

Integrating Hybrid Modeling and Multifidelity Approaches for Data-Driven Process Model Discovery

Suryateja Ravutla^a, and Fani Boukouvala^{a*}

^a Department of Chemical and Biomolecular Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA

* Corresponding Author: fani.boukouvala@chbe.gatech.edu

ABSTRACT

Modeling the non-linear dynamics of a system from measurement data accurately is an open challenge. Over the past few years, various tools such as SINDy and DySMHO have emerged as approaches to distill dynamics from data. However, challenges persist in accurately capturing dynamics of a system especially when the physical knowledge about the system is unknown. A promising solution is to use a hybrid paradigm, that combines mechanistic and black-box models to leverage their respective strengths. In this study, we combine a hybrid modeling paradigm with sparse regression, to develop and identify models simultaneously. Two methods are explored, considering varying complexities, data quality, and availability and by comparing different case studies. In the first approach, we integrate SINDy-discovered models with neural ODE structures, to model unknown physics. In the second approach, we employ Multifidelity Surrogate Models (MFSMs) to construct composite models comprised of SINDy-discovered models and error-correction models.

Keywords: Data-driven modeling, Model identification, Hybrid modeling, Multifidelity, Sparse regression

INTRODUCTION

Present modeling approaches of intricate dynamic systems rely on ordinary and/or partial differential equations (ODEs, PDEs) to describe their behaviors. These governing equations are conventionally obtained from rigorous first principles like conservation laws or derived from phenomenological knowledge-based approaches. However, many dynamic systems remain unexplored, lacking comprehensive analytical descriptions. Exploiting advances in data acquisition, digitization, and storage, data-driven 'black box' models have emerged as an alternative [1, 2]. These data-driven methods excel in regression and classification tasks, yet their resultant black-box nature commonly lacks physical insight and exhibits limitations in extrapolation beyond the training data's boundaries. Certain systems exist, such as biological processes or intricate design problems, that necessitate a deeper understanding of governing equations. Consequently, recent developments focus on the integration of data-driven techniques into modeling system dynamics. Early attempts utilized symbolic regression [3] and genetic programming algorithms [4]. However,

challenges like overfitting and the computational demands arising from their combinatorial nature limited their applicability to low-dimensional systems and small initial candidate sets of symbolic expressions. An alternative approach called SINDy [5] was proposed, that reconstructs the underlying equations based on a large-space library of candidate terms and transforming the discovery problem to sparse regression and over time, several extensions to SINDy and alternative approaches on similar ideas were also introduced [6-9]. More recently, another such approach (DySMHO) [10] was proposed that uses moving horizon optimization for identifying the governing equations. Nevertheless, the effectiveness of these techniques relies on the identification and selection of terms from an array of potential candidate terms. While an exhaustive pool covering all potential terms might aid in building a highly accurate mechanistic model, this presents a challenge, especially in cases involving complex nonlinear systems and when the system dynamics are not fully known. A potential solution is to use hybrid modeling techniques with these methods. These hybrid models (HMs), also referred to as 'grey-box' models have the advantages of both mechanistic and the

black-box models [1, 11]. Figure 1 shows the comparison between the mechanistic, grey- and black-box models. Early implementations of these hybrid models can be found in works that date back to 1980's.

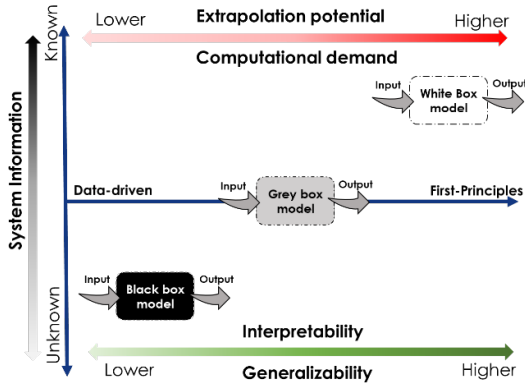


Figure 1: A comparison between white-box/mechanistic, grey-box and black-box models.

In this work, we combine a hybrid modeling paradigm with sparse regression, with the goal of simultaneous hybrid model development and model identification. Specifically, in our study, we investigate two distinct methods. In the first method, we employ the SINDy formulation to establish the initial physics-based model from the incomplete candidate library. Subsequently, we integrate this with neural ordinary differential equations (NODEs) [12-14], to improve the accuracy of the final model. Leveraging the NODE formulation to model unknown or missing physics within a mechanistic model has previously proven to be successful [12, 15, 16]. For the second approach, we employ composite structures known as Multifidelity Surrogate Models (MFSMs) [17, 18] using true data/high-fidelity (HF) and the low-fidelity (LF) data. The model output from SINDy constructed using the incomplete candidate library is treated as the LF model. In the subsequent step, we develop MFSMs that use HF and LF data to refine the model accuracy.

In the following sections of this paper, we introduce the methods for our proposed HM approaches and a workflow to build the HMs. In the subsequent sections, we utilize two non-linear case studies to test our HM models and show their prediction accuracy. Furthermore, we show our analysis on the impact of sampled data density and noise on the accuracy of the HMs, as well as the HMs extrapolation capabilities.

2. METHODS

2.1 Sparse identification of nonlinear dynamics

In [6], the authors leveraged the fact that most physical systems have only a few relevant terms that

define the dynamics, making the governing equations sparse in a high-dimensional nonlinear function space. Consider the example first-order ODE system with n states which are denoted by $\mathbf{X}$. We denote the derivative with respect to time for $\mathbf{X}$ as $\mathbf{X}'$. The right-hand side of the ODE is given by the derivative with respect to time, t , denoted by the function f ,

$$\mathbf{X}' = \frac{d\mathbf{X}}{dt} = f(\mathbf{X}) \quad (1)$$

Next, a library $\theta(\mathbf{X})$, consisting of candidate nonlinear functions of the columns of $\mathbf{X}$ is constructed. For example, the library may consist of constant, polynomial, and trigonometric terms. Finally, a sparse regression problem is set up to determine the sparse vectors of coefficients $\mathcal{E} = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n]$, which determines the nonlinearities that are active.

$$\theta(\mathbf{X}) = \begin{bmatrix} 1 & \mathbf{X} & \mathbf{X}^2 & \mathbf{X}^3 & \dots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \dots \end{bmatrix} \quad (2)$$

$$\mathbf{X}'_{SINDy} = \theta(\mathbf{X})\mathcal{E} \quad (3)$$

$$\mathbf{X}_{SINDy} = \text{ODEsolver}(\theta(\mathbf{X})\mathcal{E}, \mathbf{X}_0, t) \quad (4)$$

Discovering the mechanistic equations from state data with SINDy is subject to a) estimating the derivatives accurately with data limitations and b) formulating the candidate library $\theta(\mathbf{X})$ to contain all the terms that could potentially form an accurate mechanistic equation. While several extensions to SINDy have been proposed over the years to address the former issue, the latter is still a challenge. This is especially a major issue when we are dealing with state data from non-linear systems where the potential terms in the candidate library are not known. To address this, we propose to use the following hybrid modeling approaches.

2.2 Neural Ordinary Differential Equations for Error Correction

Neural networks (NNs) are one of the widely used ML models in data-driven modeling due to their ability to approximate complex nonlinear relationships. Since the early 90s, NNs have been used to model dynamic systems within differential equations [19]. Recently, NODEs have emerged, integrating NNs with automatic differentiation tools [12-14]. NODEs predict system derivatives directly during training, capturing both state and derivative data. The potential of this approach was shown in better capturing curvature in dynamic data when compared to data-driven models that ignore derivative information. A neural ODE is essentially a NN used to model $f(\mathbf{X})$ from Eq (1). We denote the modeled derivative by $\mathbf{X}'_{NN}$. We can obtain the predicted ODE solution $\mathbf{X}_{NN}$ by employing any preferred ODE solver.

$$\mathbf{X}'_{NN} = \frac{d\mathbf{X}_{NN}}{dt} = NN(\mathbf{X}) \quad (5)$$

$$\mathbf{X}_{NN} = \text{ODEsolver}(\mathbf{NN}, \mathbf{X}_0, t) \quad (6)$$

Training the NODE can be done either by minimizing the error between the predicted and true states $\text{Error} = \text{MSE}(\mathbf{X}, \mathbf{X}_{NN})$ by passing the NN through an ODE solver [14] or, utilize a collocation-based approach [13] and avoid using explicit ODE solver by minimizing the loss function $\text{Error} = \text{MSE}(\mathbf{X}'_{est}, \mathbf{NN}(\mathbf{X}_{est}))$. In [13], the authors show that by using the collocation based approach, training a NODE is faster. We utilize this collocation-based approach to train our HM with NODE formulation for model identification from state data. In the first step, we generate a low-fidelity (LF) mechanistic model using SINDy by assuming few terms in the candidate library $\theta(\mathbf{X})$. In the next step, we correct the error resulting from the LF model with a NODE. We utilize the following formulation shown in Eq (7).

$$\mathbf{X}'_{HM} = \frac{d\mathbf{X}_{HM}}{dt} = \theta(\mathbf{X})\mathbf{E} + \mathbf{NN}(\mathbf{X}) \quad (7)$$

$$\mathbf{X}_{HM} = \text{ODEsolver}(\theta(\mathbf{X})\mathbf{E} + \mathbf{NN}(\mathbf{X}), \mathbf{X}_0, t) \quad (8)$$

To train the NODE we take the collocation-based approach and minimize the error between the derivatives directly. In the next step, we calculate the difference between the derivatives values between the HF and the LF derivatives at the sampled data. This derivative difference corresponds to the mismatch between the true model and the LF model. A NN model is then trained to predict this difference when given the true state data. Finally, the hybrid model states $\mathbf{X}_{HM}$ are estimated by using an ODE solver.

2.3 Multifidelity Surrogate Models for Error Correction

In recent years, hybrid composite structures that can learn from both HF and LF data were proposed to improve the LF model predictions by correcting the error between the HF and LF data. These composite structures are referred to as multi-fidelity surrogate models (MFSMs) [17, 18, 20]. A widely used structure of MFSMs is $y_H = \rho(x)y_L + \delta(x)$, where y_L, y_H represent the low and high-fidelity data respectively, $\rho(x)$ is multiplicative correlation surrogate and $\delta(x)$ is the additive surrogate. It can be re-written as $y_H = F(x, y_L)$. To establish a connection between the HF and LF data, it's necessary to have both a HF model and a LF model that can produce this data. Similar to the NODE model correction approach, we use SINDy to obtain the LF model. The LF model from SINDy is then integrated using an ODE solver to obtain $\mathbf{X}_{SINDy}$ using Eq (4). We then formulate the MFSM by utilizing a NN to model the error. The NN model is trained to minimize the loss function $\text{Error} = \text{MSE}(\mathbf{X}, \mathbf{X}_{MFSM})$, and takes both time and LF states as input to predict $\mathbf{X}_{MFSM}$.

$$\mathbf{X}_{MFSM} = \mathbf{NN}(t, \mathbf{X}_{SINDy}) \quad (9)$$

2.3 Workflow for constructing hybrid models

We utilize the workflow shown in Table (1) for constructing the hybrid models under approach 1 and approach 2. Here $\mathbf{X}_{HF}$ represent the true state data. $\beta_1, \beta_2, \beta_3$ are the regularization coefficients.

Table 1: Workflow for constructing the hybrid models to correct error for NODE and MFSM formulations.

Let $[t, \mathbf{X}_{HF}]$ be the complete HF dataset	
Generate SINDy model	1. Set $t \leftarrow$ Input and $\mathbf{X}_{HF} \leftarrow$ HF output 2. Estimate the derivatives $\mathbf{X}'_{HF}$ from $[t, \mathbf{X}_{HF}]$ data. 3. Generate the feature library $\theta(\mathbf{X})$ and set the optimizer While termination criteria not true: $\min(\mathbf{X}'_{HF} - \theta(\mathbf{X})\mathbf{E} ^2) + \beta_1 (\mathbf{E})$ Tune $\mathbf{E}$
Set $\mathbf{X}'_{LF} = \theta(\mathbf{X})\mathbf{E}$ and generate data, $\mathbf{X}_{LF} \leftarrow$ LF output.	
If Approach 1: Correct LF model error with NODE formulation	1. Initialize NN, Set $\mathbf{X}_{HF} \leftarrow$ Input and $\Delta\mathbf{X}' = \mathbf{X}'_{HF} - \mathbf{X}'_{LF} \leftarrow$ output 2. While termination criteria not true: $\min(\Delta\mathbf{X}' - \mathbf{X}'_{NN} ^2) + \beta_2 \ \Phi_{NN}\ _2$ Set $\mathbf{X}'_{HM} = \theta(\mathbf{X})\mathbf{E} + \mathbf{NN}(\mathbf{X}) \leftarrow$ final model
If Approach 2: Correct LF model error with MFSM formulation	1. Initialize a NN, Set $[t, \mathbf{X}_{LF}] \leftarrow$ Input and $\mathbf{X}_{HF} \leftarrow$ output 2. While termination criteria not true: $\min(\mathbf{X}_{MFSM} - \mathbf{X}_{HF} ^2) + \beta_3 \ \Phi_{NN}\ _2$ Set $\mathbf{X}_{MFSM} = \mathbf{NN}(t, \mathbf{X}_{LF}) \leftarrow$ final model

3. RESULTS AND DISCUSSION

In this section, we first introduce the case studies that we use to test and compare the approaches from 2.1 – 2.3. In the subsequent sections, we show the analysis on the model accuracy, effect of density of data and noise on the model accuracy and the extrapolation ability of the models.

3.1 Case Studies

To test the two approaches, we utilize two non-linear case studies. The first case study is a non-isothermal continuously stirred tank reactor (CSTR) problem, and the second case study is a penicillin biosynthesis problem.

3.1.1 The non-isothermal CSTR problem

The CSTR's governing equation, Eq (11), represents mass conservation. But, in non-isothermal operating scenarios which is common in practical applications, the energy balance must also be considered. Thus, temperature is added as an extra state variable, along with obtaining temperature measurements in addition to composition data, as depicted in Eq (12). We utilize the model equations and assumptions from [10].

$$\frac{dC_A}{dt} = \frac{q}{V} (C_{A,i} - C_A) - k_0 e^{-\frac{E_a}{RT}} C_A \quad (11)$$

$$\frac{dT}{dt} = \frac{q}{V}(T_i - T) + \frac{(-\Delta H_R)}{\rho C} k_0 e^{-\frac{E_a}{RT}} C_A(t) + \frac{UA}{V\rho C}(T_c - T) \quad (12)$$

3.1.2 The penicillin biosynthesis problem

For the second case study, we chose to model the production of penicillin via yeast fermentation. The level of nonlinearity in the system differs significantly between state variables. The process is modeled by four differential equations on the following states: volume (V), concentrations of biomass (B), product (P) and substrate (S). The system of ODEs is defined as shown in Eq (14) and Eq (15). The parameter values have been taken from [12].

$$\begin{aligned} \frac{dB}{dt} &= B(\mu - D - c_L) \\ \frac{dS}{dt} &= -\sigma B + (S_f - S)D \\ \frac{dP}{dt} &= q_p B - P(D + c_1 k) \\ \frac{dV}{dt} &= F \end{aligned} \quad (13)$$

$$\begin{aligned} \mu &= \frac{\mu_m S}{K_x X + 10} \\ \sigma &= \frac{\mu}{Y_x} + \frac{q_p}{Y_p} + m_x \\ q_p &= \frac{1.5 q_{pm} S B}{4 k_p + B S \left[1 + \frac{S}{3 k_i} \right]} \\ c_L &= \frac{c_{Lmax} B \exp\left(-\frac{S}{100}\right)}{K_L + B + 1} \\ m_x &= m_{xm} \frac{B}{B + 10} \end{aligned} \quad (14)$$

3.2 Analysis with no noise in true state data

The case study shown in 3.1.1 and 3.1.2 were simulated with the initial conditions $[C_{A,0}, T_0] = [0.5 \frac{\text{mol}}{L}, 350K]$ and $[B_0, S_0, P_0, V_0] = [5 \frac{g}{L}, 525 \frac{g}{L}, 0 \frac{g}{L}, 0.2L]$ respectively, to generate 50 HF data points for each case. In the next step, the workflow shown in Table (1) was utilized to generate the Hybrid model with NODE and MSFM formulations to correct the error from SINDy – LF model. The results for both the approaches are shown below. Figure (2) shows the prediction using hybrid models with NODE MFSM formulation for case study 3.1.1 and Figure (3) shows the results for case study 3.1.2. In both Figures (2) and (3), the solid blue dots represent true (HF) state data. The green dashed line represents SINDy-LF model predictions. The solid orange line and the dotted red lines represent the HM model predictions with NODE and MFSM formulations respectively.

Figures (2) and (3) illustrate the limitation of the constructed SINDy model in accurately predicting the true states. This mismatch between the HF state data and the SINDy model stems from the candidate library's inability to comprehensively encompass all potential nonlinear terms contributing to the final mechanistic model equation. Consequently, the mechanistic model derived lacks accuracy in predicting the HF states. However, the HMs built using this SINDy model as the LF model,

employing both NODE and MFSM formulations, demonstrate the ability to accurately predict the HF state profiles. This HM structure effectively compensates for mismatch and missing terms within the LF model.

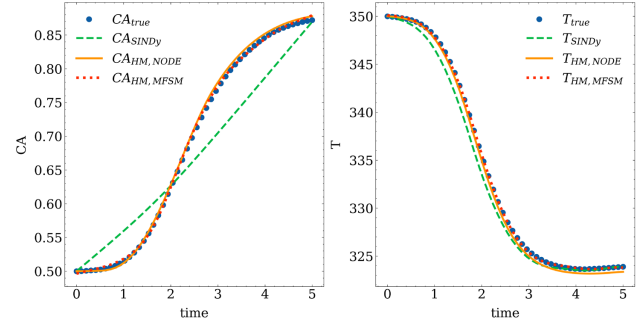


Figure 2: Concentration (C_A) and temperature (T) profiles from SINDy model, HM-NODE and HM-MFSM formulation for case study 3.1.1, compared with true state data.

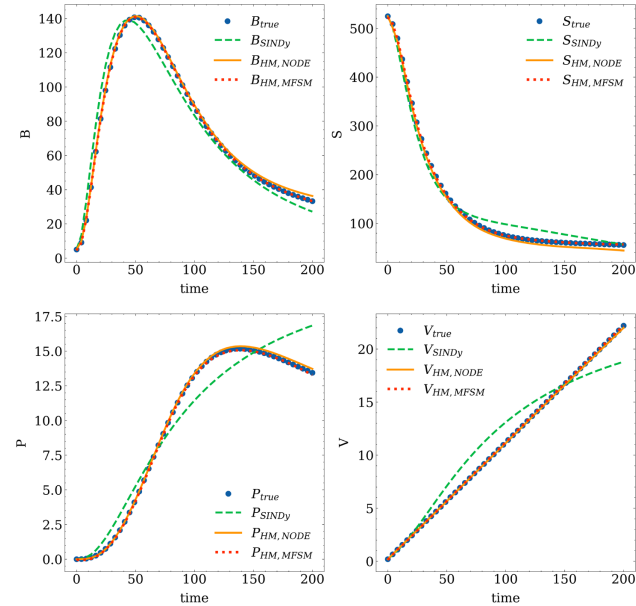


Figure 3: Volume (V), Biomass (B), product (P) and substrate (S) profiles from SINDy model, HM-NODE and HM-MFSM formulation for case study 3.1.2 with 50 HF data points, compared with true state data.

3.3 Effect of density of sampled data on HMs

To analyze the effect of density of sampled data on the HM model, we repeated the experiment from 3.2, by decreasing the amount of HF data available. For this analysis, we reduced the data size to 30 HF samples. The case study shown in 3.1.1 and 3.1.2 were simulated with the initial conditions in 3.2 to generate 30 HF data points. In the next step, using the workflow from Table (1) HMs are built. Figures (4) and (5) show the prediction from the SINDy and HMs.

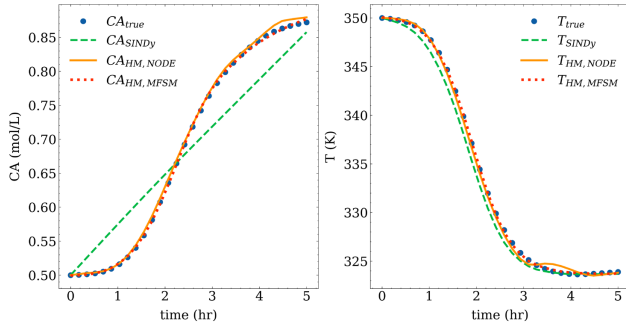


Figure 4: Concentration (C_A) and temperature (T) profiles from SINDy model, HM-NODE and HM-MFSM formulations for case study 3.1.1, with 30 HF datapoints.

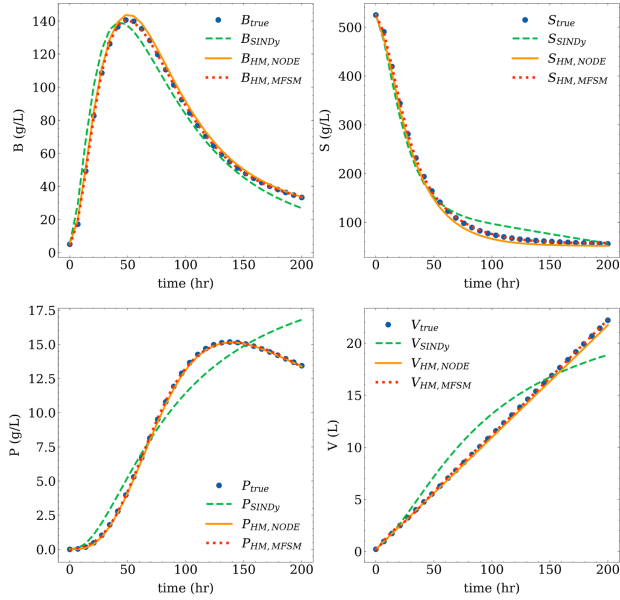


Figure 5: Volume (V), Biomass (B), product (P), substrate (S) profiles from SINDy model, HM-NODE and HM-MFSM formulations for case study 3.1.2, with 30 datapoints

We repeat the analysis once again, but this time by reducing the data size to 20 HF samples. We simulate the case studies from 3.1.1 and 3.1.2 with the same initial conditions to generate 20 HF data points to train the SINDy and HM models. Figures (6) and (7) show the prediction from the SINDy model and HMs.

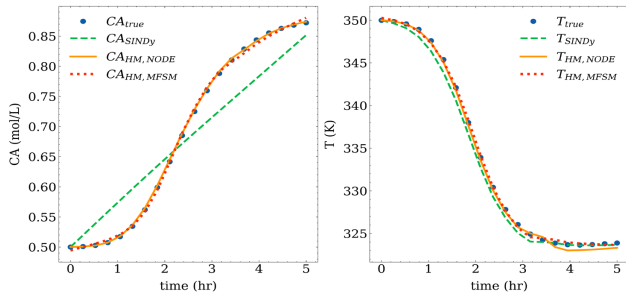


Figure 6: Concentration (C_A) and temperature (T) profiles from SINDy model, HM-NODE and HM-MFSM

formulations for case study 3.1.1, with 20 HF datapoints.

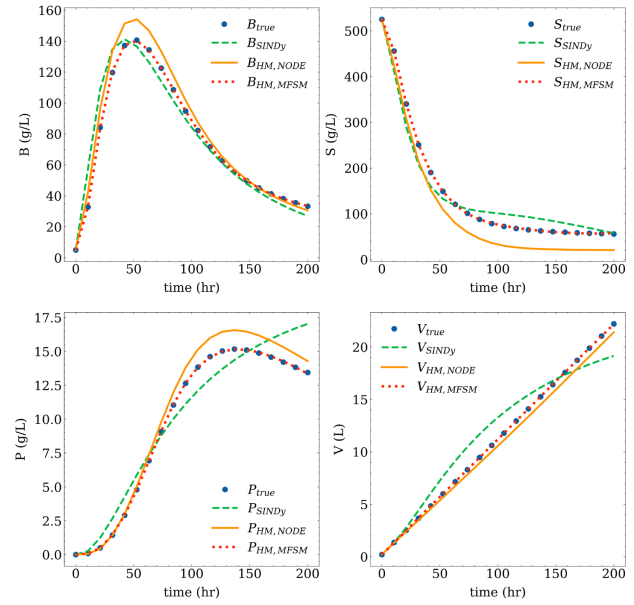


Figure 7: Volume (V), Biomass (B), product (P), substrate (S) profiles from SINDy model, HM-NODE and HM-MFSM formulations for case study 3.1.2, with 20 HF datapoints

Figures (4-7) show that the HMs constructed with both NODE and MFSM approaches are robust to the low densities of sampled data and can still predict the true states with decent accuracy and capture the profile trends better than the SINDy model. We can also notice that the decrease in density of the data affects the model accuracy, and we start to observe a slight mismatch in HM predictions, and this increases as the density of sampled data decreases. Among the two HM approaches, the HM-MFSM approach performs marginally better than the HM-NODE approach. This is because we take a two-step approach for correcting the error, and decreasing the density of data affects the LF model in step 1. Consequently, the LF model directly affects the NODE approach which is aimed at correcting the derivative space error and results in low errors as we integrate it forward in time for the state profiles. On the other hand, we see an improved fit with MFSMs in the training range. This is because the MFSM approach corrects the error in the state space and does not need integrating the states forward in time.

3.3 Extrapolation of the HMs

To analyze the extrapolation ability of the constructed HMs, we show the model predictions with a test dataset that contains input to the HM model from outside the training dataset region. We also compare the results with the SINDy model we obtained in the first step and with decreasing data density. Figure (8) shows three scenarios aimed at illustrating extrapolation using HMs for the case study 3.1.1. Specifically, Figures (8A, 8B, 8C)

correspond to instances with 50, 30, 20 HF samples, respectively. It is evident that HMs utilizing both NODE and MFSM formulations demonstrate better extrapolation capabilities. On the other hand, predictions obtained from the SINDy model show limited extrapolative power.

We can also notice that as data density decreases, the extrapolation performance of these HM with NODE formulation is slightly better than the MFSM formulation. This observation can be attributed to the manner in which error correction is implemented in both approaches. As the NODE approach corrects error in the derivatives, it has an edge in capturing profiles compared to MFSMs when extrapolating the corresponding HM.

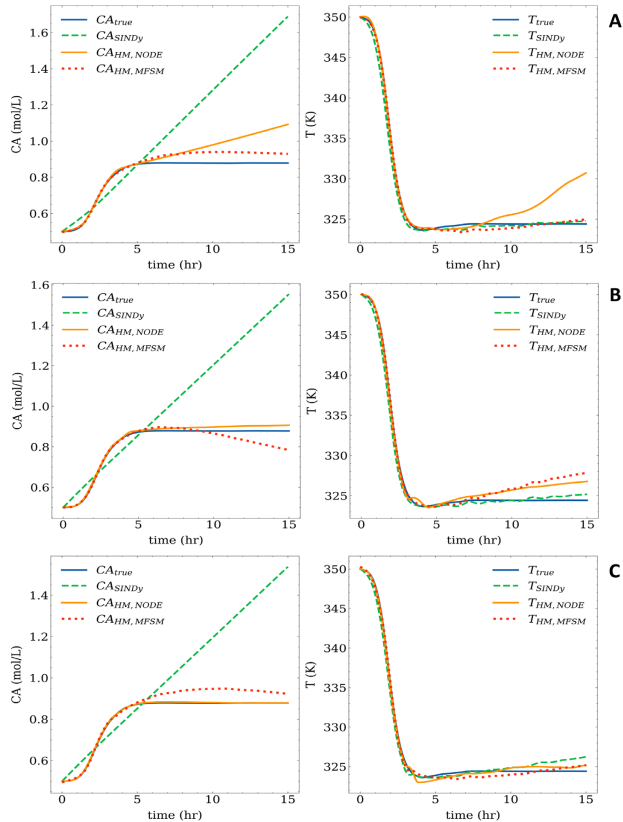


Figure 8: Extrapolated (C_A) and (T) profiles from SINDy model, HM-NODE and HM-MFSM formulations for case study 3.1.1 with 50, 30, and 20 HF samples.

Figure (9) shows the same analysis for the case study 3.1.2. Figures (9A, 9B, 9C) correspond to instances with 50, 30, 20 HF samples, respectively. We observe a similar behavior for this case study as well. HMs with NODE and MFSM formulations demonstrate better extrapolation capabilities than the SINDy. We can also observe a similar trend that shows HM with NODE formulation is slightly better than the MFSM formulation.

3.4 Effect of noise on the HMs prediction

In most practical applications, there is noise associated with the true state measurements. It is necessary to

evaluate the model performance with noise in the true state data and check for robustness. To test our approach, we replicate a practical scenario by adding noise to the simulated data. The true state dataset was modified by adding Gaussian-distributed noise to each state variable data for the two case studies 3.1.1 and 3.1.2. The noise was simulated using a normal distribution, where the mean was set to zero and the standard deviation was adjusted to represent 3%, 5% of the range of the uncorrupted data. To test the methods at challenging scenarios, we present results with 20 HF samples and varying levels of noise.

Figure (10) shows the results for this analysis. The legend (A) in Figure (10) represents the 3% noise and (B) represents 5% noise in data. We can notice the robustness of the HM approach towards the noise in data and both HM approaches are able to still predict the profiles accurately in both cases studies 3.1.1 and 3.1.2. But we can also observe that increasing the noise for the same sparsity reduces the accuracy of predicted profiles. This is expected because modeling accuracy becomes a challenge with increasing sparsity and noise. When comparing the two methods, the HM-MFSM approach performs better because the HM-MFSM structure directly fits the state values for each variable and proper tuning of the model helps in making it more robust towards noise. On the other hand, the HM-NODE fits the derivative values and estimating the derivatives in the presence of noise and sparsity becomes a challenge. While there exist many noise filtering techniques to smoothen the noisy data and mitigate this issue, we intend to include those as the future work.

4. CONCLUSIONS

In this work, we address the complexities inherent in data-driven model identification, especially when the underlying physics of the model remains unknown. We investigate the limitations of current state-of-the-art tools like SINDy when the available candidate library inadequately covers all potential terms contributing to the final mechanistic model equation. To tackle this challenge, we propose combining hybrid modeling techniques with sparse regression with the goal of simultaneous hybrid model development and model identification. To achieve this, we outline a workflow that employs two distinct methods: a) utilizing a NODE formulation, and b) employing an MFSM formulation as hybrid modeling approaches integrated with SINDy. Our study demonstrates the effectiveness of this hybrid model architecture in constructing accurate models capable of predicting profiles that predict the true data accurately, while integrating the mechanistic model knowledge.

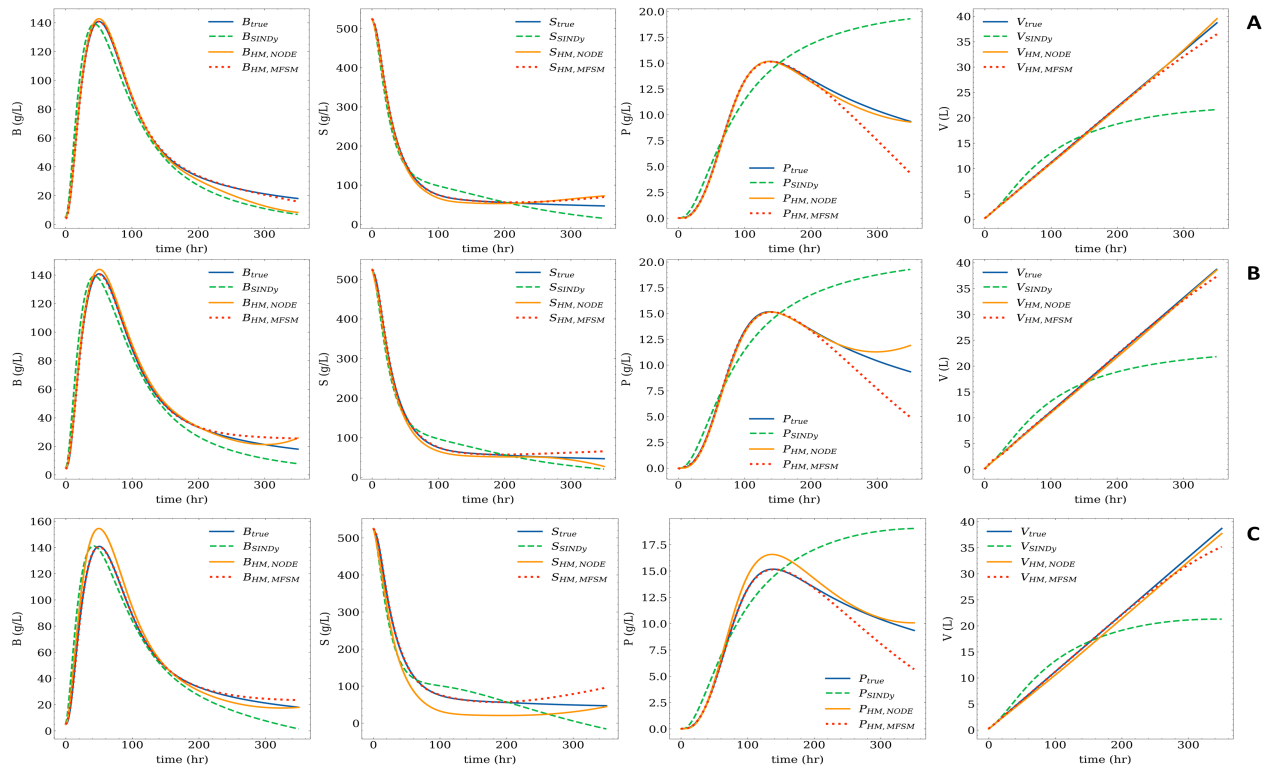


Figure 9: Extrapolated Volume (V), Biomass (B), product (P), substrate (S) profiles from SINDy model, HM-NODE and HM-MFSM formulations for case study 3.1.2 with 50, 30, and 20 HF samples.

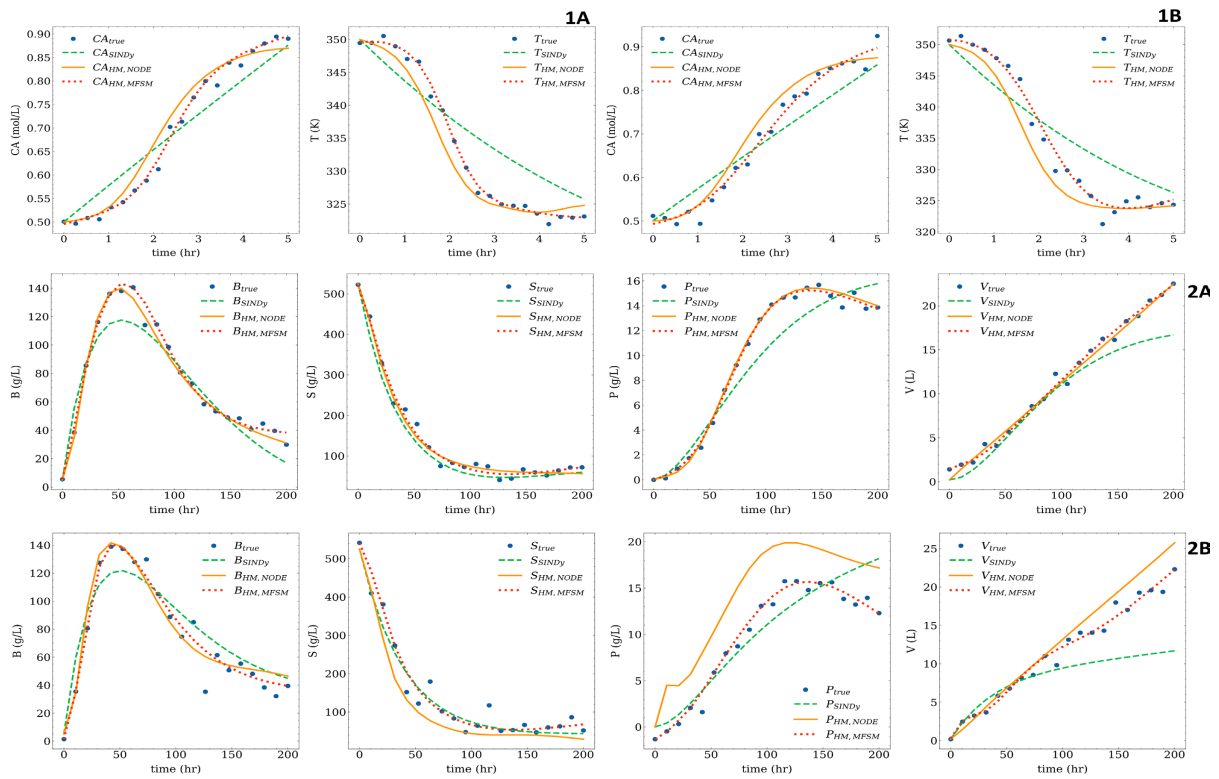


Figure 10: Predicted profiles from SINDy model, HM-NODE and HM-MFSM formulations for case study 3.1.1 (1A, 1B), and 3.1.2 (2A, 2B) with noise in the data. A, B correspond to 3% noise and 5% noise respectively.

We also showcase the robustness of hybrid models in handling low densities in sampled data and their ability to extrapolate. At lower amounts of sampled data, HM-NODE formulation can extrapolate better and HM-MFSM formulation can predict the states more accurately. Furthermore, we investigated the effect of noise in the true state data on these hybrid models. While the HM formulations could still predict the true profiles with a good accuracy, we saw that reducing the sampling data and increasing the noise can affect the model performance directly and make them less accurate. Future directions on this work will be focused on testing our methods with experimental data to formulate the models and proposing more efficient techniques to train hybrid models and improving their robustness towards data sparsity and noise.

5. ACKNOWLEDGEMENTS

The authors acknowledge support from the National Science Foundation (NSF-1944678)

REFERENCES

- Bradley, W., et al., Perspectives on the integration between first-principles and data-driven modeling. *Computers & Chemical Engineering*, 2022. **166**: p. 107898.
- van de Berg, D., et al., Data-driven optimization for process systems engineering applications. *Chemical Engineering Science*, 2022. **248**: p. 117135-117135.
- Quade, M., et al., Prediction of dynamical systems by symbolic regression. *Physical Review E*, 2016. **94**(1): p. 12214-12214.
- Koza, J., On the programming of computers by means of natural selection. *Genetic programming*, 1992.
- Brunton, S., J. Proctor, and N. Kutz. Sparse identification of nonlinear dynamics (sindy).
- Brunton, S.L., J.L. Proctor, and J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 2016. **113**(15): p. 3932-3937.
- Champion, K., et al., Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 2019. **116**(45): p. 22445-22451.
- Chen, Z., Y. Liu, and H. Sun, Physics-informed learning of governing equations from scarce data. *Nature Communications*, 2021. **12**(1): p. 6136-6136.
- Sun, F., Y. Liu, and H. Sun, Physics-informed spline learning for nonlinear dynamics discovery. *arXiv preprint arXiv:2105.02368*, 2021.
- Lejarza, F. and M. Baldea, Data-driven discovery of the governing equations of dynamical systems via moving horizon optimization. *Scientific Reports*, 2022. **12**(1): p. 11836.
- Von Stosch, M., et al., Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Computers & Chemical Engineering*, 2014. **60**: p. 86-101.
- Bradley, W. and F. Boukouvala, Two-Stage Approach to Parameter Estimation of Differential Equations Using Neural ODEs. *Industrial & Engineering Chemistry Research*, 2021. **60**(45): p. 16330-16344.
- Roesch, E., C. Rackauckas, and M.P.H. Stumpf, Collocation based training of neural ordinary differential equations. *Statistical Applications in Genetics and Molecular Biology*, 2021. **20**(2): p. 37-49.
- Chen, R.T.Q., et al., Neural ordinary differential equations. *Advances in neural information processing systems*, 2018. **31**.
- Lee, D., A. Jayaraman, and J.S. Kwon, Development of a hybrid model for a partially known intracellular signaling pathway through correction term estimation and neural network modeling. *PLoS Computational Biology*, 2020. **16**(12): p. e1008472.
- Lai, Z., et al., Structural identification with physics-informed neural ordinary differential equations. *Journal of Sound and Vibration*, 2021. **508**: p. 116196.
- Ravutla, S., J. Zhai, and F. Boukouvala, Hybrid Modeling and Multi-Fidelity Approaches for Data-Driven Branch-and-Bound Optimization, in *Computer Aided Chemical Engineering*. 2023, Elsevier. p. 1313-1318.
- Meng, X. and G.E. Karniadakis, A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. *Journal of Computational Physics*, 2020. **401**: p. 109020-109020.
- Lagaris, I.E., A. Likas, and D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 1998. **9**(5): p. 987-1000.
- Guo, M., et al., Multi-fidelity regression using artificial neural networks: efficient approximation of parameter-dependent output quantities. *Computer methods in applied mechanics and engineering*, 2022. **389**: p. 114378-114378.

© 2024 by the authors. Licensed to PSEcommunity.org and PSE Press. This is an open access article under the creative commons CC-BY-SA licensing terms. Credit must be given to creator and adaptations must be shared under the same terms. See <https://creativecommons.org/licenses/by-sa/4.0/>

