integrated into a hybrid approach like in the early work of Doyle et al. (2003) on particle size distribution modeling in polymerization.

The present research embraced a similar hybrid modeling approach for the electrochemical process because the reduction of CO_2 on a flat copper catalyst is subject to unpredictable variations in catalyst activity and deactivation. Reaction rates were modeled with a polynomial kernel support vector regression (SVR) model using GC data from many runs. For our real-time control objective, this reaction rate model was inserted into a first-principles, dynamic gas-phase concentration model to predict the time-evolution of the product concentrations (Çıtmacı et al., 2022). The SVR model takes surface potential, rotation speed, current, and the cumulative integral of the current as inputs. These inputs are transformed into polynomial powers and the corresponding outputs are normalized between 0 and 1, depending on the minimum and maximum values in the training data sets for each input. Because of catalyst activity uncertainties across runs, standard deviations of the data were used to build the data-based models by integrating most-likelihood estimation methods into an artificial neural network for production rate estimation. This model was then used to calculate the most energy-efficient set-points with IPOPT (Çıtmacı et al., 2022; Luo et al., 2022). The SMIP greatly facilitated the data transmission and management of data across many runs for this kind of control and optimization modeling development effort. It was significant to have flexible and ready access to operating data from the beginning of the development effort to facilitate model evaluation and development with easy data selection and the management of training and testing sets. It cannot be understated how access to operating data facilitated the synergistic process of developing further insights and the understanding of the reaction

that went into building a better model of the physical phenomena and process.

Remark 4. Process and measurement noise is an important aspect that is addressed best with operational data that reflects the noise experienced. A particularly important source of noise in our experimental setup occurred because the current to the cylindrical electrode continuously fluctuated within a small range due to the adhesion of gas bubbles on the surface of the electrode. The rotation of the electrode created further complications with fluctuations in the mixing of the liquid that not only affected the gas phase at the surface of the electrode but also the mass transfer at the gas-liquid interface. This noise in the current and surface potential calculations (since the surface potential is a function of the current) was addressed using the average of every two consecutive current values as input to the ML model. Additionally, the output from the SVR model was input into the gas-phase dynamic mass balance (i.e., the output of the SVR model was entered the right-hand-side of an ordinary differential equation) further reducing the impact of the noise on the gas-phase ethylene concentration estimation (the dependent variable of the differential equation).

Remark 5. The overall model development effort was conducted offline with open-loop experimental data collected over a year. As a potential future work, it is possible to set up an auto-ML model update cycle using the SMIP as data from new experiments are generated. This, however, requires being able to identify runs that proceeded without any abnormal situations or conditions. It also requires a mechanism to determine if the new data are appropriate for updating the model. Finally, there will need to be a methodology to separating data in training, testing and validation data sets. We can see that the SMIP provides the infrastructure and services needed to develop an automated ML-model update cycle.

7. Conclusion

This paper addresses our experience with the digitalization of the UCLA experimental electrochemical reactor using the Smart Manufacturing Innovation Platform (SMIP). Smart Manufacturing fundamentals and concepts were explained and relevant examples to demonstrate the value were presented. Specifically, Smart Manufacturing building block tools and infrastructure were applied to exploit operational data from the experimental electrochemical reactor to overcome a lack of fundamental understanding and to demonstrate that the reactor could be controlled. The development of a hybrid, first-principles and data-centered model, leveraged the operational data. It was in turn used to develop and demonstrate process optimization and real-time estimation-based control. A key Smart Manufacturing development was the automation of gas chromatography composition measurements that could be used as on-line, real-time measurements for feedback control. In addition to addressing operational technology modeling and application requirements, the paper also explained the integrated IT capabilities including connectivity, data transfer, use of SM Profiles, and the use of Docker containers. The value of integrated Operational and Information technologies is uniquely explained with examples. The OT and IT features offered by the SMIP significantly accelerated data acquisition and analysis, as well as machine learning modeling efforts, while keeping the proprietary data safe. The access to and use of operating data at the onset of the effort facilitated the understanding of the reaction kinetics and the phenomenological and operational noise while building the models. This resulted in a better model for control and a robust demonstration of what is needed to control an electrochemical reactor. Lastly, the project demonstrated reusability of data and models using Smart Manufacturing concepts of Profiles and Docker Application Containers. The overall approach implemented on the UCLA electrochemical reactor is applicable to other experimental and industrial reactors as well as other unit operation processes. Once the sensor, actuator and reactor Profiles are developed, the automation, connectivity, and contextualization can be adopted by other experimental groups working on other processes with minimal training and effort. The automated modeling tools are available as Docker Containers.

Disclaimer

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.

Acknowledgment

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office Award Number DE-EE0007613.

References

Autolab, M., 2013. Autolab SDK User Manual. Methrom Autolab M.B..
Botcha, B., Wang, Z., Rajan, S., Natarajan, G., Bukkapatnam, S., Mananthanwar, A., Scott, M., Schneider, D., Korambath, P., 2018. Implementing the transformation of discrete manufacturing systems into smart manufacturing platforms. In: Proceedings of the ASME International Manufacuturing Science and Engineering Conference, vol. 51371, V003T02A009.
Brito, G., Valente, M.T., 2020. REST vs. GraphQL: a controlled experiment. In: Proceedings of IEEE International Conference on Software Architecture, pp. 81–91.
Christofides, P.D., Davis, J.F., El-Farra, N.H., Clark, D., Harris, K.R.D., Gipson, J.N., 2007. Smart plant operations: vision, progress and challenges. AIChE J. 53, 2734–2741.
Çıtmacı, B., Luo, J., Jang, J.B., Canuso, V., Richard, D., Ren, Y.M., Morales-Guio, C.G., Christofides, P.D., 2022. Machine learning-based ethylene concentration estimation, real-time optimization and feedback control of an experimental electrochemical reactor. Chem. Eng. Res. Des. 185, 87–107.
Davis, J., Malkani, H., Dyck, J., Korambath, P., Wise, J., 2020. Chapter 4—Cyberinfrastructure for the democratization of smart manufacturing. In: Soroush, M., Baldea, M., Edgar, T.F. (Eds.), Smart Manufacturing: Concepts and Methods. Elsevier, pp. 83–116.
Doyle, F.J., Harrison, C.A., Crowley, T.J., 2003. Hybrid model-based approach to batch-to–batch control of particle size distribution in emulsion polymerization. Comput. Chem. Eng. 27 (8), 1153–1163.
Edgar, T.F., Pistikopoulos, E.N., 2018. Smart manufacturing and energy systems. Comput. Chem. Eng. 114, 130–144.
Eilers, P.H., Boelens, H.F., 2005. Baseline correction with asymmetric least squares smoothing. Leiden Univ. Med. Centre Rep. 1, 1–24.
Elsayed, K., Lacor, C., 2012. Modeling and pareto optimization of gas cyclone separator performance using RBF type artificial neural networks and genetic algorithms. Powder Technol. 217, 84–99.
Grambow, C.A., Pattanaik, L., Green, W.H., 2020. Deep learning of activation energies. J. Phys. Chem. Lett. 11 (8), 2992–2997.
Haas, T., Schubert, C., Eickhoff, M., Pfeifer, H., 2020. BubCNN: bubble detection using faster RCNN and shape regression network. Chem. Eng. Sci. 216, 115467.
Hartig, O., Pérez, J., 2018. Semantics and complexity of GraphQL. In: Proceedings of the World Wide Web Conference. International World Wide Web Conferences Steering Committee, Lyon, France, pp. 1155–1164.
Kang, H.S., Lee, J.Y., Choi, S., Kim, H., Park, J.H., Son, J.Y., Kim, B.H., Noh, S.D., 2016. Smart manufacturing: past research, present findings, and future directions. Int. J. Precis. Eng. Manufacturing-Green Technol. 3, 111–128.
Komp, E., Valleau, S., 2020. Machine learning quantum reaction rate constants. J. Phys. Chem. A 124 (41), 8607–8613.
Korambath, P., Wang, J., Kumar, A., Davis, J., Graybill, R., Schott, B., Baldea, M., 2016. A smart manufacturing use case: furnace temperature balancing in steam methane reforming process via Kepler workflows. Procedia Comput. Sci. 80, 680–689.

Kumar, A., Baldea, M., Edgar, T.F., 2017. A physics-based model for industrial steam-methane reformer optimization with non-uniform temperature field. Comput. Chem. Eng. 105, 224–236.
Luo, J., Canuso, V., Jang, J.B., Wu, Z., Morales-Guio, C.G., Christofides, P.D., 2022. Machine learning-based operational modeling of an electrochemical reactor: handling data variability and improving empirical models. Ind. Eng. Chem. Res. 61, 8399–8410.
Mistry, A., Franco, A.A., Cooper, S.J., Roberts, S.A., Viswanathan, V., 2021. How machine learning will revolutionize electrochemical sciences. ACS Energy Lett. 6 (4), 1422–1431.
Phuyal, S., Bista, D., Bista, R., 2020. Challenges, opportunities and future directions of smart manufacturing: a state of art review. Sustain. Futures 2, 100023.
Pistikopoulos, E.N., Diangelakis, N.A., Oberdieck, R., Papathanasiou, M.M., Nascu, I., Sun, M., 2015. PAROC—An integrated framework and software platform for the optimisation and advanced model-based control of process systems. Chem. Eng. Sci. 136, 115–138.
Rad, B.B., Bhatti, H.J., Ahmadi, M., 2017. An introduction to docker and analysis of its performance. Int. J. Comput. Sci. Netw. Secur. 17, 228.
Ren, Y.M., Ding, Y., Zhang, Y., Christofides, P.D., 2021. A three-level hierachical framework for additive manufacturing. Digit. Chem. Eng. 1, 100001.
Saudagar, M., Ye, M., Al-Otaibi, S., Al-Jarba, K., 2019. Smart manufacturing: hope or hype. Chem. Eng. Prog. 115 (6), 43–48.
Scipy, 2022. Scipy peak prominences API. https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.peak_prominences.html, Accessed: 2022-07-12.
Simeone, A., Caggiano, A., Boun, L., Deng, B., 2019. Intelligent cloud manufacturing platform for efficient resource sharing in smart manufacturing networks. Procedia Cirp 79, 233–238.
Wächter, A., 2009. Short tutorial: Getting started with IPOPT in 90 min. In: Dagstuhl Seminar Proceedings, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. Dagstuhl, Germany, pp. 1–17.
Wächter, A., Biegler, L.T., 2003. Line Search Filter Methods for Nonlinear Programming: Local Convergence. Technical Report. Technical Report RC23033 (W0312-090), TJ Watson Research Center, Yorktown.
Wächter, A., Biegler, L.T., 2005. Line search filter methods for nonlinear programming: motivation and global convergence. SIAM J. Optim. 16 (1), 1–31.
Wu, Z., Tran, A., Rincon, D., Christofides, P.D., 2019. Machine learning-based predictive control of nonlinear processes. Part I: theory. AlChE J. 65 (11), e16729.
Yang, S., Navarathna, P., Ghosh, S., Bequette, B.W., 2020. Hybrid modeling in the era of smart manufacturing. Comput. Chem. Eng. 140, 106874.
Zhu, L.-T., Chen, X.-Z., Ouyang, B., Yan, W.-C., Lei, H., Chen, Z., Luo, Z.-H., 2022. Review of machine learning for hydrodynamics, transport, and reactions in multiphase flows and reactors. Ind. Eng. Chem. Res. 61, 9901–9949.