IMECE2021/69657

NEURAL DIFFERENTIAL EQUATIONS FOR INVERSE MODELING IN MODEL COMBUSTORS

Xingyu Su
Tsinghua University
Beijing 100084, China
su-xy19@mails.tsinghua.edu.cn

Weiqi Ji
Massachusetts Institute of Technology
Cambridge, MA 02139
weiqiji@mit.edu

Long Zhang
Tsinghua University
Beijing 100084, China
long-zha19@mails.tsinghua.edu.cn

Wantong Wu
Tsinghua University
Beijing 100084, China
wwt15@mails.tsinghua.edu.cn

Zhuyin Ren *
Tsinghua University
Beijing 100084, China
zhuyinren@mail.tsinghua.edu.cn

Sili Deng †
Massachusetts Institute of Technology
Cambridge, MA 02139
silideng@mit.edu

ABSTRACT

Monitoring the dynamics processes in combustors is crucial for safe and efficient operations. However, in practice, only limited data can be obtained due to limitations in the measurable quantities, visualization window, and temporal resolution. This work proposes an approach based on neural differential equations to approximate the unknown quantities from available sparse measurements. The approach tackles the challenges of nonlinearity and the curse of dimensionality in inverse modeling by representing the dynamic signal using neural network models. In addition, we augment physical models for combustion with neural differential equations to enable learning from sparse measurements. We demonstrated the inverse modeling approach in a model combustor system by simulating the oscillation of an industrial combustor with a perfectly stirred reactor. Given the sparse measurements of the temperature inside the combustor, upstream fluctuations in compositions and/or flow rates can be inferred. Various types of fluctuations in the upstream, as well as the responses in the combustor, were synthesized to train and validate the algorithm. The results demonstrated that the approach can efficiently and accurately infer the dynamics of the unknown inlet boundary conditions, even without assuming the types of fluctuations. Those demonstrations shall open a lot of opportunities in utilizing neural differential equations for fault diagnostics and model-based dynamic control of industrial power systems.

Keywords: Neural Differential Equations, Inverse Problems, Unsteady Combustion, Auto-differentiation, Machine Learning.

1. INTRODUCTION

Oscillation in combustors can degrade the efficiency and safety of the combustion system. Unsteadiness in the upstream is one of the major causes of the oscillation, and thus monitoring the upstream conditions is important for control and operation. However, in practice, it is difficult to measure the upstream conditions due to the challenges in diagnostic at harass environment and the constraints of cost. Therefore, computational tools aiming at the inference of unknown information based on available measurements have been developed and are often termed as inverse models [1, 2, 3]. Conventional inverse modeling methods are usually limited to a few unknown parameters and suffering from the curse of dimensionality since the computational cost increases significantly as the number of unknown quantities increases.

In the present work, we propose to use neural networks

---

* Address all correspondence to this author.
† Address all correspondence to this author.

for approximating the high dimensional unknown quantities and train the neural networks efficiently with available measurements by differential programming. Traditional supervised learning methods requires a large number of training datasets covering a wide range of conditions to achieve good extropotation capability outside of the training datasets. But for the neural differential equation that physical constraints are imposed during the training by augmenting the neural network models with the governing differential equations, the training can be viewed as self-supervised learning to avoid the need for obtaining labeled training data, in this case, the unmeasurable, which are intrinsic obstacles for modeling industrial combustors with data-driven approaches. Therefore, the approach doesn't suffer from the extropotation risk of typical machine learning model.

We demonstrated the inverse modeling approach in a model combustor system by simulating oscillating combustion in an industrial hot air heater with a perfectly stirred reactor (PSR). Industrial combustion devices such as hot air heaters are widely used for burning non-standard low-calorific value (NLCV) gases, such as blast furnace gas and coke oven gas, which are prone to combustion instability. Zhang et al. [4] studied the oscillating combustion in reactors resulting from periodic fuel burnt out, fuel replenishment, and ignition governed by the subtle mixing-chemical kinetics interaction or due to the inlet flow rate fluctuation under industrial conditions. Zhang et al. utilized forward modeling to investigate the oscillation of combustion via unsteady PSR simulation. Given the initial conditions and boundary conditions, one can integrate the governing ordinary differential equations (ODEs) to get the temporal evolution profile of temperature and species concentrations. However, in practical industrial scenarios, it is often difficult to measure the boundary conditions with satisfying accuracy, especially for species concentrations. The experimental data that can be effectively obtained are the temperature and flow velocity at the combustor outlet. In this study, we try to infer the inlet fluctuations in compositions and/or flow rate given the sparse measurements of the temperature at the combustor outlet. The fluctuation in the inlet conditions is usually time-dependent, such that one needs to use a large number of data points to resolve the evolution of the inlet conditions, and thus the inverse modeling involves high dimensional unknown parameters.

This work aims to tackle the challenges in achieving scalability of the data-driven inverse modeling of unknown measurements in model combustors using techniques developed for training deep neural network models. Specifically, we first parameterize the fluctuations using a neural network model. We then employ the stochastic gradient descent (SGD) optimizer to learn the inlet boundary conditions in an unsteady PSR. The SGD optimizer is widely applied in the deep learning communities for its efficiency as well as scalability in dealing with large datasets, and its robustness in optimizing high-dimensional non-convex neural network models. Our recent work [5] has also shown that

a chemical reaction network is equivalent to a neural network with a single hidden layer. Similarly, solving ordinary differential equations (ODEs) of reaction network models is equivalent to solving infinite-depth deep residual networks [6], which further rationalizes exploiting SGD in training ODEs.

However, to exploit neural networks in combustion simulation optimizations, one needs a well-established software ecosystem that can efficiently and accurately compute the gradient of simulation output to model parameters. For instance, the finite difference method (often termed as the "brute-force method") usually suffers both the computational inefficiency due to the cost sharply rising with the number of parameters and the numerical inaccuracy due to the truncation error [7]. Conversely, gradient evaluation methods based on auto-differentiation (AD) have shown both efficiency and accuracy in the training of large-scale deep neural network models, with a large range of datasets. Many open-source AD packages have been developed in the last decade, including TensorFlow [8], Jax and PyTorch [9] in Python, ForwardDiff.jl and Zygote.jl [10] in Julia. Thus, this work utilized a recently developed differentiable combustion simulation package Arrhenius.jl [11] to enable SGD for combustion model optimizations and applies Arrhenius.jl to infer the boundary conditions of the model combustor described by PSR. Specifically, Arrhenius.jl, the differential programming ecosystem in Julia, was employed to encode and solve the governing ODEs for a perfectly stirred reactor, conduct auto-differentiation, and train the neural network models. Various types of fluctuations in the upstream, as well as the responses at the combustor's outlet, were synthesized to train and validate the algorithm. The results demonstrated that the approach can efficiently and accurately infer the dynamics of the inlet flow and calorific fluctuations.

This paper is structured as follows: we first introduce the package of Arrhenius.jl, the governing equations for the combustor, and the inverse modeling using neural differential equations; we then demonstrate the approach in inverse modeling during two representative scenarios: fuel switching and inflow fluctuations. Finally, we conclude and discuss the opportunities and challenges of employing the current approach for the dynamic control of industrial power systems.

2. METHODS
2.1 Arrhenius.jl

Arrhenius.jl is built using the programming language of Julia to leverage the rich ecosystems of auto-differentiation and differential equation solvers. Arrhenius.jl does two types of differentiable programming: (i) it can differentiate elemental computational blocks. For example, it can differentiate the reaction source term with respect to the kinetic and thermodynamic parameters as well as species concentrations; (ii) it can differentiate the entire simulator in various ways, such as solving the

adjoint sensitivities [12] as did in CHEMKIN and Cantera [13]. The first type of differentiation plays the role of the basis of the second type of high-level differentiation. Arrhenius.jl offers the core functionality of chemical reaction simulations in native Julia programming, which means users can readily build the applications on top of Arrhenius.jl and exploit various approaches to do high-level optimizations.

Figure 1 shows a schematic diagram of the structure of the Arrhenius.jl package. The package reads in the reaction mechanism files in YAML format as the same as those in the Cantera community, which contains the kinetic models, thermodynamic, and transport databases. The core functionality of Arrhenius.jl is to compute the reaction source terms and mixture properties, such as heat capacities, enthalpies, entropies, Gibbs free energies, etc. Furthermore, Arrhenius.jl makes it possible for users to define neural network models as submodels and augment them with existing physical models. For example, one can use a neural network submodel to represent the unknown chemical reaction pathways and exploit various scientific machine learning methods to train the neural network models, such as neural ordinary differential equations [6, 14] and physics-informed neural network models [15, 16]. One can then implement the governing equations for different applications with these core functionalities and solve the governing equations using classical numerical methods or neural-network-based solvers. Arrhenius.jl provides solvers for canonical combustion problems, such as simulating the auto-ignition in constant volume/pressure reactors and oxidation in perfectly stirred reactors.

Compared to the legacy combustion simulation packages, Arrhenius.jl can not only provide predictions given the physical models but also the capability of optimizing model parameters or boundary conditions given experimental measurements. By efficiently and accurately evaluating the gradient of the solution outputs to the model parameters, experimental data can be incorporated into the simulation pipeline to enable data-driven modeling with deep learning algorithms.

2.2 Governing Equations for the Model Combustor

Following the work of [4, 17], we employ a non-adiabatic unsteady PSR to reveal the complex dynamics exhibited in industrial hot air heaters. The chemical kinetic model employed is DRM22, a reduced version of GRIMECH 1.2 obtained by Kazakov et al. [18], which consists of 22 species and 104 reactions.

The hot air heater in this study operates under constant atmospheric pressure. The heater chamber is cylindrical with a constant volume of 13.82 m^3 as detailed in [4]. The mass flow rate of the inlet mixture m_a is 2.75 kg/s, and the residence time τ_{res} is calculated according to Eq. 1,

$$\tau_{res} = \frac{\rho V}{m_a}, \tag{1}$$

where ρ is the density of the mixture in the reactor and V is the volume of the hot air heater. With the assumption of perfect mixing in the reactor, the dynamics inside the combustor can be modeled by a non-adiabatic PSR governed by the following set of ODEs,

$$\frac{d\mathbf{Y}}{dt} = \frac{\mathbf{Y}_{in} - \mathbf{Y}}{\tau_{res}} + \mathbf{S}(\mathbf{Y}), \tag{2}$$

$$\frac{dT}{dt} = \frac{\sum_i q_i R_i}{\rho C_p} - \frac{T - T_{in}}{\tau_{res}} - \frac{Q_{loss}(T - T_a)}{\rho C_p}, \tag{3}$$

where $\mathbf{Y}_{in}$ and T_{in} are the species mass fraction and temperature of the inlet stream, T_a is the ambient temperature, Q_{loss} is the heat loss coefficient, T is the mixture temperature inside the chamber, C_p is the specific heat of the mixture inside the reactor, R_i is the $i-th$ reaction rate, q_i is the exothermicity of the $i-th$ reaction, and the vector $\mathbf{S}(\mathbf{Y})$ refers to the chemical reaction source terms of the compositions. The combustion characteristics of PSR are determined by the boundary conditions of τ_{res}, $\mathbf{Y}_{in}$, T_{in}, P, T_a and Q_{loss} and the initial conditions.

This work considers two kinds of NLCV gases, the coke oven gas and the blast furnace gas. For both two kinds of fuels, the hot air heater is operated under the following conditions: $P = 101325Pa$, $T_{in} = 300K$, $T_a = 873K$, $Q_{loss} = 4JK^{-1}m^{-3}s^{-1}$, and $\tau_{res} = 1.0s$. The heat loss coefficient Q_{loss} is estimated by calculating the equivalent thermal resistance of the heater wall. Typical compositions of the NCLV gases applied in this study are listed in Table 1. The initial conditions of the PSR are the adiabatic and isobaric chemical-equilibrium state of the inlet mixture. Equations (2-3) are numerically integrated with the ODE solver $TRBDF2()$ in the Julia package DifferentialEquations.jl [12].

TABLE 1. Typical fuel compositions of coke oven gas (COG) and blast furnace gas (BFG) used (volume fraction) [4]

Fuel composition		H2	CO	CH4	N2
Coke oven gas	COG1	50.0%	9.0%	30.0%	11.0%
	COG2	60.0%	9.0%	25.0%	6.0%
Blast furnace gas	BFG	2.5%	22.5%	0.0%	75.0%

2.3 Inverse Modeling

As discussed in the introduction, inverse modeling mainly involves unknown inlet conditions. To alleviate the high dimensionality of the time-dependent inlets, we use a neural network

3
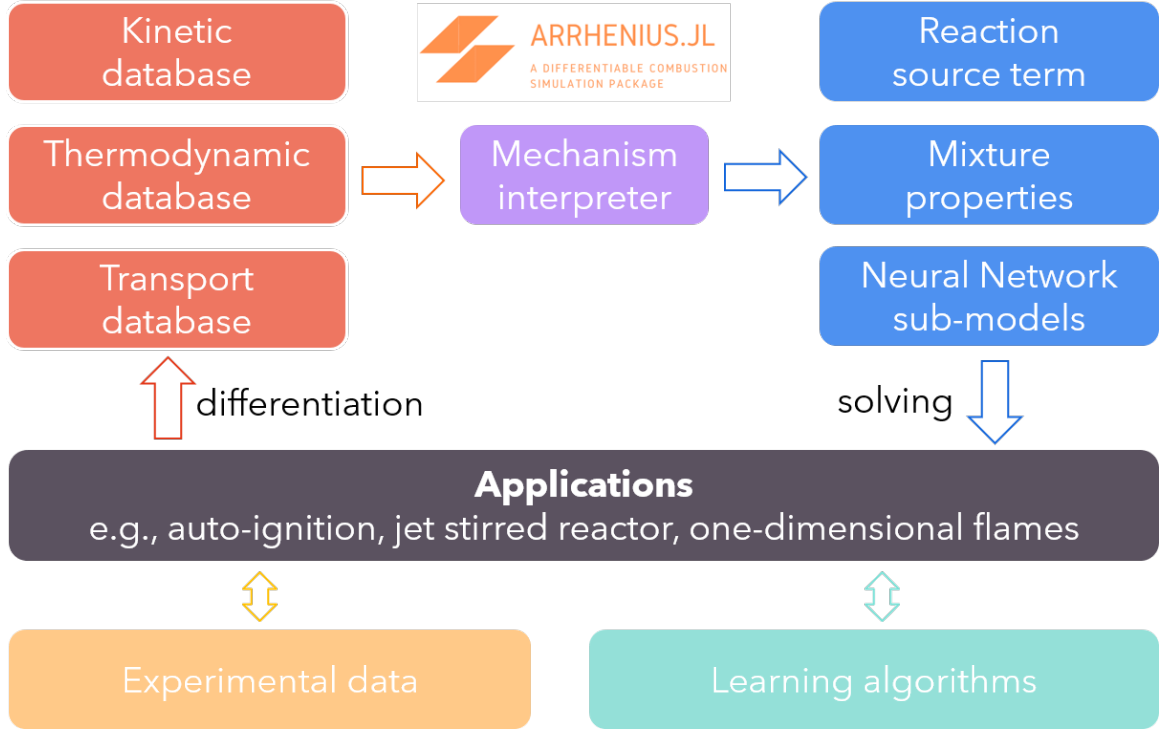
FIGURE 1. Schematic diagram showing the structure of the Arrhenius.jl [11] package.

model to represent the inlet conditions. The neural network takes the input of time and outputs the mixture compositions and mass flow rate of the inlet stream. Recent work of physics-informed neural network [15] has demonstrated that the neural network representation of time-dependent state variable not only tackles the curse of dimensionality but also has the potential to learn the physics information in the training data and predict the state variable beyond the training time domain. Therefore, this work employs a neural network model to infer the unknown inlet conditions, as outlined in Fig. 2. The true inlet conditions $\hat{\boldsymbol{u}_{in}}$ are essentially unknown, and the optimization process will make the state variables $\boldsymbol{u}$ fit with the obervations $\hat{\boldsymbol{u}}$, and thus we get the $NN_{inlet}(t;\tilde{\theta})$ as the best estimation of the boundary conditions.

Without loss of generality, we can rewrite Eqs. 2 and 3 as

$$\frac{d\boldsymbol{u}}{dt} = f(\boldsymbol{u}, \boldsymbol{u}_{in}), \tag{4}$$

where f refers to the right hand side of the ODEs. $\boldsymbol{u}$ refers to the vector of state variables $[\boldsymbol{Y}, T]$. We then use a neural network model $NN_{inlet}(t;\theta)$ to model the inlet mixture, i.e.,

$$\boldsymbol{u}_{in} = NN_{inlet}(t;\theta), \tag{5}$$

where θ refers to all of the weights and bias of the neural network

model.

$$\frac{d\boldsymbol{u}}{dt} = f(\boldsymbol{u}, NN_{inlet}(t;\theta)). \tag{6}$$

For a given initial condition $\boldsymbol{u}_0$, the ODEs of Eq. 6 can be solved with a stiff ODE solver from DifferentialEquations.jl and results in the following solution array:

$$\boldsymbol{u}(t) = \text{ODESolve}(f, NN_{inlet}, \boldsymbol{u}_0, t; \theta). \tag{7}$$

We can then compare the predicted evolution of compositions inside the combustor with available measurements and guide the optimization of the neural network model. Specifically, with observations $\hat{\boldsymbol{u}}$ as the training data, one can define the loss function as the mean square error (MSE) between the solution array and the training data via

$$Loss = MSE(\boldsymbol{u}, \hat{\boldsymbol{u}}). \tag{8}$$

We employ the algorithm of ForwardDiffSensitivity to compute the gradient of the loss function to the neural network weights. ForwardDiffSensitivity is an implementation of the discrete forward sensitivity analysis through ForwardDiff.jl [10].

4
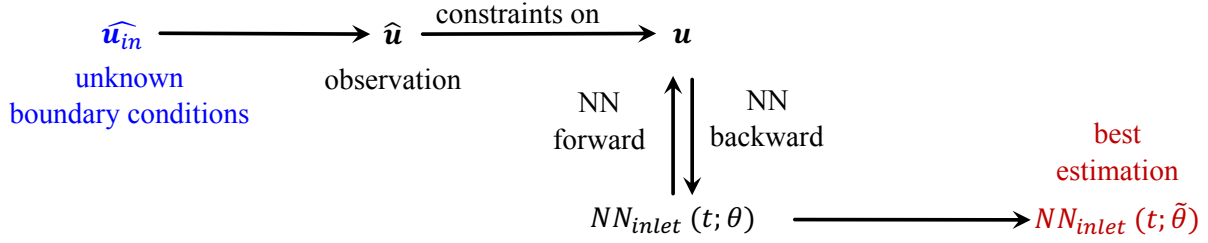
FIGURE 2. Schematic diagram of the inference of boundary conditions via neural networks.

Note that accurate gradient computation is crucial for training neural network models as the optimization problem is usually stiff and it is difficult to reach a good minimum with noisy gradient.

With the computed gradient, we then employ the SGD optimizer to optimize the neural network models. SGD optimizer is usually very sensitive to the hyper-parameters, such as the learning rate, initialization, etc. We will present the details of the optimizer in the next section. We then treat the neural network that minimizes the mismatch between the predicted and measured composition evolution inside the combustor as the best estimation of the inlet conditions.

3. RESULTS

Non-steady inlet boundary conditions are common causes for the oscillation of temperature and species concentrations inside the combustor. Industrial operations using the NLCV fuels are mainly concerned with two types of combustion instabilities. One is due to fuel switching and the other is due to the flow rate and calorific fluctuations in the inlet flow. In this section, we demonstrate the present method in the two representative cases with industry relevant amplitudes and frequencies. We also design different initialization method and different noise levels to demonstrate the robustness of the optimization process. We will first demonstrate the identification of fuel switching from the measurement of temperature history and then demonstrate the inference of inlet flow rate and calorific fluctuations from the measurement of temperature and CO profiles.

3.1 Inverse Modeling for Fuel Switching

In combustion devices such as hot air heaters, coke oven gas can be directly ignited at a normal temperature, for it has half of the calorific value of natural gas, about 4.75 kWh/m^3. On the contrary, blast furnace gas with one-tenth the calorific value of natural gas, about 0.95 kWh/m^3, needs to be burnt at a higher temperature. During the fuel switch, the distinct combustion characteristic between the two gases can induce oscillating combustion, and thus cause the vibration of the heater body and the instability of the heated air. In addition, extinction occasion-

ally occurs since the switching temperature is usually determined based on empirical rules and thus it is not readily known and may not be optimal during the fuel-switching stage. Therefore, identifying the transition process during fuel switching would be beneficial for further optimizing the operation and avoiding unexpected extinction.

Here, we simulate the fuel switching from COG1 to COG2 with a linear switching function to synthesize the measurement dataset. As shown in Fig. 3(a), the volume fraction of COG2 at the inlet transitions from 0 to 1 to represent the fuel switching process from COG1 to COG2 in an industrial combustor. As detailed in [4], the burning of COG fuels will show instability from mixing-chemistry interaction, with an oscillation frequency of around 20 Hz. Therefore, the simulation time range here is chosen to be [0, 0.5s] with 100 data points in between, and the fuel switch process occurs between 0.15 s and 0.35 s. Such time interval in the dataset is sufficiently small to resolve the oscillation inside the combustor.

In practice, it is often the case that we can only measure the temperature profiles while it is difficult to measure the species concentrations. Therefore, to simulate this challenging scenario, we only employ the temperature profiles into the loss functions for training the neural network model, i.e.,

$$Loss = MSE\left(T, \widehat{T}\right). \tag{9}$$

Since the neural differential equations involve stiff combustion chemistry, differentiating a stiff neural ODE is usually much more computationally expensive than a non-stiff problem, as pointed out in [14]. In addition, the computational cost scales with the number of neural network parameters. To reduce the burden of automatic differentiation, the neural network model should be as concise as possible. Here, a simple neural network model with a single hidden layer with five hidden nodes is employed. The activation functions used are $\tanh(x)$ for the hidden layer and $\exp(x)$ for the output layer.

The popular first-order optimizer of Adam [19] is adopted for the optimization. The optimization proceeds step-by-step with a decaying learning rate. Specifically, we first employ a

(a) inlet conditions   (b) state variables
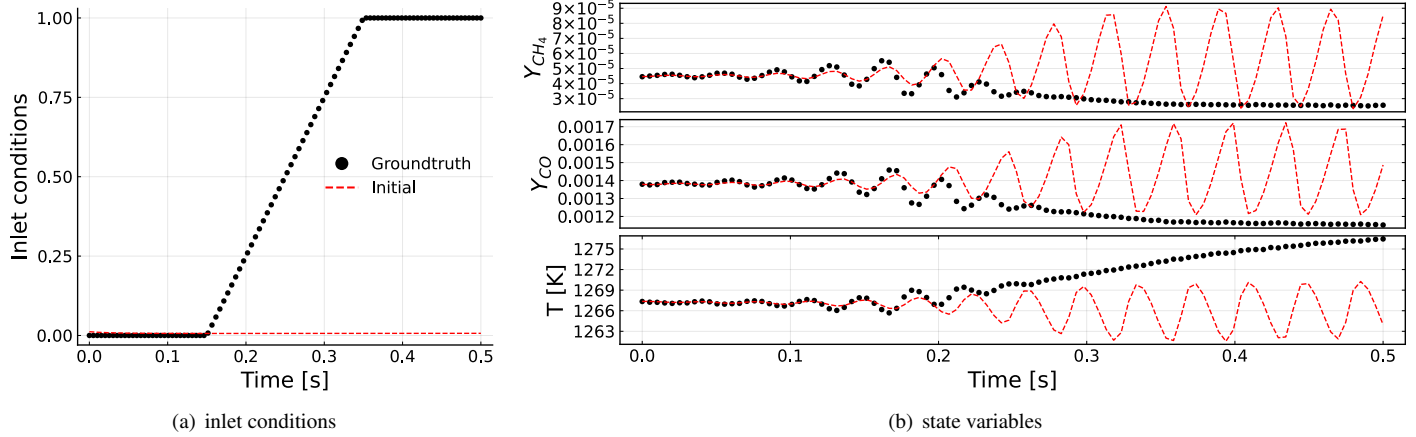
FIGURE 3.   Initial states of the fuel switching case. (a) The ground truth and the initialized inlet conditions of COG2 ratio. (b) The ground truth and predicted species and temperature profiles based on the inlet conditions in (a).

(a) inlet conditions   (b) state variables
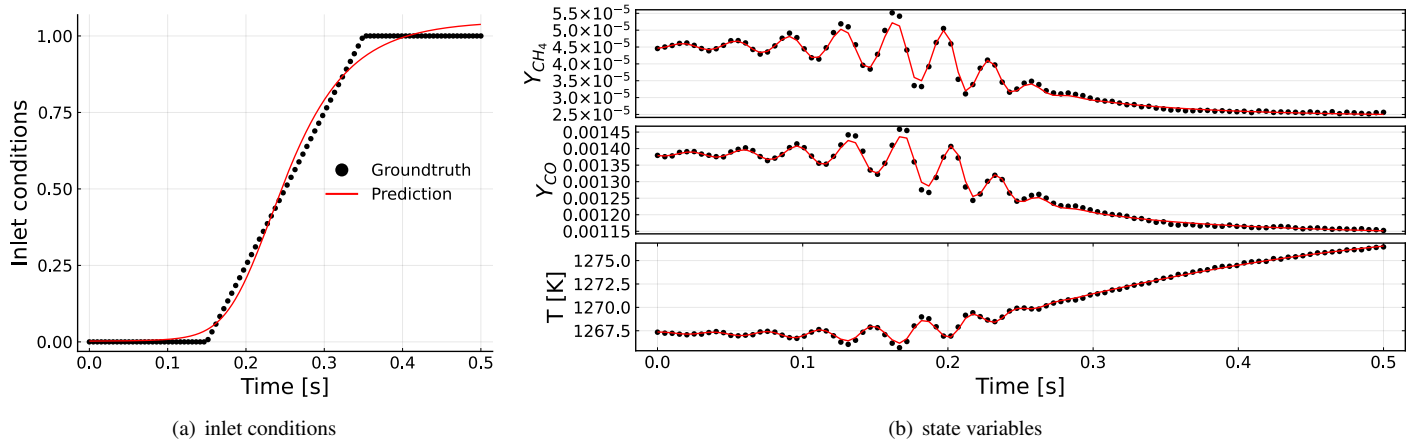
FIGURE 4.   Trained results of the fuel switching case. (a) The ground truth and the inferred inlet conditions of COG2. (b) The ground truth and predicted species and temperature profiles based on the inlet conditions in (a).

learning rate of 1e-1 to capture the most important information with 100 training epochs. We then employ 1e-2 for another 100 epochs and 5e-3 for extra 200 epochs to fine-tune the model. Figure 5 presents the history of the loss function. As can be seen, the loss decreases stably, which indicates that the learning rate is properly chosen. The loss function reaches a plateau after 300 epochs, and thus 400 epochs are sufficient for the training.

The initialization of the neural network models is achieved by assuming a steady inlet with only COG1. Even though the inlet condition is steady, oscillation is observed in the combustor due to intrinsic instability, as indicated by the dashed line shown in Fig. 3(b). Similar oscillation is found in the evolution of the state variables corresponding to the ground truth inlet condition, where fuel switches from COG1 to COG2. Shown as the symbols, 1% of noise has been added to represent the mea-
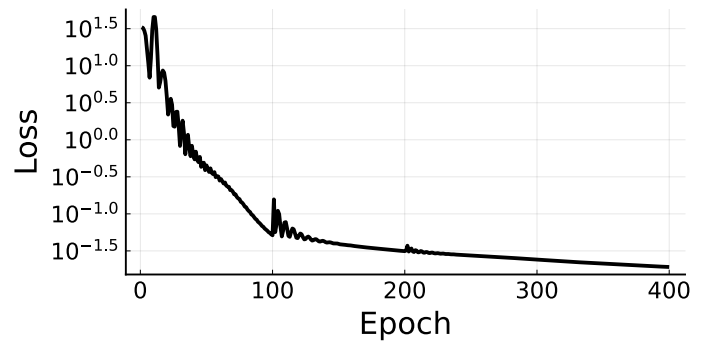
FIGURE 5.   The loss history of the fuel switching case.

6

surement uncertainty. Such synthesized measurements are also shown in Fig. 3(b) and utilized to train neural network models to infer the inlet conditions. Therefore, the lines in Figs. 3(a) and 3(b) demonstrate the inferred inlet conditions and predicted state variables, respectively. As can be seen, the inferences and predictions agree very well with the ground truth. In addition, the predicted species profiles for methane and carbon monoxide also agree well with the ground truth, although they are not included in the loss function. Such capability of revealing hidden information demonstrates the good generalization capability of the approach. The training takes about 2 hours on a Linux server with a single CPU core, and the training can be potentially accelerated by improving the weight initialization strategy and parallel computing [14].

3.2 Inverse Modeling for Inflow Fluctuations

Another important mechanism for oscillating combustion behaviors is the uncertain inlet conditions with flow and calorific fluctuations during the stable burning of the BFG gas. According to Zhang et al., [17], typical inflow fluctuations of hot air heater consist of 10% of fluctuation in inlet flow rates and 3% of fluctuation in inlet calorific value, with frequencies around 1 Hz and 0.05 Hz, respectively. In this study, we employ a sine function with 20% of noise added to synthesize the "ground truth" inlet boundary conditions, as shown in Fig. 6(a). The inlet flow rate fluctuation is characterized by the fluctuations in the resident time $\tau_{res} = \rho V / m_a$, which has a nominal value of $\bar{\tau}_{res} = 1.0s$ and frequency of 1 Hz. For simplicity, we assume that the fluctuation of inlet calorific value is due to the composition variation in the fuel blend of COG1 and BFG. Thus, the volume fraction of BFG can well characterize the change in the inlet calorific value. To reduce the computational cost, the frequency of calorific fluctuation is set to be 0.2 Hz and the time range for the ODE integration is [0, 5 s] for demonstration. In addition, to demonstrate the robustness and capability of the feature learning of the current method, 10% of noise is added to the state variables obtained at the presumed inlet boundary conditions to be utilized as the training dataset for the neural network.

The structure of the neural network NN_{inlet} is similar to that mentioned in the fuel switching case, but has two output neurons corresponding to the two inlet variables in this case, i.e., the volume fraction of BFG and the residence time τ_{res}. Besides, to capture the periodic inlet boundary conditions, we employ $\sin(x)$ as the activation function for the hidden layer. This means that the neural network inlet NN_{inlet} is a linear combination of sine functions

$$NN_{inlet}(t)_j = \exp\left(\sum_i (A_{ij} \sin(a_i x + b_i) + B_{ij}) \right), \qquad (10)$$

where $i = 1, 2, ..., 5$, $j = 1, 2$, correspond to the indices of neurons

in the hidden layer and the output layer. a_i, b_i are the weights and biases of the mapping from the input layer to the hidden layer. A_{ij} and B_{ij} are the weights and biases of the mapping from the hidden layer to the output layer.

In practice, the fluctuation frequency and amplitude in the temperature profile can provide insights for the initialization of the neural network. For example, Fig. 6(a) shows the NN_{inlet} initialized with similar frequencies and amplitudes as the temperature profiles, and the corresponding state variables are shown in Fig. 6(b) with the dashed red lines.

However, due to the complexity of the coupling between flow and calorific fluctuations, it is difficult to identify the boundary conditions with only temperatures as the observed data. Therefore, temperature along with CO mass fractions are used to constrain the neural networks via the loss function

$$Loss = MSE\left([T, Y_{CO}], [\hat{T}, \hat{Y}_{CO}] \right). \qquad (11)$$

Similar to the case of fuel switching, we gradually decaying the learning rate. We first employ a learning rate of 5e-3 to capture the most important information with 100 training epochs. Then a learning rate of 2e-3 for another 100 epochs and 1e-3 for extra 200 epochs are employed to fine-tune the results. The training results are shown in Fig. 7, which takes about 30 minutes to train with a single thread. The losses of the training process are shown in Fig. 8.

As can be seen from Figs. 7(a) and 7(b), the inferred inlet conditions and the predicted temperature evolution with the inferred inlet conditions agree with the ground truth very well. In addition, the predicted species profiles for methane and other species also agree well with the ground truth. Such capability of revealing coupled hidden inlet conditions further shows the potential of the approach in fault diagnostics for sourcing the causes of combustion oscillation.

4. CONCLUSION

This work presents a machine learning approach for inverse modeling of inlet boundary conditions of a model combustor. The approach is demonstrated to recover the inlet conditions under two common modes of unsteady inlets: fuel switching and inlet flow rate/calorific fluctuations. Using synthesized data, the results show that the approach can accurately infer the temporal evolution of the inlet boundary conditions from limited measurements of temperature and/or CO profiles. In addition, the temporal profiles of methane inside the combustor can also be accurately inferred although methane is not exposed to the training process. Therefore, the approach is shown to efficiently and accurately reveal the hidden dynamics of the upstream compositions and flow rate.
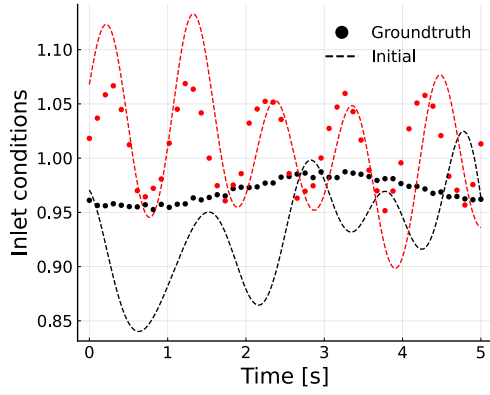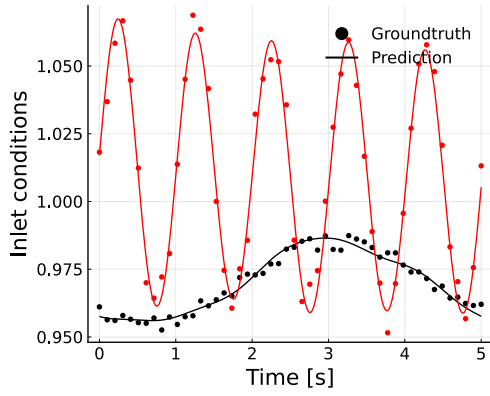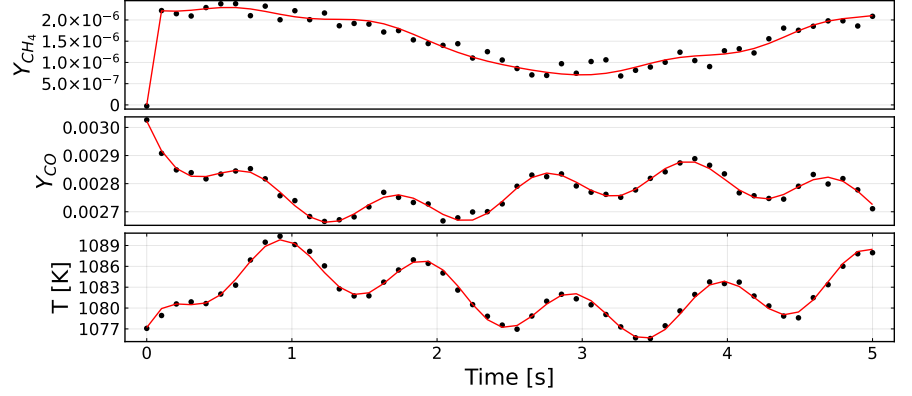
(a) inlet conditions
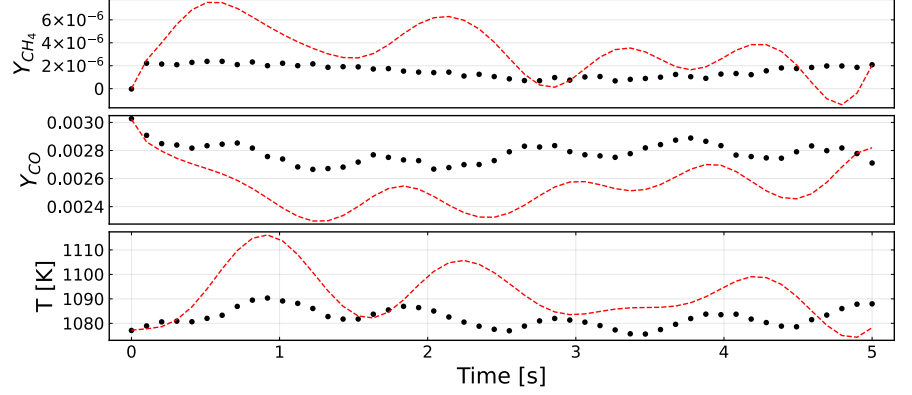
(b) state variables

FIGURE 6. Initial states of the inflow fluctuating case. (a) The ground truth and the initialized inlet conditions of the volume fraction of BFG (black) and the flow rate represented by τ_{res} (red). (b) The ground truth and predicted species and temperature profiles based on the inlet conditions in (a).
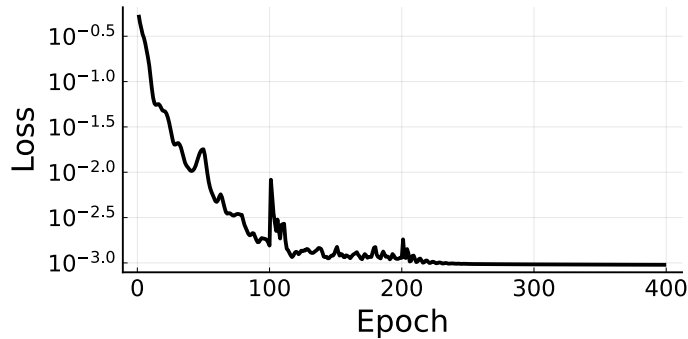

(a) inlet conditions

(b) state variables

FIGURE 7. Trained results of the inflow fluctuating case. (a) The ground truth and the inferred inlet conditions of the volume fraction of BFG (black) and the flow rate represented by τ_{res} (red). (b) The ground truth and predicted species and temperature profiles based on the inlet conditions in (a).

FIGURE 8. The loss history of the inflow fluctutating case.

Furthermore, the proposed approach has shown its scalability and capability to be utilized in realistic industry systems. Modern combustors always work near the limit conditions for the best performance in energy conversion efficiency or emission level control. Therefore, subtle inlet fluctuations might lead to bad working conditions or even serious equipment failures. As for the hot air heater mentioned above, inlet fluctuations would lead to combustion instability and decrease the life expectancy of some specific machinery components. When unexpected outlet oscillation occurs, one can apply the proposed method to investigate the influence of inlet boundary conditions with limited sparse measurements of the combustor outlet and take relevant actions according to that. If further improvements make it possible for on-the-fly prediction of inlet conditions, it would be very

useful in the embedded controller of reacting flow devices since this algorithm could obtain the inlet conditions that are necessary for some active control methods.

The current limitation of the algorithm depends on how well the ODEs can reproduce the behaviors of the practical devices. So further validation and verification of its performance with experimental data is needed, especially in multi-variables fluctuating inlet conditions with notable noises.

While this work provides a point estimation of unknown inlet conditions, Bayesian neural networks [20] can be readily augmented into the approach to estimating the uncertainties of the inverse modeling. The estimated uncertainty will greatly facilitate the development of dynamic control of industrial combustion systems.

ACKNOWLEDGMENT

REFERENCES

[1] Cui, M., Mei, J., Zhang, B.-W., Xu, B.-B., Zhou, L., and Zhang, Y., 2018. "Inverse identification of boundary conditions in a scramjet combustor with a regenerative cooling system". *Applied Thermal Engineering,* *134*, pp. 555–563.

[2] Perakis, N., Strauß, J., and Haidn, O. J., 2019. "Heat flux evaluation in a multi-element ch4/o2 rocket combustor using an inverse heat transfer method". *International Journal of Heat and Mass Transfer,* *142*, p. 118425.

[3] Lee, K. H., and Baek, S. W., 2011. "Inverse estimation of inlet parameters in an axisymmetric cylindrical combustor with thermal radiation effect". *International journal of heat and mass transfer,* *54*(25-26), pp. 5386–5394.

[4] Zhang, L., Xie, W., and Ren, Z., 2020. "Combustion stability analysis for non-standard low-calorific gases: Blast furnace gas and coke oven gas". *Fuel,* *278*, p. 118216.

[5] Ji, W., and Deng, S., 2021. "Autonomous discovery of unknown reaction pathways from data by chemical reaction neural network". *The Journal of Physical Chemistry A,* *125*(4), pp. 1082–1092.

[6] Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D., 2018. "Neural ordinary differential equations". *arXiv preprint arXiv:1806.07366*.

[7] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M., 2018. "Automatic differentiation in machine learning: a survey". *Journal of machine learning research,* *18*.

[8] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al., 2016. "Tensorflow: A system for large-scale machine learning". In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), pp. 265–283.

[9] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. "Pytorch: An imperative style, high-performance deep learning library". *arXiv preprint arXiv:1912.01703*.

[10] Revels, J., Lubin, M., and Papamarkou, T., 2016. "Forward-mode automatic differentiation in julia". *arXiv preprint arXiv:1607.07892*.

[11] Ji, W., and Deng, S., 2021. Arrhenius.jl: A differentiable combustion simulation package. https://github.com/DENG-MIT/Arrhenius.jl.

[12] Rackauckas, C., and Nie, Q., 2017. "Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in julia". *Journal of Open Research Software,* *5*(1).

[13] Goodwin, D. G., Moffat, H. K., and Speth, R. L., 2009. Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes.

[14] Kim, S., Ji, W., Deng, S., and Rackauckas, C., 2021. "Stiff neural ordinary differential equations". *arXiv preprint arXiv:2103.15341*.

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

[16] Ji, W., Qiu, W., Shi, Z., Pan, S., and Deng, S., 2020. "Stiff-pinn: Physics-informed neural network for stiff chemical kinetics". *arXiv preprint arXiv:2011.04520*.

[17] Zhang, L., Xue, Y., Xie, Q., and Ren, Z., 2021. "Analysis and neural network prediction of combustion stability for industrial gases". *Fuel,* *287*, p. 119507.

[18] Kazakov, A., and Frenklach, M., 1994. http://www.me.berkeley.edu/drm/.

[19] Kingma, D. P., and Ba, J., 2014. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980*.

[20] Gal, Y., and Ghahramani, Z., 2016. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In international conference on machine learning, PMLR, pp. 1050–1059.