A Hybrid Science-Guided Machine Learning Approach for Modeling Chemical and Polymer Processes

Niket Sharma and Y.A. Liu
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061, U.S.A.

Abstract

This chapter provides a comprehensive examination of hybrid modeling in chemical and polymer processes, employing a science-guided machine learning (SGML) approach to fuse scientific knowledge with data analytics. We introduce the concept of hybrid SGML and outline our motivation for exploring this innovative approach. A critical review of the broad applications of SGML in chemical engineering highlights the growing complexity and diversity in methodologies, making it challenging for newcomers to navigate the field. To address this, we offer a systematic classification of hybrid SGML methodologies, distinguishing between models where machine learning complements scientific understanding and vice versa. We delve into various applications of machine learning to augment science-based models, discussing direct serial and parallel hybrid modeling, inverse modeling, reduced-order modeling, and the quantification of uncertainty in process models, including the discovery of process governing equations. Each category is explored in detail, evaluating their requirements, strengths, and limitations, and suggesting potential areas of application with specific focus on polyolefin manufacturing. Similarly, we examine how scientific principles can enhance machine learning models, discussing the design, learning, and refinement processes. The study discussing the challenges and opportunities that lie ahead for the hybrid SGML approach in the modeling of chemical and polymer processes, signaling a promising direction for future research and application in this interdisciplinary field.

This is a preprint version of our chapter 11 [133] from the book - *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing. Please cite the original work if referenced* [144] and is also an extended version of the study [131].

11.1 Introduction

This chapter presents a broad perspective of hybrid modeling combining the scientific knowledge and data analytics in chemical and polymer processes with a science-guided machine learning (SGML) approach. Section 11.1 introduces the hybrid SGML approach and describes our motivation for writing this chapter. Section 11.2 gives a review of the broad applications of hybrid SGML approach in chemical engineering. As the number of reported methodologies and applications continues to rise significantly, it is hard for a person unfamiliar with the subject to identify the appropriate approach for a specific application. This leads to our key focus in Sections 11.3 to 11.5, beginning with a systematic classification and exposition of hybrid SGML methodologies in Section 11.3. We divide the approach into two major categories: ML complements science, and science complements ML. Section 11.4 explains different categories of applying ML to complement science-based models, and presents expositions of direct serial and parallel hybrid modeling and their combinations, inverse modeling, reduced-order modeling, quantifying uncertainty in the process and even discovering governing equations of the process model. We discuss their requirements, strengths and limitations, suggest potential areas of applications, and present illustrative workshops from polyolefin manufacturing. Section 11.5 focuses on different categories of applying scientific principles to complement ML models. We discuss the science-guided design, learning and refinement, together with their requirements, strengths and limitations, as

well as their potential applications and application workshops to polyolefin manufacturing. Section 11.6 present a workshop on reduced-order model for a polystyrene process using Aspen Multi-Case and Aspen AI Model Builder. Section 11.7 describes the challenges and opportunities for hybrid SGML approach for modeling chemical and polymer processes.

Modeling of many physiochemical systems requires detailed scientific knowledge of the system which is not always feasible for complex processes. We make some assumptions when modeling the system with first principles that ultimately leads to some knowledge gaps in describing the original system. Even for the systems where the scientific knowledge is sufficient to model the system and there is limited data to estimate the multiple parameters of a first-principle model. We often apply data-based models to study the systems where scientific data are available since they are more accurate in prediction. However, data-based/machine learning models are *black-box models* which can over-fit the data and also produce scientifically inconsistent results. For better accuracy, ML models also require more data which is not always feasible for many problems. Therefore, it is important to integrate science-based knowledge and data-based knowledge for an accurate and scientifically consistent prediction, which we will refer to as ***hybrid science-guided machine learning (SGML) approach.***

The most popular hybrid SGML approach that is being practiced in different fields of science is to combine a data-based ML model with a science-based first-principle model. However, there are more ways to combine scientific knowledge and data-based knowledge. In this work, we focus on both aspects of science complementing ML, and ML complementing science.

In our development of the hybrid SGML approach, we have benefited from two latest references. In their 2017 article, Karpatne et al. [1] suggest the theory-guided data science as a new paradigm for scientific discovery from data. They classify the theory-guided data science methods into different categories, such as theory-guided design of models, initialization, theory-guided refinement of data science outputs, hybrid models of theory of data science, and augmenting theory-based models using data science. In their 2020 article, Willard et al. [2] classify the integration of physics-based modeling with ML methodology according to the modeling objectives. The latter include, for example, improving the predictions beyond physical models, downscaling the complexity of physics-based models, generating data, quantifying uncertainty, and discovering governing equations of the data-based model.

The objective of this chapter is to present a review and exposition of scientific and engineering literature relating to the hybrid SGML approach, and propose a systematic classification of hybrid SGML models focusing on both science complementing ML models, and ML complementing science-based models. This work differentiates itself from several recent reviews of hybrid modeling in bioprocessing and chemical engineering through the following contributions: (1) presentation of a broader hybrid SGML methodology of integrating science-guided and data-based models, and not just the direct combinations of first-principle and ML models; (2) classification of the hybrid model applications according to their methodology and objectives, instead of their areas of applications; (3) identification of the themes and methodologies which have not been explored much in bioprocessing and chemical engineering applications, like the use of scientific knowledge to help improve the ML model architecture and learning process for more scientifically consistent solutions; and (4) illustrations of the use of these hybrid SGML methodologies applied to industrial polymer processes, such as inverse modeling and science-guided loss which have not been applied previously in such applications.

11.2 Applications of Hybrid SGML Approach in Chemical Engineering

The integration of science-based models with data-based models has appeared in various fields like fluid mechanics [3], turbulence modeling [4], quantum physics [5], climate science [6], geology [7] and biological sciences [8,132].

This study focuses on applications of hybrid SGML methodologies in bioprocessing and chemical engineering. Among the earliest applications is *the direct hybrid modeling* involving the integration of first-principle model with data-based neural networks [9]. Psichogios and Unger [10] combine a first-principle model based on prior process knowledge with a neural network, which serves as an estimator of unmeasured process parameters that are difficult to model from first principle. They apply the hybrid model to a fed-batch bioreactor, and the integrated model has better properties than the standard "black-box" neural network models. In particular, *the integrated model is able to interpolate and extrapolate much more accurately, is easier to analyze and interpret, and requires significantly fewer training examples*. Thompson and Kramer [11] later demonstrate how to integrate simple process model and first-principle equations to improve the neural network predictions of cell biomass and secondary metabolite in a fed-batch penicillin fermentation reactor when trained on sparse and noisy process data.

Agarwal [12] develops a general qualitative framework for identifying the possible ways of combining neural networks with the prior knowledge and experience embedded in the available first-principle models, and discusses *the direct hybrid modeling with series or parallel configuration* to combine the outputs of the science-based model and the ML model. Asprion, et al. [13] present the term, *grey-box modeling*, for optimization of chemical processes. They consider the case where a predictive model is missing for a process unit within a larger process flowsheet, and use measured operating data to set up hybrid models combining physical knowledge and process data. They report results of optimization using different gray-box models for process simulators applied to a cumene process. Actually, in a number of earlier studies, Bohlin and his coworkers have explored in details the concepts of gray-box identification for process control and optimization, and Bohlin has summarized the concepts, tools and applications of grey-box hybrid modeling in an excellent book [14].

Over the years, we have seen a growing number of applications of hybrid modeling in bioprocessing and chemical engineering as part of the advances in smart manufacturing [15-17].

In their 2021 paper, Sansana et al. [16] discuss mechanistic modeling, data-based modeling, hybrid modeling structures, system identification methodologies, and applications. They classify their hybrid model into parallel, series, surrogate models (which are simpler mathematical representations of more complex models and similar to reduced-order models that we discuss below), and alternate structures (which include gray-box modeling mentioned above). In the alternate structures, they refer to some applications of semi-mechanistic model structures where the best hybrid model is selected using optimization concepts. They also classify the hybrid models based on some of the chemical industry applications into analysis of model-plant mismatch [17], model transfer, feasibility analysis and predictive maintenance, apart from the previous mentioned applications like process control, monitoring and optimization.

Von Stosch et al. [18] have used the term, *hybrid semi-parametric modeling*, in their 2014 review, and have summarized applications in bioprocessing for monitoring, control, optimization, scale-up and model reduction. They emphasize that the application of hybrid semi-parametric techniques does not automatically lead to better results, but that rational knowledge integration has potential to significantly improve model-based process design and operation.

Qin and Chiang [19] review the advances in statistical machine leaning and process data analytics that can provide efficient tools in developing future hybrid models. In a latest paper, Qin et al. [20] propose a

statistical learning procedure integrating with process knowledge to handle a challenging problem of developing a predictive model for process impurity levels from more than 40 process variables in an industrial distillation system. Both studies highlight the power of statistical machine leaning for developing future hybrid process models.

A survey of the literature has shown applications of hybrid modeling in bioprocesses [21-27], chemical and oil and gas process industries [28-32], and polymer processes [33,34] for more accurate and scientifically consistent predictions. This survey has also shown many topical focuses of applications in bioprocessing and chemical engineering, including process control [35-38], design of experiments [39,40], process development and scale-up [41,42], process design [43] and optimization [13,44,45].

In a recent study, Zhou et al. [46] present *a hybrid approach for integrating material and process design* that holds much promise in process and product design. Cardillo et al. [47] demonstrate the importance of hybrid models in silico production of vaccines to accelerate the manufacturing process. Chopda et al.[23] apply integrated process analytical techniques, and modeling and control strategies to enable the continuous manufacturing of monoclonal antibodies. McBride et al. [48] classify the hybrid modeling applications in different separation processes in chemical industry, namely, distillation [49-51], crystallization [52,53], extraction [54-56], floatation [57,58], filtration [59,60] and drying [61,62]. Venkatasubramanian [63] gives an excellent exposition of the current state of development and applications of artificial intelligence in chemical engineering. The author highlights the intellectual challenges and rewards for developing the conceptual frameworks for hybrid models, mechanism-based causal explanations, domain-specific knowledge discovery engines, and analytical theories of emergence, and presents examples from optimizing material design and process operations.

In an excellent edited volume, Glassey and Stosch [64] discuss some of the key strengths of hybrid modeling in chemical processes, particularly in the prediction of scientifically consistent results beyond the experimentally tested process conditions, which is crucial for process development, scale-up, control and optimization. They also identify some challenges. For example, incorrect fundamental knowledge in a science-based model could impose bias on predictions, thus the underlying assumptions used in a model are important for analysis. Also, time and accuracy of parameter estimation is critical when deciding on a hybrid modeling strategy.

Herwing and Portner in their latest book showcase the applications of hybrid modeling in digital twins for smart biomanufacturing [65].

A recent patent by Chan et al. [66] presents Aspen Technology's approach on asset optimization using integrated modeling, optimization and artificial intelligence. In a later white paper, Beck and Munoz [67] describe Aspen Technology's current focus on hybrid modeling, combining AI and domain expertise to optimize assts. In particular, based on their application experience in in chemical industries, Aspen Tech have classified hybrid models into three categories: AI-driven, first-principle driven and reduced-order models [67]. They define *an AI-driven hybrid model* as an empirical model based on plant or experimental data and use first principles, constraints and domain knowledge to create a more accurate model. Examples of AI-driven models are inferential sensors or online equipment models. They define *a first-principle driven hybrid model* as an existing first-principle model augmented with data and AI to improve model's accuracy and predictability, which has seen many applications in bioprocessing and chemical engineering. Lastly, they define *a reduced-order model* where we use ML to create an empirical data-based model based on data from first-principle process simulation runs, augmented with constraints and domain expertise, in order to build a fit-for-purpose low-dimensional model that can run more quickly. With reduced-order models, we can extend the scale of modeling from units to the plant-wide models that can be deployed faster.

11.3 A Classification and Exposition of Hybrid SGML Models

As we have seen thus far, the majority of work in hybrid model applications in bioprocessing and chemical engineering focuses on the direct combination of science-based and data-based models. In this article, we portray a broad perspective of the combination of scientific knowledge and data analysis in bioprocessing and chemical engineering as inspired by some of the applications in physics and other areas [1,2]. We categorize these hybrid SGML applications in chemical process industry into two major categories, namely, ML compliments science and science compliments ML, together with their sub-categories based on the methodologies and objectives of hybrid modeling as illustrated in Figure 11.1. We also classify the applications in bioprocessing and chemical engineering according to our hybrid SGML approach. We present examples in several areas of SGML which have not been explored much thus far, and which have great potential for process improvement and optimization.

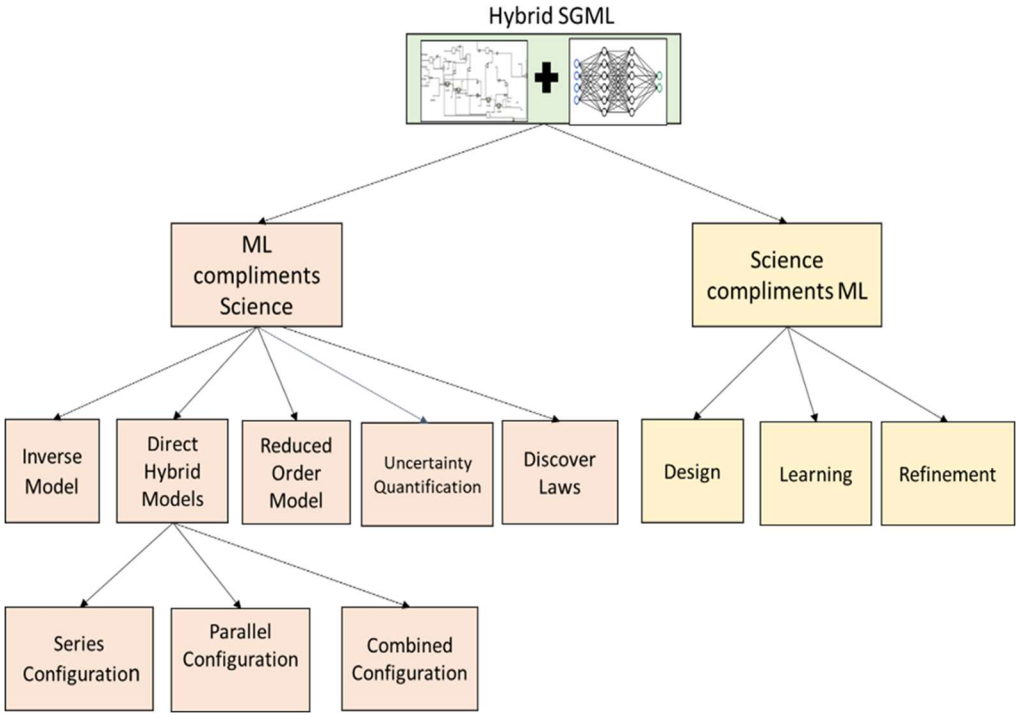

Figure 11.1 Classification of hybrid SGML models

11.4 ML Compliments Science

We can integrate a first-principle scientific model with a data-based model to improve the model accuracy and consistency. In the following, we introduce the sub-categories of direct hybrid modeling, inverse modeling approach, reducing model complexity, quantifying uncertainty in the process, and discovering governing equations.

11.4.1 Direct Hybrid Modeling

A direct hybrid model combines the output of a first-principle or science-based model with the output of a data-based ML model to improve the prediction accuracy of dependent variables. These combinations could occur in a series configuration, a parallel configuration, or a series-parallel configuration. The

direct hybrid modeling strategy is the most widely used approach in hybrid modeling in bioprocessing and chemical engineering.

11.4.1a Parallel Direct Hybrid Model

Figure 11.2 illustrates the concept of a parallel direct hybrid model. The science-based model may use the initial conditions and boundary conditions as inputs to make a prediction (Ym), while the ML model uses dynamic time-varying data to make the predictions (YML). We then combine both outputs directly or with assigned weights (w1, w2) to achieve higher prediction accuracy. We can determine the weights by least squares optimization to minimize the total sum of squares of errors for the difference between the plant and the hybrid model.

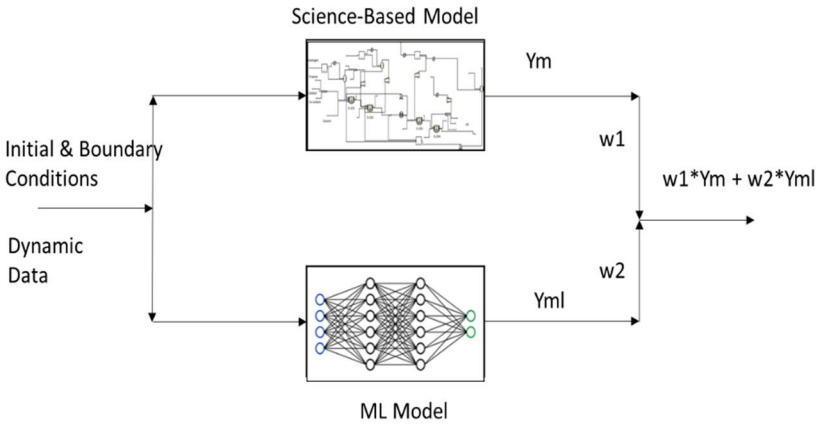

Figure 11.2. Parallel direct hybrid model: Ym and YML are model predictions, and w1 and w2 are weights.

Galvanauskas et al. [68] combine directly the data-based neural networks for kinetics and viscosity predictions with the first-principle mass balance ordinary differential equations to optimize the production rate of an industrial penicillin process. Chang et al. [33] showcase a parallel hybrid model for the dynamic simulation of a batch free-radical polymerization of methyl methacrylate. They combine an approximate rate function for the concentration of the immeasurable initiator concentration with a black-box time-dependent or recurrent neural network model [9] of the dependent variables representing the mass and moment balance equations of the polymerization reactor. They use the resulting hybrid neural network and rate function (HNNRF) model to optimize the batch polymerization system, identifying the optimal recipe or operating conditions of the batch polymerization system.

Hybrid residual modeling or *parallel direct hybrid residual model* is a class of the parallel direct hybrid model, where we use a first-principle or science-based process model to quantify the time-dependent prediction error or residual, Yres, between plant data Y(t) and science-based model prediction Ym as a function of process variables [41,69-71]. Figure 11.3 illustrates the concept of the parallel direct hybrid residual model. The correction to the model output taking care of the prediction error or residual of the ML model in the hybrid residual configuration improves the model accuracy over the non-residual configuration of Figure 11.2.

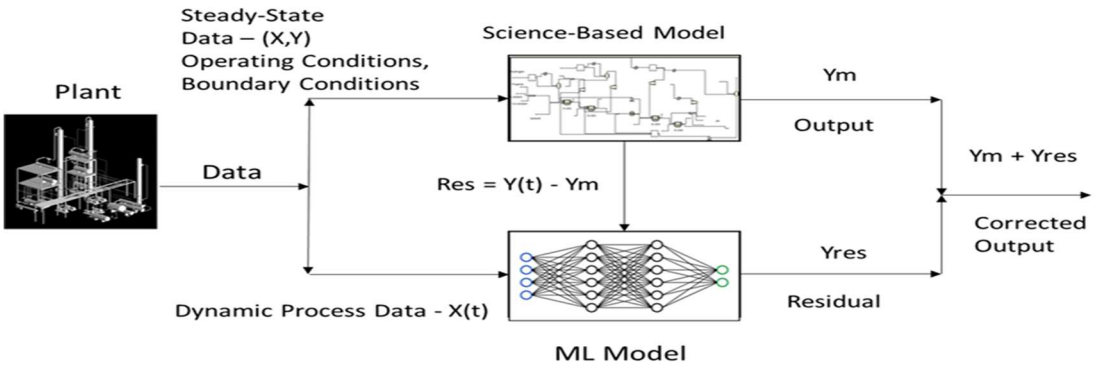

Figure 11.3. Parallel direct hybrid residual model: Ym represents model outputs, Res are the time-dependent prediction errors or residues between plant data Y(t) and science-based model outputs Ym, and Ym + Yres are the corrected model outputs

We recommend that the use of hybrid models will generally perform better than standalone ML model for applications like process development. This follows because hybrid models are better at extrapolation, while standalone ML models can be adequate for prediction in a steady running plant.

Tian et al. [69] develop a hybrid residual model for a batch polymerization reactor. First, they develop a simplified process model based on polymerization kinetics, and mass and energy balances to predict the monomer conversion, number-average molecular weight MWN, and weight-average molecular weight MWW. This first-principle process model cannot predict these product quality targets accurately because of its neglect of the gel effect at high monomer conversion and other factors. Next, the authors develop a parallel configuration of three data-based, time-dependent or recurrent neural networks [9] trained by process data to predict the residuals of monomer conversion, MWN and MWW of the simplified first-principle process model. The predicted residuals are added to the predictions from the simplified process model to form the final hybrid model predictions. Because of focus in batch process control is on the end-of-batch product quality targets, the use of time-dependent or recurrent neural networks can usually offer good long-range predictions. Therefore, the resulting hybrid residual model performs well in many batch process control and optimization applications [41,43,69-71].

Simutis and Lubnert [36] present another application of the direct hybrid modeling methodology to state estimation for bioprocess control. This work combines a first-principle state Kalman filter based on mass balances of biomass, substrate and product, and an ML-based observation model for quantifying relationship between less established variables and measurements. Recently, Ghosh et al. 72-73] apply the parallel hybrid modeling framework in process control, where they combine first-principle models with data-based model built by applying subspace identification for better prediction of batch polymer manufacturing and seed crystallization system. Hanachi et al. [74] showcase the application of direct hybrid modeling methodology for predictive maintenance. They combine a physics-based model with a data-based inferential model in an iterative parallel combination for predicting manufacturing tool wear.

11.4.1b Series Direct Hybrid Model

Figure 11.4 illustrates the series direct hybrid model. The science-based process model serves to augment the data needs of the ML model, while the ML model can help in estimating the parameters of the science-based model. Babanezhad et al. [75] consider the computational fluid dynamics (CFD) for two-phase flows in chemical reactors, and couple science-based CFD results to a ML model based on an adaptive network-based fuzzy inference system (ANFIS). Once the ML model captures the pattern of the

CFD results, they use the hybrid model for process simulation and optimization. Some features calculated from a science-based CFD model can *augment the data* as inputs to a ML model. Chan et al. [66] have discussed the advantages of data augmentation by combining simulation and plant data to generate a more accurate data-based analysis. In an application to crude distillation in petroleum refining, Mahalec and Sanchez [51] use a science-based model to calculate the internal reflux to augment other plant data as inputs to a ML model, in order to calculate the relationship to the product true boiling point curves for quality analysis. The data augmentation in series hybrid models is more relevant when some feature measurements are missing in the original data, so we use a first-principle model to calculate those features and then augment those calculated data to the ML model to study the combined multivariate effects. The goal is more towards causal effect of the added science model features and less towards improving accuracy. If we find that some missing feature measurements cause a mismatch between a science-based model and the actual plant, data augmentation may improve the training performance of the hybrid model.

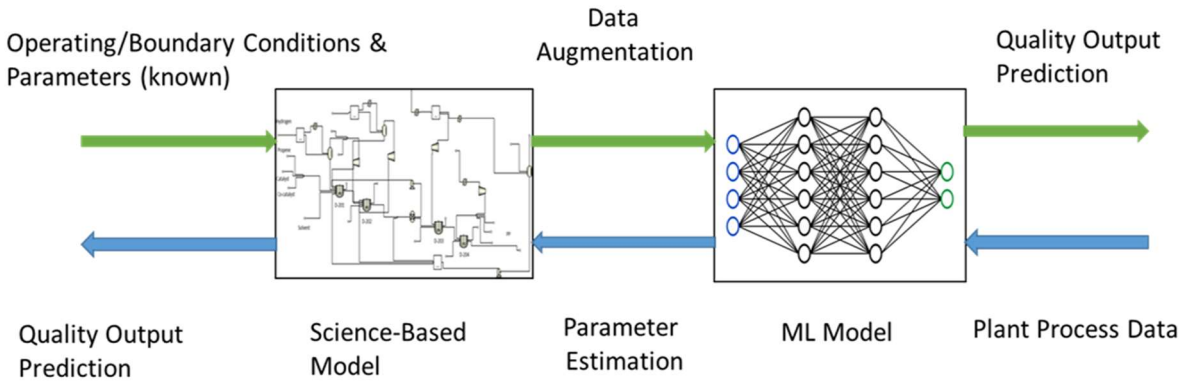

Figure 11.4. Series direct hybrid model

Krippl et al. [76] present the hybrid modeling of an ultrafiltration process where they calculate the flux using a ML model to act as an input to a science-based model. Similarly, Luo et al. [29] develop a hybrid model for a fixed-bed reactor for ethylene oxidation, integrating first-principle reaction kinetics and reactor model with a ML catalyst deactivation model. The latter is developed with support vector regression from operating data, assuming the deactivation property decreasing monotonically with time. With the hybrid model, the prediction error is less than 5% for the prediction of an industrial reactor. The approach can predict the production more accurately and have more reliable extrapolation.

Figure 11.4 shows that a ML model can also help in *estimating the parameters* of the science-based model. Mantovanelli et al. [77] develop a hybrid model for an industrial alcoholic fermentation process, combining first-principle mass and energy balance equations for a series of five fermenters with a data-based, functional link network [75] to identify the kinetic parameters of the fermentation reactors trained by plant data. The hybrid model includes the effect of temperature on the fermentation kinetics and show good nonlinear approximation capability. Sharma and Liu [78] show how to use plant data to estimate kinetic parameters of first-principle models for industrial polyolefin processes. In a recent study, Bangi and Kwong [79] estimate process parameters in hydraulic fracturing process using deep neural network which are then input to a first-principle model. Finally, we note that as illustrated in Figure 11.4, we can interchangeably use a science-based model or a ML model first in the hybrid

framework, depending on if we require to add more features to augment the dataset or to estimate model parameters.

11.4.1c Series-Parallel or Combined Direct Hybrid Model

Figure 11.5 shows a combined direct hybrid model, where we use the steady-state data from the plant to estimate the unknown parameters of a science-based process model and then uses the hybrid residual modeling strategy of Figure 11.3 for prediction. This series-parallel combination or feedback system can improve model predictions depending on the application.

Bhutani et al. [80] present a definitive study comparing first-principle, data-based and hybrid models applied to an industrial hydrocracking process. In particular, they couple a first-principle hydrocracking model based on pseudocomponents with data-based neural network models of different configurations of Figures 11.3 to 11.5 that quantify the variations in operating conditions, feed quality and catalyst deactivation. The neural network component of the hybrid model either provides updated model parameters in the first-principle process model connected in series, or correct predictions of the first-principle process models. The hybrid models are able to represent the behavior of an industrial hydrocracking unit to provide accurate and consistent predictions in the presence of process variations and changing operating scenarios.

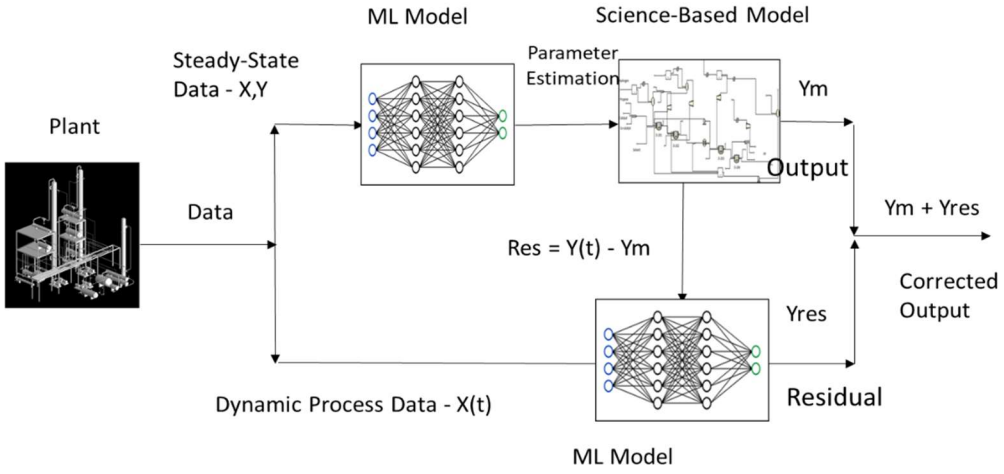

Figure 11.5 Combined Direct Hybrid Model: Ym are outputs,
Yres are residuals, and Ym+ Tres are corrected outputs

Song et al. [81] also apply the direct hybrid model configurations of Figure 11.3 to 11.5 to an industrial hydrocracking process and analyze the strengths and weaknesses of these configurations. They call a model *a mechanism-dominated model* if the accuracy of its outputs is mainly dominated by the available theoretical knowledge used to develop the model; and they also call a model *a data-dominated model* if the accuracy of its outputs is mainly dominated by the quality of the training data and the performance of the resulting data-based model. In particular, they give both the first-principle model and the series direct hybrid model of Figure 11.4 as examples of mechanism-dominated models, and cite the data-based model, parallel direct residual model of Figure 11.3, and the combined direct hybrid model of Figure 5 as examples of data-dominated models.

In their work, Song et al. [81] combine a mechanism-dominated model with a data-dominated model as a hybrid direct model of Figure 11.2, with the weighting factors for the outputs of two individual models being determined in an adaptive fashion. For their application, Song et al. work with a mechanism-

dominated model of an industrial hydrocracking process based on kinetic lumping [80,81], and with a data-dominated model based on a self-organizing map (SOM) followed by a convolutional neural network (CNN), with both being trained by simulated process data based on Aspen HYSYS [81]. They evaluate the performance of the hybrid model for operational optimization of the hydrocracking producing different product scenarios. While this study includes new conceptual development, it needs much simplification of its relatively complex methodology to make it readily applicable by data scientists and practicing engineers.

In a recent study, Chen and Lerapetritou [17] demonstrate how to use partial correlation analysis from multivariate statistics and mutual information analysis from information theory to identify and improve the plant-model mismatch in using a direct combined hybrid model for a pharmaceutical manufacturing process. As the authors state, implementing this plant-model mismatch strategy requires active excitation of variables online in order to capture the corresponding response data from the plant, which is often difficult to perform in manufacturing plants and in experimental settings, and could benefit from new development in computing and information technology.

Lima et al. [82] propose a semi-mechanistic model building framework based on selective and localized model extensions. They use a symbolic reformulation of a set of first-principle model equations in order to derive hybrid mechanistic–empirical models. The symbolic reformation permits the addition of empirical elements selectively and locally to the model. They apply the approach to the identification of a non-ideal reactor and to the optimization of the Otto–Williams benchmark reactor.

This combined strategy is generally more useful for the case where the science-based model has unknown parameters. We could use ML to determine these unknown parameters and then apply a hybrid residual ML approach. By doing so, we could improve the model prediction accuracy as well.

11.4.1d Workshop 11.1 - An Application of Combined Direct Hybrid Modeling to Polymer Manufacturing

The objective of this workshop is to predict polymer melt index using a combined direct hybrid modeling methodology to build a more accurate and scientifically consistent quality sensor.

We apply the combined direct modeling strategy to an industrial polyethylene process for the prediction of melt index. We build a first-principle steady-state model of a Mitsui slurry high-density polyethylene (HDPE) process by following the methodology and kinetic parameters presented in Supplement 5.1b of Chapter 5. For this application, it is easier to first estimate the complex multisite Ziegler-Natta polymerization kinetic parameters using steady-state production targets (Section 5.5), and then convert the resulting steady-state simulation model based on Aspen Plus to a dynamic simulation model using Aspen Plus Dynamics (Sections 7.7 and 7.8). The resulting dynamic simulation model has similar independent process variables, including the feed flow and compositions and the reactor operating conditions. For less complex applications, dynamic data could be used for parameter estimation. The following equations relate the residue (Res) or the difference between the plant and model values of the melt index ($MI_{Plant} - MI_{Model}$) as a function of independent process variables, $f(X_{Process})$. Additionally, we wish the *MI* value predicted by the hybrid model, MI_{Hybrid}, matches the plant value, MI_{Plant}:

$$MI_{Plant} - MI_{Model} = Res = f(X_{Process}) \qquad (11.1)$$

$$MI_{Hybrid} = MI_{Model} + Res \qquad (11.2)$$

We consider an industrial slurry HDPE process similar to that in Workshop 9.3, Section 9.4.2 of Chapter using actual plant data from LG Petrochemicals in South Korea. We build a dynamic model following the procedure describe in Chapters 7 and 9. We use tasks in Aspen dynamics (Section 7.5) to simulate grade change (Section 7.6) and simulate plant data similar to the industrial process. In the Aspen dynamic model (**Plant_HDPE_Hybrid.dynf**), we use the Melt_Index value in the stream R1OUT to calculate the MI model. We copy the model data to a csv file containing the plant data.

Using the csv, we calculate the difference between the MI Plant and the MI Model, and label the difference as Res MI in the column in the sheet (***Data_Hybrid2.xlsx***).

Next, we use a ML model Random Forest Regressor [83] to predict the residual as a function of input process variables. We load the data and then split the dataset into test and train subsets. We follow the details of training a ML model described in Sections 10.3.2 and 10.6 of Chapter 10. Note that when using ensemble models, normalizing the data is not mandatory. Then, we train the random forest regressor model. We predict the Res MI using the model. The predicted residual (yt) can be output in the form of the csv and then be copied to a combined data file (***hybrid_result.xlsx***).

Figure 11.6 shows a part of the code, ***Hybrid_Direct_HPDE.ipynb***, available in the supplement under ***Workshop 11.1.***

```
df = pd.read_excel('Data_Hybrid.xlsx')
df.head()

X = df.iloc[:,1:10]

Y1 = df.iloc[:,12]

#Splitting data into test-train
X_train, X_test, Y1_train, Y1_test = train_test_split(X, Y1)

#Random Forest Model

from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(n_estimators=100)

rf.fit(X_train, Y1_train)

cv = cross_val_score(rf, X_train, Y1_train, cv = 10,
                     scoring='neg_mean_squared_error')
cv_score = cv.mean()
rmse_train = np.sqrt(abs(cv_score))
print(rmse_train)
#print(cv_score)

Y_rf = rf.predict(X_test)
rmse = np.sqrt(mean_squared_error(Y1_test, Y_rf))
print(rmse)
#
yt = rf.predict(X)
```

Figure 11.6 A part of the ML Python code, ***Hybrid_Direct_HPDE.ipynb***.

We can do final calculations of the plant data using pandas data-frame. As the dataset is not very large, we just explain the csv details in the Excel file, ***Hybrid_results.xlsx***, available in the supplement for ***Workshop 11.1***.

The ML predictions are labelled as ML Predicted Res MI. The we add the ML predicted Res to the MI Model to give the Predicted Hybrid MI.

Hybrid Predicted MI = Model MI + ML Res

We can then calculate the difference b/w the 'Hybrid Predicted MI' and the 'Plant MI' calling that Hybrid Res. And calculate the final RMSE of the predictions which comes out to be 0.17 for the hybrid model compared to a RMSE of 1.74 for a standalone model MI

We can plot the model results. Figure 11.7 compares the predictions of the first-principle dynamic simulation model (in red) with the plant data with grade transitions (in green). We see much deviation between the model predictions and plant data. We compare the MI values from the model with the plant data and calculate the error residuals. *The root-mean-squares-error (RMSE) value of the model residual is to 1.7* for the actual MI data with a standard deviation of 5.1.

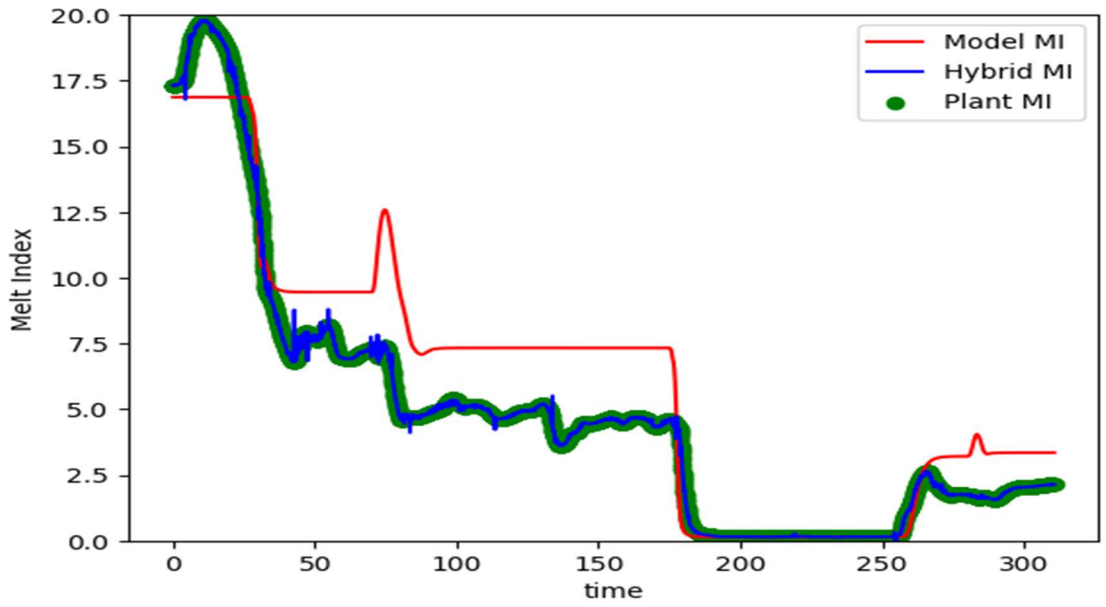

Figure 11.7 Melt index prediction of a combined direct hybrid model compared to the first-principle model and plant data

Figure 11.6 shows that the hybrid model predictions (with a RMSE value of 0.17) match the plant data much better than a first-principle dynamic simulation model alone. We note that a data-based model alone has also a similar accuracy, but it may give scientifically inconsistent results for predictions beyond process operating data which the model uses. Thus, the hybrid model is not only accurate, but also gives scientifically consistent results beyond current operating range.

11.4.2 Inverse Modeling

In *inverse modeling*, we use the output of a system to infer its corresponding input or independent variables; this is different from the *forward modeling* where we use the known independent variables to predict the output of the system [2]. Figure 11.8 illustrates the inverse modeling framework. We see that in the traditional data-based approach, we use process variable data (X) and quality target data (Y) to train and test a ML model. Because the plant does not measure most quality targets continuously, we

can apply a science-based process model, developed by first principles and validated by plant data, to predict and augment the quality target data (Y) for given process variable (X).

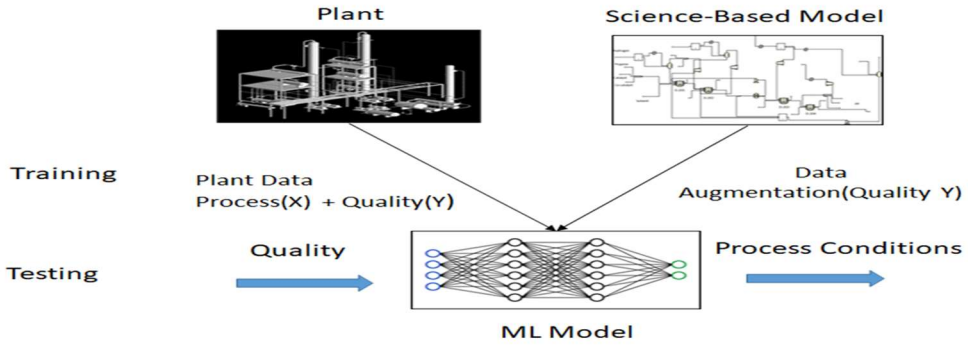

Figure 11.8 Inverse Modeling framework.

One of the earliest applications of inverse modeling for chemical process was by Savkovic-Stevanovic et al. [84]. They use a neural network controller for product composition control of a distillation plant based on the process inverse dynamic model relating the product composition to the reflux flow rate. The results illustrate the feasibility of using neural network for learning nonlinear dynamic model of the distillation column from plant input-output data. Their results also demonstrate the importance to take the time-delay of the plant into account.

Pharmaceutical product design and development typically uses the design of experiments (DOE) and response surface modeling (RSM) for steady-state process modeling, while neglecting the process dynamics and time delays. Tomba et al. [85] demonstrate how to use the inverse modeling concept to generate process understanding with dynamic process models, quantifying the impact of temporal deviations and production dynamics. Specifically, they perform data-based, latent variable regression model inversion to find the best combination of raw materials and process variables to achieve the desired quality targets. The authors propose to combine design-of-experiments studies with hybrid modeling for process characterization.

Recently, Bayer al. [86] apply the inverse modeling approach to Escherichia coli fed-batch cultivations, evaluating the impact of three critical process variables. They compare the performance of a hybrid model to a pure data-driven model and the widely adopted RSM of the process endpoints, and show the superior behavior of the hybrid model compared to the pure black-box approaches for process characterization. The inverse modeling methodology makes the decision-making process in pharmaceutical product development faster, while minimizing the number of experiments and reducing the raw material consumption.

Raccuglia et.al. [87] train the ML learning model using reaction data to predict reaction outcomes for the crystallization of templated vanadium selenites. They demonstrate the use of ML to assist material discovery using data from previously unsuccessful or failed material synthesis experiments. The resulting ML model outperforms traditional human strategies, and successfully predicts conditions for new organically templated, inorganic product formation with a success rate of nearly 90%. Significantly, they show that inverting the machine-learning model reveals new hypotheses regarding the conditions for successful product formation.

There is a growing interest in the inverse approach to material deign, in which the desired target properties are used as input to identify the atomic identity, composition and structure (ACS) that exhibit

such properties. Liao et al. [88] present a metaheuristic approach to material design that incorporates the inverse modeling framework.

Venkatasubramanian [61] also mentions the importance of inverse problem being solved by the application of artificial intelligence in chemical engineering processes.

Note the inverse modeling approach may lead to non-unique solutions which can give a range of predictions of input parameters within the operating range. By adding additional constraints to the input parameters (such as their operating range), we may obtain a unique solution.

11.4.3 Workshop 11.2 - An Application of Inverse Modeling to Polymer Manufacturing

We illustrate the application of an inverse modeling approach that integrates steady-state and dynamic simulation models of a Mitsui slurry HDPE process, developed from first principles and validated by plant data, with a data-based ML model. The goal is to predict the operating conditions for producing new polymer grades, given the desired product quality targets, such as melt index (MI), polymer density (Rho), polydispersity index (PDI) and polymer production rate (P). The details of the steady-state simulation model are available in Supplement 5.1b of Chapter 5.

We first estimate the polymerization kinetic parameters from plant production targets in a steady-state model using Aspen Polymers based on our methodology in Section 5.7. This results in a validated Aspen Polymers steady-state simulation model. Next, we convert the steady-state model to a dynamic model using Aspen Plus Dynamics following Sections 7.7 and 7.8. We use the dynamic model to simulate the product quality data for different process operating conditions, which include the data characterizing the polymer grade transitions.

Then, we use a Python-based, ensemble machine learning regression model [89] to regress the simulated data, with the simulated product quality data as input, and the process operating conditions (flow rates of all input streams) as the output. Given the desired quality targets for a new polymer grade, we apply the trained ML model to predict the operating conditions for the new polymer grade.

When loading the data in inverse modeling, we use the quality variable as the inputs (X) and the process variables as outputs (Y1). We use the stacked ensemble regression model for prediction. We use a combination of ensemble regression models using stacking technique for the prediction of the operating conditions. We used the tree regression models like the gradient boosting, ada boosting, Random forest and Xgboost regression model for the stacked regression algorithm. We choose the combination of the regression models by first individually fitting the regression models and then the regressor which performs best is chosen as the Meta regressor while the other three regressors are chosen as the Initial regressors.

Figure 11.9 shows the stacked model code ***inverse_HDPE.py***. In the figure, variable **stregr** is the stacked regression model, **Y_stregr** is the model prediction, and the RMSE for inverse modeling is **predc.**

```
df = pd.read_excel('inverse_HDPE_feature.xlsx')

X = df.iloc[:,14:18]
Y1 = df.iloc[:,1]

X_train, X_test, Y1_train, Y1_test = train_test_split(X, Y1)


from mlxtend.regressor import StackingRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import RandomForestRegressor
import xgboost as xgb

gb = GradientBoostingRegressor()
adb = AdaBoostRegressor()
xgr = xgb.XGBRegressor()
rf = RandomForestRegressor(n_estimators=100)

stregr = StackingRegressor(regressors=[adb,gb,xgr],
                           meta_regressor=rf)

cv = cross_val_score(stregr, X_train, Y1_train, cv = 5,scoring='r2')
cv_score = cv.mean()
print(cv_score)

stregr.fit(X_train,Y1_train)
Y_stegr = stregr.predict(X_test)
print (stregr.score(X_test,Y1_test))
print(np.sqrt(mean_squared_error(Y1_test, Y_stegr)))
```

Figure 11.9 ML Python code ***inverse_HDPE.py*** for Workshop 11.2

The stacked ML model predictions give a low RMSE of 0.9 when compared to actual plant data for a standard deviation of 20. We predict all process variables for the parallel HDPE process using the stacked regression model as listed in Table 11.1. The table consists of the mean and standard deviation of each of the process variables from the actual data and the RMSE and nRMSE predictions, defined in Eqs. (10.2) and (10.3).

Table 11.1 Process variable prediction for parallel HDPE process using inverse modeling

Predicted variable (kg/hr)	Data Mean (kg/hr)	Data Standard deviation (kg/hr)	RMSE (test) kg/hr	Normalized RMSE, nRMSE (%)
H2	52	21	1.04	2
C2	8873	569	68.5	0.772005
CAT	26	5.6	1.03	3.961538
HX	22356	2734	219	0.979603
C3	51	44	2.83	5.54902
T	84	0.3	0.11	0.130952
P	3.1	0.7	0.2	6.451613

H2/C2	0.95	0.4	0.01	1.052632
C3/C4	0.4	0.37	0.014	3.5

Figure 11.10 illustrates that the inverse modeling approach predicts the hydrogen feed flow rate with a high accuracy (low RMSE = 0.9) when compared to actual plant data for a standard deviation of 20. Thus, if we want to produce a new polymer grade given its quality targets, we can predict the operating conditions required to produce that polymer grade using the inverse modeling approach.

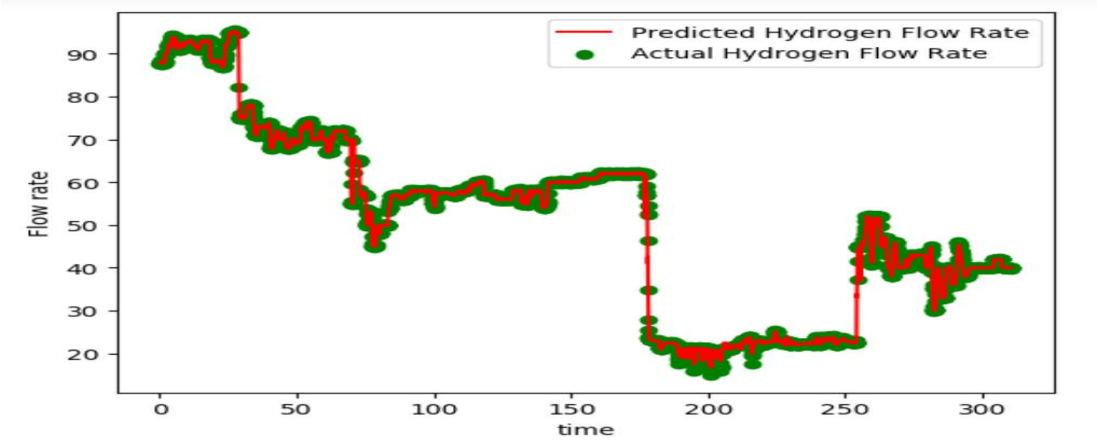

Figure 11.10 Hydrogen feed predictions from inverse modeling of product quality features

11.4.4 Reduced-Order Models

Reduced-order models (ROMs) are simplified models that represent a complex process in a computationally inexpensive manner, but also maintain high degree of accuracy of prediction in simulating the process. In bioprocessing and chemical engineering, we can apply the ROM methodology to simulate complex processes and then use ML models to optimize the processes. See Figure 9. We can use ROMs to simulate different scenarios and sensitivities in order to generate process data, which in turn can be combined with ML models to build accurate soft sensors to predict quality variables. This approach helps to make sure that the ML model is trained on process data with multiple variations which is not possible in a steady plant run. Hence, data-based sensors will be accurate for any future process optimization, scale up etc. and it is also easier to deploy such models online.

In one of the earliest applications of ROM, MacGregor et al. [90] apply a PLS ML model of a polyethylene using process data simulated from a process model to develop inferential prediction models for polymer properties. This application involves a high-pressure tabular reactor system producing low-density polyethylene, in which all the fundamental polymer properties are extremely difficult to measure and are usually unavailable, and some on-line measurements such as the temperature profile down the reactor and the solvent flow rate are available on a frequent basis. The dimensionality reduction aspects of PLS facilitates the development of a multivariate statistical control plot for monitoring the operating performance of the reactors.

Model reduction can be achieved through dimensional reduction methods like principal component analysis. Another approach is to apply the residual combination with ML model for a ROM model, or to build a ML-based surrogate model for the full- order model. Reduced-order models have been called *surrogate models* in the context of grey-box modeling techniques where first-principle models are

combined with data-based optimization techniques. Rogers and Lerapetritou [91,92] propose the use of surrogate models as reduced-order models that approximate the feasibility function for a process in order to evaluate the flexibility and operability of a science-based process model, since it is difficult to directly evaluate the feasibility due to black-box constraints.

In a recent study, Abdullah et al. [93] showcase a data-based reduced-order modeling of non-linear processes that have time-scale multiplicity to identify the slow process state variables that can be used in a dynamic model. Agarwal et al. [94] use ROM for modeling pressure swing adsorption process where they use a low-dimensional approximation of a dynamic partial differential equation model, which is more computationally efficient. In another study, Kumar et al. [45] use a reduced-order steam methane reformer model to optimize furnace temperature distribution. In a recent study, Shafer et al. [95] use a reduced-dimensional dynamic model for the optimal control of air separation unit. The model combines compartmentalization to reduce the number of differential equations with artificial neural networks to quantify the nonlinear input–output relations within compartments. This wok reduces the size of the differential equation system by 90%, while limiting the additional error in product purities to below 1 ppm compared to a full-order stage-by-stage model.

Kumari et al. [96] use data based reduced order methods for computational fluid dynamic modeling applied to a case study of super critical carbon dioxide rare event. They propose a k-nearest neighbor (kNN)-based parametric reduced-order model (PROM) for consequence estimation of rare events to enhance numerical robustness with respect to parameter change. Recently, many operator-theoretic modeling identification and model reduction approaches like the Koopman operators have been applied to integrate first-principle knowledge into finding relationship among multiple process variables in chemical processes. Koopman operator offers great utility in data-driven analysis and control of nonlinear and high-dimensional systems. Narsingham and Kwon [97] develop a new local Dynamic Mode Decomposition (DMD) method to better capture local dynamics which does temporal clustering of snapshot data using mixed integer nonlinear programming. The developed models are subsequently used to compute approximate solutions to the original high-dimensional system and to design a feedback control system of hydraulic fracturing processes for the computation of optimal pumping schedules.

Our focus on ROM is more towards using the science-based model to simulate process data that can be used by ML models to derive empirical correlations for process optimization. ROM are particularly useful in chemical processes for dynamic optimization of a complex large-scale process.

11.4.5 Workshop 11.3 - An Application of Reduced-Order Modeling to Polymer Manufacturing

The objective of this example is to illustrate the application of ROM methodology for process development of HYPOL polypropylene production.

The details of the steady-state simulation model are available in Supplement 5.1a of Chapter 5. The Hypol process is complex with series of reactors, separators and recycle loops. The process has many operating variables, such as feed flow rates of propylene, hydrogen to each reactor, and temperature and pressure in each reactor. It is critical to quantify the effects of operating variables on the polymer quality targets, particularly melt index, in order to design or optimize the process. To achieve this, we need multivariate process data which are not usually available in a steady running plant. Hence, we use the ROM methodology.

We model the HYPOL polypropylene production process following the methodology of Section 5.5, and then run multiple steady-state simulations to generate multivariate data with varying operating variables and the corresponding melt index predictions.

In the Aspen steady-state model, we use sensitivity analysis to generate process data by varying the process conditions using the Sensitivity Analysis tool. We vary the temperature, pressure of each reactor, the input feed flow rates within operating ranges to generate sensitivity data. Table 11.2 lists the process and quality variables.

Table 11.2 Process and quality variables of Hypol process

Process variable and quality target	Description
C31, C32, C33, C34	Propylene monomer flow in each of the reactors (R1, R2,R3,R4)(kg/hr)
H21, H22, H23, H24	Hydrogen flow in each of the reactors (R1,R2,R3,R4) (kg/hr)
CAT	Catalyst flow in the first reactor (kg/hr)
HX1	Solvent flow in the first reactor (kg/hr)
C24	Ethylene co-monomer flow in the 4th reactor (kg/hr)
T1, T2, T3, T4	Temperature in each of the reactors (R1, R2, R3, R4) C
P1, P2, P3, P4	Pressure in each of the reactors (R1, R2, R3, R4) Bar
MI (quality variable)	Melt Index (quality variable)

We compile the data in a spreadsheet (***ROM_data.xlsx***) and then use them to fit a random forest ML model [89] model using the same procedure as described in previous examples and chapter 10.

We use a random forest ML model to train the simulated data to predict the melt index as a function of the process variables and also understand the causality of important features affecting the polymer quality. The empirical ML model can serve as an approximate quality sensor. We can use it to predict the melt index at varying process variables. See Figure 11.11a.

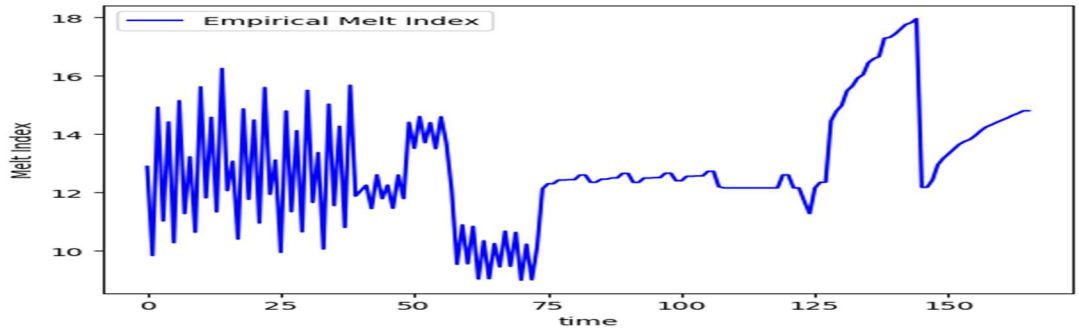

Figure 11.11a Melt Index prediction at varying process variables

For process development, it is also useful to study the relative importance of variables in predicting the output. The random forest ML model also decides the relative importance of different operating variables in reducing the mean decrease in “node impurity”, which is a measure of how much each operating variable feature reduces the variance in the model. Figure 11.10b illustrates that the ROM

calculates the important features like hydrogen flow rate (H24) and the temperature to the fourth reactor (R4T) as the most important variables affecting the melt index, which can then be used to find the optimum conditions to produce polymer of a specified melt index value and to improve the process design for a new process.

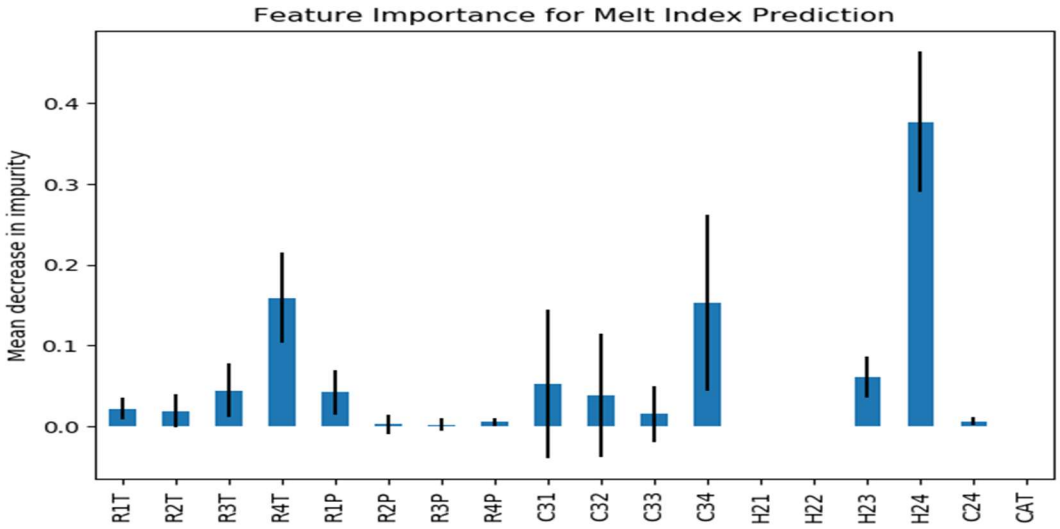

Figure 11.11b Feature importance for melt Index prediction: RxT and RxP refer to the temperature and pressure of reactor x; C3x and H2x represent the mass flow rates of propylene and hydrogen to reactor x; C24 is the mass flor rate of ethylene to reactor 4; and CAT is the catalysts mass flow rate.

Figure 11.12 shows the ML Python code for the feature importance, and the complete code for the example is available in the supplement as ***ROM_Hypol_PP.ipynb***.

```
import time
import numpy as np

start_time = time.time()
importances = rf.feature_importances_
std = np.std([
    tree.feature_importances_ for tree in rf.estimators_], axis=0)
elapsed_time = time.time() - start_time

print(f"Elapsed time to compute the importances: "
      f"{elapsed_time:.3f} seconds")

feature_names = [f'feature {i}' for i in range(X.shape[1])]
feature_names = X.columns

import pandas as pd
forest_importances = pd.Series(importances, index=feature_names)

fig, ax = plt.subplots()
forest_importances.plot.bar(yerr=std, ax=ax)
ax.set_title("Feature Importance for Melt Index Prediction")
ax.set_ylabel("Mean decrease in impurity")
fig.tight_layout
```

Figure 11.12 ML Python code for ranking feature importance

11.4.6 Hybrid SGML Modeling for Uncertainty Quantification

A science-based model can produce results with some uncertainties which can be quantified by ML-based techniques. The uncertainties in science-based models arise from uncertainty in model parameters, and boundary and initial conditions. In some cases, the model bias and assumptions can be a source of uncertainty as well. We can use the predictions from a calibrated model to quantify uncertainties. Data-based ML models like Gaussian process, neural networks etc. are used to help build a surrogate model that defines a relation between model inputs and outputs which can then be used to quantify the uncertainty.

Because of uncertainty in process inputs and process states in a chemical process model, the uncertainty propagates to the process outputs as well. The uncertainty in a science-based model due to any of the parameters or any of the prior knowledge can be used by a ML model to quantify uncertainty in a chemical process as shown in Figure 11.13.

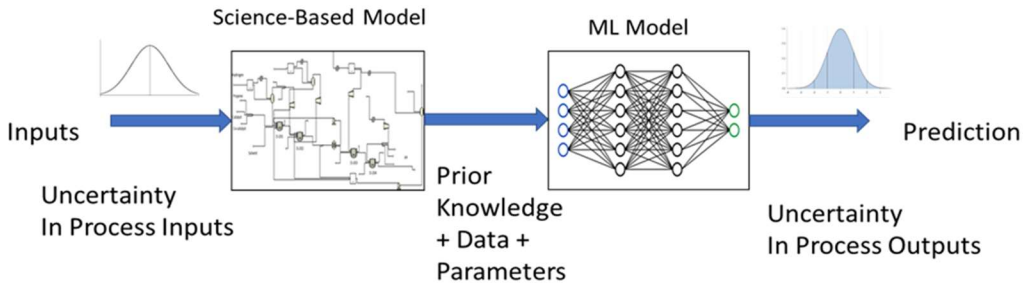

Figure 11.13 Uncertainty quantification modeling framework

This surrogate data- based ML modeling reduces the computational expense of Monte Carlo methods, which are traditionally used for uncertainty quantification (UQ) [98].

Because of uncertainty in process inputs and process states in a chemical process model, the uncertainty propagates to the process outputs as well. Duong et al. [99] uses UQ for process design and sensitivity analysis of complex chemical processes using the polynomial chaos theory. Fenila et al. [100] utilize UQ for electrochemical synthesis, where they calculate simulation uncertainties and global parameter sensitivities for the hybrid model. UQ has also been applied to understand complex reaction mechanisms. Proppe et al. [101] showcase kinetic simulations in discrete-time space considering the uncertainty in free energy and detecting regions of uncertainty in reaction networks. UQ techniques are popular in the field of catalysis and material science as they are used to quantify the uncertainty of models based on density functional theory [102,103]. In another study, Boukouval and Lerapetritou [104] demonstrate the feasibility analysis of a science-based process model over a multivariate factor space. They use a stochastic data-based model for feasibility evaluation, referred to as Kriging and develop an adaptive sampling strategy to minimize sampling cost while maintaining feasibility.

11.4.7 Workshop 11.4 - An Application of SGML Modeling to Uncertainty Quantification in Polymer Manufacturing

The objective of this workshop is to quantify the uncertainty of the chemical process model in predicting the melt index for an industrial HDPE process.

This uncertainty in prediction may result from the estimated kinetic parameters of the process, which propagates to the quality output as well. We simulate the data using Aspen Dynamic model similar to strategy illustrated in Sections 7.6 and 7.7. Then, we use the simulated data from the first-principle

model and fit a ML model to predict the Melt Index. We make use of the concept of prediction intervals to showcase the uncertainty in prediction.

In this case, we calculate the prediction intervals using a gradient boosting ML model [89]. In this case, we use the concept of prediction intervals to determine the range of model prediction. We use the quantile regression loss with gradient boosting model to predict the prediction intervals [105]. We define a lower and an upper quantile according to the desired prediction interval. We consider the uncertainty in the prediction of melt index given by the range of the 90% prediction interval which implies that there is 90% likelihood that the ML model prediction will lie in the given range.

The simulated data are compiled in the spreadsheet and follows the same methodology to load data. Then, we calculate the prediction intervals with the help of Python sklearn libraries as shown in the python code shown in Figure 11.14. In the code, we define the upper and lower quantile values and then define the three gradient boosting models with the upper and lower models defined by quantile loss, while the middle model has the default mean squared loss as shown.

Figure 11.15 shows a part of the ML Python code for uncertain quantification.

The full code is available as ***UQ_HDPE.ipynb*** in the supplement to this chapter.

We then predict the results for each model and plot them using the matplotlib library plots.

```
from sklearn.ensemble import GradientBoostingRegressor
# Set lower and upper quantile
LOWER_ALPHA = 0.1
UPPER_ALPHA = 0.9
# Each model has to be separate
lower_model = GradientBoostingRegressor(loss="quantile",
                                        alpha=LOWER_ALPHA)
# The mid model will use the default loss
mid_model = GradientBoostingRegressor(loss="ls")
upper_model = GradientBoostingRegressor(loss="quantile",
                                        alpha=UPPER_ALPHA)

# Fit models
lower_model.fit(X_train, y_train)
mid_model.fit(X_train, y_train)
upper_model.fit(X_train, y_train)
```

Figure 11.14 ML Python sklearn code to calculate prediction intervals

```
# Record actual values on test set
predictions = pd.DataFrame(y_test)
# Predict
predictions['lower'] = lower_model.predict(X_test)
predictions['mid'] = mid_model.predict(X_test)
predictions['upper'] = upper_model.predict(X_test)

y_lower = predictions['lower']
y_mid = predictions['mid']
y_upper = predictions['upper']

y_l = lower_model.predict(X_test)

#print (rf.score(X_test,Y1_test))
rmse_l = np.sqrt(mean_squared_error(y_test, y_l))
print(rmse_l)

y_m = mid_model.predict(X_test)

#print (rf.score(X_test,Y1_test))
rmse_m = np.sqrt(mean_squared_error(y_m, y_l))
print(rmse_m)


fig = plt.figure()
#plt.plot(xx, f(xx), 'g:', label=r'$f(x) = x\,\sin(x)$')
plt.plot(t, y, 'g.', markersize=10, label=u'Observations')
plt.plot(t, y_mid, 'r-', label=u'Prediction')
plt.plot(t, y_upper, 'k-')
plt.plot(t, y_lower, 'k-')
plt.fill(np.concatenate([t, t[::-1]]),
         np.concatenate([y_upper, y_lower[::-1]]),
         alpha=.5, fc='b', ec='None', label='prediction interval')
plt.xlabel('Time')
plt.ylabel('Melt Index')
plt.ylim(-10, 20)
plt.legend(loc='upper right')
plt.show()
```

Figure 11.15 A part of the ML Python code for the uncertainty quantification example

Figure 11.16 illustrates the uncertainty in the prediction of melt index given by the range of the 90% prediction interval which implies that there is a 90% likelihood that the ML model prediction will lie in the given range. The resulting RMSE value lies within 1.2 to 1.5, with the standard deviation of melt index data equals 5.1. In the figure, we see that the prediction interval is the area between the two black lines represented by the upper quantile (95^{th} percentile) and the lower quantile (5^{th} percentile). From the figure, we see a larger prediction interval that means a higher uncertainty in prediction for time less than 100 hours compared to the later stage because of a more appreciable change in MI in that interval. Thus, uncertainty quantification (UQ) helps in making better process decisions after knowing the error estimate of the model.

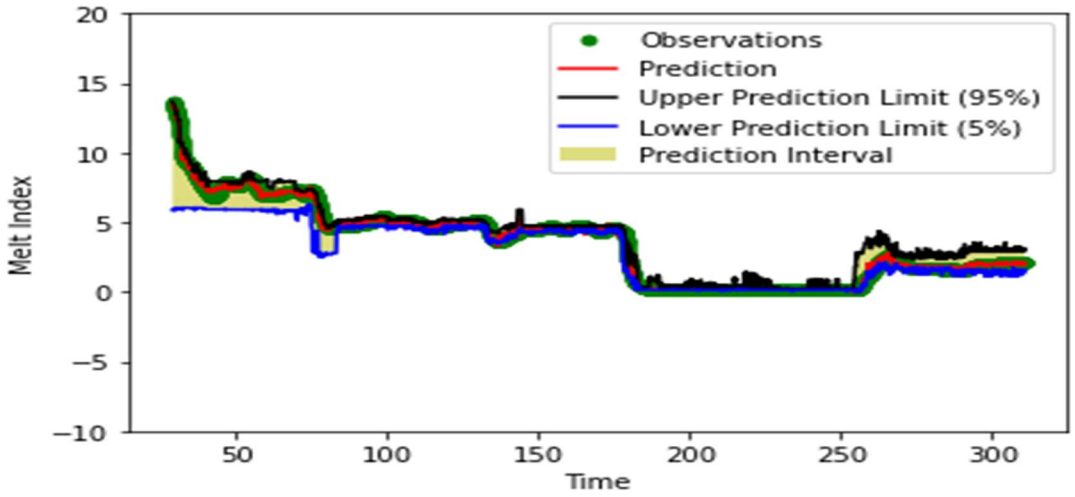

Figure 11.16. Uncertainty quantification of melt Index prediction of a slurry HDPE process

11.4.8 Hybrid SGML Modeling to Aid in Discovering Scientific Laws Using ML

One way in which ML can help science-based modeling is by discovering new scientific laws which governs the system. There is a growing application of ML in physics to rediscover or discover physical laws mainly by data-driven discovery of partial differential equations. ML can be used to develop an empirical correlation which can be used as a scientific law in a science-based model, or ML can be used to solve the partial differential equation defining scientific laws as illustrated in Figure 11.17.

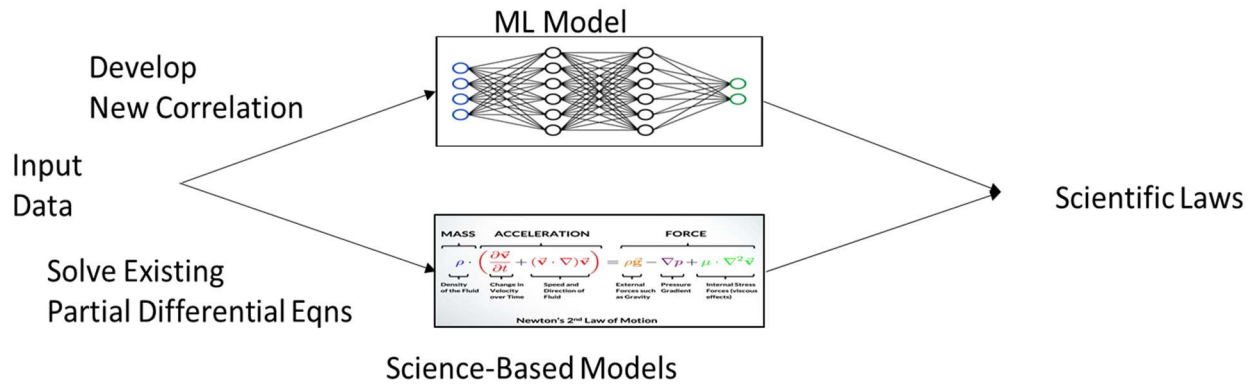

Figure 11.17 Discovering scientific laws

Rudy et al. [106] showcase the discovery of physical laws like the Navier-Stokes equation and the reaction-diffusion equation in chemical processes by a sparse regression method governing the PDE by using a system of time series measurements. Langley et al. [107] present the applications of ML in rediscovering some of the chemistry laws, such as the law of definite proportions, law of combining volumes, determination of atomic weights and many others.

Another important application of ML is to discover some of the thermodynamic laws which can be useful in defining the phase equilibrium and critical for an accurate science-based process model. Nentwich et al. [108] use data-based mixed adaptive sampling strategy to calculate the phase composition, instead of the complex equation-of-state models. Thus, ML application can have promising use in discovering more accurate physical and chemistry laws that govern the chemical process. This

methodology can be used to obtain the functional form of scientific laws as well as the estimation of the parameters of existing laws. Brunton et al. [109] demonstrate a novel framework to discovering governing equations underlying a dynamic system simply from data measurements, leveraging advances in sparsity techniques and machine learning. These scientific laws calculated by ML-based models can then be utilized in first-principle model to improve accuracy as well as reduce model complexity.

11.5 Science Compliments ML

Referring to Figure 11.1, we can also improve ML models using scientific knowledge. We can improve *the generalization or extrapolation capability* and reduce the scientific inconsistency of ML models by using scientific knowledge in designing the ML models. The scientific knowledge can also help in improving the architecture of the data-based ML model or the learning process of the ML model and even with the final post-processing of the ML model results.

11.5.1 Science-Guided Design

In science-guided design, we choose the model architecture based on scientific knowledge. For a neural network, we can decide the intermediate variables expressed as hidden layers based on scientific knowledge of the system. This helps in improving the interpretative ability of the models. Figure 11.18 illustrates a neural network model whose architecture like the number of neurons, hidden layers, activation layers etc. can be decided by prior scientific knowledge.

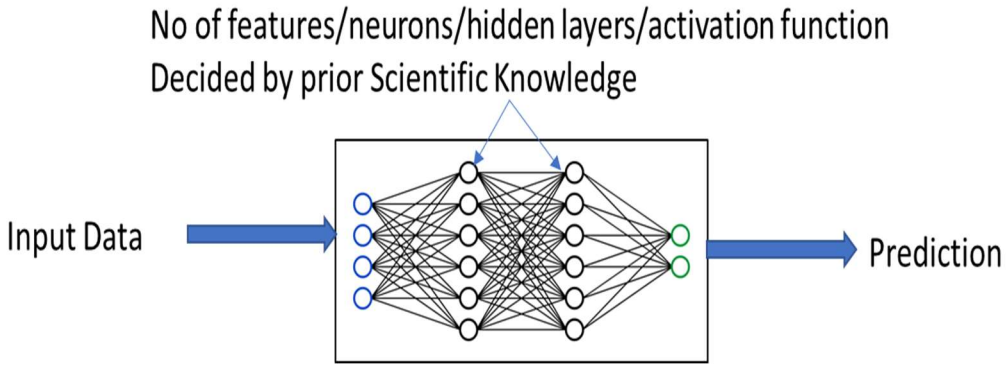

Figure 11.18 Science-guided design framework of neural network architecture

In a bioprocess application, Rodriguez-Granrose et al. [39] use the design of experiments (DOE) to create and evaluate a neural network architecture. They use DOE to evaluate activation functions and neurons on each layer to optimize the neural network. In their recent study, Wang et al. [110] design their theory-infused neural networks based on adsorption energy principles for interpretable reactivity prediction. The use of the novel neural differential equation [111] to solve a first-principle dynamic system represents a hybrid SGML approach, where the architecture of ML model is influenced by the system and finds applications in continuous time series models and scalable normalizing flows. The derivative of hidden state is parameterized using a neural network and the output of the network is computed using a differential equation solver. In a recent study, Jaegher et al. [112] use the neural differential equation to predict the dynamic behavior of electro-dialysis fouling under varying process conditions. In a recent application of this theme in chemical process for model predictive control, Wu et al. [113] use prior process knowledge to design the recurrent neural network (RNN) structure [9]. They showcase a methodology to design the RNN structure using prior scientific knowledge of the system and also employ weight constraints in the optimization problem of the RNN training process. Reis et al. [114]

discuss the concept of incorporation of process-specific structure to improve process fault detection and diagnosis.

Fuzzy artificial neural networks (ANN) is a class of neural networks which utilize prior scientific knowledge of the system to formulate rules mapped on to the structure of the ANN [9,115]. The weights of the ANN connecting the process input to output can be connected to physical process variables [63]. Apart from making the models more scientifically consistent with prior knowledge, they also reduce computational complexity and provides interpretable results. The use of prior knowledge also makes them suitable for extrapolation. Fuzzy ANN have been particularly useful for applications in process control [116]. Simutis et al. use fuzzy ANN system for industrial bioprocess monitoring and control [117,118]. They also illustrate the application of fuzzy ANN process control expert to perform appropriate control actions based on process trends for bioprocess optimization and control [119].

Sparse Identification of Nonlinear Dynamics (SINDy) is another data-based modeling method that utilizes scientific knowledge for improving the model performance with the algorithms [109]. Bhadiraju et al. [120] have used the SINDy algorithm to identify the Non Linear Dynamics of a chemical process system(CSTR). They used sparse regression in combination with feature selection to identify accurate models in an adaptive model identification methodology which requires much less than data that current methods. In a similar study Bhadiraju et al. [121] have a modified adaptive SINDy approach that is helpful in cases of plant model mismatch and does not require retraining and hence computationally less expensive.

11.5.2 Science-Guided Learning

Here, we make use of the scientific principles to improve the scientific consistency of data-based models by modifying the machine learning process. We do this by modifying the loss function, constraints and even the initialization of ML models based on scientific laws. Specifically, in order to make the ML models physically consistent we make the loss function of neural network model incorporate physical constraints [2]. A loss function in ML measures how far an estimated value is from its true value. A loss function maps decisions to their associated costs. Loss functions are not fixed, they change depending on the task in hand and the goal to be met. We can define a loss function (based on the mean squared error, MSE) of the ML model ($Loss_M$) for regression to calculate the difference between the true value (Y_{true}) and the model predicted value (Y_{pred}). Likewise, we can define a loss function for a science-based model ($Loss_{SC}$), which is a function of the model predicted value (Y_pred) consistent with science-based loss. We include a weighting factor λ to express the relative importance of both loss terms. We write the overall loss function (Loss) as:

$$Loss = Loss_M(Y_{true} - Y_{pred}) + \lambda Loss_{SC}(Y_{pred}) \quad (11.3)$$

Figure 11.19 illustrates the concept of science-guided loss function.

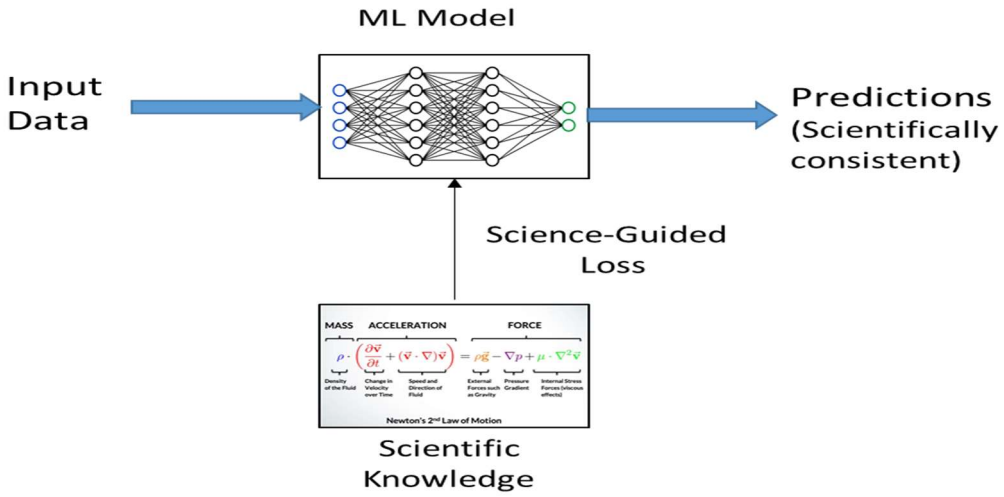

Figure 11.19 Science-guided loss function representation

A science-guided initialization helps in deriving an initial choice of parameters before a model is trained so that it improves model training and also prevents from reaching a local minimum, which is the concept of transfer learning. Thus, we can use the data from a science-based model to pre-train a ML model based on this concept of initialization [1,2,7]. This concept has been utilized in chemical process model in the form of process similarity and developing new process models through migration. In particular, Lu et al. [122] introduce the concept of process similarity, and classify it into attribute-based and model-based similarities. They present a model migration strategy to develop a new process model by taking advantage of an existing base model, and process attribute information. Adapting existing process models can allow using fewer experiments for the development of a new process model, resulting in a saving of time, cost, and effort. They apply the concept to predict the melt-flow-length in injection molding and obtain satisfactory results.

In another study on the similar concept, Yan et al. [123] use a Bayesian method for migrating a ML Gaussian process regression model. They showcased an approach of an iterative model migration and process optimization for an epoxy catalytic reaction process.

Recently, Kumar et al. [124] try to optimize the Non-Newtonian fluid flow for industrial processes like crude oil transportation using a physics- based loss function for the shear stress calculation for more accurate flow predictions. In another study on the similar principle, Pun et al. [125] apply physics-informed neural networks for more accurate and transferable atomistic modeling of materials.

11.5.3 Workshop 11.5 - An Illustrative Example of Science-Guided Learning

The objective of this example is to illustrate the application of the science-guided loss function in the slurry HDPE process for the industrial HDPE process described in Section 2.1.4.

The goal is to predict the melt index of the polymer. The plant only measures the polymer melt index as the quality output, but we also want the data-based ML model to predict the scientifically consistent polymer density values.

We express polymer density as a function of the melt index using some empirical correlations and modify the loss function (based on the mean squared error, MSE) to consider density as well. See Eq.

(11.4) below. We then train a deep neural network (DNN) model to predict the melt index of the polymer.

$$Loss = Loss_M(MI_{true} - MI_{pred}) + \lambda Loss_{SC}(\rho(MI_{true}) - \rho(MI_{pred})) \quad (11.4)$$

The Python process of loading the data is similar and we also normalize/preprocess the data.

We use the tensor flow framework for training the multilayer neural network model as shown in the python code shown in Figure 11.20. The DNN has 2 hidden layers with 64 and 2 neurons, respectively. It uses the Rectified Linear Unit (ReLU) transfer function.

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.layers.experimental import preprocessing


model = tf.keras.models.Sequential([
   tf.keras.layers.Dense(64, activation='relu',input_shape=(n_features,)),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(32, activation='relu'),
  tf.keras.layers.Dense(1)
])
```

Figure 11.20 The ML Python code for a deep neural network for predicting melt index

In order to modify the loss function, we define a custom loss function by the ML Python code shown in Figure 11.21.

```
def custom_loss_function(y_pred,y_true):
    k1 = 0.001

    #y_pred = tf.convert_to_tensor_v2(y_pred)
    y_true = tf.cast(y_true, y_pred.dtype)
    #loss = tf.reduce_mean(tf.square(y_pred - y_true), axis=-1)
    loss = tf.reduce_mean((tf.square(y_pred - y_true)) +
                    k1*(tf.math.log(abs(y_pred)) - tf.math.log(abs(y_true))))
    return loss
```

Figure 11.21 The ML Python code for defining the loss function of Eq. (11.4)

We then train and optimize the neural network and also output the RMSE and prediction using the ML Python code shown in Figure 11.22.

```
model.compile(optimizer='adam',loss= custom_loss_function)
history = model.fit(X_train,Y1_train, epochs=500)


Y_p = model.predict(X)
df1 = pd.DataFrame(Y_p)
df1.to_excel("SGloss.xlsx")
```

Figure 11.22 The ML Python code to optimize the deep neural network model for melt index

Figure 11.23 illustrates that the SGML hybrid model calculates the melt index, resulting in a RMSE of the melt Index that is slightly higher (RMSE = 0.8) (standard deviation of data= 5) compared to a standalone ML model. In addition to predicting the melt index values, the hybrid SGML model is simultaneously predicting the polymer density correctly within the physically consistent range of 0.94-0.97 g/c. By contrast, the density estimates by the ML model alone result in density values greater than 1, which is physically inconsistent.

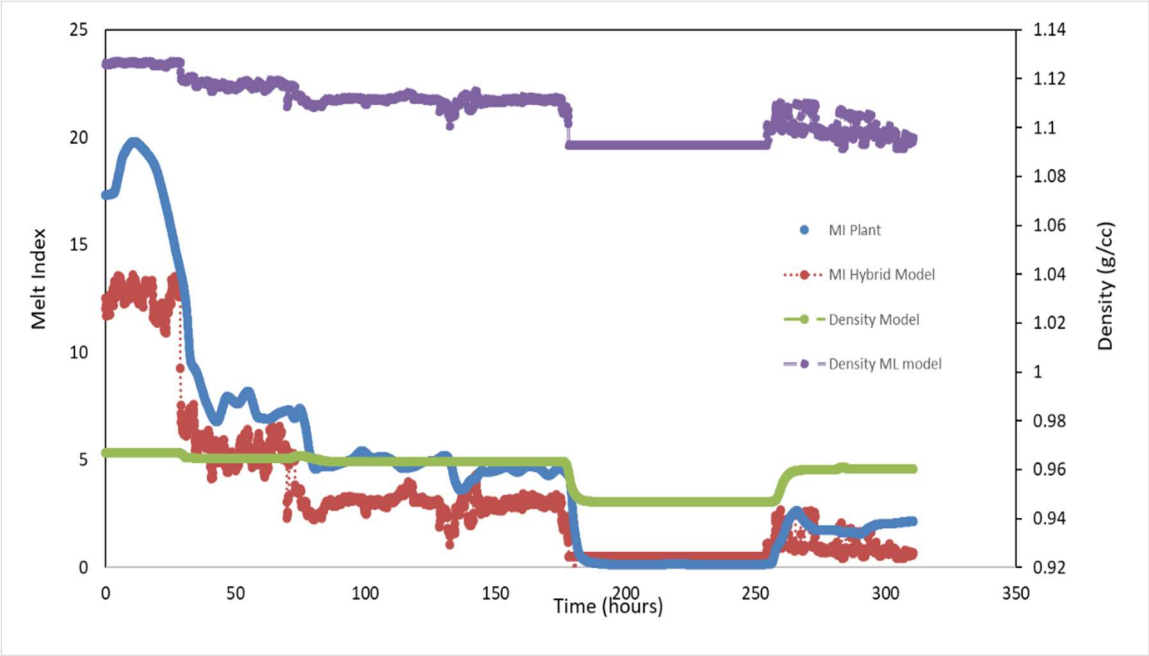

Figure 11.23 Melt index and polymer density prediction with a ML model with a science-guided loss function

11.5.4 Science-Guided Refinement

By science-guided refinement, we mean the post-processing of ML model results based on scientific principles. This post-processing of results of the ML model using science-based models can be useful to the design and prediction of material structure [116]. Thus, the discovery of materials forms the basis of chemical process development from which the manufacturing process of any compound can be designed. This is different than the serial direct hybrid model discussed in Section 4.1.2. In particular, we

use the science-based model to merely test the scientific consistency of the ML model results. Hautier et al. [117] use first-s models based on density functional model to refine the results of probabilistic ML models to discovery ternary oxides. Figure 11.24 illustrates the science-guided refinement framework.

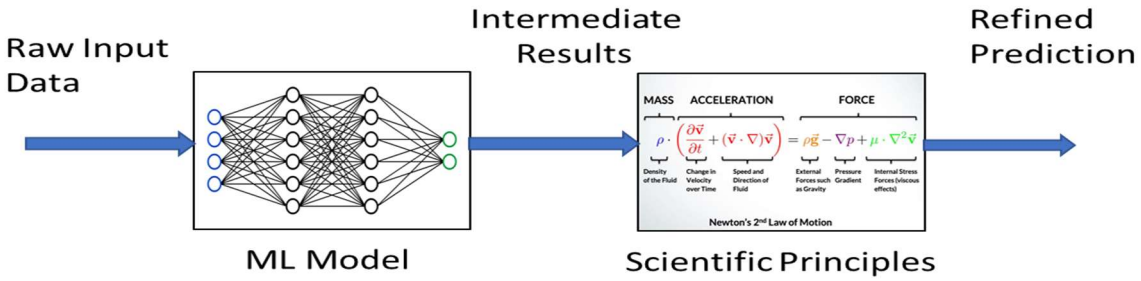

Figure 11.24 Science-guided refinement framework

Another application for science-guided learning is for data generation. ML techniques like generalized adversarial networks (GAN) are useful for generating data in an unsupervised learning. GANs do have a problem of high sample complexity [2] which can be reduced by incorporating some science-based constraints and prior knowledge. Cang et al. [118] apply ML models to predict the structure and properties of materials and use the results of the ab initio calculations to refine the ML model results. They generate more imaging data for property prediction using a convolution neural network and introduce a morphology constraint form scientific principles, while training of the generative models so that it improves the prediction of the structure- property model.

Thus, some of these methodologies of having science complimenting ML have much potential for future applications to chemical and polymer processes.

11.6 Workshop 11.5 – Reduced-Order Model for a Polystyrene Process Using Aspen Multi-Case and Aspen AI Model Builder

In Sections 11.4.4 and 11.4.5, we discussed the principles of a reduced-order model (ROM) and presented a workshop of developing a ROM for polyolefin manufacturing.

The objective of this workshop is to illustrate two useful AspenTech software tools for developing hybrid SGML model for polyolefin and other chemical processes.

11.6.1 Introduction to Aspen Multi-Case and Aspen AI Model Builder

In Section 11.4.5, we use an Aspen Plus simulation model of a HYPOL polypropylene process, and run multiple steady-state simulations to generate multivariate process data with varying operating variables and the corresponding melt index predictions. We can speed up the process data generation by using the software tool, Aspen Multi-Case, in conjunction with Aspen Plus to rapidly simulate alternative process scenarios in parallel and leveraging high-performance computing, machine learning, data analysis and visualization tools. We refer the reader to an online demand webinar by Mofar and Baker [129] that explains the principles and practice of using Aspen Multi-Case. This webinar demonstrates how Aspen Multi-Case can help to: (1) leverage the computing power available to run Aspen Plus and Aspen HYSYS cases in a fraction of the time; (2) employ visualization to analyze multiple cases and evaluate tradeoffs on quality, economics, safety and sustainability; and (3) share the simulation files and results seaMLessly with other engineers and stakeholders. We will show an application below.

Another useful tool to the Aspen AI Model Builder. This is a SaaS (Software as a Service) product, that is a software distribution model in which a cloud provider hosts applications and makes them available to end users over the internet. In particular, referring to reference [130], we see that since August 1, 2020, AspenTech continues to roll out periodic updates of new features and modeling tools for Aspen AI Model Builder which immediately become available to all of our users. These include, for example, creating a reduced-order sensor, planning, equipment or production optimization model, and creating an AI-driven hybrid model. We illustrate its application to developing a hybrid reduced-order model for a polystyrene process below.

11.6.2 Developing a Hybrid Reduced-Order Model (ROM) for a Polystyrene Process

We consider a polystyrene production process with three reactors in series and three distillation columns for separation. We plan to build a hybrid ROM model for the whole process

We reduce the process to one hierarchy with only main input and output streams. We do this by selecting the whole process and then right-click to move to a hierarchy. We make sure we add the input and output streams to the process correctly. Figure 11.25 shows the original polystyrene process flowsheet.

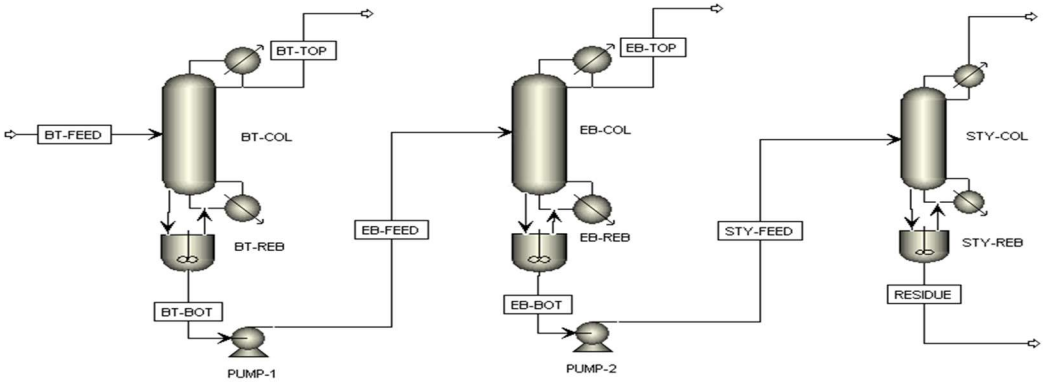

Figure 11.25 Original process flowsheet of a polystyrene process

We use Aspen Multi-Case to generate the dataset from the process model. Figure 11.26 shows the interface of the Aspen Multi-Case for our problem.

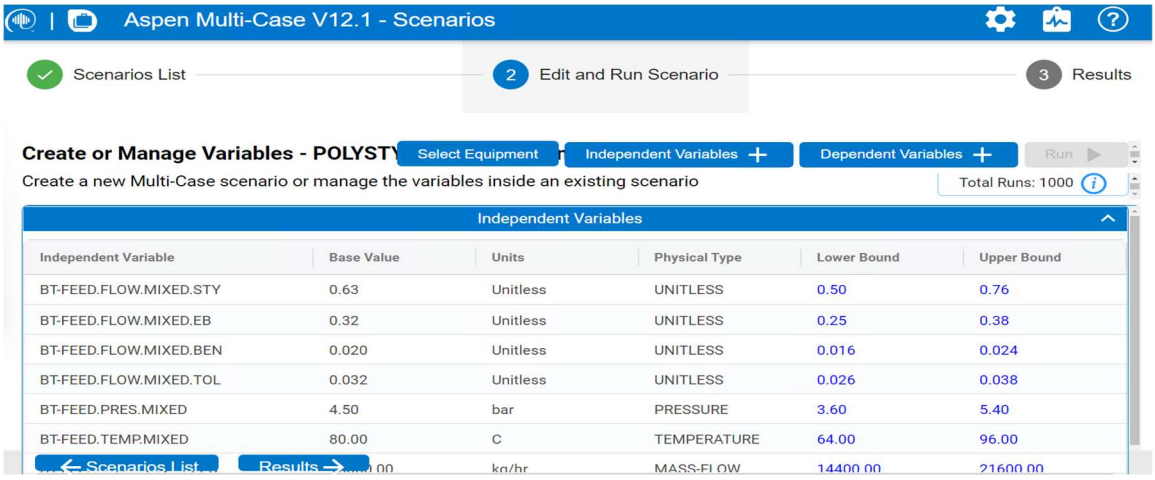

Figure 11.26 Interface for Aspen Multi-Case for data generation for ROM

The software generates the dataset by sampling the independent and dependent variables. We choose run scenario by selecting the whole process hierarchy and the process generates the default independent and dependent stream variables, which are temperature, pressure, mass flow and mass fraction of the input and output streams. We can add more independent variables as needed for our hybrid ROM model. We can also add more equipment variables like reactor temperature, pressure, etc. as needed for our hybrid ROM model.

We edit the lower and upper bounds for the independent variables as required. We then define the number of runs in the Multi-Case. After the model runs, we can download the dataset in the form of ****.json*** model file.

Then we use Aspen AI Model builder to build the data-base model. Figure 11.27 illustrates the variable interface of the Aspen AI Model Builder. Note the work flow as indicated at the top of the interface figure: Import data -> Manage variables -> Clean data ->Build model -> Validate model.

We first "import data", that is, the ****.json*** model file from Aspen Plus Multi-Case. The model "manages variables" by automatically identifying the independent and dependent stream variables. Then, the software "cleans data" in the form of data required for model building. To "build model", we can add the engineering constraints, which include the overall mass balance and physical constraints like mass fraction equal to 1.

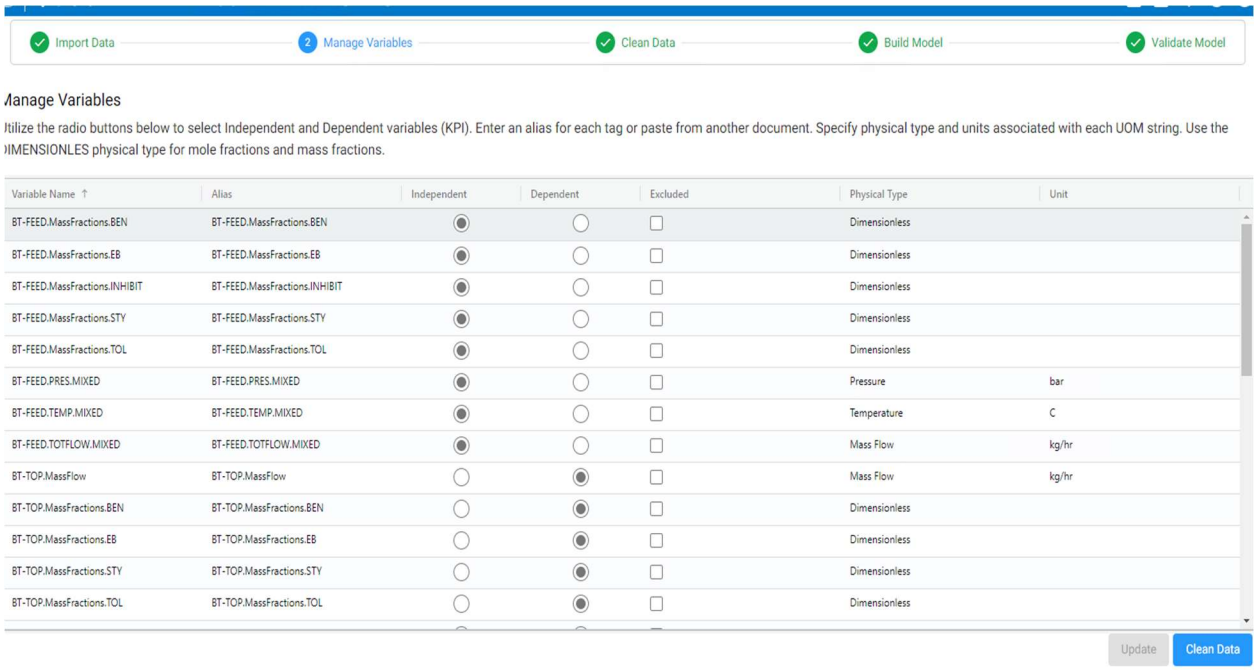

Figure 11.27 Variable interface of Aspen AI Model Builder

Figure 11.28 illustrates the interface for model building in Aspen AI Model Builder. We on the right side of the figure that we choose "*Lasso CV*" as the ML method. The word "Lasso" stands for Least Absolute Shrinkage and Selection Operator, and the word "CV" means cross validation. *Lasso regression* is a type of linear regression that uses shrinkage, where data values are shrunk towards a central point, like the mean. It is a statistical formula for the regularization of data models and feature selection to avoid overfitting of the data, especially when the trained and test data are much varying. Specifically, regularization adds a "penalty" term to the best fit derived from the trained data, to achieve a lesser variance with the tested data and also restricts the influence of predictor variables over the output

variable by compressing their coefficients. *Cross-validation* is a technique for evaluating ML models by training several ML models on subsets of the available input data and evaluating them on the complementary subset of the data. Use cross-validation to detect overfitting, i.e., failing to generalize a pattern.

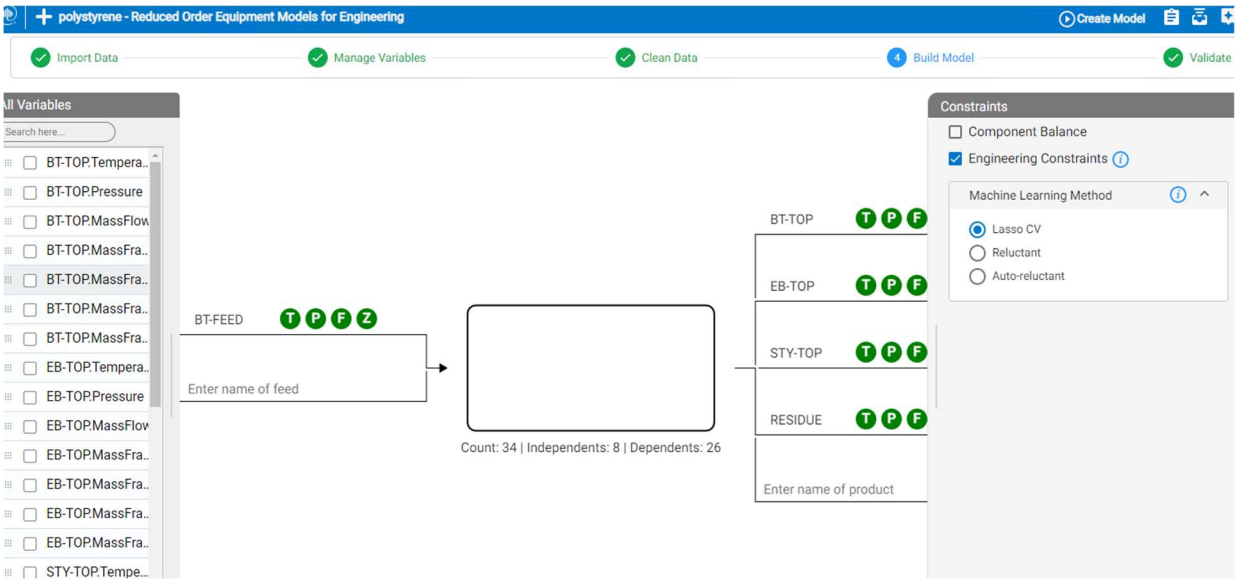

Figure 11.28 Model building in Aspen AI Model Builder

In Figure 11.28, we see names of feed and product. Figure 11.29 shows the model validation results in Aspen AI Model Builder. This figure shows that the R2 values of dependent variables are above 0.96656, and the Q2 values after across validation are above .9961. We note that we define both R2 and V2 values previously in Eqs. (9.19) and (9.20) in Section 9.14. In the figure, we also see the root-mean squared error (RMSE) values. These results indicate a very good model prediction.

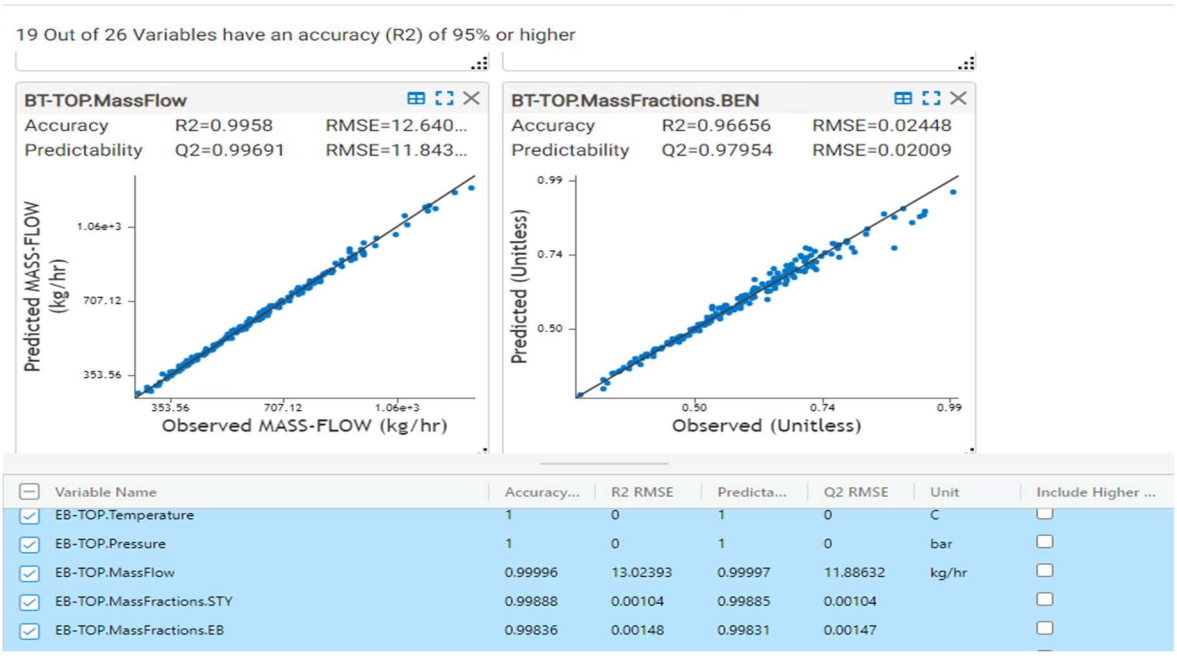

Figure 11.29 Model validation results in Aspen AI Model Builder

Next, we download the hybrid model results from the side tab in the format of AspenTech hybrid model file, ***.athm.*** We deploy the data-based ROM in Aspen Plus by following the path: Customize -> Manage Hybrid Models -> Browse for available models, ***.ahtm*** -> open. Figure 11.30 shows the resulting process flowsheet of th hybrid reduced-order model (ROM) for the polystyrene process.

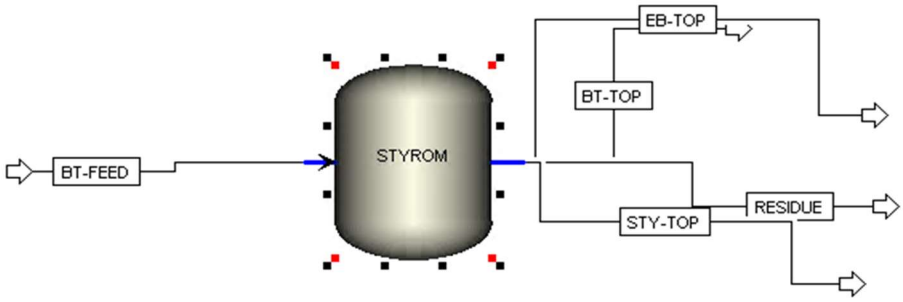

Figure 11.30 Process flowsheet of ROM for polystyrene process

When we run the hybrid ROM on Aspen Plus, we see that the hybrid ROM runs much faster than the original process model.

We can even set up an optimization problem using the ROM model to test the faster and easier convergence for the model. For this case, we set up a small optimization problem to maximize the polystyrene polymer flow in the bottom residue by varying the feed flow rate, and the model converges very fast with results displayed in Figure 11.31.

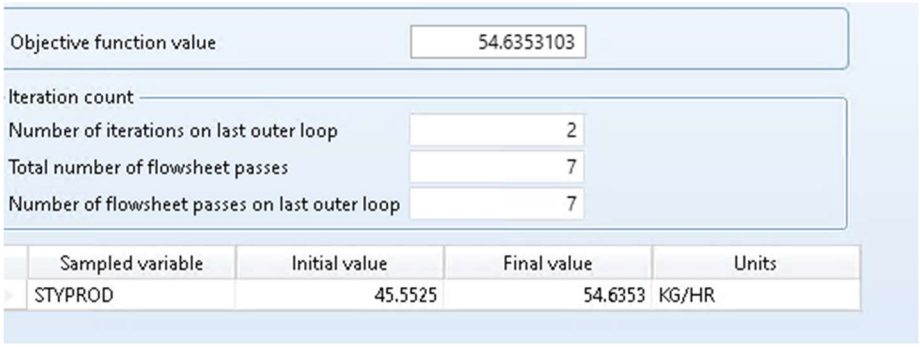

Figure 11.31 Results of optimization study using the hybrid reduced-order model (ROM).

This process can be repeated form more complex plant flowsheets for easy analysis and optimization of the complex process using a hybrid reduced-order model. We conclude this workshop and present the challenges and opportunities of hybrid SGML approach for modeling chemical and polymer processes below.

11.7 Challenges and Opportunities of Hybrid SGML Approach for Modeling Chemical and Polymer Processes

Along with all the merits of using the SGML methodology there are challenges as well. Incorrect fundamental knowledge and the assumptions of the science-based first-principle model will lead to

inaccurate hybrid model, so it is important for the scientific model to be very accurate. Lack of engineer/scientists having expertise in both domain knowledge and machine learning. Computation infeasibility in certain modeling approaches like inverse modeling.

Data cleaning, preprocessing, feature engineering maybe difficult in certain cases but may be imperative in science-based model parameter estimation hence in these cases the hybrid models may increase the complexity compared to a stand-alone ML models like Neural Network which may not require feature engineering. Model predictions must not only be accurate but also with lower uncertainty which may be difficult for certain hybrid model methods.

There is a lot of scope of using hybrid SGML methodologies in chemical process modeling, summarizing here some of the opportunities and areas where they can be beneficial. As we have seen Hybrid SGML models are useful for extrapolation and predicting beyond operating range, hence it will be particularly useful for processes development. Process fault diagnosis and anomaly detection is one such area where data-based methods have been used extensively, thus there is opportunity to combine scientific knowledge as well to make the anomaly detection process more scientifically consistent.

Table 11.2 summarizes all the hybrid SGML models and their requirements, advantages, limitations and potential applications.

Tabl1 11.2 Summary of th SGML approach to modeling chemical and polymer processes

Hybrid SGML modeling		Science-based model/ knowledge	ML model	Advantages	Limitations	Potential applications
ML compliments science (base model: science-based)						
Direct hybrid modeling	Series	Science-based model (SBM)	Regression	Extrapolation; parameter estimation; data augmentation	Limited by data for parameter estimation, interpolation, scientific knowledge dominated	Kinetic estimation[77,78] Soft sensor[86] Process optimization[75] Process design[51] Process modeling[29,76]
	Parallel	SBM	Regression	Improved accuracy of prediction, interpolation	Scientific consistency depends on SBM, extrapolation; data-dominated	Process scale-up[41] Process optimization[33,68] Process control[36,71,72] Soft sensor[33] Process monitoring[68] Predictive maintenance[66,74]
	Series–parallel	SBM	Regression	Higher accuracy, interpolation	Increased model complexity, data-dominated	Process optimization[80,81] Process monitoring and control[58] Plant-model mismatch[17]
Inverse modeling		SBM	Probabilistic, regression	Computationally cheaper inverse problem	Lower generality of the model	Product design and development[85] Polymer grade change Material design[87]
Reduced-order models		SBM	Regression	Fast online deployment, reduce model complexity	Higher bias, limited by SBM accuracy	Process optimization at plant scale[45,94] Dynamic modeling[93,95] Soft sensor[90] Feasibility analysis[91,92]
Uncertainty quantification		SBM	Probabilistic	Gives real error estimate and solution space	Limited by SBM assumptions, parameters	Process design and development[99] Reaction kinetics[100] Feasibility analysis[104]
Discovering scientific law		SBM	Regression, probabilistic	System stability interpretability	Limited by data size/availability	Physical chemistry[107] Fluid dynamics[106] Thermodynamics phase equilibrium[108]
Science compliments ML (base model: ML)						
Science-guided design		Laws or SBM	Deep neural network (DNN); neural differential equation	Scientifically consistent and interpretable	Requires deep scientific knowledge of system	Dynamic system[112] Process control[113,118] Process monitoring[154]
Science-guided learning		Laws	DNN	Scientifically consistent and interpretable	Possible lower prediction accuracy	Process design and development[122,123] Process monitoring Process flow[124]
Science-guided refinement		SBM	Probabilistic	Less effort in feature selection	Limited by SBM assumptions, parameters	Process design Discovery of materials[125–128]

11.8 Conclusion

We present a broad perspective of hybrid modeling with a science-guided machine learning (SGML) approach and its application in bioprocessing and chemical engineering. We give a detailed review and

exposition of the hybrid SGML modeling approach and its applications, and classify the approach into two categories. The first refers to the case where a data-based ML model compliments and makes the first-principle science-based model more accurate in prediction, and the second corresponds to the case where scientific knowledge helps make the ML model more scientifically consistent. We point out some of the areas of SGML which have not been explored much in chemical process modeling and have potential for further use like in the areas where Science can help improve the data-based model by improving the model design, learning and refinement. We also illustrate some of these applications of the hybrid SGML methodologies for industrial polymer/chemical process improvement.

Thus, based on our review, we recommend that the use of hybrid models will perform better than standalone ML for applications like process development, since they are better at extrapolation while standalone ML models which can be adequate for prediction in a steady running plant.

This chapter is published with Wiley publication in the book *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing by Liu & Sharma [133-147].*

11.9 Bibliography

1. Karpatne, A.; Atluri, G; Faghmous, J. H.; Steinbach, M.; Banerjee, A.; Ganguly, A. Shekhar, S. Samatova, N.; Kumar, V. (2017). Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data. *IEEE Transactions on Knowledge and Data Engineering*. **29,** 2318.
2. Willard, J.; Jia, X.; Xu, S.; Steinbach, M.; Kumar, V. (2003). Integrating Physics-Based Modeling with Machine Learning: A Survey. arXiv preprint arXiv:2003.04919. 2020.
3. Muralidhar, N.; Bu, J.; Cao, Z.; He, L.; Ramakrishnan, N.; Tafti, D.; Karpatne, A. (2020) PhyNet: Physics Guided Neural Networks for Particle Drag Force Prediction in Assembly. *Proceedings of the 2020 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics. pp.559-567.
4. Bode, M.; Gauding, M.; Lian, Z.; Denker, D.; Davidovic, M.; Kleinheinz, K.; Jitsev, J.; Pitsch, H. (2019). Using Physics-Informed Super-Resolution Generative Adversarial Networks for Subgrid Modeling in Turbulent Reactive Flows. arXiv preprint arXiv:1911.11380.
5. Schütt, K. T.; Kindermans, P. J.; Sauceda, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K. R. Schnet (2017). A Continuous-Filter Convolutional Neural Network for Modeling Quantum Interactions. arXiv preprint arXiv:1706.08566.
6. Faghmous, J. H.; Kumar, V. (2014). A Big Data Guide to Understanding Climate Change: The Case for Theory-Guided Data Science. *Big Data*., **2,** 155.
7. Karpatne, A.; Watkins, W.; Read, J.; Kumar, V. (2017). Physics-Guided Neural Networks (PGNN): An Application in Lake Temperature Modeling. arXiv preprint arXiv:1710.11431.
8. Lee, D.; Jayaraman, A.; Kwon, J. S. (2020). Development of a Hybrid Model for a Partially Known Intracellular Signaling Pathway through Correction Term Estimation and Neural Network Modeling. *PLoS Computational Biology*, **16**, e1008472.
9. Baughman, D. R.; Liu, Y. A. (1995). *Neural Networks in Bioprocessing and Chemical Engineering*. Academic Press, San Diego, CA.
10. Psichogios, D. C.; Ungar, L. H. (1992). A Hybrid Neural Network-First Principles Approach to Process Modeling. *AIChE Journal*, **38**, 1499.
11. Thompson, M. L.; Kramer, M. A. (1994). Modeling Chemical Process Using Prior Knowledge and Neural Networks. *AIChE Journal*, **40**,1328.
12. Agarwal, M. (1997). Combining Neural and Conventional Paradigms for Modelling, Prediction and Control. *International Journal of Systems Science,* **28**, 65.

13. Asprion, N.; Böttcher, R.; Pack, R.; Stavrou, M. E.; Höller, J.; Schwientek, J.; Bortz, M. (2019). Gray-Box Modeling for the Optimization of Chemical Processes. *Chemie Ingenieur Technik,* **91**, 305.
14. Bohlin, T. P. (2006). *Practical Grey-Box Process Identification: Theory and Applications*. Springer Science & Business Media.
15. Yang, S.; Navarathna, P.; Ghosh, S.; Bequette, B. W. (2020). Hybrid Modeling in the Era of Smart Manufacturing. *Computers & Chemical Engineering,* **140**, 106.
16. Sansana, J.; Joswiak, M. N.; Castillo, I.; Wang, Z.; Rendall, R.; Chiang, L. H.; Reis, M. S. (2021). Recent Trends on Hybrid Modeling for Industry 4.0. *Computers & Chemical Engineering*. **11**, 107.
17. Chen, Y.; Ierapetritou, M. (2020). A Framework of Hybrid Model Development with Identification of Plant-Model Mismatch. *AIChE Journal,* **66**, e16996.
18. Von Stosch, M.; Oliveira, R.; Peres, J.; de Azevedo, S. F. (2014). Hybrid Semi-Parametric Modeling in Process Systems Engineering: Past, Present and Future. *Computers & Chemical Engineering*. **60**, 86.
19. Qin, S. J.; Chiang, L. H. (2019). Advances and Opportunities in Machine Learning for Process Data Analytics. *Computers & Chemical Engineering*. **126**, 465.
20. Qin, S. J.; Guo, S.; Li, Z.; Chiang, L. H.; Castillo, I.; Braun, B.; Wang, Z. (2021). Integration of Process Knowledge and Statistical Learning for the Dow Data Challenge Problem. *Computers & Chemical Engineering*. **153**, 107451.
21. O'Brien, C. M.; Zhang, Q. Daoutidis, P.; Hu, W. S. (2021). A Hybrid Mechanistic-Empirical Model for in silico Mammalian Cell Bioprocess Simulation. *Metabolic Engineering*. **66**, 31.
22. Pinto, J.; de Azevedo, C. R.; Oliveira, R., von Stosch, M. (2019). A Bootstrap-Aggregated Hybrid Semi-Parametric Modeling Framework for Bioprocess Development. *Bioprocess and Biosystems Engineering*,**42**, 1853.
23. Chopda, V.; Gyorgypal, A.; Yang, O.; Singh, R.; Ramachandran, R.; Zhang, H.; Tsilomelekis, G.; Chundawat, S. P.; Ierapetritou, M. G. (2021). Recent Advances in Integrated Process Analytical Techniques, Modeling, and Control Strategies to Enable Continuous Biomanufacturing of Monoclonal Antibodies. *Journal of Chemical Technology & Biotechnology,* http:/doi.org/10.1002/jctb.6765
24. Zhang, D.; Del Rio-Chanona, E. A.; Petsagkourakis, P.; Wagner, J. (2019). Hybrid Physics-Based and Data-Driven Modeling for Bioprocess Online Simulation and Optimization. *Biotechnology and Bioengineering.* **116,** 2919.
25. Al-Yemni, M.; Yang, R. Y. (2005). Hybrid Neural-Networks Modeling of an Enzymatic Membrane Reactor. *Journal of the Chinese Institute of Engineers,* **28,** 1061.
26. Chabbi, C.; Taibi, M.; Khier, B. (2008). Neural and Hybrid Neural Modeling of a Yeast Fermentation Process. *International Journal of Computational Cognition,* **6,** 42.
27. Corazza, F. .; Calsavara, L. P.; Moraes, F. F.; Zanin, G. M.; Neitzel, I.(2005). Determination of Inhibition in the Enzymatic Hydrolysis of Cellobiose Using Hybrid Neural Modeling. *Brazilian Journal of Chemical Engineering,* **22,** 19.
28. Azarpour, A.; Borhani, T. N.; Alwi, S. R.; Manan, Z. A.; Mutalib, M. I. (2017). A Generic Hybrid Model Development for Process Analysis of Industrial Fixed-Bed Catalytic Reactors. *Chemical Engineering Research and Design,* **117**, 149.
29. Luo, N.; Du, W.; Ye, Z.; Qian, F. (2012). Development of a Hybrid Model for Industrial Ethylene Oxide Reactor. *Industrial & Engineering Chemistry Research*. **51**, 6926.
30. Zahedi, G.; Lohi, A.; Mahdi, K. A. (2011). Hybrid Modeling of Ethylene to Ethylene Oxide Heterogeneous Reactor. *Fuel Processing Technology*. **92,**1725.
31. Simon, L. L.; Fischer, U.; Hungerbühler, K. (2006). Modeling of a Three-Phase Industrial Batch Reactor Using a Hybrid First-Principle Neural-Network Model. *Industrial & Engineering Chemistry Research*. **45**,7336.

32. Bellos, G. D.; Kallinikos, L. E.; Gounaris, C. E.; Papayannakos, N.G. (2005). Modelling of the Performance of Industrial HDS Reactors Using a Hybrid Neural Network Approach. *Chemical Engineering and Processing: Process Intensification*. **44**, 505.
33. Chang, J. S.; Lu, S. C.; Chiu, Y. L. (2007). Dynamic Modeling of Batch Polymerization Reactors via the Hybrid Neural-Network Rate-Function Approach. *Chemical Engineering Journal*. **130**, 19.
34. Hinchliffe, M.; Montague, G.; Willis, M.; Burke, A. (2003). Hybrid Approach to Modeling an Industrial Polyethylene Process. *AIChE Journal*. **49**, 3127.
35. Madar, J.; Abonyi, J.; Szeifert, F. (2005). Feedback Linearizing Control Using Hybrid Neural Networks Identified by Sensitivity Approach. *Engineering Applications of Artificial Intelligence*.**36**, 343.
36. Simutis, R.; Lübbert, A. (2017). Hybrid Approach to State Estimation for Bioprocess Control. *Bioengineering*. **4**, 21.
37. Cubillos, F.; Callejas, H.; Lima, E. L.; Vega, M. P. (2001). Adaptive Control Using a Hybrid-Neural Model: Application to a Polymerization Reactor. *Brazilian Journal of Chemical Engineering*. **18**, 113.
38. Doyle III, 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*. Computers & Chemical Engineering.* **27**, 1153.
39. Rodriguez-Granrose, D.; Jones, A.; Loftus, H.; Tandeski, T.; Heaton, W.; Foley, K.T.; Silverman, L. (2021). Design of Experiment (DOE) Applied to Artificial Neural Network Architecture Enables Rapid Bioprocess Improvement. *Bioprocess and Biosystems Engineering*. **44**, 1301.
40. Brendel, M.; Marquardt, W. (2008). Experimental Design for the Identification of Hybrid Reaction Models from Transient Data. *Chemical Engineering Journal*. **141**, 264.
41. Bollas, G. M.; Papadokonstadakis, S.; Michalopoulos, J.; Arampatzis, G.; Lappas, A. A.; Vasalos, I. A.; Lygeros, A. (2003). Using Hybrid Neural Networks in Scaling up an FCC Model from a Pilot Plant to an Industrial Unit. *Chemical Engineering and Processing: Process Intensification*. **42**, 697.
42. Von Stosch, M.; Hamelink, J. M.; Oliveira, R. (2016). Hybrid Modeling as a QbD/PAT Tool in Process Development: An Industrial E. coli Case Study. *Bioprocess and Biosystems Engineering*. **39,** 773.
43. Iwama, R.; Kaneko, H. (2021). Design of Ethylene Oxide Production Process Based on Adaptive Design of Experiments and Bayesian Optimization. *Journal of Advanced Manufacturing and Processing*. **3**, e10085.
44. Zhang, S.; Wang, F.; He, D.; Jia, R. (2012). Batch-to-Batch Control of Particle Size Distribution in Cobalt Oxalate Synthesis Process Based on Hybrid Model. *Powder Technology*. **224**, 253.
45. Kumar, A.; Baldea, M.; Edgar, T. F. (2016). Real-Time Optimization of an Industrial Steam-Methane Reformer under Distributed Sensing. *Control Engineering Practice*. **54**, 140.
46. Zhou, T.; Gani, R.; Sundmacher, K. (2021). Hybrid Data-Driven and Mechanistic Modeling Approaches for Multiscale Material and Process Design. *Engineering*. **7**, 1231.
47. Cardillo, A. G.; Castellanos, M. M.; Desailly, B.; Dessoy, S.; Mariti, M.; Portela, R. M.; Scutella, B.; von Stosch, M.; Tomba, E.; Varsakelis, C. (2021). Towards *in silico* Process Modeling for Vaccines. *Trends in Biotechnology*, **39**,1120.
48. McBride, K.; Sanchez, Medina EI.; Sundmacher, K. (2020). Hybrid Semi-Parametric Modeling in Separation Processes: A Review. *Chemie Ingenieur Technik*, **92**, 842.
49. Safavi, A. A,; Nooraii, A.; Romagnoli, J. A. (1999). A Hybrid Model Formulation for a Distillation Column and the On-Line Optimisation Study. *Journal of Process Control*. **9**, 125.
50. Mahalec, V. (2018). Hybrid Modeling of Petrochemical Processes. In *Hybrid Modeling in Process Industries*, CRC Press, 129-165.

51. Mahalec, V.; Sanchez, Y. (2012). Inferential Monitoring and Optimization of Crude Separation Units via Hybrid Models. *Computers & Chemical Engineering*. **45**, 15.
52. Peroni, C. V.; Parisi, M.; Chianese, A. (2010). A Hybrid Modelling and Self-Learning System for Dextrose Crystallization Process. *Chemical Engineering Research and Design*. **88,** 1653.
53. Xiong, Z.; Zhang, J. (2005). A Batch-to-Batch Iterative Optimal Control Strategy Based on Recurrent Neural Network Models. *Journal of Process Control*. **15**, 11.
54. Zhang, S.; Chu, F.; Deng, G.; Wang, F. (2019). Soft Sensor Model Development for Cobalt Oxalate Synthesis Process Based on Adaptive Gaussian Mixture Regression. *IEEE Access*, **7**, 118749.
55. Nentwich, C.; Winz, J.; Engell, S. (2019). Surrogate Modeling of Fugacity Coefficients Using Adaptive Sampling. *Industrial & Engineering Chemistry Research*, **58**, 18703.
56. Kunde, C.; Keßler, T.; Linke, S.; McBride, K.; Sundmacher, K.; Kienle, A. (2019).Surrogate Modeling for Liquid–Liquid Equilibria Using a Parameterization of the Binodal Curve. *Processes*. **7**, 753.
57. Côte, M.; Grandjean, B. P.; Lessard, P.; Thibault, J. (1995). Dynamic Modelling of the Activated Sludge Process: Improving Prediction Using Neural Networks. *Water Research*. **29**, 995.
58. Hwang, T. M.; Oh, H.; Choi, Y. J.; Nam, S. H.; Lee, S.; Choung, Y. K. (2009). Development of a Statistical and Mathematical Hybrid Model to Predict Membrane Fouling and Performance. *Desalination*. **247**, 210.
59. Piron, E.; Latrille, E.; Rene, F. (1997). Application of artificial neural networks for crossflow microfiltration modelling: "black-box" and semi-physical approaches. *Computers & Chemical Engineering*. **21**, 1021.
60. Zbiciński, I.; Strumiłło, P.; Kamiński, W. (1996). Hybrid Neural Model of Thermal Drying in a Fluidized Bed. *Computers & Chemical Engineering*. **20**, S695.
61. Cubillos, F. A.; Alvarez, P. I.; Pinto, J. C.; Lima, E. L. (1996). Hybrid-Neural Modeling for Particulate Solid Drying Processes. *Powder Technology*. **87**, 153.
62. Venkatasubramanian, V. (2019). The Promise of Artificial Intelligence in Chemical Engineering: Is it Here, Finally. *AIChE Journal*. **65**, 466.
63. Glassey, J.; Von Stosch, M. editors. (2018). *Hybrid Modeling in Process Industries*. CRC Press. Boca Raton, Florida.
64. Kahrs, O.; Marquardt, W. (2008). Incremental Identification of Hybrid Process Models. *Computers & Chemical Engineering*, **32**, 694.
65. Herwig, C.; Porter, R.; Moller, J., eds. (2021). *Digital Twins: Tools and Concepts for Smart Biomanufacturing*. Springer, New York.
66. Chan, W. K.; Fischer, B.; Varvarezos, D.; Rao, A.; Zhao, H. Inventors; Aspen Technology Inc, assignee. (2020). Asset Optimization Using Integrated Modeling, Optimization, and Artificial Intelligence. United States patent application US 16/434,793.
67. Beck, R.; Munoz, G. (2020). Hybrid Modeling: AI and Domain Expertise Combine to Optimize Assets., https://www.aspentech.com/en/resources/white-papers/hybrid-modeling-ai-and-domain-expertise-combine-to-optimize-assets/?src=blog-global-wpt
68. Galvanauskas, V.; Simutis, R.; Lübbert, A. (2004). Hybrid Process Models for Process Optimization, Monitoring and Control. *Bioprocess and Biosystems Engineering*. **26**, 393.
69. Tian, Y.; Zhang, J.; Morris, J. (2001). Modeling and Optimal Control of a Batch Polymerization Reactor Using a Hybrid Stacked Recurrent Neural Network Model. *Industrial & Engineering Chemistry Research*. **40**, 4525.
70. Su, H. T.; McAvoy, T. J.; Werbos, P. (1992). Long-term Predictions of Chemical Processes Using Recurrent Neural Networks: A Parallel Training Approach. *Industrial & Engineering Chemistry Research*. **31**, 1338.

71. Hermanto, M. W.; Braatz, R. D.; Chiu, M. S. (2011). Integrated Batch-to-Batch and Nonlinear Model Predictive Control for Polymorphic Transformation in Pharmaceutical Crystallization. *AIChE Journal*. 57, 1008.
72. Ghosh, D.; Hermonat, E.; Mhaskar, P.; Snowling, S.; Goel, R. (2019). Hybrid Modeling Approach Integrating First-Principle Models with Subspace Identification. *Industrial & Engineering Chemistry Research*. **58**, 13533.
73. Ghosh, D.; Moreira, J.; Mhaskar, P. (2021). Model Predictive Control Embedding a Parallel Hybrid Modeling Strategy. *Industrial & Engineering Chemistry Research*. **60**, 2547.
74. Hanachi, H.; Yu, W.; Kim, I. Y.; Liu, J.; Mechefske, C. K. (2019). Hybrid Data-Driven Physics-Based Model Fusion Framework for Tool Wear Prediction. *The International Journal of Advanced Manufacturing Technology*. **101**, 2861.
75. Babanezhad, M.; Behroyan, I.; Nakhjiri, A. T.; Marjani, A.; Rezakazemi, M.; Shirazian, S. (2020). High-Performance Hybrid Modeling Chemical Reactors Using Differential Evolution Based Fuzzy Inference System. *Scientific Reports*. **10**, 1-1.
76. Krippl, M.; Dürauer, A.; Duerkop, M. (2020). Hybrid Modeling of Cross-Flow Filtration: Predicting the Flux Evolution and Duration of Ultrafiltration Processes. *Separation and Purification Technology*. **248**, 117064.
77. Mantovanelli, I. C.; Rivera, E. C.; Da Costa, A. C.; Maciel, F. R. (2007). Hybrid Neural Network Model of an Industrial Ethanol Fermentation Process Considering the Effect of Temperature. *Applied Biochemistry and Biotechnology*. **137**, 817.
78. Sharma, N.; Liu, Y. A. (2019). 110th Anniversary: An Effective Methodology for Kinetic Parameter Estimation for Modeling Commercial Polyolefin Processes from Plant Data Using Efficient Simulation Software Tools. *Industrial & Engineering Chemistry Research*. **58**, 14209.
79. Bangi, M. S.; Kwon, J. S. (2020) Deep Hybrid Modeling of Chemical Process: Application to Hydraulic Fracturing. *Computers & Chemical Engineering*. **134**,106696.
80. Bhutani, N.; Rangaiah, G. P.; Ray, A. K. (2006). First-Principle, Data-Based, and Hybrid Modeling and Optimization of an Industrial HydrocrackingUnit. *Industrial & Engineering Chemistry Research*. **45**,7807.
81. Song, W.; Du, W.; Fan, C.; Yang, M.; Qian, F. (2021). Adaptive Weighted Hybrid Modeling of Hydrocracking Process and Its Operational Optimization. *Industrial & Engineering Chemistry Research*. 60, 3617.
82. Lima, P. V.; Saraiva, P. M.; Group, G.P. (2007). A Semi-Mechanistic Model Building Framework Based on Selective and Localized Model Extensions. *Computers & Chemical Engineering*. **31**, 361.
83. Breiman L. (2001). Random Forests. *Machine Learning*. 45, 5.
84. Savkovic-Stevanovic, J. (1996). Neural Net Controller by Inverse Modeling for a Distillation Plant. *Computers & Chemical Engineering*. **20**, S925.
85. Tomba E, Barolo M, García-Muñoz S. (2014). In-Silico Product Formulation Design through Latent Variable Model Inversion. *Chemical Engineering Research & Design*, **92**, 534.
86. Bayer, B.; von Stosch, M.; Striedner, G.; Duerkop, M. (2020). Comparison of Modeling Methods for DoE-Based Holistic Upstream Process Characterization. *Biotechnology Journal*. **15**, 1900551.
87. Raccuglia, P.; Elbert, K. C.; Adler, P. D.; Falk, C.; Wenny, M. B.; Mollo, A.; Zeller, M; Friedler, S. A.; Schrier, J.; Norquist, A. J. (2016). Machine-Learning-Assisted Materials Discovery Using Failed Experiments. *Nature*. **533**, 73.
88. Liao, T. W.; Li, G. (2020). Metaheuristic-Based Inverse Design of Materials–A Survey. *Journal of Materiomics*. **6**, 414.
89. Zhou, Z. H. (2012). *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC Machine Learning.

90. MacGregor, J. F.; Skagerberg, B.; Kiparissides, C. (1992). Multivariate Statistical Process Control and Property Inference Applied to Low Density Polyethylene Reactors. In *Advanced Control of Chemical Processes. IFAC Symposia Series*. 155-159.
91. Rogers, A.; Ierapetritou, M. (2015). Feasibility and Flexibility Analysis of Black-Box Processes. Part 1: Surrogate-Based Feasibility Analysis. *Chemical Engineering Science*. **137**, 986.
92. Rogers, A.; Ierapetritou, M. (2015). Feasibility and Flexibility Analysis of Black-Box Processes. Part 2: Surrogate-Based Flexibility Analysis. *Chemical Engineering Science*. **137**, 1005.
93. Abdullah, F.; Wu, Z.; Christofides, P. D. (2021). Data-Based Reduced-Order Modeling of Nonlinear Two-Time-Scale Processes. *Chemical Engineering Research and Design.* **166**, 1.
94. Agarwal, A.; Biegler, L. T.; Zitney, S. E. (2009). Simulation and Optimization of Pressure Swing Adsorption Systems Using Reduced-Order Modeling. *Industrial & Engineering Chemistry Research.* **48**, 2327.
95. Schäfer, P.; Caspari, A.; Kleinhans, K.; Mhamdi, A.; Mitsos, A. (2019). Reduced Dynamic Modeling Approach for Rectification Columns Based on Compartmentalization and Artificial Neural Networks. *AIChE Journal*. **65**, e16568.
96. Kumari, P.; Bhadriraju, B.; Wang, Q.; Kwon, J.S. (2021). Development of Parametric Reduced-Order Model for Consequence Estimation of Rare Events. *Chemical Engineering Research and Design*. **169**, 142.
97. Narasingam, A.; Kwon, J. S. (2017). Development of Local Dynamic Mode Decomposition with Control: Application to Model Predictive Control of Hydraulic Fracturing. *Computers & Chemical Engineering*. **106**, 501.
98 Zhang, J.; Yin, J.; Wang, R. (2020). Basic Framework and Main Methods of Uncertainty Quantification. *Mathematical Problems in Engineering*. https://doi.org/10.1155/2020/6068203.
99. Duong, P. L.; Ali, W.; Kwok, E.; Lee, M. (2016). Uncertainty Quantification and Global Sensitivity Analysis of Complex Chemical Process Using a Generalized Polynomial Chaos Approach. *Computers & Chemical Engineering*. **90**, 23.
100. Francis-Xavier, F.; Kubannek, F.; Schenkendorf, R. (2021). Hybrid Process Models in Electrochemical Syntheses under Deep Uncertainty. *Processes*. **9**,704.
101. Proppe, J.; Husch, T.; Simm, G. N.; Reiher, M. (2017). Uncertainty Quantification for Quantum Chemical Models of Complex Reaction Networks. *Faraday Discussions*. **195**, 497.
102. Parks, H.L.; McGaughey, A. J.; Viswanathan, V. (2019). Uncertainty Quantification in First-Principle Predictions of Harmonic Vibrational Frequencies of Molecules and Molecular Complexes. *Journal of Physical Chemistry C*. **123**, 4072.
103. Wang, S.; Pillai, H. S.; Xin, H. (2020). Bayesian Learning of Chemisorption for Bridging the Complexity of Electronic Descriptors. *Nature Communications*, **11**, 6132.
104. Boukouvala, F.; Ierapetritou, M. G. (2012). Feasibility Analysis of Black-Box Processes Using an Adaptive Sampling Kriging-Based Method. *Computers & Chemical Engineering*. **36**, 358.
105. Ghenis, M. Quantile Regression- From Linear Regression to Deep Learning. https://towardsdatascience.com/quantile-regression-from-linear-models-to-trees-to-deep-learning-af3738b527c3, accessed August 5, 2021
106. Rudy, S. H.; Brunton, S. L.; Proctor, J. L.; Kutz, J. N. (2017).Data-Driven Discovery of Partial Differential Equations. *Science Advances*. **3**, e1602614.

107. Langley, P.; Bradshaw, G. L.; Simon, H. A. (1983). Rediscovering Chemistry with the BACON System. In: Michalski, R.S.; Carbonell, J.G.; Mitchell, T.M. (eds) *Machine Learning*. *Symbolic Computation*. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-12405-5_10Springer, Berlin, Heidelberg. 307.

108. Nentwich, C.; Engell, S. (2019). Surrogate Modeling of Phase Equilibrium Calculations Using Adaptive Sampling. *Computers & Chemical Engineering*. **126**, 204.

109. Brunton, S. L.; Proctor, J. L.; Kutz, J. N. (2016). Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems. *Proceedings of the National Academy of Sciences*. **113**, 3932.

110. Wang, S. H.; Pillai, H. S.; Wang, S.; Achenie, L. E.; Xin, H. (2021). Infusing Theory into Machine Learning for Interpretable Reactivity Prediction. arXiv preprint arXiv:2103.15210.

111. Chen, R. T., Rubanova, Y.; Bettencourt, J.; Duvenaud, D. (2018). Neural Ordinary Differential Equations. arXiv preprint arXiv:1806.07366.

112. De Jaegher, B.; Larumbe, E.; De Schepper, W.; Verliefde, A.; Nopens, I. (2020). Colloidal Fouling in Electrodialysis: A Neural Differential Equations Model. *Separation and Purification Technology*. **249**, 116939.

113. Wu, Z.; Rincon, D.; Christofides, P. D. (2020). Process Structure-Based Recurrent Neural Network Modeling for Model Predictive Control of Nonlinear Processes. *Journal of Process Control*. **89**, 74.

114. Reis, M. S.; Gins, G.; Rato, T. J. (2019). Incorporation of Process-Specific Structure in Statistical Process Monitoring: A Review. *Journal of Quality Technology*. **51**, 407.

115. Hayashi, Y.; Buckley, J. J.; Czogala, E. (1993). Fuzzy Neural Network with Fuzzy Signals and Weights. *International Journal of Intelligent Systems*. **8**, 527.

116. Brown, M.; Harris, C. J. (1994). *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall, Hoboken, New Jersey.

117.Simutis, R.; Havlik, I.; Schneider, F.; Dors, M.; Lübbert, A. (1995). Artificial Neural Networks of Improved Reliability for Industrial Process Supervision. *IFAC Proceedings Volumes*. **28**, 59.

118. Simutis, R.; Lübbert, A. (2015). Bioreactor Control Improves Bioprocess Performance. *Journal of Biotechnology.* **10**, 1115.

119. Schubert, J.; Simutis, R.; Dors, M.; Havlik, I.; Lübbert, A. (1994). Bioprocess Optimization and Control: Application of Hybrid Modelling. *Journal of Biotechnology*. *35*, 51.

120. Bhadriraju, B.; Narasingam, A.; Kwon, J. S. (2019). Machine Learning-Based Adaptive Model Identification of Systems: Application to a Chemical Process. *Chemical Engineering Research and Design*. **152**, 372.

121. Bhadriraju, B.; Bangi, M. S.; Narasingam, A.; Kwon, J. S. (2020). Operable Adaptive Sparse Identification of Systems: Application to Chemical Processes. *AIChE Journal*. **66**, e16980.

122. Lu, J.; Yao, K.; Gao, F. (2009). Process Similarity and Developing New Process Models through Migration. *AIChE Journal*. **55**, 2318.

123. Yan, W.; Hu, S.; Yang, Y.; Gao, F.; Chen, T. (2011). Bayesian Migration of Gaussian Process Regression for Rapid Process Modeling and Optimization. *Chemical Engineering Journal*. **166**, 1095.

124. Kumar, A.; Ridha, S.; Narahari, M.; Ilyas, S. U. (2021). Physics-Guided Deep Neural Network to Characterize Non-Newtonian Fluid Flow for Optimal Use of Energy Resources. *Expert Systems with Applications.* **19**, 115409.

125. Pun G. P.; Batra, R.; Ramprasad, R.; Mishin, Y. (2019). Physically Informed Artificial Neural Networks for Atomistic Modeling of Materials. *Nature Communications*. **10**, 1.

126. Fischer, C. C.; Tibbetts, K. J.; Morgan, D.; Ceder, G. (2006). Predicting Crystal Structure by Merging Data Mining with Quantum Mechanics. *Nature Materials*. **5**, 641.

127. Hautier, G.; Fischer, C. C.; Jain, A.; Mueller, T.; Ceder, G. (2010). Finding Nature's Missing Ternary Oxide Compounds Using Machine Learning and Density Functional Theory. *Chemistry of Materials*. **22**, 3762.

128. Cang, R.; Li, H.; Yao, H.; Jiao, Y.; Ren, Y. (2018). Improving Direct Physical Properties Prediction of Heterogeneous Materials from imaging Data via Convolutional Neural Network and a Morphology-Aware Generative Model. *Computational Materials Science*. **150**, 212.

129. Mofar, W.; Baker, L. (2021). Aspen Technology On-Demand Seminar: StreaMLine Concurrent Simulation Scenarios to Solve Problems Faster Using Aspen Multi-Case. https://www.aspentech.com/en/resources/on-demand-webinars/streaMLine-concurrent-simulation-scenarios-to-solve-problems-faster, Accessed May 14, 2022.

130. Aspen Technology, Inc. (2022). What Is New in AI Model Builder. AspenTech online help. https://aimodelbuilder.aspentech.ai/assets/AspenAIModelBuilderHelp/HybridModelingApplication.htm#htML/whatsnew.htm?TocPath=_____2, Accessed May 14, 2022.

131. Sharma, N., & Liu, Y. A. (2022). A hybrid science-guided machine learning approach for modeling chemical processes: A review. *AIChE Journal*, *68*(5), e17609. https://doi.org/10.1002/aic.17609

132. Nguyen, X. D. J., Sharma, N., Liu, Y. A., Lee, Y., & McDowell, C. C. (2023). Analyzing the occurrence of foaming in batch fermentation processes using multiway partial least square approaches. *AIChE Journal*, *69*(12), e18250. https://doi.org/10.1002/aic.18250

133. Liu, Y. A., & Sharma, N. (2023). *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing*. Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831

134. Liu, Y. A., & Sharma, N. (2023). Introduction to Integrated Process Modeling, Advanced Control, and Data Analytics in Optimizing Polyolefin Manufacturing. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 1, pp. 1-40). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch1

135. Liu, Y. A., & Sharma, N. (2023). Selection of Property Methods and Estimation of Physical Properties for Polymer Process Modeling. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 2, pp. 41-86). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch2

136. Liu, Y. A., & Sharma, N. (2023). Reactor Modeling, Convergence Tips, and Data-Fit Tool. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 3, pp. 87-114). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch3

137. Liu, Y. A., & Sharma, N. (2023). Free Radical Polymerizations: LDPE and EVA. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 4, pp. 115-162). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch4

138. Liu, Y. A., & Sharma, N. (2023). Ziegler–Natta Polymerization: HDPE , PP , LLDPE, and EPDM. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 5, pp. 163-265). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch5

139. Liu, Y. A., & Sharma, N. (2023). Free Radical and Ionic Polymerizations: PS and SBS Rubber. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 6, pp. 267-319). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch6

140. Liu, Y. A., & Sharma, N. (2023). Improved Polymer Process Operability and Control Through Steady-State and Dynamic Simulation Models. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 7, pp. 321-379). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch7

141. Liu, Y. A., & Sharma, N. (2023). Model-Predictive Control of Polyolefin Processes. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 8, pp. 381-476). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch8

142. Liu, Y. A., & Sharma, N. .2023. Application of Multivariate Statistics to Optimizing Polyolefin Manufacturing. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 9, pp. 477-531). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch9

143. Liu, Y. A., & Sharma, N. (2023). Applications of Machine Learning to Optimizing Polyolefin Manufacturing. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 10, pp. 553-650). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch10.

144. Liu, Y. A., & Sharma, N. (2023). A Hybrid Science-Guided Machine Learning Approach for Modeling Chemical and Polymer Processes. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 11, pp. 651-698). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch11

145. Sharma, N. and Liu, Y., 2019, November. Polyolefin Process Modeling and Monitoring. In *2019 AIChE Annual Meeting*. AIChE.

146. Sharma, N. and Liu, Y., 2020, November. Polyolefin Process Improvement Using Machine Learning. In *2020 Virtual AIChE Annual Meeting*. AIChE

147. Sharma, N., 2022, November. Polyolefin Property Estimation using Process Modeling and Machine Learning in Industry. In *2022 AIChE Annual Meeting*. AIChE.