Machine Learning with Applications 3 (2021) 100014

Contents lists available at ScienceDirect

Machine Learning with Applications

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

White-box Machine learning approaches to identify governing equations for overall dynamics of manufacturing systems: A case study on distillation column

Renganathan Subramanian [a], Raghav Rajesh Moar [a], Shweta Singh [b,c,*]

[a] Department of Chemical Engineering, Indian Institute of Technology, Madras, TN, India
[b] Agricultural & Biological Engineering, Purdue University, West Lafayette, IN, USA
[c] Environmental and Ecological Engineering, Purdue University, West Lafayette, IN, USA

ARTICLE INFO

Keywords:
Machine learning
ASPEN dynamics
Distillation column
SINDy
Dynamic equation
Symbolic regression
Genetic programming

ABSTRACT

Dynamical equations form the basis of design for manufacturing processes and control systems; however, identifying governing equations using a mechanistic approach is tedious. Recently, Machine learning (ML) has shown promise to identify the governing dynamical equations for physical systems faster. This possibility of rapid identification of governing equations provides an exciting opportunity for advancing dynamical systems modeling. However, applicability of the ML approach in identifying governing mechanisms for the dynamics of complex systems relevant to manufacturing has not been tested. We test and compare the efficacy of two white-box ML approaches (SINDy and SymReg) for predicting dynamics and structure of dynamical equations for overall dynamics in a distillation column. Results demonstrate that a combination of ML approaches should be used to identify a full range of equations. In terms of physical law, few terms were interpretable as related to Fick's law of diffusion and Henry's law in SINDy, whereas SymReg identified energy balance as driving dynamics.

1. Introduction

Mathematical models have formed the foundation of chemical engineering and manufacturing systems for several decades in order to design, optimize and control the processes (Rasmuson et al., 2014). Among these models, identification of system dynamics is a critical component, that mainly relies on first principal methods to understand the underlying mechanisms driving dynamics. With the increased focus on reducing greenhouse gas emissions to combat climate change, there is an increased push towards adopting novel manufacturing processes such as biobased or waste recycling. In order to design and safely operate these novel systems, understanding the mechanisms driving the dynamics is crucial.

However, discovering governing dynamical equations from first principles is difficult and slow as we need to simplify the model to make it more tractable while ensuring that the skeletal form still preserves the salient features of the complete model (Carrier, 1967; Wollkind & Dichone, 2018). This can sometimes take many decades to establish equations such as in fluid dynamical equations (Lin & Segel, 1988), which can delay the adoption of novel manufacturing units. Sometimes, the system is highly complex and deriving a single governing dynamical equation for the whole system seems infeasible such as in the distillation column units (Kumar et al., 1983). Further, system identification of nonlinear systems poses another set of challenges where mainly black box models such as artificial neural networks have been used (Krishnapura & Jutan, 1997; Prasad & Bequette, 2003). While these models can give a good predictive model for state variables, it does not provide insights into the mechanisms driving the dynamics. With the existing challenges of system identification using first principle approach and lack of insights into mechanisms using black-box models, data-driven approach such as machine learning (ML) using white-box models to identify governing dynamical equations have been proposed recently. This data-driven discovery of governing equations has been shown to perform well for extracting the equations of fluid dynamics such as NVS (Raissi & Karniadakis, 2018), Lorenz systems (Brunton et al., 2016), chemical reaction kinetics (Hoffmann et al., 2019) networks and Burgers equation (Rudy et al., 2019) or also used for extracting reduced kinetic equations that can capture whole dynamics (Harirchi et al., 0000). Different classes of ML algorithm that have been used are symbolic regression (Bongard & Lipson, 2007; Schmidt & Lipson, 2009), sparse identification of nonlinear dynamics based on lasso regression (Schaeffer et al., 2013), neural networks informed by physics (Lee et al., 2018a), equation-free modeling (Bindal et al., 2006), genetic programming (Koza, 1992) etc. The recent studies

* Corresponding author at: Agricultural & Biological Engineering, Purdue University, West Lafayette, IN, USA.
E-mail addresses: ch16b058@smail.iitm.ac.in (R. Subramanian), ch16b045@smail.iitm.ac.in (R.R. Moar), singh294@purdue.edu (S. Singh).

https://doi.org/10.1016/j.mlwa.2020.100014
Received 3 October 2020; Received in revised form 4 November 2020; Accepted 24 November 2020
Available online 4 December 2020

that have shown promising results on identifying governing equations for dynamics of physical systems from fast data-driven approach may also prove beneficial to identify governing dynamical equations for complex manufacturing units. Therefore, we propose to use white-box ML approach to identify governing dynamical equations for complex manufacturing systems.

For the white-box ML algorithms, a basic guiding principle is to relate the depiction of dynamics using the state space representation of each system variables as a combination of variables to capture the dynamics. So far, most of these algorithms have been tested on well-known equations (such as Lorenz, NVS etc.) or for extraction of dynamics in cases where the mechanism is easy to elucidate (such as rate kinetics dependent on the law of mass action in Hoffmann et al., 2019). In this work, we demonstrate the efficacy of such data-driven ML approach on a complex manufacturing unit of distillation column, which is a ubiquitous unit in chemical, pharmaceutical and food manufacturing systems. The system studied is an extractive distillation column for binary separation. We test two different white box ML algorithms: SINDy and SymReg and compare the identified governing dynamical equations for prediction accuracy and interpreting the terms for mechanisms that drive dynamics. The standard procedure of cross-validation is followed to identify the best model, and the model is tested for accuracy using Root Mean Square Error (RMSE). The results from prediction of dynamics are promising, as both SINDy and SymReg show good predictions. We also test different scenarios (such as changing the operation parameters for distillation column), in order to test the robustness of algorithms, which also shows promising results. Finally, in terms of mechanism elucidation, we find that SINDy and SymReg capture different mechanisms due to the nature of operation of these methods. Hence, we propose that a combination of white-box ML approaches should be used to identify governing equations for unknown physical systems and use the results to narrow down physical testing to verify the mechanisms. This approach will result in fine-tuning and cost savings for designing experiments to verify physical laws governing the dynamics of unknown systems.

Rest of the paper is organized as follows: In Section 2, we discuss related work on modeling the dynamics of distillation column and examples of ML approach for identification of governing dynamics. In Section 3, we explain our methodology and the white-box ML approaches — SINDy and SymReg. Next in Section 4, we explain the set up of the physical system and simulations using dynamic process flow simulation of the extractive distillation column built in ASPEN dynamics to generate data set followed by details of model training and selection in Section 5. In Section 6, we discuss our results and observations from testing, along with explanations on the differences for SINDy and SymReg for identified equations. We finally discuss the key takeaways and the prospects for future research on extending white-box ML approaches for manufacturing systems in Section 7.

2. Related work on modeling dynamics of distillation column and machine learning applications for dynamics

A review on extractive distillation by Gerbaud(Gerbaud et al., 2019), explains the current status of process design, operation, optimization and control of extractive distillation columns. Dynamics in distillation columns arise primarily due to liquid and vapor holdups (effected by fluid dynamics or pressure dynamics), tray hydraulics and changes in physical properties (effected by thermodynamics). A rigorous model for distillation columns dynamics was developed by (Gani et al., 1986). This generalized dynamical model is derived from first principles and thoroughly models tray hydraulics and holdups. It involves a set of non-linear ordinary differential equations written for every tray based on material and energy balances, vapor–liquid equilibrium models, Murphee tray efficiency coefficient, Bernoulli equation for friction losses, tray hydraulics models (Bolles, 1988; Lockett & Banik, 1986; Stichlmair & Hofer, 1978), froth density correlation (Bennett et al., 1983), flow over weir given by the modified Francis weir formula (Green & Perry, 2007) and pressure drop correlations derived (Richardson et al., 1983). An alternate rate-based model developed by Retzbach (1986) (Retzbach, 1986) has mass transfer correlations, hydraulic and pressure drop correlations. This models multi-component mixtures and is used extensively in simulations. These highly complex models describe the dynamics precisely and is capable of making accurate predictions. However, these models involve simultaneously solving nonlinear differential equations, numerical integrations, parameter fitting and root-finding methods and present challenges for use in real-time predictions similar to the CFD limitations (Ding et al., 2019; Skogestad, 1992). Overviews of different rigorous models developed with different underlying assumptions are available in (McAvoy & Wang, 1986; Rademaker, 1975; Rosenbrock, 1962; Tolliver & Waggoner, 1980). Some simplifications have been used for distillation dynamics (Berber & Karadurmus, 1989; Choe & Luyben, 1987; Gani et al., 1986; Kapoor & McAvoy, 1988; Pantelides et al., 1988), however these models are not very accurate and give large deviations, especially in low pressure (vacuum) and high-pressure columns (Choe & Luyben, 1987).

Recently with the advent in machine learning, non-parametric models have been developed for the distillation process. These data-driven models use input–output data to identify the system without using first principles and can thus be made less complex. ANN (artificial neural networks) are increasingly being employed for this purpose (Macmurray & Himmelblau, 1995). Further, these data driven models have been shown to perform better than simplified models in terms of predictive capabilities and are computationally faster than rigorous models due to their parallelizability. These methods also do not require detailed domain-specific theory and assumptions. Unlike the previous models which are built either for individual trays or by treating groups of trays as sections, ANN models can be built for the entire column also (Singh et al., 2005, 2007). Other recent applications and potential of using machine learning approaches in chemical engineering have also been discussed by Lee et al. (2018b) and Venkatsubramanian (Venkatasubramanian, 2019). However, data-driven approaches using ANN, yield black box models which, despite having very good predictive capabilities do not provide much insight about the underlying mechanisms driving dynamics, thus limiting insights into design for improvement. This might be a drawback in systems which evolve or whose parameters (which were not modeled but kept as constants during the training phase) change. The developed model, unlike in the case of first principles becomes obsolete and needs to be developed again from scratch. Hence, there is a trade-off between utilizing first principle based dynamical equations and black-box data driven models which can be overcome by using the white-box ML approach such as SINDy (Brunton et al., 2016) or Symbolic Regressions (Zames et al., 1981) that can provide underlying equations governing dynamics. However, how good these white-box ML approaches are in elucidating the basic mechanisms driving dynamics of manufacturing systems is still an open question which we address in this paper. The white-box ML approach is expected to fill the need for creating models capturing dynamics which are (1) simple and computationally inexpensive (2) have good predicting abilities (capture the complexities of the process) (3) provide insights about the governing mechanisms for dynamical predictions (4) require little domain knowledge during the development phase and (5) can be modified fast according to the changes in the system, which is crucial for evolving manufacturing systems. SINDy method has been shown to perform extremely well for systems such as fluid dynamics/chemical kinetics (Brunton et al., 2016; Hoffmann et al., 2019) and can balance model complexity with model accuracy while SymReg method also generates models without any assumptions on model structure and has recently been successfully applied on prediction of continuous (/mechanical) dynamical systems (Gout et al., 2018; Quade et al., 2016). The strengths of these approaches motivates our study to develop an approach for using these algorithms on manufacturing systems and also test their efficacy. We explain these methods in Sections 3.1 and 3.2 .

3. Methodology

In order to extract the governing equations for overall dynamics of the distillation column, we follow a hybrid methodology. We chose two white box machine learning approaches, SINDy and SymReg, and applied these techniques on time series data generated for distillation column in ASPEN dynamics. Hence, our methodology utilizes mechanistic models to generate time series data and ML methods to construct governing equations depicting the mechanistic dynamics. The results from two different ML approaches are then compared for performance and identification of mechanisms driving the dynamics. We first describe the ML algorithms SINDy (Section 3.1) and SymReg (Section 3.2) followed by simulation details of distillation column dynamics (Section 4) and model selection in Section 5. In this work, we do not modify the original SINDy and SymReg algorithm because our goal was to test the hypothesis of the efficacy of these algorithms in extracting underlying mechanisms that govern dynamics for systems with unknown governing principles.

3.1. Sparse identification of non-linear dynamics: White box machine learning approach 1

Sparse Identification of Non-Linear Dynamics or SINDy is a sparse regression based ML methodology that works under the assumption that the governing equations of most dynamical systems can be described in sparse dimensions that can capture the essential dynamics of the complex system. Hence, these equations can be considered sparse in the function space, and the system is expected to evolve on a low dimensional manifold. SINDy algorithm then utilizes an optimization approach to identify these equations from time series data. Here, we consider systems whose governing equations are non-linear ODEs of the form given in Eq. (1).

$$\frac{\mathrm{d}(\mathbf{x}(\mathbf{t}))}{\mathrm{dt}} = \mathrm{f}(\mathbf{x}(\mathbf{t}), \mathrm{u}(\mathrm{t})) \tag{1}$$

In Eq. (1), $\mathbf{x}(\mathbf{t}) \in \mathbb{R}^n$ denotes the state of the system at time t, $\mathrm{u}(t) \in \mathrm{R}$ is the external forcing variable at time t and $\mathrm{f}(\mathbf{x}(\mathrm{t}), \mathrm{u}(\mathrm{t}))$ is a linear combination of non-linear functions of $\mathrm{x}(\mathrm{t})$ and $\mathrm{u}(\mathrm{t})$. Hence, the equation obtained for capturing the dynamics is given by Eq. (2), where θs are non-linear functions called the candidate terms of $f(x,u)$, and ξs are the coefficients of the terms. The state of the system is given by $\mathbf{x}(\mathrm{t})$ in Eq. (3) where $x_1, \ldots x_n$ are the n states of the system corresponding to n state variables.

$$\dot{\mathbf{x}}_\mathbf{t} = \Sigma_{i=1}^{k} \xi_i \theta_i(\mathbf{x}(t), u(t)) \tag{2}$$

$$\mathbf{x}(\mathbf{t}) = \begin{bmatrix} x_1(t) & x_2(t) & \ldots x_n(t) \end{bmatrix} \tag{3}$$

SINDy algorithm builds on the assumption that most of ξs in Eq. (2) can be 0 if the system displays dynamics that can be captured by a few functions. Hence, the goal of algorithm is to identify the very few non-zero coefficients for terms which make up $f(x,u)$ from a very large set of candidates. To obtain this sparse set of functions, the SINDy algorithm utilizes a penalty for model complexity instead of brute forces combinatorial search, thus ensuring not to overfit the data. As a result, the final governing equation is given by Eq. (4).

$$\dot{\mathrm{X}}_\mathrm{t} = \Theta(\mathbf{X}(\mathrm{t}), \mathrm{u}(\mathrm{t}))\Xi \tag{4}$$

where, $\Theta(\mathbf{X}(\mathrm{t}), \mathrm{u}(\mathrm{t})) \in \mathbb{R}^{m \times k}$ and can be expressed as $\begin{bmatrix} \theta_1(\mathbf{X}(\mathrm{t}), \mathrm{u}(\mathrm{t})) & \theta_2(\boldsymbol{X}(\mathrm{t}), \mathrm{u}(\mathrm{t})) \ldots \end{bmatrix}$ and $\Xi \in \mathbb{R}^{k \times n}$ can be expressed as $\begin{bmatrix} \xi_1 & \xi_2 & \ldots \xi_n \end{bmatrix}$, m is the length of time series data, n is the number of state variables and k is the number of non-zero terms.

The sparse matrix Ξ is obtained by solving the least squares optimization problem for n state variables. A regularization term (α) is added to the objective function to implement sparsity. The ideal regularization to force sparsity would be minimizing the L_0 norm of the coefficients (number of non zero terms in the vector). But this an NP-hard problem (K. Natarajan, 1995), hence a "Least Absolute Shrinkage and Selection Operator" or LASSO approach is used which minimizes the L_1 norm and also produces sparse solutions (Donoho & Elad, 2003). The LASSO optimization problem is given by Eq. (5).

$$\xi_i^* = \underset{\xi_i}{\operatorname{argmin}} \left\| \dot{\mathbf{x}}_\mathbf{i} - \Theta\left(\mathbf{X}_\mathbf{t}, \mathrm{u}_\mathrm{t}\right) \xi_i \right\|_2 + \alpha \|\xi_i\|_1 \qquad \mathrm{i} = 1, 2 \ldots n \tag{5}$$

In Eq. (5), α is the regularization parameter which has to be tuned in order to achieve a trade-off between accuracy and sparsity. This optimization problem can be solved by the standard convex optimization algorithms. We have used coordinate descent algorithm which is available as a prebuilt function in the *scikit* Python library. The capability of the algorithm to capture the dynamics of the system depends mainly on the candidate functions, $\Theta(\mathbf{X}(\mathbf{t}), \mathbf{u}(\mathbf{t}))$ provided. Some prior knowledge of the functional form that may govern dynamics of the process might help identify these candidate functions where domain knowledge becomes an important aspect for applying ML appropriately. In case, no prior knowledge is available, various combination of functions such as polynomial, trigonometric etc, can be used and allow the algorithm to identify correct functional representation; however, this is a computationally expensive approach. Fortunately, with the advent of computational power, this is no longer a limiting factor and has led to explorations of data driven approaches for identifying the dynamics of these systems.

3.2. Symbolic Regression: White Box Machine Learning Approach 2

In Symbolic Regression (SymReg), the function is determined using genetic programming, which is an evolutionary algorithm that builds and tests candidate functions out of simple building blocks. That is, unlike SINDY in SymReg, no particular model is provided as a starting point to the algorithm. Instead, initial expressions are formed by randomly combining mathematical building blocks such as constants, mathematical operators and state variables. These functions are then modified according to a set of evolutionary rules and generations of functions are tested until a pre-determined accuracy is achieved or any other criteria of termination are satisfied. As compared to the SINDy approach, SymReg is a bottom up approach that allows the algorithm to build functions without any set rules.

As depicted in Fig. 1 symbolic regression algorithm constructs a population of parse trees, which gradually evolves to optimal algebraic expressions expressing the functional input–output relationship of the data. The equation for capturing the dynamics is given by Eq. (6), where γs are non-linear functions generated from a set of operators such as (+, sin(), cos(), MyFunction(), ...) for $f(x)$ similar to ζ. The state of the system is given by $x(t)$ as in Eq. (3). Notice how there are no ξs in Eq. (6) (as compared to Eq. (2)) because all the generated functions from SymReg are included in the final identified dynamics equation.

Gplearn api in python was used to implement the SymReg optimization problem with fitness measured using root mean square error (rmse) and taking into account a parsimony coefficient of 0.01. (parsimony coefficient is set low because algorithm has tendency to generate simple results). Similar to SINDy, some prior knowledge of the mechanisms that may govern dynamics of the process might help to limit the operator functions to be used in SymReg, where again domain knowledge becomes an important aspect for applying ML appropriately.

$$\dot{\mathbf{x}}_\mathbf{t} = \Sigma_{\mathrm{i}=1}^{k} \gamma_\mathrm{i}(\mathbf{x}(\mathrm{t}), \mathrm{u}(\mathrm{t})) \tag{6}$$

We do not modify SINDy and SymReg in order to test the applicability of original algorithms purely on data without any domain knowledge to identify system dynamics. After the set up of the problem and system, the next step was to obtain appropriate time series data that captures the dynamics of the physical system for which we use mechanistic approach and simulations, described in next section.

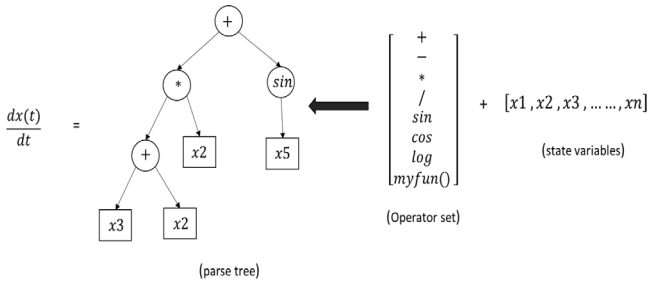


Fig. 1. An example of a parse tree generated for the expression $\dot{x} = x_2^2 + x_3 x_2 + sin(x_5)$.

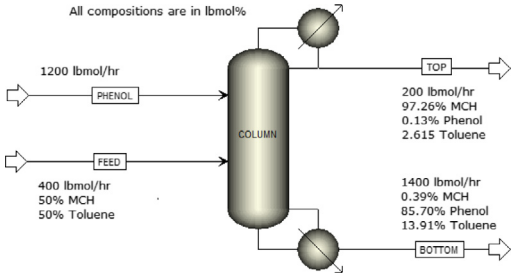


Fig. 2. Process Flow Diagram — Distillation Column.

4. Simulations: Data Generation for applying Machine Learning to Extract Governing Dynamical Equations

4.1. System selection and data set up for machine learning

We selected a simple extractive distillation column shown in Fig. 2 to extract governing equation for distillation dynamics using SINDy and SymReg. The column was modeled to recover methylcyclohexane (MCH) from a mixture of MCH and toulene. Since MCH (Boiling Point = 101° C) and toluene (Boiling Point = 110.6° C) have very close boiling points, these cannot be separated by a conventional distillation column. Therefore, phenol (Boiling Point = 181.7° C) is used which has a higher affinity towards toluene to alter the relative volatility and promote separation. An equimolar mixture of MCH and toluene forming the feed stream and a pure phenol stream are fed to the distillation column. MCH is extracted as the overhead product while toluene and phenol leave as the bottom products. The column was modeled as a RadFrac unit. The specifications of the distillation column used are listed in Appendix A, Table 1, and the feed conditions are given in Appendix A, Table 2.

In order to generate time series data, the ASPEN model for the column was exported to Aspen Dynamics for running dynamic simulations. The first-principle based mechanistic model has 2403 variables and 1848 equations as identified by Aspen Dynamics, however structure of all these equations are not known. The dynamics was captured using a perturbations to the phenol feed rate while the feed mixture flow rate was kept constant. The phenol feed perturbation was implemented by executing a Task in Aspen Dynamics. The perturbation was a random mix of step changes, linear ramps and sigmoidal ramps with a time period of 1 h each and amplitudes between 1000 lbmol/hr to 3000 lbmol/hr generated randomly with a uniform probability distribution function. The simulation was run for 100 h with a calculation step size of 0.01 h (See details in Section 1, Appendix A) . This allows system to show dynamics related to changes in extracting agent flow rate. The goal of SINDy and SymReg was then to extract the governing equations for predicting the dynamics of whole system in response to these perturbations using the time series generated from mechanistic model.

Table 1
Different operating conditions tested for dynamics equation.

Parameter	System 1	System 2	System 3	System 4
Reflux Ratio	6	8	8	8
Toluene Feed	200	200	200	400
MCH Feed	200	200	400	200

Valid range of phenol flow rates were obtained through sensitivity analysis (Appendix A, Fig. 1).

Operating Conditions : In order to define the system, the following variables were fixed as operating condition parameters : Reflux ratio, toluene feed rate, MCH feed rate, distillation column sizing, tray geometry, reboiler geometry and sizing, condenser geometry and sizing, reboiler duty and condenser heat transfer coefficients. These conditions play a crucial role in operation of selected distillation column hence fixing these parameters would allow us to identify the governing equations for mechanisms that drive the dynamics of flow streams. Further, in order to test the robustness of the equations extracted, the structure of the obtained equations were compared across different operating conditions obtained by altering some of these parameters. The different operating conditions tested are listed in Table 1 which forms four different systems for which governing equations have been extracted.

State Variables of the System : To study the dynamics, we selected the set of state variables which change with the perturbations and are not fixed as operating conditions. Hence, from the ML algorithm, the result will be a system of ODEs that can describe the evolution of the whole system as state space dynamics for these variables. For the system under consideration, we initially chose the following variables as state space variables : overhead stream temperature (TOP_T), overhead stream Phenol flow rate (Top_{Ph}), OVERHEAD Stream — Toluene Flow Rate (Top_{Tol}), OVERHEAD Stream — MCH Flow Rate (Top_{MCH}), BOTTOMS Stream — MCH Flow Rate (Bot_{MCH}), BOTTOMS Stream — Phenol Flow Rate ($\mathrm{Bot_{Ph}}$), BOTTOMS Stream — Toluene Flow Rate $\mathrm{Bot_{Tol}}$, BOTTOMS Stream Temperature ($\mathrm{Bot_T}$), Condenser Duty ($\mathrm{Q_{cond}}$), Reboiler Vapor Flow Rate (Vap Reb), Stage 1/Condenser Pressure ($\mathrm{P_1}$), Stage 22/Reboiler Pressure ($\mathrm{P_{22}}$).

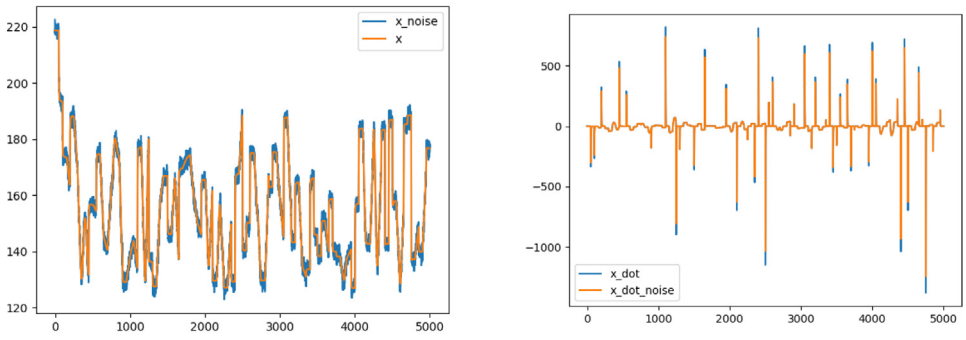
These variables hold significance in terms of column requirements as the equations developed can later be used for studying dynamics of a specific extent of separation, quality of product, ensure pressure in the column within safety limits or estimate energy requirements. Hence, ODEs for these variables will make these applications possible.

4.2. Data pre-processing (noisy data)

Since, performance of both the methods also depends on the quality of the data, we filtered the derivatives and variables to remove any noise by applying numerical differentiation on time series data to obtain the derivatives. In order to test the effect of differentiation on any noise present in data, we added a normally distributed white noise of mean zero and variance based on SNR(Sound to Noise Ratio) of 40 dB to the simulated data so as to simulate sensor error during industrial data collection as seen in Fig. 3(a). But Fig. 3(b) shows that the numerical differentiation using total variance regularization method generated the same differentiated values as the differentiation method was developed to reduce the effect of noise in the data (Chartrand, 2011). Therefore, we perform all the model fitting on original time series data obtained from simulation.

5. Algorithm implementation, model selection and testing

After selection of state variables and generation of data, the SINDy algorithm was implemented in Python 3.6.5 using the libraries - *pandas*, *numpy*, *sklearn*, *scipy*, *matplotlib* and *itertools*. Similarly the SymReg Algorithm was implemented in Python 3.6.5 using the libraries - *pandas*,



(a) simulated data vs noisy data

(b) $\dot{x}$ for simulated vs noisy data

Fig. 3. Removal of noise effect after differentiation : Both Simulated and Noisy Data show same results.

numpy, *sklearn*, *matplotlib*, *gplearn (API Version 0.4.1)*. For both the algorithms we have used numerical differentiation with total variance regularization method developed in (Chartrand, 2011) to obtain the derivatives of the variables. And have split the time series data in the ratio 3:1:1 for training, cross validation (CV) and testing. The function selection input for both algorithms is as follows :

Candidate Functions input to SINDy for Governing Dynamics : Once the state variables were selected, the next input to the SINDy algorithm is the set of candidate functions to capture the dynamics of systems. The state variables were first mean shifted and auto scaled before generating the candidate functions. We used 360 candidate functions of the form given in Eq. (7), which corresponds to second order polynomial function built using x_j^s.

$$\begin{aligned} &f_i = x_1^{a_1^{(i)}} x_2^{a_2^{(i)}} \cdots x_{14}^{a_{14}^{(i)}} \qquad i = 1, 2 \ldots k \\ &\text{where,} \\ &\Sigma_{j=0}^{12} a_j^i \leq 2 \\ &-2 \leq a_j^{(i)} \leq 2 \\ &a_j^{(i)} \in \mathbb{Z} \\ &x_0 : \text{Perturbed Input} \end{aligned} \tag{7}$$

And 70 candidate functions of the form $\sin(x_j)$, $\cos(x_j)$, $ln(|x_j|)$, e^{x_j} and $\sqrt{|x_j|} \forall\ j = 1, 2 \ldots 14$. These functions were chosen without using any strong understanding of the system to check if the algorithm can work with very little to no domain knowledge for choosing functions representing dynamics of system.

Operator set ζ for SymReg Governing Dynamics : In case of SymReg, operators where chosen so as to generate similar kind of candidate functions mentioned above and at the same time cover a broad set of mathematical functional space. Therefore set ζ includes following rules for building functional relationship between variables provided : {'add', 'sub', 'mul', 'div', 'sin', 'cos', 'log', 'sqrt', 'abs', 'neg', 'inv', 'tan' }. Using these mathematical operations, the algorithm builds functions using the variables given to represent the dynamics.

5.1. Model selection

Generally, model selection based on highest Cross Validation Accuracy for white box machine learning algorithm results in models with too many terms making it difficult to interpret their physical meaning and resulting in overfitting. Therefore, a penalty for number of terms is added to balance between accuracy and interpretability.

SINDy Model selection: Inline with the above discussion, a selection score based on model complexity was defined for model selection by SINDy. Based on this fitness score given by Eq. (8), the model with the highest score was selected.

$$\mu ln\left(R_{CV}^2\right) - \lambda k \tag{8}$$

In Eq. (8), λ and μ are parameters for the score function and k denotes the number of terms in the obtained equation and R_{CV}^2 is the cross validation R^2 accuracy. For SINDy, values for both λ and μ are based on AIC (Akaike information criterion) calculated for variables that showed clear elbow plot indicating decline in gain in prediction accuracy for adding more parameters.

SymReg Model selection: In SymReg similar approach to Eq. (8) was used for model selection. Here, the values of λ and μ for the fitness score was determined using the parsimony coefficient parameter that is the API for SymReg calculates internally λ and μ based on parsimony coefficient parameter.

5.2. Model testing and evaluation

The governing equations or models for dynamics extracted from both SINDy and SymReg were tested for (a) accuracy of predicting $\dot{\mathbf{x}}(t)$ given $\mathbf{x}(t)$ and $\mathbf{u}(t)$, (b) comparison of structure of ODEs for capturing the mechanisms of dynamics for 4 different systems simulated and (c) comparing the equations obtained from SINDy and SymReg . The results of these tests along with their interpretations are available in Section 6 and Appendix B. The data selection for testing is described below :

Random Test Data: Tests the accuracy of the developed model on the 20% data selected randomly and excluded from training. This gives an idea about the overfitting and the predictive ability of the model under conditions similar to which the training data was obtained. Low success under this test could indicate overfitting during training of model.

Long Time Accuracy: In this testing, the dynamical system is run for a longer time (250 h) than the training time (100 h) to generate test data. This will help identify long time dynamic effects or time based evolution of the system which could have been missed by the algorithm.

Outside Perturbation Region-: In this test, we created a new data set by changing the feed perturbation region and testing the model on this new data. This checks if the model was able to capture the complete mechanism of dynamics of the system so that it can also perform well outside the perturbation region. Low accuracy under this test would indicate incompleteness of the model in terms of missing critical state variables or insufficient candidate functions.

ODE Structural Comparison-: 4 additional system variation were created as mentioned in Table 1 by altering the operating conditions. The model was trained on these 4 systems. The structure of the equations obtained were compared across these 4 systems for similar terms (only for the presence or absence of

Table 2
SINDy: Training and test R^2 values for the systems 2 & 3.

Variable	System 2						System 3					
	Low regularization			High regularization			Low regularization			High regularization		
	Train	Test	N	Train	Test	N	Train	Test	N	Train	Test	N
Top F ($\dot{x_0}$)	0.99	0.986	25	0.979	0.979	14	0.959	0.942	39	0.93	0.897	19
Top T ($\dot{x_1}$)	0.99	0.985	27	0.978	0.977	15	0.965	0.943	36	0.933	0.894	18
Top MCH ($\dot{x_2}$)	0.99	0.986	28	0.979	0.979	14	0.96	0.943	39	0.93	0.898	19
Top Ph ($\dot{x_3}$)	0.282	0.315	6	0.282	0.315	6	0.435	0.389	50	0.377	0.358	40
Top Tol ($\dot{x_4}$)	0.962	0.974	25	0.959	0.969	18	0.954	0.924	41	0.866	0.712	9
Bot T ($\dot{x_5}$)	0.972	0.966	50	0.854	0.835	19	0.875	0.774	60	0.748	0.669	23
Bot MCH ($\dot{x_6}$)	0.975	0.977	45	0.827	0.815	22	0.856	0.772	63	0.698	0.58	22
Bot Ph ($\dot{x_7}$)	0.904	0.871	57	0.718	0.632	23	0.795	0.755	78	0.544	0.495	23
Bot Tol ($\dot{x_8}$)	0.873	0.722	50	0.723	0.5	15	0.77	0.769	63	0.58	0.53	21
Cond Q ($\dot{x_9}$)	0.972	0.956	36	0.958	0.948	14	0.955	0.926	50	0.909	0.858	20
Vap Reb ($\dot{x_{10}}$)	0.914	0.866	37	0.844	0.783	20	0.861	0.822	64	0.686	0.651	20
P1 ($\dot{x_{11}}$)	0.976	0.96	31	0.963	0.939	14	0.952	0.908	32	0.927	0.862	19
P22 ($\dot{x_{12}}$)	0.965	0.948	30	0.947	0.923	16	0.932	0.902	52	0.873	0.812	18

Table 3
SINDy: Training and test R^2 values for the systems 4 & 1.

Variable	System 4						System 1					
	Low regularization			High regularization			Low regularization			High regularization		
	Train	Test	N	Train	Test	N	Train	Test	N	Train	Test	N
Top F ($\dot{x_0}$)	0.983	0.957	53	0.944	0.921	21	0.989	0.963	37	0.97	0.955	16
Top T ($\dot{x_1}$)	0.985	0.981	52	0.951	0.951	19	0.989	0.966	40	0.964	0.949	16
Top MCH ($\dot{x_2}$)	0.974	0.963	39	0.942	0.926	20	0.989	0.963	36	0.97	0.955	16
Top Ph ($\dot{x_3}$)	0.624	0.605	34	0.625	0.605	34	0.562	0.565	43	0.511	0.554	30
Top Tol ($\dot{x_4}$)	0.97	0.609	33	0.933	0.704	20	0.892	0.826	8	0.892	0.826	8
Bot T ($\dot{x_5}$)	0.878	0.734	70	0.733	0.696	27	0.98	0.889	41	0.892	0.491	18
Bot MCH ($\dot{x_6}$)	0.832	0.7	67	0.649	0.49	24	0.863	0.794	48	0.759	0.66	31
Bot Ph ($\dot{x_7}$)	0.792	0.493	87	0.792	0.493	87	0.832	0.762	48	0.703	0.504	23
Bot Tol ($\dot{x_8}$)	0.803	0.389	88	0.803	0.389	88	0.952	0.866	30	0.923	0.89	14
Cond Q ($\dot{x_9}$)	0.969	0.964	50	0.935	0.93	18	0.966	0.926	50	0.93	0.904	15
Vap Reb ($\dot{x_{10}}$)	0.889	0.875	67	0.758	0.864	38	0.91	0.843	51	0.822	0.677	21
P1 ($\dot{x_{11}}$)	0.976	0.934	53	0.939	0.901	25	0.982	0.97	43	0.968	0.946	20
P22 ($\dot{x_{12}}$)	0.958	0.935	51	0.91	0.908	21	0.977	0.948	38	0.958	0.906	21

Table 4
SymReg: Training and test R^2 values for the 4 systems.

Variable	System 1			System 2			System 3			System 4		
	Train	Test	N	Train	Test	N	Train	Test	N	Train	Test	N
Top F ($\dot{x_0}$)	0.717	0.694	2	0.773	0.760	2	0.493	0.471	2	0.680	0.662	2
Top T ($\dot{x_1}$)	0.715	0.672	2	0.687	0.670	2	0.493	0.472	2	0.692	0.654	2
Top MCH ($\dot{x_2}$)	0.714	0.675	2	0.760	0.730	2	0.494	0.471	2	0.551	0.502	2
Top Ph ($\dot{x_3}$)	0.386	0.463	2	0.558	0.433	2	0.414	0.396	2	0.452	0.345	2
Top Tol ($\dot{x_4}$)	−0.002	−0.002	1	0.001	0.002	1	−0.011	0.001	1	0.006	0.008	1
Bot T ($\dot{x_5}$)	0.000	0.000	1	0.332	0.296	2	0.000	0.000	1	0.198	0.203	2
Bot MCH ($\dot{x_6}$)	0.000	0.000	1	−0.001	0.000	1	0.189	0.198	2	0.109	0.139	2
Bot Ph ($\dot{x_7}$)	0.254	0.213	2	0.201	0.192	2	0.198	0.210	2	0.001	0.001	1
Bot Tol ($\dot{x_8}$)	0.251	0.202	2	0.260	0.198	2	0.333	0.360	2	0.072	0.073	2
Cond Q ($\dot{x_9}$)	0.689	0.623	2	0.621	0.645	2	0.652	0.602	2	0.431	0.441	2
Vap Reb ($\dot{x_{10}}$)	0.452	0.443	2	0.338	0.314	2	0.326	0.363	2	0.479	0.473	2
P1 ($\dot{x_{11}}$)	0.738	0.737	2	0.772	0.7605	2	0.568	0.523	2	0.640	0.655	2
P22 ($\dot{x_{12}}$)	0.728	0.713	2	0.770	0.748	2	0.989	0.963	2	0.622	0.582	2

terms and not for the similarity of regression coefficients). Our test hypothesis was that if the algorithm is able to extract the governing mechanisms for dynamics of the system, irrespective of the operating conditions, the equation would contain the same terms and differ only in the parameter values.

6. Results and discussions

6.1. Derivative predictions : SINDy and symreg in 4 simulated system

Results for testing of extracted DEs to predict $\dot{x}(t)$ using test data $x(t)$ for all 4 simulated systems (Table 1) are shown in Tables 2 and 3 for SINDy and in Table 4 for SymReg. Interpretations are discussed below.

SINDy: In case of SINDy, we trained and tested the models for two values of α, corresponding to low and high regularization. Training and test column for all 4 systems show high accuracy at both high and low regularization, except for the phenol flow rate in the top feed (state variable Top Ph). We also observed that reducing the regularization increases the accuracy in the test data. This trend is seen across variables and till very small regularization parameter values. This indicates that we are unable to capture enough information from the data using the provided candidate functions and number of terms as accuracy keeps on increasing with lowering regularization that leads to increasing number of functions included in the model. This could either indicate insufficient candidate function and state variables or absence of a low dimensional function space representation for the system.

Table 5
SINDy: Long Time and testing outside the training perturbation region.

Variable	Long time		Outside training perturbation	
	Low α	High α	Low α	High α
Top F ($\dot{x_0}$)	0.948	0.945	0.778	0.803
Top T ($\dot{x_1}$)	0.953	0.944	0.704	0.817
Top MCH ($\dot{x_2}$)	0.951	0.946	0.819	0.799
Top Ph ($\dot{x_3}$)	0.198	0.198	−0.588	−0.588
Top Tol ($\dot{x_4}$)	0.516	0.523	0	0.19
Bot T ($\dot{x_5}$)	0.95	0.813	−4.94	−0.477
Bot MCH ($\dot{x_6}$)	0.931	0.76	0	0.13
Bot Ph ($\dot{x_7}$)	0.786	0.585	−9.034	−9.034
Bot Tol ($\dot{x_8}$)	0.75	0.514	−30.744	−30.744
Cond Q ($\dot{x_9}$)	0.949	0.922	0.626	0.789
Vap Reb ($\dot{x_{10}}$)	0.852	0.78	−0.498	−1.424
P1 ($\dot{x_{11}}$)	0.868	0.86	0.24	0.783
P22 ($\dot{x_{12}}$)	0.851	0.877	0	0.726

Table 6
SymReg: Long time and testing outside the training perturbation region.

Variable	Long time	Outside training perturbation
Top F ($\dot{x_0}$)	0.681	0.501
Top T ($\dot{x_1}$)	0.663	0.657
Top MCH ($\dot{x_2}$)	0.650	0.510
Top Ph ($\dot{x_3}$)	0.423	0.466
Top Tol ($\dot{x_4}$)	−0.004	0.000
Bot T ($\dot{x_5}$)	−0.001	0.000
Bot MCH ($\dot{x_6}$)	0.124	0.213
Bot Ph ($\dot{x_7}$)	0.211	0.197
Bot Tol ($\dot{x_8}$)	0.201	0.223
Cond Q ($\dot{x_9}$)	0.489	0.452
Vap Reb ($\dot{x_{10}}$)	0.550	0.482
P1 ($\dot{x_{11}}$)	0.631	0.341
P22 ($\dot{x_{12}}$)	0.638	0.341

Ways to analyze and possibly overcome this are discussed in Section 7. However, high accuracy in predicting dynamics of key state variables such as Top MCH from extracted ODEs even at high regularization was promising if the goal is to build a simplified predictive models for dynamics of key state variables.

SymReg: We see that for SymReg the accuracy is low as compared to SINDy across all variables for all systems because of the simple form of predicted DEs (2 or 1 term in Equation). Therefore, based on R^2 accuracy SymReg is underfitting and SINDy performs better comparatively. This happens because of (1) the spiky nature (close to zero at most places and with sudden high values in between) of the derivatives which results in difficult learning for machine learning algorithm and (2) learning difference of SymReg and SINDy as discussed in Section 6.3.2. Also across all the four systems derivative predictions for the variables Top Tol (x4), Bot T (x5), and Bot MCH (x6) is bad , $R^2 = 0$, that is simple average value is predicted for these variables. This is because these variable show fast dynamics and that is they reach steady state faster than other variables. In other words derivatives for these variables are more spiky than other variables. However, SINDy performs really well for predicting the dynamics of these variables with complex function. This shows that SymReg is getting stuck in local optima (here average value), thus failing to learn the complex function needed to capture the dynamics of these variables.

Hence, overall we observed that SINDy performed better than SymReg for predicting the dynamics of most state variables under all four different operating conditions tested for the distillation column.

6.2. Derivative predictions beyond the training time and outside training perturbation region: SINDy and symreg

Results in Tables 5 and 6 show the performance of extracted ODEs from the algorithm for accuracy of predictions beyond the time of training datasets (long time evolution accuracy) for System 1, for SINDy and SymReg respectively. It is clearly seen that for long time simulations, both algorithms give test accuracies similar to those given in Tables 3 and 4 for System 1. Hence, we chose to show only 1 system here, as the differences in performance will be expected to be similar based on similar functions for these algorithms across the systems. In Fig. 4 for SINDy we can see that the model performs well for predicting dynamics on test data generated from long time simulations beyond the training time as indicated by high R^2 value. This is the case for all the variables. This cements the fact that the evolution of the system with time (if present) has been captured well by the extracted DEs. If this were not the case the model performance would have deteriorated with longer tests. On the other hand, consistent with lower performance of SymReg, Fig. 5 shows lower prediction accuracy for long term simulations.

Results for outside perturbation region testing for system 1 are also listed in Tables 5 and 6 for SINDy and SymReg respectively. We find that the performance is sub-par in the region outside the training perturbation for most of the states for SINDy model as in Fig. 6(b). Also, for SINDy, with higher regularization, the model marginally improves as opposed to all the previous observations where the model kept getting better on the test set with decreasing regularization. This indicates that the available variables and candidate functions are over-fitting for the state of the system in the training region. It can also be indicative of other dynamic regimes present outside the perturbation region, which can be resolved by including new state variables or candidate functions to train new models in different perturbation regions that will allow capturing the dynamics pertinent to that regime. Hence, this can be used to build piecemeal functions for different regimes and gain insights into overlapping mechanisms for governing dynamics. However, it is not clear what range of perturbations will be enough to capture all dynamic regimes.

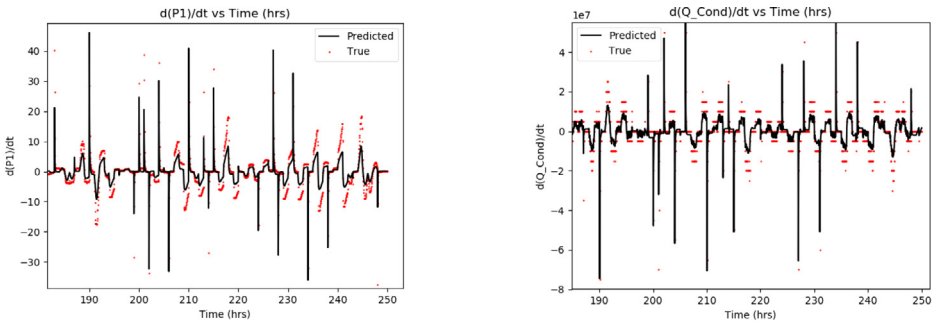
In contrast to SINDy, we see that the accuracy for outside the perturbation for SymReg is at par with basic System 1 and long time simulation for System 1 (Fig. 7). This indicates that even though SymReg might be underfitting but the underlying structure identified by it is valid for outside perturbation range thereby eliminating the point of other dynamic regime for this outside perturbation range data. This further validated our observation that SINDy may have over-fitted for a specific dynamic regime.

6.3. Identified Governing Equations: Structural Comparison and Physical Interpretations

6.3.1. ODE structural comparison across different systems

The governing Ordinary Differential Equations (ODEs) obtained for the simulated distillation column under 4 different operating conditions (Table 1) were compared with each other for similarity in the terms selected. This comparison can be interpreted as dynamic equivalent of sensitivity analysis in steady state systems since the results here depict the impact of changing operating conditions on governing equations for dynamics in the studied distillation column. We also present comparison between the equations obtained from SINDy and SymReg.

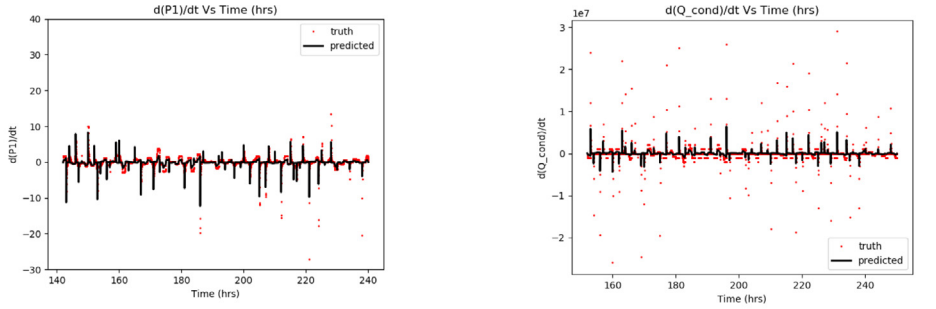
SINDy: The number of similar terms in ODEs for two levels of regularization along with the total number of terms is provided in Appendix B, section 2.2. Tables in section 2.2 of Appendix B also have a list of the terms that were repeated maximum number of times across the 4 systems tested. If there is a unique governing dynamical law that determines the overall dynamics of distillation column and SINDy appropriately captured it, we would expect most of the terms in the ODEs to be repeated across the system to depict similar driver of dynamics. However, this was not observed. Hardly 10% of the total terms were common across 3 systems where the feed compositions were altered. This could mean that we have not completely described our system with the current set of states or there is not a truly unique governing law for overall dynamics of distillation column. We need



(a) Condenser Pressure Derivative vs Time - Good Predictions ($R^2 = 0.949$)

(b) Reboiler Duty Derivative vs Time - Good Predictions ($R^2 = 0.852$)

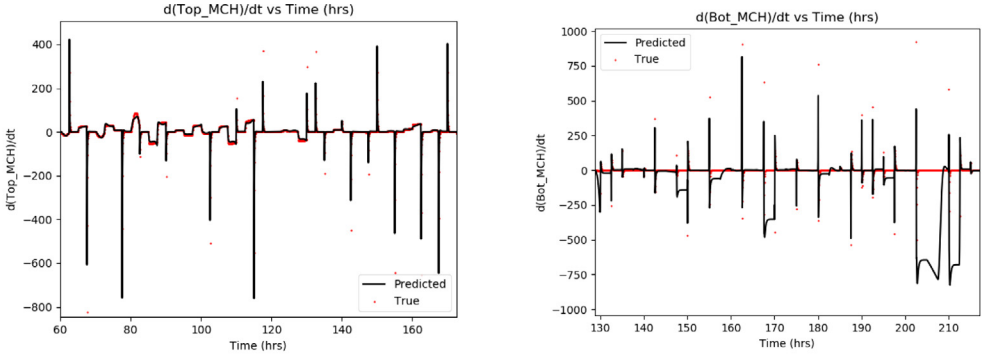
Fig. 4. SINDy Model Performance on Predicting $\dot{x}(t)$ for Test Data from Long Time Simulations.



(a) Condenser Pressure Derivative vs Time - OK Predictions ($R^2 = 0.489$)

(b) Reboiler Duty Derivative vs Time - OK Predictions ($R^2 = 0.550$)

Fig. 5. SymReg Model Performance on Predicting $\dot{x}(t)$ for Test Data from Long Time Simulations.



(a) Outside Training Perturbation - Good Predictions ($R^2 = 0.799$)

(b) Outside Training Perturbation - Poor Predictions ($R^2 = 0.130$)

Fig. 6. SINDy Model Performance on Predicting $\dot{x}(t)$ for Data Outside Training Perturbation.

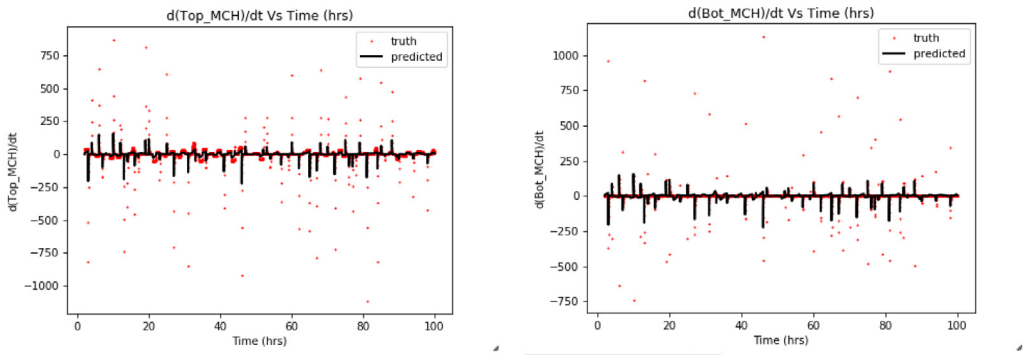
to look for variables which are crucial in deciding the dynamics by performing sensitivity analyses on the operating conditions too.

However, by reducing regularization we noticed that the fraction of terms retained across the systems either increased or remains the same in most cases. This indicates that by increasing the number of candidate functions selected, they are able to explain the model better, even if only by a small increment. This result correlates with the prediction accuracy explained earlier which kept improving with smaller regularization. We also find the same terms repeating across all 4 systems more commonly. The system with a different Reflux Ratio (which is the only column specification varied, System 1) had no common terms with other systems under high regularization but had an increasing number of common terms under low regularization. This could further indicate that the system might not be truly sparse in function space, highlighting the possible limitations of using SINDy in identifying the complex dynamics of unknown system without some knowledge about functional space that may govern the dynamics of these systems. A similar analysis was carried out between the training set and the test set with phenol feed outside the training perturbation region. The results of this analysis are listed in Table 7

SymReg: Table 8 contains common form of derivative equations obtained across all the 4 systems. We observed that for most of the state variables the ODE had structure of the form in Eq. (9).

$$\dot{x} = c * (x_a - x_b) \tag{9}$$

where c is some constant. Similar structures were obtained even after increasing the population size at each generation and depth of initial generation. Across all the systems, variable Cond Q ($\dot{X}_9$) had ODE



(a) Outside Training Perturbation - ok Predictions ($R^2 = 0.510$)

(b) Outside Training Perturbation - Ok Predictions ($R^2 = 0.213$)

Fig. 7. SymReg Model Performance on Predicting $\dot{x}(t)$ for Data Outside Training Perturbation.

Table 7
SINDy: Structural similarity of ODEs.

Variable	Low α		High α	
	Common	Total	Common	Total
Top F ($\dot{x_0}$)	5	23	5	14
Top T ($\dot{x_1}$)	4	24	1	3
Top MCH ($\dot{x_2}$)	7	28	4	14
Top Ph ($\dot{x_3}$)	1	6	1	6
Top Tol ($\dot{x_4}$)	13	28	5	18
Bot T ($\dot{x_5}$)	18	39	5	16
Bot MCH ($\dot{x_6}$)	21	39	3	18
Bot Ph ($\dot{x_7}$)	19	35	6	13
Bot Tol ($\dot{x_8}$)	21	37	4	12
Cond Q ($\dot{x_9}$)	13	36	3	14
Vap Reb ($\dot{x_{10}}$)	17	35	5	17
P1 ($\dot{x_{11}}$)	5	30	3	9
P22 ($\dot{x_{12}}$)	17	34	3	11

Table 8
SymReg: Structural similarity of ODE's.

Variable	Common form
Top F ($\dot{x_0}$)	$x5 - x10$
Top T ($\dot{x_1}$)	$x5 - x10$
Top MCH ($\dot{x_2}$)	$x5 - x10$
Top Ph ($\dot{x_3}$)	$x5 - x10$
Top Tol ($\dot{x_4}$)	*Constant*
Bot T ($\dot{x_5}$)	*Constant*
Bot MCH ($\dot{x_6}$)	$x5 - x10$
Bot Ph ($\dot{x_7}$)	$x5 - x10$
Bot Tol ($\dot{x_8}$)	$x5 - x10$
Cond Q ($\dot{x_9}$)	$x10 - x5$
Vap Reb ($\dot{x_{10}}$)	$x5 - x10$
P1 ($\dot{x_{11}}$)	$x5 - x10$
P22 ($\dot{x_{12}}$)	$x5 - x10$

structure in negation of other state variables. This clearly indicates that whenever value of Top F or Top T increased based on Phenol perturbation then Cond Q value decrease and vice versa. Such simple interpretative results can be seen directly from simple white-box models. Structure for Top Tol ($\dot{X}_4$) and Bot T ($\dot{X}_5$) had a constant predicted for all the four systems. These constants corresponded to the average value of these state variables. Overall, SymReg shows more similarity across systems for capturing dynamics of variables than SINDy.

ODE Structural Comparison for SINDy vs SymReg : Our next comparison focused on comparing the equations obtained for same state variable from SINDy and SymReg. Fig. 8 shows the dynamics prediction for $x2$ (Top MCH) using the equation obtained from SINDy (Eq. (10)) and SymReg (Eq. (11)). It can be seen in Figure that SINDy performs better on prediction of dynamics with low RMSE while SymReg has high RMSE. SymReg captures the pattern but misses on the magnitude by large values. However, the SINDy equation is much more complex and may be overfitting and has low interpretability while SymReg is clearly underfitting and has easier interpretability. According to SymReg (Eq. (11)), the dynamics of Top MCH flow is determined by bottom Toluene and vapor flow rate in reboiler which simply means that the flow rate of MCH obtained as top product is determined by how much toluene is extracted in bottom and reboiling returning vapor to the column. Such simplistic understanding of dynamics was not feasible for SINDy equations.

$$\dot{x}_{2-SINDy} = 0.409x_{10}x_{12}^{-1} - 0.01188x_8x_{13} - 0.6272x_8x_{12}^{-1} - 1.278x_6x_8^{-1} - 0.3459x_8^2 + 0.05691x_5x_8^{-1} + 0.04383x_4^{-1}x_3^{-1} + 0.1908x_4x_6 + 0.03016x_3^{-1}x_{13}^{-1} + 0.06078x_3^{-1}x_{11}^{-1} - 0.1535x_0^{-1}x_8 - 0.03093sin(x_4) - 0.0147sinx_6 - 0.002557cosx_4 \quad (10)$$

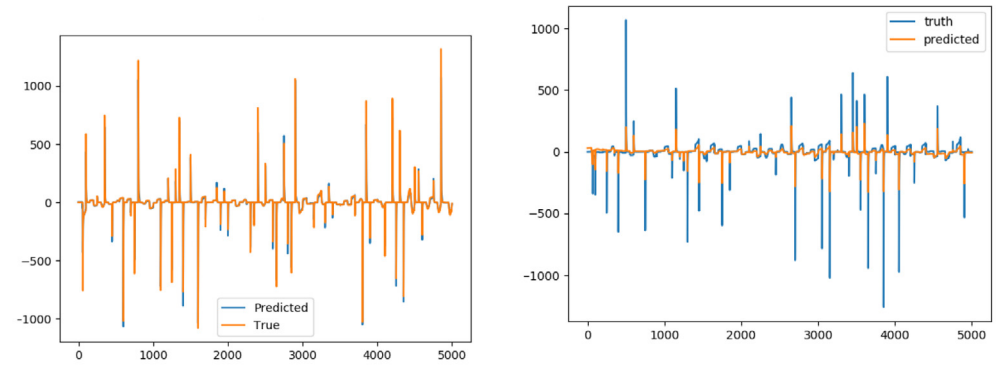
$$\dot{x}_{2-SymReg} = x_5 - x_{10} \quad (11)$$

6.3.2. *Why symreg can generate simpler functions as compared to SINDy ?*

In SINDy a single model function (comprising of all candidate functions of all state variables) is optimized. Whereas in SymReg multiple models functions (parse trees) generated from either some or all the state variables are optimized. For example in Fig. 9 the spiked shaped figure indicates the objective function space of a sparse model function. In Fig. 9(a) the SINDy model function (dotted circle) consists of all the state variables (here two) which is then optimized, The complete circle (optimized model) is stuck at the point of local optima (red circle). In Fig. 9(b) SymReg generates multiple initial model functions consisting of either some of the state variables (two dotted models on both the axis) or all state variables. Those functions are evolved until one of them reaches a local optima (red circle). Since smaller model functions are generated therefore there is a chance that smaller model functions can also reach local optimum which is selected if the overall prediction accuracy is best among the selected trees.

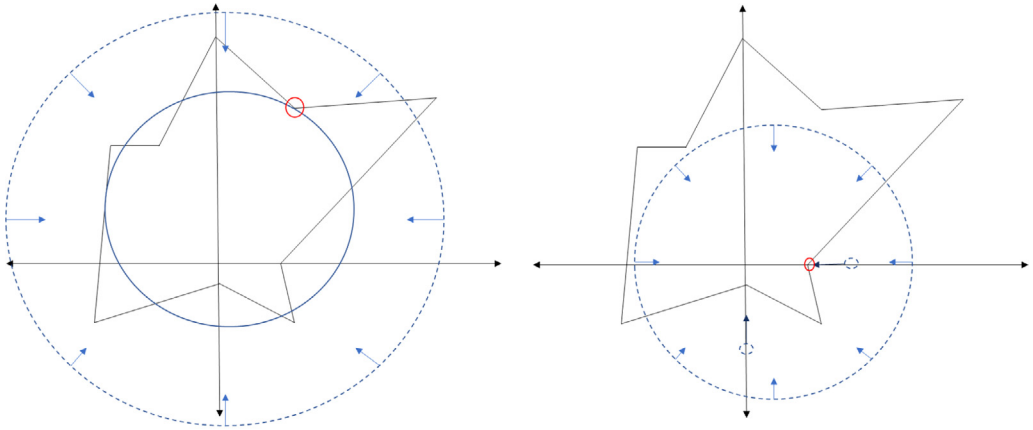
6.4. *Physical interpretation of ODEs*

In order to relate the extracted ODEs to a governing physical law for dynamics of distillation column, structure of ODEs was analyzed for System 2 (Equations under high regularization and are shown in Appendix B, section 2.1). While these ODEs are very complex to interpret for a single physical law, it is still a win for representing the dynamics of this system using one equation for each state variable as compared to over 1000 complex equations that relate the dynamics of system. However, there was no direct interpretation of most of the terms in physical sense for both the models.



(a) $\dot{x}2$ predicted using Equation 10, RMSE = 14.06 (b) $\dot{x}2$ predicted using Equation 11, RMSE = 40.34

Fig. 8. Comparison of SINDy and SymReg for derivative prediction based on RMSE and parsimony.



(a) Sindy Algorithm Working (b) SymReg algorithm working

Fig. 9. Working depiction of both the algorithms, dotted lines are initial model functions, and red circle indicates local optima. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

For **SINDy** some of the terms such as $sin(Top_{Tol})$ which represents sin of Toluene concentration in Top flow is physically not interpretable. Some of the commonly recurring terms that we found physically relevant were: $Conc^2$ which basically meant that second order terms in concentration were found relevant for controlling the dynamics. These second order terms were related to the possible diffusion of two components or cross diffusion driving the dynamics. This can be because of fick's law of diffusion acting on both the component involved. For example one of the terms in Appendix B, section 2.1 is Bot_{MCH}/Bot_{Tol} which is the ratio of concentration of MCH and Toluene in bottom flow. Appearance of this term in the equation driving the MCH in top stream denotes some relationship between diffusivity difference of MCH and Toluene in the extracting component Phenol. While the form of equation is surprising because the functional form did not give the fick's law of diffusion which actually needs concentration variation rather than just concentration, the appearance of this term provides some hope of these data driven approaches to learn about the governing mechanisms of dynamics in unknown complex systems. Another term that we related to a physical law is ratio of concentration and Pressure such as the terms of form Bot_{Tol}/P_{22}. This term represents concentration of Toluene in bottom feed and pressure of the last plate. We related this term to the Henry's law which relates concentration of a solute in liquid phase to the partial pressure of the solute in gas phase. This term probably represents the relationship that the dynamics of extraction is driven by concentration in bottom stream for toluene which is related to the pressure on plates where the component may exist in vapor phase.

For **SymReg**, the form of Eq. (9) makes it clear that difference of those two variables is changing at the same instance as $\dot{x_s}$ (of most of the state variables). Though the magnitude of spike is not learned by this simple form of SymReg algorithm. For example in Fig. 9(b), $\dot{x_2}$ for system 3 is predicted as x5-x10. Looking at the variables it states that $\dot{Top_{MCH}} = BotT - VapReb$, that is change in flowrate = temperature - Energy. Though it makes sense that the flowrate of component will be effected by temperature and energy provided or based on flowrate and temperature, energy required can be calculated but here the model does not provide more further interpretation. Hence, SymReg provides a minimal equation that captures the dynamics well but does not provide complex relationships between different governing variables. Perhaps, a parsimonious physical relationship is derived between the key variables driving the dynamics which can be useful for the goal of predicting dynamics but not understanding the full mechanisms driving the dynamics.

Overall the gas–liquid mass transfer in these type of complex systems are interconnected and complex, hence it is difficult to pin-point one single mechanism driving dynamics. However, it is encouraging to see some functional forms that may be related to physical laws being picked up by SINDy in these equations and simple equation in SymReg that have key driving variables. We conclude from these observations that in order to be able to identify the laws for complex engineered systems such as distillation column, better functional formulation rules incorporating domain knowledge must be developed to be used with properly tweaked machine learning algorithms. SINDy will be preferred in such cases as it performed well in capturing these complex functions.

In case, no underlying functional law is available, we believe SymReg will be a good starting algorithm to understand the relationship for key variables driving the dynamics.

7. Conclusions

In this work, we have demonstrated the strengths and limitations of machine learning (ML) based approach to identify governing equations for dynamics of complex manufacturing systems such as distillation column. Comparison of two widely used white-box ML approaches — SINDy and SymReg on extracting distillation column dynamics, shows promising results for prediction of dynamics using the model extracted from data which is simpler to interpret. On comparison for prediction of dynamics it was observed that SINDy performs better in prediction than SymReg based on R^2 values. The results for prediction using ODEs extracted by SINDy on the test data generated from mechanistic models were very encouraging with most variables showing more than 80% accuracy. However, outside the perturbation range, the equations did not perform very well which may be because of the change in dynamic regime. If the training data set only captured a particular dynamic regime, it cannot capture the dynamics in a different regime. However, this is still an un-resolved question from mechanistic perspective, that if DEs capture true physical mechanisms this should provide insights into impending regime change as well. On the other hand, while SymReg showed lower prediction power as compared to SINDy, it gave much simpler equations for dynamics and performed better than SINDy on outside perturbation range. This may be due to overfitting during training the model using complex functions in SINDy while SymReg starts from simpler function.

From physical interpretation perspective of the equations obtained, in SINDy, it was encouraging to see terms such as $Concentration^2$ and ratio of concentration with pressure. The prior can be related to fick's law of diffusion for two components in the column whereas the later can be related to the Henry's law controlling the solubility of the components in the mixture controlled by pressure at different plates in the column. In comparison, some of the SymReg equations can be related to conservation laws like energy conservation but not much can be interpreted due to the simplicity except for the basic variables that may be driving the dynamics.

Overall, with SINDy being overfitting and SymReg being underfitting, some crucial state variables or functional forms might be being missed in these algorithms. However, if the aim of the work is to obtain simpler equations that can capture the overall non-linear dynamics for a particular system, the algorithms perform well for predicting dynamics with reasonable accuracy. Additionally, it was clearly seen that SINDy outperformed SymReg except in case of outside perturbation range predictions. But, in order to understand the true underlying physical mechanisms governing dynamics, the SINDy algorithm perhaps need to be provided with functional forms determined by domain expert and SymReg must be tweaked to generate these complex forms. Such an approach was used in identifying the reaction kinetics equations (Hoffmann et al., 2019), where the authors provided functional form determined by "law of mass action" which is a known physical law that drives rate kinetics and mechanisms. To improve on the distillation column differential equation identification, such knowledge about relationship between top and bottom feed, temperature and pressure need to be used to construct appropriate functions. This is challenging for the distillation column system because there are several heuristic based equations that are used in design of the separation system along with iterative numerical computations that utilize mass and energy balances at plate scale, highlighting some limitations of ML approach for identifying mechanisms for complex manufacturing systems.

One interesting finding from extracting these DEs is the simplified relationship that was obtained between component flow rate in top stream to the component flow rates in the bottom flow rate along with the pressure of last plate by SINDy. In actual distillation column design, there is a mass balance equation solved for each plate that finally relates the component concentration in top stream to the bottom stream. Use of this one simplified equation captures this whole dynamics. Hence, we conclude this to be one of the major strength of machine learning approach for analyzing the dynamics of manufacturing unit operation. Based on the accuracy of prediction within certain time steps, a moving time window to train the model would be more appropriate. Our expectation of SINDy and SymReg generating same governing equations may also be far fetched, given both these algorithms use different approach to capture dynamics. However, the simple equations generated can be used complementary to identify key variables for overall dynamics of unknown system using SymReg and key functions (from set of expected functions) to capture the dynamics using SINDy.

In summary, different ML algorithms may need to be used in parallel to discover the laws of dynamics for complex systems. While, we cannot claim that ML is a panacea to identifying governing equations for mechanisms driving the dynamics of complex unit operations, this study certainly shows the strength of ML in identifying equations that can be very useful for predicting dynamics. The key challenge is providing appropriate functions and mathematical rules using domain knowledge to the algorithm. Such integration of ML and chemical engineering sciences will be powerful in understanding dynamics of novel complex systems along with using this knowledge for robust design and operations of emerging manufacturing systems.

CRediT authorship contribution statement

Renganathan Subramanian: SINDy Implementation, Data Curation, Simulations, Analysis, Visualization, Writing. **Raghav Rajesh Moar:** SymReg Implementation, Data Curation, Analysis, Visualization, Writing. **Shweta Singh:** Conceptualization, Funding Acquisition, Writing - review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Authors acknowledge the funding support provided by Purdue Undergraduate Research Experience (PURE) and Indian Institute of Technology (IIT), Madras, India for partial support of research participation for first and second authors and U.S. National Science Foundation (CBET-1805741). We also thank feedback from anonymous reviewers and editors that have helped improved this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.mlwa.2020.100014.

References

Bennett, D. L., Agrawal, R., & Cook, P. J. (1983). New pressure drop correlation for sieve tray distillation columns. *AIChE Journal*, *29*(3), 434–442. http://dx.doi.org/10.1002/aic.690290313.

Berber, R., & Karadurmus, E. (1989). Dynamic simulation of a distillation column separating a multicomponent mixture. *Chemical Engineering Communications*, *84*(1), 113–127. http://dx.doi.org/10.1080/00986448908940338.

Bindal, A., Ierapetritou, M. G., Balakrishnan, S., Armaou, A., Makeev, A. G., & Kevrekidis, I. G. (2006). Equation-free, coarse-grained computational optimization using timesteppers. *Chemical Engineering Science*, *61*(2), 779–793. http://dx.doi.org/10.1016/j.ces.2005.06.034.

Bolles, W. L. (1988). Distillation tray fundamentals, by M. J. Lockett, 1986, 226 pages, Cambridge University Press, Cambridge, England and New York, $54.50 (U.S.). *The Canadian Journal of Chemical Engineering*, *66*(1), 173–174. http://dx.doi.org/10.1002/cjce.5450660130.

Bongard, J., & Lipson, H. (2007). Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, *104*(24), 9943–9948. http://dx.doi.org/10.1073/pnas.0609476104.

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*(15), 3932–3937. http://dx.doi.org/10.1073/pnas.1517384113, arXiv:https://www.pnas.org/content/113/15/3932.full.pdf.

Carrier, G. F. (1967). IV. Training in applied mathematics research. *SIAM Review*, *9*(2), 347–365. http://dx.doi.org/10.1137/1009065.

Chartrand, R. (2011). Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics*, http://dx.doi.org/10.5402/2011/164564.

Choe, Y. S., & Luyben, W. L. (1987). Rigorous dynamic models of distillation columns. *Industrial & Engineering Chemistry Research*, *26*(10), 2158–2161. http://dx.doi.org/10.1021/ie00070a038.

Ding, Y., Zhang, Y., Ren, Y. M., Orkoulas, G., & Christofides, P. D. (2019). Machine learning-based modeling and operation for ALD of SiO_2 thin-films using data from a multiscale CFD simulation. *Chemical Engineering Research and Design*, *151*, 131–145. http://dx.doi.org/10.1016/j.cherd.2019.09.005.

Donoho, D. L., & Elad, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences*, *100*(5), 2197–2202. http://dx.doi.org/10.1073/pnas.0437847100, arXiv:https://www.pnas.org/content/100/5/2197.full.pdf.

Gani, R., Ruiz, C., & Cameron, I. (1986). A generalized model for distillation columns—I. *Computers & Chemical Engineering*, *10*(3), 181–198. http://dx.doi.org/10.1016/0098-1354(86)85001-3.

Gerbaud, V., Rodriguez-Donis, I., Hegely, L., Lang, P., Denes, F., & You, X. (2019). Review of extractive distillation. Process design, operation, optimization and control. *Chemical Engineering Research and Design*, *141*, 229–271. http://dx.doi.org/10.1016/j.cherd.2018.09.020.

Gout, J., Quade, M., Shafi, K., Niven, R. K., & Abel, M. (2018). Synchronization control of oscillator networks using symbolic regression. *Nonlinear Dynamics*, *91*(2), 1001–1021.

Green, D. W., & Perry, R. H. (2007). *Perry's chemical engineers' handbook* (8th ed.). McGraw-Hill Education, URL: https://www.amazon.com/Perrys-Chemical-Engineers-Handbook-Eighth/dp/0071422943?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0071422943.

Harirchi, F., Kim, D., Khalil, O., Liu, S., Elvati, P., Baranwal, M., Hero, A., & Violi, A. (0000). On sparse identification of complex dynamical systems: A study on discovering influential reactions in chemical reaction networks. http://web.eecs.umich.edu/~mayankb/docs/JFUE-D-20-00031.pdf.

Hoffmann, M., Fröhner, C., & Noé, F. (2019). Reactive SINDy: Discovering governing reactions from concentration data. *The Journal of Chemical Physics*, *150*(2), Article 025101. http://dx.doi.org/10.1063/1.5066099.

K. Natarajan, B. (1995). Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, *24*, 227–234. http://dx.doi.org/10.1137/S0097539792240406.

Kapoor, N., & McAvoy, T. (1988). An analytical approach to approximate dynamic modeling of distillation towers. In *Dynamics and control of chemical reactors and distillation columns* (pp. 99–104). Elsevier, http://dx.doi.org/10.1016/b978-0-08-034917-6.50019-2.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection (Complex adaptive systems)*. A Bradford Book, URL: https://www.amazon.com/Genetic-Programming-Computers-Selection-Adaptive/dp/0262111705?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0262111705.

Krishnapura, V. G., & Jutan, A. (1997). Arma neuron networks for modeling nonlinear dynamical systems. *The Canadian Journal of Chemical Engineering*, *75*(3), 574–582. http://dx.doi.org/10.1002/cjce.5450750311, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cjce.5450750311, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cjce.5450750311.

Kumar, S., Wright, J., & Taylor, P. (1983). Modelling and dynamics of an extractive distillation column. In *1983 American control conference*. IEEE, http://dx.doi.org/10.23919/acc.1983.4788098.

Lee, J. H., Shin, J., & Realff, M. J. (2018a). Machine learning: Overview of the recent progresses and implications for the process systems engineering field. *Computers & Chemical Engineering*, *114*, 111–121. http://dx.doi.org/10.1016/j.compchemeng.2017.10.008.

Lee, J. H., Shin, J., & Realff, M. J. (2018b). Machine learning: Overview of the recent progresses and implications for the process systems engineering field. *Computers & Chemical Engineering*, *114*, 111–121. http://dx.doi.org/10.1016/j.compchemeng.2017.10.008, URL: http://www.sciencedirect.com/science/article/pii/S0098135417303538.

Lin, C. C., & Segel, L. A. (1988). *Mathematics applied to deterministic problems in the natural sciences*. Society for Industrial and Applied Mathematics, http://dx.doi.org/10.1137/1.9781611971347.

Lockett, M. J., & Banik, S. (1986). Weeping from sieve trays. *Industrial & Engineering Chemistry Process Design and Development*, *25*(2), 561–569. http://dx.doi.org/10.1021/i200033a038.

Macmurray, J., & Himmelblau, D. (1995). Modeling and control of a packed distillation column using artificial neural networks. *Computers & Chemical Engineering*, *19*(10), 1077–1088. http://dx.doi.org/10.1016/0098-1354(94)00098-9.

McAvoy, T., & Wang, Y. (1986). Survey of recent distillation control results. *ISA Transactions*, *25*(1), 5–21, URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0022472526&partnerID=40&md5=dd48440dd66e25a35142de78b9c2891b, cited By 13.

Pantelides, C., Gritsis, D., Morison, K., & Sargent, R. (1988). The mathematical modelling of transient systems using differential-algebraic equations. *Computers & Chemical Engineering*, *12*(5), 449–454. http://dx.doi.org/10.1016/0098-1354(88)85062-2.

Prasad, V., & Bequette, B. W. (2003). Nonlinear system identification and model reduction using artificial neural networks. *Computers & Chemical Engineering*, *27*(12), 1741–1754. http://dx.doi.org/10.1016/S0098-1354(03)00137-6, URL: http://www.sciencedirect.com/science/article/pii/S0098135403001376.

Quade, M., Abel, M., Shafi, K., Niven, R. K., & Noack, B. R. (2016). Prediction of dynamical systems by symbolic regression. *Physical Review E*, *94*(1), Article 012214.

Rademaker, O. (1975). *Dynamics and control of continuous distillation units*. Elsevier Scientific Pub. Co, URL: https://www.amazon.com/Dynamics-control-continuous-distillation-units/dp/0444412344?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0444412344.

Raissi, M., & Karniadakis, G. E. (2018). Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, *357*, 125–141. http://dx.doi.org/10.1016/j.jcp.2017.11.039.

Rasmuson, A., Andersson, B., Olsson, L., & Andersson, R. (2014). *Mathematical modeling in chemical engineering*. Cambridge: Cambridge University Press, http://dx.doi.org/10.1017/CBO9781107279124, URL: https://www.cambridge.org/core/books/mathematical-modeling-in-chemical-engineering/DCAFF7943523A402266EB50E2507A152.

Retzbach, B. (1986). Control of an extractive distillation plant. *IFAC Proceedings Volumes*, *19*(15), 225–230. http://dx.doi.org/10.1016/s1474-6670(17)59426-4.

Richardson, J., Coulson, J., & Sinnott, R. K. (1983). Chemical engineering, vol. 6: An introduction to design. In: Chemical engineering technical series. Pergamon. URL: https://www.amazon.com/Chemical-Engineering-Introduction-Design-Technical/dp/0080229700?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0080229700.

Rosenbrock, H. (1962). The transient behaviour of distillation columns heat exchangers — a historical and critical review. *Transactions of the Institution of Chemical Engineers*, *40*(6), 376–384, cited By 11.

Rudy, S., Alla, A., Brunton, S. L., & Kutz, J. N. (2019). Data-driven identification of parametric partial differential equations. *SIAM Journal on Applied Dynamical Systems*, *18*(2), 643–660. http://dx.doi.org/10.1137/18m1191944.

Schaeffer, H., Caflisch, R., Hauck, C. D., & Osher, S. (2013). Sparse dynamics for partial differential equations. *Proceedings of the National Academy of Sciences*, *110*(17), 6634–6639. http://dx.doi.org/10.1073/pnas.1302752110.

Schmidt, M., & Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science*, *324*(5923), 81–85. http://dx.doi.org/10.1126/science.1165893.

Singh, V., Gupta, I., & Gupta, H. (2005). ANN based estimator for distillation—inferential control. *Chemical Engineering and Processing: Process Intensification*, *44*(7), 785–795. http://dx.doi.org/10.1016/j.cep.2004.08.010.

Singh, V., Gupta, I., & Gupta, H. (2007). ANN-based estimator for distillation using Levenberg–Marquardt approach. *Engineering Applications of Artificial Intelligence*, *20*(2), 249–259. http://dx.doi.org/10.1016/j.engappai.2006.06.017.

Skogestad, S. (1992). Dynamics and control of distillation columns - A critical survey. *IFAC Proceedings Volumes*, *25*(5), 11–35. http://dx.doi.org/10.1016/s1474-6670(17)50966-0.

Stichlmair, J., & Hofer, H. (1978). Mitreißen von Flüssigkeit aus der Zweiphasenschicht von Kolonnenböden. *Chemie Ingenieur Technik*, *50*(7), 553. http://dx.doi.org/10.1002/cite.330500717.

Tolliver, T., & Waggoner, R. (1980). Distillation column control; a review and perspective from the cpi. *35*, (pt 1), (pp. 83–106). URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0019103881&partnerID=40&md5=d089a08f1794e11a99e8dde099ce5936, cited By 5,

Venkatasubramanian, V. (2019). The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal*, *65*(2), 466–478. http://dx.doi.org/10.1002/aic.16489, URL: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.16489.

Wollkind, D. J., & Dichone, B. J. (2018). *Comprehensive applied mathematical modeling in the natural and engineering sciences: Theoretical predictions compared with data*. Springer, URL: https://www.amazon.com/Comprehensive-Mathematical-Modeling-Engineering-Sciences-ebook/dp/B07DQR96TV?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=B07DQR96TV.

Zames, G., Ajlouni, N., Ajlouni, N., Ajlouni, N., Holland, J., Hills, W., & Goldberg, D. (1981). Genetic algorithms in search, optimization and machine learning. *Information Technology Journal*, *3*(1), 301–302.