2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Dynamic System Identification from Scarce and Noisy Data using Symbolic Regression

Benjamin Cohen[1], Burcu Beykal[1,2], and George Bollas[1]

Abstract— **A framework for dynamic system model identification from scarce and noisy data is proposed. This framework uses symbolic regression via genetic programming with a gradient-based parameter estimation step to identify a differential equation model and its parameters from available system data. The effectiveness of the method is demonstrated by identifying four synthetic systems: an ideal plug flow reactor (PFR) with an irreversible chemical reaction, an ideal continuously stirred tank reactor (CSTR) with an irreversible chemical reaction, a system described by Burgers' Equation, and an ideal PFR with a reversible chemical reaction. The results show that this framework can identify PDE models of systems from broadly spaced and noisy data. When the data was not sufficiently rich, the framework discovered a surrogate model that described the observations in equal or fewer terms than the true system model. Additionally, the method can select relevant physics terms to describe a system from a list of candidate arguments, providing valuable models for use in controls applications.**

I. INTRODUCTION

A model can have a profound effect on the entire lifecycle of a process, so effective model discovery is vital to the success of a project. Data-driven modeling approaches (neural networks [1], Gaussian processes [2], and others [3], [4]) can transform data into effective models that predict dynamics well. These methods can integrate into control systems [5], [6], help engineers design chemical processes [7], and require little to no physics to train [8]. Despite their success, physics-agnostic models fail to predict dynamics well outside the bounds of their training data and do not explain the phenomena responsible for the dynamics they capture.

To overcome some of the limitations of traditional data-driven modeling methods, engineers can fuse these approaches with domain expert knowledge to generate models consistent with the phenomena that drive the dynamics of a process. Physics-informed machine learning (PIML) leverages domain expert knowledge and rich data sets to create models. Models created using PIML approaches are capable of extrapolating beyond the bounds of their training data better than their physics-agnostic counterparts [9], [10], learn dynamics from scarce data [10], [11], and lend themselves well to control applications [5].

Despite the injection of physics into data-driven modeling methods, many PIML modeling methods still create black-box models that can be difficult to interpret. The black-box nature of a model makes it difficult to trust, justify, and understand, especially outside the bounds of its training data. One subset of PIML methods that can overcome the black-box nature of many data-driven models is the subset that can yield human-interpretable models. Symbolic regression (SR) is one such method that automatically discovers mechanistic or semi-empirical models from data and a physics-informed argument set.

Two approaches in SR are genetic programming (GP) and parametric regression algorithms. SR via parametric regression linearly combines terms of a predefined candidate library to generate a parsimonious model that agrees with the training data. SR via constrained regression [12] and sequential threshold ridge regression [13], [14] are fast methods that simultaneously discover a model shape and its parameters. These methods can identify models expressed as algebraic and differential equations. The work done by Rudy et al. [15], and related works [16], [17], can recover partial differential equation (PDE) models from measurements but struggle to identify meaningful models once the data becomes too noisy. In response to this limitation, Messenger and Bortz [18] extended the work by Rudy et al. to consider the weak form of PDEs. This approach integrates data using a data-driven Galerkin method. The integration helps overcome sensitivity to noise but model discovery depends on low integration error which grows rapidly as the spacing between samples increases.

SR via GP searches a symbol space for an optimal expression by evolving a population of expressions represented as binary trees [19]. GP and other evolutionary algorithms can discover physics models from data [14], [20]. SR via GP can also discover differential equations by either approximating derivatives from data [21], [22] or integrating evolving expressions [23]. The flexibility of GP means that it can discover ordinary differential equation models from data that exhibit wide ranges of quantity and quality. Genetic algorithms can also extend other data-driven modeling methods to find PDE models [24]. To the best of the authors' knowledge, current methods of identifying PDE models using genetic algorithms fail when confronted with limited and noisy data.

This work was supported by the Pratt & Whitney Institute for Advanced Systems Engineering (P&W-IASE) of the University of Connecticut. Any opinions expressed herein are those of the authors and do not represent those of the sponsor.

B. Cohen is with the Department of Chemical and Biomolecular Engineering, University of Connecticut, Storrs, CT 06269, USA benjamin.2.cohen@uconn.edu

B. Beykal is with the Department of Chemical and Biomolecular Engineering, University of Connecticut, Storrs, CT 06269, USA and the Center for Clean Energy Engineering, University of Connecticut Storrs, CT 06269, USA burcu.beykal@uconn.edu

G. Bollas is with the Department of Chemical and Biomolecular Engineering, University of Connecticut, Storrs, CT 06269, USA george.bollas@uconn.edu

This work focuses on a novel framework using SR via GP that can identify informative PDE models when only spatially scarce and potentially noisy data is available. This framework takes an incomplete system model, data, a physics-informed argument set, and instructions for parameter inclusion and returns a parsimonious, descriptive, and dynamic model. A bi-level optimization problem generates this model. The upper-level optimization problem seeks to minimize an information-theoretic criterion by exploring a symbol space using GP. The information-theoretic criterion should consider both the complexity of the evolving models and how well those models agree with the data. The lower-level optimization problem minimizes the difference between the predicted and measured state variables by searching for an optimal parameter set for a fixed model structure.

Section II presents the mathematical background and the formulation of the bi-level optimization problem. Section III describes the synthetic experiments that tested the framework's ability to identify useful PDE models from data and discard irrelevant physics terms. Section IV provides case study results and discussion, and Section V concludes with insights and future work.

II. MODEL DISCOVERY FRAMEWORK

This section summarizes the mathematical background of the framework. A generalized formulation of the framework that can automatically discover a PDE model and its parameters from data is introduced. The PDE model should describe the t, time-, and z, space-, dependent state variables $\mathbf{x}(t, z)$ in the form below,

$$\begin{aligned} \frac{\partial \mathbf{x}(t,z)}{\partial t} &= \mathbf{s}(\mathbf{x}(t,z);\boldsymbol{\theta}) \\ \mathbf{x}(t,z) &= \mathbf{x}_0 \in \mathbb{R}^{n_s} \ \text{ at } \ t=0 \\ \mathbf{x}(t,z) &= \mathbf{x}_f \in \mathbb{R}^{n_s} \ \text{ at } \ z=0 \\ \frac{\partial \mathbf{x}(t,z)}{\partial z} &= \mathbf{0} \ \text{ at } \ z=l \\ 0 &\le t \le t_f \\ 0 &\le z \le l \end{aligned} \tag{1}$$

with n_s equal to the number of state variables and the function $\mathbf{s} : \mathbb{R}^{n_s} \mapsto \mathbb{R}^{n_s}$. Time-dependent data, $\mathbf{y}(t) \in \mathbb{R}^{n_s \times M}$ is collected for all n_s state variables at M measurement locations $\mathbf{z} = [z_1, ...z_M]$. y_m represents the data collected at z_m with $m = 1, ..., M$.

The framework identifies a set of expressions in the symbol space, $\mathbf{s} \in \mathcal{S}$, and its n_p parameters, $\boldsymbol{\theta} \in \mathbb{R}^{n_p}$, that minimizes an information-theoretic criterion, $\mathcal{I}$. This can be formulated as a bi-level optimization problem shown in equation (2).

The upper bounds on the parameters, $\boldsymbol{\theta}_{ub}$, and the lower bounds on the parameters, $\boldsymbol{\theta}_{lb}$, constrain the parameter space. An operator, aug$^{\theta}$, augments each candidate set of expressions, $\mathbf{s}_c \in \mathcal{S}$, with parameters using a physics-informed recipe. The optimization problem is formulated in this way to separate the parameter search from the expression search, reducing the size of the symbol space allowing for faster model discovery.

$$\begin{aligned} &\mathbf{s}^*(\mathbf{x}(t,z);\boldsymbol{\theta}) \in \arg\min_{\mathbf{s}\in\mathcal{S}} \quad \mathcal{I}(\mathbf{s}(\mathbf{x}(t,z);\boldsymbol{\theta})) \\ &\quad \text{s.t.} \\ &\qquad \boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^{n_p}} \quad \sum_{m=1}^{M} \int_0^{t_f} \frac{(\mathbf{y}_m(t) - \mathbf{x}(t,z_m))^2}{t_f} dt \\ &\qquad \text{s.t.} \\ &\qquad\quad \frac{\partial \mathbf{x}(t,z)}{\partial t} - \mathbf{s}(\mathbf{x}(t,z);\boldsymbol{\theta}) = \mathbf{0} \\ &\qquad\quad \boldsymbol{\theta}_{lb} - \boldsymbol{\theta} \le \mathbf{0} \\ &\qquad\quad \boldsymbol{\theta} - \boldsymbol{\theta}_{ub} \le \mathbf{0} \end{aligned} \tag{2}$$

A. Symbolic Regression via Genetic Programming

The upper level objective, $\mathcal{I}(\mathbf{s})$, is minimized by SR via GP. GP begins searching $\mathcal{S}$ by initializing a population of random differential equations represented as expression trees. Each node in an expression tree is assigned from a physics-informed argument set, $\mathcal{P}$, or a primitive set, $\mathcal{M}$.

The GP searches $\mathcal{S}$ by subjecting the population of expression trees to genetic operations. In each generation, the genetic operations, cross-over, and mutation, randomly operate on individuals within the population. This process generates a new population of expression trees. Individuals from both populations then compete in a tournament selection to select the fittest individuals that will make up the next generation. The genetic operations iteratively evolve the population until generation count exceeds the generation limit.

When $n_s > 1$, a co-evolution scheme searches $\mathcal{S}$ for a combination of n_s expressions that satisfy the objective function. In this case, an individual is considered to be a set of n_s expressions that evolve from n_s populations. Each population has a $\mathcal{P}$, and a $\mathcal{M}$. An individual's fitness depends on each of its expressions and how well those expressions, together, minimize $\mathcal{I}$.

B. Gradient-Based Parameter Estimation

The lower-level optimization problem aims to provide optimal parameters, $\boldsymbol{\theta}^*$, for each $\mathbf{s}_c$ proposed by the GP. A gradient-based optimization scheme attempts to give each $\mathbf{s}_c$ the best opportunity to minimize $\mathcal{I}$ by finding $\boldsymbol{\theta}^*$. One of two parameter estimation schemes is used to find $\boldsymbol{\theta}^*$. For experiments A and B, discussed later, a collocation scheme in time and central finite difference scheme in space integrated $\mathbf{s}_c$ and an interior point solver estimated the parameters. For experiments C and D, also discussed later, a nonlinear least squares algorithm estimated the parameters using a trust region reflective solver. Each $\mathbf{s}_c$ was integrated using a backward differentiation scheme with a preconditioned Krylov iterative linear solver. Neither solver can guarantee optimality so the framework checks the feasibility of each $\mathbf{s}_c$ by integrating $\mathbf{s}_c$ with its $\boldsymbol{\theta}^*$ returned by the lower-level optimization problem.

C. Information Theoretic Criterion

The framework requires an $\mathcal{I}(\mathbf{s})$ that considers both the complexity of an expression and the degree to which the complete system model agrees with $\mathbf{y}$. Fitness evaluation criteria like the mean squared error and R^2 are great candidates to quantify how well a model agrees with the data but do not consider the complexity of a model. The Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) both satisfy the requirements to consider the complexity of the model and the fit to $\mathbf{y}$. In this work, the $\mathcal{I}(\mathbf{s})$ is the BIC because the BIC penalizes the complexity of a model slightly more than the AIC. The BIC can be written as:

$$BIC = n_d \log mse + c_s \log n_d \tag{3}$$

with n_d representing the number of data points, mse representing the mean squared error between the data and the model prediction (also the lower-level optimization function), and c_s representing the complexity of $\mathbf{s}_c$. We consider the complexity of $\mathbf{s}_c$ to be equal to the number of terms in $\mathbf{s}_c$ that are in $\mathcal{P}$.

III. SYNTHETIC EXPERIMENTS

Four flow systems designed to test the flexibility of the framework and it scapacity to identify concise and descriptive differentation equation models from broadly spaced and noisy data were considered. For each of these flow systems, a physics informed argument set, $\mathcal{P}$, a parameter augmentation operator, aug^{θ}, and a primitive set $\mathcal{M}$, helped to build human-interpretable models. The initial and boundary conditions for all of the experiments are shown below:

$$\begin{aligned} \mathbf{x}(t,z) &= \mathbf{0} \text{ at } t = 0 \\ \mathbf{x}(t,z) &= \mathbf{x}_f \text{ at } z = 0 \\ \frac{\partial \mathbf{x}(t,z)}{\partial z} &= \mathbf{0} \text{ at } z = l \end{aligned} \tag{4}$$

with $l = 1$ representing the length of the system, and $\mathbf{x}_f$ representing the system's feed condition. We tasked the framework with discovering the set of expressions, $\mathbf{s}$, using data collected at $z = l$ from synthetic experiments with added Gaussian noise.

A. Plug Flow Reactor

Experiment A tested the framework on an ideal, isothermal plug flow reactor (PFR) home to a single, irreversible chemical reaction, $2A \rightarrow B$. A PFR is a chemical reactor in which three phenomena drive the changes to the concentration of a chemical species: bulk fluid flow, diffusion of the species through the bulk fluid, and a chemical reaction. The state variable of this PFR is the concentration of A. The argument set, $\mathcal{P}_A = \{x, \frac{\partial x}{\partial z}, \frac{\partial^2 x}{\partial z^2}, x_f, v\}$, can describe a chemical reaction, convection, diffusion, feed concentration of A, and flow velocity through the reactor. The parameter augmentation operator, aug^{θ}_{A}, multiplies each term in $\mathcal{P}_A$ by a parameter and raises the term used to describe a chemical reaction expressed as x to the power of a parameter. The primitive set, $\mathcal{M}_A = \{+, \times, -\}$, allowed the framework to add, subtract, and multiply terms. The experimental feed conditions, x_f and v are provided in TABLE I.

	System		System
	Admissible Inputs		Measured Outputs
Experiments	x_f	v	y
A-1	1.00	0.50	$x(t,z)$ at $z = l$
A-2	2.00	0.55	

TABLE I

EXPERIMENT A

B. Continuously Stirred Tank Reactor

The synthetic system in experiment B is an ideal, isothermal continuously stirred tank reactor (CSTR) home to the same chemical reaction of experiment A. A CSTR is a homogeneous chemical reactor in which the concentration of a chemical species depends on the influent and effluent bulk fluid velocity and a chemical reaction. Again, the state variable is the concentration of A. The argument set, parameter augmentation operator, and primitive set for experiment B are identical to those for experiment A. The values for x_f and v are provided in TABLE II.

	System		System
	Admissible Inputs		Measured Outputs
Experiments	x_f	v	y
B-1	2.00	1.00	$x(t,z)$ at $z = l$
B-2	1.00	2.00	
B-3	2.00	0.55	

TABLE II

EXPERIMENT B

C. Burgers' Equation

Burgers' equation models the synthetic system of experiment C, and the state variable is the fluid velocity. Burgers' equation is a nonlinear PDE that describes dynamic viscous fluid convection and diffusion in one spatial dimension. The argument set, $\mathcal{P}_C = \{x, x^2, \frac{\partial x}{\partial z}, \frac{\partial^2 x}{\partial z^2}, x_f\}$, contains terms that can describe the fluid velocity, convection, diffusion, and the feed flow velocity. The parameter augmentation operator, aug^{θ}_{C}, multiplies each term in $\mathcal{P}_C$ by a parameter. The values for x_f are provided in TABLE III.

	System	System
	Admissible Inputs	Measured Outputs
Experiments	x_f	y
C-1	3.00	$x(t,z)$ at $z = l$
C-2	1.50	

TABLE III

EXPERIMENT C

D. Plug Flow Reactor with Reversible Reaction

The synthetic system in experiment D is a PFR with a reversible chemical reaction $2A \rightleftharpoons B$. In this experiment, there are two state variables: x_1 is the concentration of A, and x_2 is the concentration of B. The two primitive sets are identical. $\mathcal{M}_{D,1} = \mathcal{M}_{D,2} = \mathcal{M}_A$. The argument set for the first expression, $\mathcal{P}_{D,1} = \{x_1, x_1^2, x_2, x_2^2, \frac{\partial x_1}{\partial z}, \frac{\partial^2 x_1}{\partial z^2}\}$, contains terms that can describe a chemical reaction, and convection and diffusion of A through the reactor. The argument set for the second expression, $\mathcal{P}_{D,2} = \{x_1, x_1^2, x_2, x_2^2, \frac{\partial x_2}{\partial z}, \frac{\partial^2 x_2}{\partial z^2}\}$, is similar to $\mathcal{P}_{D,1}$, but describes the convection of diffusion of

B through the reactor rather than A. The parameter augmentation operator, aug_D^θ is identical to aug_C^θ. The experimental pairs of $\mathbf{x}_f$ are provided in TABLE IV.

	System Admissible Inputs		System Measured Outputs
Experiments	$x_{f,1}$	$x_{f,2}$	y
D-1	1.50	0.00	$x(t,z)$ at $z=l$
D-2	1.00	0.00	
D-2	0.50	0.10	

TABLE IV

EXPERIMENT D

IV. RESULTS AND DISCUSSION

A. Plug Flow Reactor

Experiment A tested the framework on a system that is modeled by a quasi-linear PDE. To successfully identify the PFR equation, the framework needed to linearly combine the convection, diffusion, and reaction terms. The framework also needed to correctly identify nonlinear reaction term, $x^{2.00}$. The model discovered by the framework matches the model used to generate the synthetic data and both are shown in TABLE V and in Fig. 1.

Conclusion 1: The framework can identify quasi-linear models from broadly spaced and noisy data without predefined exponent terms.

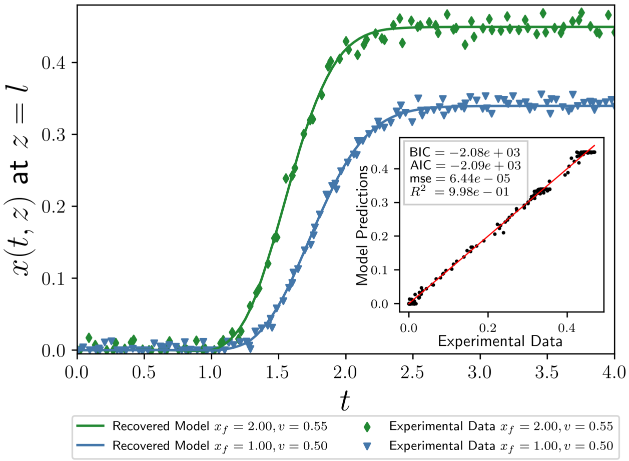

Fig. 1. The discovered model for x at the outlet of the PFR compared to the synthetic data from experiment A. The inset parity plot shows strong agreement between the discovered model and the experimental data.

B. Continuously Stirred Tank Reactor

Experiment B tested the framework's ability to disregard incorrect physics. The set up for experiment B was identical to the set up for experiment A, but the data was generated by the CSTR model rather than the PFR model. To correctly identify the CSTR model, the framework needed to determine that the convection and diffusion terms in $\mathcal{P}_B$ were not useful in describing the CSTR. The model discovered by the framework matches the model used to generate the synthetic data and both are shown in TABLE V and in Fig. 2.

Conclusion 2: The framework can reject physics in the physics-informed argument set if those physics are not useful in describing a system model.

To correctly identify the CSTR model, the framework also needed sufficiently rich data. When data from only two of the three experiments were provided to the framework, it found models of equal or lesser complexity that agree with the data just as well as the underlying model. It was not until the framework was given data from all three of the experiments shown in TABLE II that it could discover the underlying CSTR model. Even with all three experiments, the framework still struggles to correctly identify the CSTR model and its parameters, evident in the difference of parameters in TABLE II between the generating model and the recovered model. This is an issue of model identifiability, which can be explored with methods such as those discussed in Han et al. [25].

Conclusion 3: The framework can find a model consistent with the data and equally or less complex than the underlying model when the broadly spaced and noisy data are not sufficiently rich to identify the underlying model.

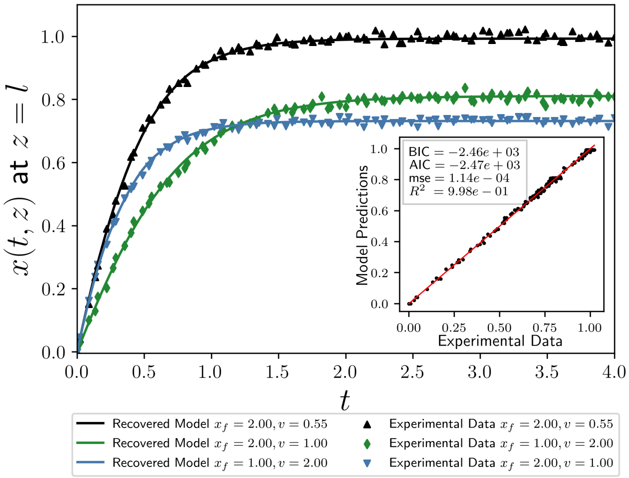

Fig. 2. The discovered model for x at the outlet of the CSTR compared to the synthetic data from experiment B. The inset parity plot shows strong agreement between the discovered model and the experimental data.

C. Burgers' Equation

Experiments A and B tested the framework's ability to identify a quasi-linear PDE model and an ODE model. Experiment C tested the framework's ability to identify a nonlinear PDE model without any predefined combinations of terms. The framework successfully identified the nonlinear convection term. The model used to generate the synthetic data and the model discovered by the framework are shown in TABLE V and in Fig. 3.

Conclusion 4: The framework can identify nonlinear combinations of terms from the argument set from broadly

Experiment	True Expression	Recovered Expression
A - PFR	$0.01\frac{\partial^2 x}{\partial z^2} - 1.00v\frac{\partial x}{\partial z} - 1.00x^{2.00}$	$0.01\frac{\partial^2 x}{\partial z^2} - 0.99v\frac{\partial x}{\partial z} - 0.99x^{1.95}$
B - CSTR	$1.00v(1.00x_f - 1.00x^{1.00}) - 1.00x^{2.00}$	$1.00v(0.99x_f - 1.03x^{1.12}) - 0.96x^{1.98}$
C - Burgers'	$0.01\frac{\partial^2 x}{\partial z^2} - 0.10x\frac{\partial x}{\partial z}$	$0.01\frac{\partial^2 x}{\partial z^2} - 0.10x\frac{\partial x}{\partial z}$
D - PFR Rev. Rxn	$0.01\frac{\partial^2 x_1}{\partial z^2} - 0.50\frac{\partial x_1}{\partial z} - 0.9x_1^2 + 0.2x_2$ $0.01\frac{\partial^2 x_2}{\partial z^2} - 0.50\frac{\partial x_2}{\partial z} + 0.45x_1^2 - 0.1x_2$	$0.01\frac{\partial^2 x_1}{\partial z^2} - 0.50\frac{\partial x_1}{\partial z} - 0.71x_1^2$ $0.01\frac{\partial^2 x_2}{\partial z^2} - 0.50\frac{\partial x_2}{\partial z} + 0.35x_1^2$

TABLE V
RECOVERED MODELS

spaced and noisy data without prior term combination definitions.

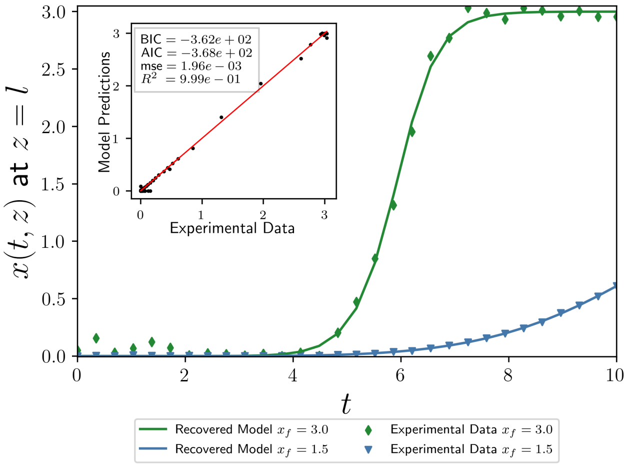

Fig. 3. The discovered model for x at the outlet of the system described by Burgers' equation compared to the synthetic data from experiment C. The inset parity plot shows strong agreement between the discovered model and the experimental data.

D. Plug Flow Reactor with Reversible Reaction

Each of the other experiments identifies a differential equation model when $n_s = 1$. Experiment D tests the framework by tasking it with identifying a PDE model when $n_s = 2$. The framework failed to identify the exact generating expressions but still discovered an informative model. It correctly identified the flow terms of the PFR and their parameters. The first parameter, 0.01, is the correct coefficient of diffusivity, and the second parameter, 0.50, is the correct flow velocity through the reactor. The framework also correctly discovered conservation of mass. Although the domain expert knowledge provided to the framework did not include conservation of mass, the framework identified a reaction scheme that correctly conserved mass. Finally, although the discovered model is incorrect, it approximates the available data with fewer terms than the underlying model. The framework may need better experimental design or additional sensor locations to uniquely identify the underlying model. Both the discovered model and the model used to generate the synthetic data are provided in TABLE V and Fig. 4 and Fig. 5.

Conclusion 5: The framework can find a model consistent with physics even if it fails to identify the underlying physics model of broadly spaced and noisy data.

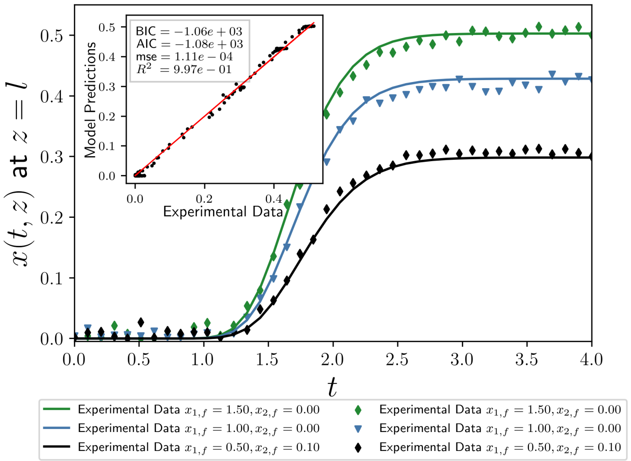

Fig. 4. The discovered model for x_1 at the outlet of the PFR compared to the synthetic data from experiment D. The inset parity plot shows strong agreement between the discovered model and the experimental data.

E. Discussion

The results in this work demonstrate several advantages of using the framework to discover PDE models. The framework can combine domain expert knowledge with broadly spaced and potentially noisy data to identify underlying physics or surrogate models. It can help identify PDE models of systems when the state variable is difficult to measure in frequent intervals in space. The method is also not constrained to predefined combinations of the argument set. SR via GP can combine terms flexibly to create expressions that are not linear combinations of the argument set.

Although the framework can discover PDE models, it is computationally expensive. The framework can be parallized to reduce time, but takes on the order of half an hour to find simple models like the PFR and Burgers' equation models and two hours to find the reversible reaction model using eight threads running on an Intel Xeon processor and 32 GB. The algorithm is executed in python version 3.9.7. The computational cost grows with model complexity and symbol space size. This cost growth can become intractable as the complexity of a model or the number of variables increases. The framework is also limited in its integration scheme. If an

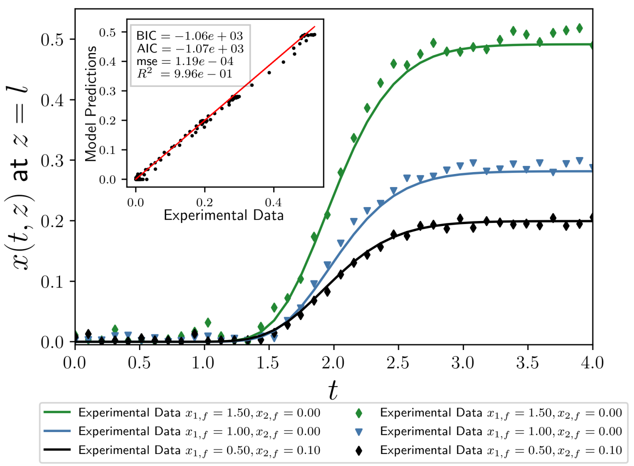

Fig. 5. The discovered model for x_2 at the outlet of the PFR compared to the synthetic data from experiment D. The inset parity plot shows strong agreement between the discovered model and the experimental data.

integration scheme cannot integrate a proposed expression, then the fitness of that expression may not reflect its actual fitness. Similarly, an inadequate parameter estimation scheme can effect the framework's ability to find models. More robust integration and parameter estimation methods may help to overcome this problem, but often at the cost of additional computational expense.

V. Conclusions and Future Work

A novel framework for the discovery of partial differential equation models from broadly spaced and noisy data is introduced. This framework can leverage data and domain expert knowledge to generate clear, concise, and human-interpretable differential equation models. It is flexible enough to identify free-form expressions and find nonlinear parameters that are not specified a-priori. When experimental data was not sufficiently rich, the framework identified a surrogate model that described the data in fewer mathematical terms than the underlying model.

This work presents simple test cases, and further investigation of the framework is warranted. Better design of experiments can aid the framework in correct model discovery. Information theory and optimal design of experiments can improve model and parameter identifiability. The practical and theoretical limitations of the framework must also be considered. In future work, the affect of system complexity on the quality of the models discovered by the framework will be explored. Exploration of alternative methods in SR and parameter estimation may also be valuable in broadening the complexity of models discoverable by this tool.

VI. Acknowledgments

This project was sponsored by the Pratt & Whitney Institute of Advanced Systems Engineering (P&W-IASE) of the University of Connecticut and Pratt & Whitney. Any opinions expressed herein are those of the authors and do not represent those of the sponsor.

References

[1] S. S. Haykin and S. S. Haykin, *Neural networks and learning machines*. Prentice Hall/Pearson, 2009.

[2] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[3] H. chaandar Ravichandar, I. Salehi, B. P. Baillie, G. M. Bollas, and A. Dani, "Learning stable nonlinear dynamical systems with external inputs using gaussian mixture models." IEEE, 6 2018, pp. 4825–4830.

[4] K. McBride and K. Sundmacher, "Overview of surrogate modeling in chemical process engineering," *Chemie Ingenieur Technik*, vol. 91, pp. 228–239, 3 2019.

[5] S. Piche, B. Sayyar-Rodsari, D. Johson, and M. Gerules, "Nonlinear model predictive control using neural networks," *IEEE Control Systems*, vol. 20, pp. 53–62, 6 2000.

[6] B. Beykal, N. A. Diangelakis, and E. N. Pistikopoulos, "Continuous-time surrogate models for data-driven dynamic optimization," in *Computer Aided Chemical Engineering*. Elsevier, 2022, vol. 51, pp. 205–210.

[7] A. Sobester, A. Forrester, and A. Keane, *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons L.td., 2008.

[8] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, "Integrating scientific knowledge with machine learning for engineering and environmental systems," 3 2020.

[9] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, pp. 422–440, 6 2021.

[10] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2 2019.

[11] Z. Chen, Y. Liu, and H. Sun, "Physics-informed learning of governing equations from scarce data," *Nature Communications*, vol. 12, p. 6136, 12 2021.

[12] Z. T. Wilson and N. V. Sahinidis, "The alamo approach to machine learning," 5 2017.

[13] S. L. Brunton, J. L. Proctor, J. N. Kutz, and W. Bialek, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 113, pp. 3932–3937, 2016.

[14] H. Vaddireddy, A. Rasheed, A. E. Staples, and O. San, "Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data," *Physics of Fluids*, vol. 32, p. 015113, 1 2020.

[15] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Science Advances*, vol. 3, 4 2017.

[16] M. Hoffmann, C. Fröhner, and F. Noé, "Reactive sindy: Discovering governing reactions from concentration data," *The Journal of Chemical Physics*, vol. 150, p. 025101, 1 2019.

[17] K. Kaheman, J. N. Kutz, and S. L. Brunton, "Sindy-pi: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 476, p. 20200279, 10 2020.

[18] D. A. Messenger and D. M. Bortz, "Weak sindy for partial differential equations," 7 2020.

[19] J. R. Koza, *Genetic Programming On the Programming of Computers by Means of Natural Selection*. Bradford, 1992.

[20] S.-M. Udrescu and M. Tegmark, "Ai feynman: A physics-inspired method for symbolic regression," *Science Advances*, vol. 6, 4 2020.

[21] J. Bongard and H. Lipson, "Automated reverse engineering of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 104, pp. 9943–9948, 6 2007.

[22] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, pp. 81–85, 4 2009.

[23] T. W. Cornforth and H. Lipson, "Inference of hidden variables in systems of differential equations with genetic programming," *Genetic Programming and Evolvable Machines*, vol. 14, pp. 155–190, 6 2013.

[24] Y. Chen, Y. Luo, Q. Liu, H. Xu, and D. Zhang, "Symbolic genetic algorithm for discovering open-form partial differential equations (sga-pde)," *Physical Review Research*, vol. 4, p. 023174, 6 2022.

[25] L. Han, Z. Zhou, and G. M. Bollas, "Model-based analysis of chemical-looping combustion experiments. part i: Structural identifiability of kinetic models for nio reduction," *AIChE Journal*, vol. 62, pp. 2419–2431, 7 2016.