AIAA JOURNAL
Vol. 56, No. 6, June 2018

Common Proper Orthogonal Decomposition-Based Spatiotemporal Emulator for Design Exploration

Shiang-Ting Yeh,* Xingjian Wang,† Chih-Li Sung,‡ Simon Mak,‡ Yu-Hung Chang,* Liwei Zhang,§
C. F. Jeff Wu,¶ and Vigor Yang**
Georgia Institute of Technology, Atlanta, Georgia 30332

DOI: 10.2514/1.J056640

The present study develops a data-driven framework trained with high-fidelity simulation results to facilitate decision making for combustor designs. Its core is a surrogate model employing a machine-learning technique called kriging, which is uniquely combined with data-driven basis functions to extract and model the coherent structures underlying the flow dynamics. This emulation framework encompasses a sensitivity analysis of key design attributes, physics-guided classification of design parameter sets, and flow evolution modeling for a efficient design survey. A sensitivity analysis using Sobol indices and a decision tree is incorporated into the framework to better inform the model. The novelty of the proposed approach is the construction of the model through common proper orthogonal decomposition, allowing for data reduction and extraction of common coherent structures. As a specific example, the spatiotemporal evolution of the flowfields in swirl injectors is considered. The prediction accuracy of the mean flow features for new swirl injector designs is assessed, and the flow dynamics is captured in the form of power spectrum densities. The framework also demonstrates the uncertainty quantification of predictions, providing a metric for model fit. The significantly reduced computation time required for evaluating new design points enables an efficient survey of the design space.

Nomenclature

$\boldsymbol{c}$	=	design setting (parameter set)
D	=	Sobol response variance over design range
f	=	flow property
h	=	liquid-film thickness at injector exit
L	=	injector length
$\mathcal{M}$	=	linear transformation scaling spatial features
p	=	design parameters
$\hat{p}_j$	=	jetlike proportion of training dataset
$\hat{p}_s$	=	swirl-like proportion of training dataset
R	=	Gaussian correlation function
R_n	=	injector radius
T	=	temperature, K
t	=	time, s
$\boldsymbol{x}$	=	spatial coordinate
Z	=	zero-mean Gaussian process
α	=	liquid-film spreading angle at injector exit
β_j	=	time-varying coefficients
ΔL	=	distance between injector inlet and headend
δ	=	inlet slot width
θ	=	tangential inlet angle
μ	=	mean
ρ	=	density, kg/m^3
ϕ_j	=	spatial basis functions

Subscripts

n	=	number of simulations
u	=	subset of design parameters $u \subseteq \{1, \ldots, p\}$

I. Introduction

FOR high-performance power generation and propulsion systems, such as those of airbreathing and rocket engines, physical experiments are extremely expensive due to the harsh requirements of operating conditions and the high level of system complexities [1–3]. In addition, it is difficult to gain insight into the underlying mechanisms of the physiochemical processes involved because of the reliance upon optical diagnostics for experimental measurements [4–6].

High-fidelity simulations can be employed to capture more salient features of the flow and combustion dynamics in engines [7,8]. These computations, however, are often too expensive and time consuming for design survey purposes. In the development process for a propulsion engine, achieving an optimal design requires models capable of evaluating designs and identifying tradeoffs in a timely manner. Furthermore, the formulation of such models requires understanding of key physics and incorporation of decision making to resolve multiple, potentially conflicting, requirements. The present study, as a specific example, treats a simplex swirl injector, which is a central component of many airbreathing and rocket combustion devices [9–11]. The model contains rich flow physics. Each high-fidelity calculation of the three-dimensional flow evolution using the large-eddy simulation (LES) technique takes about 500,000 CPU hours to obtain statistically meaningful data for a grid of 4 million mesh points [12,13]. Given the number of geometric attributes and operating conditions to be surveyed, the design space exploration necessitates a prohibitive number of sample points. The situation can be substantially improved by using design of experiments (DOE) [14] statistical methodologies to determine the ideal training dataset for surrogate modeling. With the identification of sensitive injector parameters, the sample size can be further reduced based on semiempirical approaches, such as the recommendation of 10 design points for each parameter proposed by Loeppky et al. [15].

Kriging, a technique originating in the field of geostatistics [16], is a powerful machine-learning tool for interpolation and prediction. The key idea is to model unobserved responses using a Gaussian process (GP) governed by a preset covariance function. The response surface of the trained kriging model can then be obtained by applying

Received 11 September 2017; revision received 8 January 2018; accepted for publication 21 February 2018; published online 30 April 2018.

*Graduate Student, School of Aerospace Engineering.
†Research Engineer, School of Aerospace Engineering; xingjian.wang@gatech.edu (Corresponding Author).
‡Graduate Student, School of Industrial and Systems Engineering.
§Research Engineer, School of Aerospace Engineering.
¶Professor and Coca-Cola Chair in Engineering Statistics, School of Industrial and Systems Engineering.
**William R. T. Oakes Professor and Chair, School of Aerospace Engineering.

data-tuned weights to radial basis functions centered at observed points. Thus kriging can use function values sampled at a set of input parameters and approximate well the entire function surface over its domain. With appropriately chosen parameters, the kriging model provides the best linear unbiased estimator of the responses at designs that have not been simulated [14]. Furthermore, the resulting posterior distribution of this prediction will also be Gaussian. For problems of modeling spatiotemporal flow evolution, the observed points over the entire design space are sparse because the daunting computational costs limit the number of affordable simulation cases. Conventional machine-learning techniques relying on "big data" over the design space would fail. Rather, the big data lies within the flowfield information, which encompasses a wide range of length and time scales. The proposed methodology combines machine-learning techniques with domain knowledge of the physical system to build an accurate emulator model [17]. The inclusion of flow physics allows the data-driven model to be physically interpretable with enhanced emulation performance.

The objective of the present study is to develop a kriging model capable of treating different spatial grids while capturing dynamic information [17]. The spatial and temporal resolution of all simulation cases is very fine, making direct use of the raw data for training a predictor computationally demanding. Not only is there a need to incorporate data generated from different spatial grids [18–20] and use data-reduction methods, but the kriging model must also be extended to multiple, functional outputs. A handful of studies have been published on multiple-output kriging [21,22] and functional outputs, including wavelet decomposition [23] and knot-based GP models [24]. For data with fine spatiotemporal resolution, unfortunately, these types of methods are inappropriate because of the substantially increased computation time required [17].

Here, an emulation framework for a spatiotemporal surrogate model is presented using a simplex swirl injector for demonstration. A reduced-order model is developed and implemented to handle large-scale spatiotemporal datasets with practical turnaround times for design iteration. Recent reduced-order model studies have not focused on modeling and predicting spatiotemporal flowfields but rather focusing on closure terms [25–30]. Prior attempts at using Gaussian process models [31] and decomposition techniques with Galerkin projection leveraging radial basis functions have shown some success for unsteady flowfields [32–36]. The proposed model is trained with datasets that have been classified based on established physics to reap the benefits of incorporating machine-learning techniques into the framework. The model can accurately retain the rich set of physics from LES-based high-fidelity simulations and predict flow structures.

The present study develops an integrated framework that incorporates state-of-the-art statistical methods, machine-learning algorithms, and a physics-driven data reduction method to obtain a surrogate model over a broad range of design space. The emulation framework relies on proper orthogonal decomposition (POD) [37] (also known as the Karhunen–Loeve decomposition in the theory of stochastic processes [38]) to extract the flow physics and reduce the data by representing the flowfield with basis functions. This technique can be combined with kriging to build an efficient and physics-driven emulator. The common POD (CPOD) analysis is introduced and conducted by means of a common grid generated from simulations of the geometries designated by a DOE. Although this approach is similar to that of Higdon et al. [39] for generalizing a POD expansion, the novel technique developed herein directly addresses the need for a set of common basis functions required for a kriging model. In our companion paper on basic theories [17], the statistical properties of a broader class of CPOD-based emulators are considered.

The present work applies machine-learning techniques and investigates the practical performance of the emulator with respect to flow physics. The emulated flowfield is validated against an LES simulated flowfield to demonstrate how the flow structures and injector characteristics are captured by the model. In addition, the model allows for spatiotemporal uncertainty quantification. This metric can be used to verify the model and quantify underlying flow properties.

The paper is structured as follows. Section II provides the physical model, describing the baseline configurations, the design points designated by the DOE, the high-fidelity simulation technique, and the simulation results. Section III discusses the data-driven emulation framework proposed for the design methodology and surrogate model. Section IV details the application of the framework while assessing the surrogate model using performance metrics, root-mean-square errors, and the power spectral density (PSD) of simulated and predicted flowfields. Finally, Sec. V concludes with a summary and directions for future work.

II. Physical Model Description and Simulations

A. Swirl Injectors

Figure 1 shows a schematic of a simplex swirl injector representative of those commonly used in applications like liquid-fueled propulsion engines [9,10]. The five parameters that define the geometry are injector length L, injector radius R_n, inlet slot width δ, tangential inlet angle θ, and the distance between the inlet and the headend ΔL. These design parameters play an important role in determining the injector performance, including the thickness h, and spreading angle α, of the liquid film at the injector exit. The selection of these design parameters is dependent upon engine requirements. Table 1 shows the design space and the ranges of each parameter in the present work. To generalize the emulator framework, a broad range of these parameters is chosen. The injector length covers a broad range, including those of small upper-stage and large first-stage engines, or about 22.7 and 93 mm for the RD-0110 [40] and RD-170 engines [41], respectively.

Liquid oxygen at a temperature of 120 K is delivered tangentially into the injector through inlets. The operating pressure is 100 atm, which is typical of contemporary liquid rocket engines. The ambient gas is oxygen at 300 K. The flow dynamics of this class of injectors have been systematically investigated in detail by Zong and Yang [12] and Wang et al. [13]. Here, we first conduct a set of high-fidelity simulations based on conditions in the design space described in Table 1, then extract the common flow structures for surrogate modeling.

B. Design of Experiments

Given the design space in Table 1, if 10 variations are assigned for each design parameter, the total number of design points is 10^5 for a full factorial design. It is impractical to perform so many simulations, due to the extensive computing resource required to acquire usable data. A DOE methodology is therefore required to reduce the number of design points and still capture the prominent features in the design space. To this end, the maximum projection (MaxPro) design proposed by Joseph et al. [42] is implemented for good space-filling properties and GP modeling predictions. Thirty points in the expected range of $5–10p$ ($6p$ rule with $p = 5$, which is the number of design parameters) points, as suggested by Loeppky et al. [15], and commonly used in computer experiment literature, are simulated over the entire design space. The accuracy of prediction should always be checked to determine whether additional simulations are needed (see Loeppky et al. [15]). Figure 2 shows a two-dimensional projection of the 30 simulation runs by MaxPro design, which gives representative design points distributed to fill the two-dimensional projection of the design space. Good space-filling properties are observed for all parameters.

C. High-Fidelity Simulation

An integrated theoretical and numerical framework is established and implemented to treat supercritical fluid flows and combustion

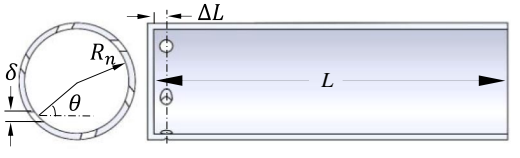

Fig. 1 Schematic of a swirl injector.

Table 1 Design space for injector geometric parameters

Parameter	Value
L, mm	20–100
R_n, mm	2.0–5.0
θ, deg	45–75
δ, mm	0.5–2.0
ΔL, mm	1.0–4.0

Table 2 Injector geometrics at design points colored blue in Fig. 2

Design	L, mm	R_n, mm	θ, deg	δ, mm	ΔL, mm
A (swirl)	20.0	3.22	52.9	0.52	3.42
B (jetlike)	41.9	3.05	65.5	1.57	1.00
C (swirl)	43.1	5.00	70.0	0.50	2.79
D (jetlike)	37.7	2.82	45.8	1.17	3.80

over a broad range of fluid thermodynamic states [43–45]. Turbulence closure is achieved using LES. Thermodynamic properties are evaluated by fundamental thermodynamics theories in accordance with the modified Soave–Redlich–Kwong equation of state. Transport properties are estimated using extended corresponding-state principles [43]. The numerical scheme is a density-based finite volume methodology along with a dual-time-step integration technique. The overall algorithm is self-consistent and robust, with implementation of a preconditioning scheme and a detailed treatment of general fluid thermodynamics [44,45].

Owing to the demanding computational requirements of three-dimensional simulations, only a cylindrical sector with periodic boundary conditions in the azimuthal direction is simulated. The objective is to ensure that the emulation captures the liquid-film development within and in the downstream region of the injector simulation. The azimuthal flow dynamics is not a major concern because the focus is on developing a model that retains the physics of any spatiotemporal flowfield. The discrete injection orifices are converted into an axisymmetric slot through dynamic similarity. A multiblock domain decomposition technique, combined with a message passing interface for parallel computing, is applied to improve computational efficiency. A typical simulation takes about 30,000 CPU hours on a single Intel Xeon processor to obtain statistically significant data, with a total span of 30 ms physical time, after reaching a fully developed state (~24 ms). The simulated data are sampled every 30 computational time steps, with 1 μs between time steps. According to the Nyquist criterion, a temporal resolution of 16.5 kHz is achieved.

D. High-Fidelity Simulation Results

Thirty high-fidelity simulations at design points defined by MaxPro were conducted. To isolate the effect of injector parameters, the mass flow rate for all runs is fixed at 0.15 kg/s. The first two design points designated by MaxPro are chosen as the baseline geometries: A and B in Table 2. The benchmark points used for assessing the accuracy of the emulator model are obtained by offsetting the design parameters of these two points.

Figures 3 and 4 show the instantaneous distributions of the temperature and density for two neighboring design points (C and D in Table 2), which were selected to indicate different flow features in the design space. The key flow structures include the swirling liquid film along the wall due to centrifugal force, liquid accumulation near the injector headend and associated flow recirculation, and a conical liquid sheet spreading outward at the injector exit propelled by azimuthal momentum and a hollow gas core in the center region [12,13].

Various flow physics are observed. The film thickness for design C is much thinner than for design D, with a larger spreading angle at the injector exit (34.6 deg as compared to 29.2 deg for design D). Among the 30 design points, some act like swirling flows, as in design C; whereas others behave like jet flows, as in design D. For convenience, the critical value of the spreading angle that separates swirling from jetlike flows is chosen to be 30 deg; this angle is considered to be an empirical indicator of whether the liquid stream has significant radial penetration in the downstream region. When this angle is not achieved, the liquid does not have enough radial momentum to spread outward. The 30 simulation runs are thus divided into two subgroups: swirling (spreading angle above 30 deg) and jetlike (spreading angle below 30 deg) flows. In the next section, a machine-learning technique, known as the decision tree, is introduced to identify the jet-swirl dichotomy. This directly influences the feature extraction and kriging processes described in the following sections, as it changes

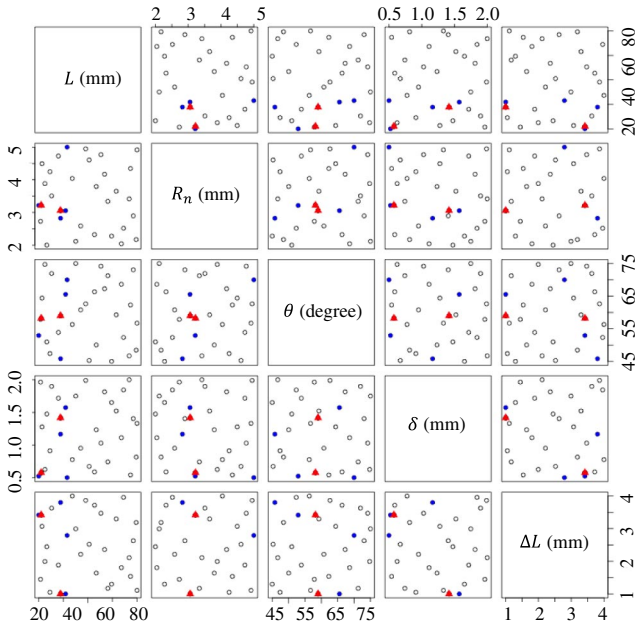

Fig. 2 Two-dimensional projections of design points: benchmark points (triangles), and baseline and neighboring points (filled circles).

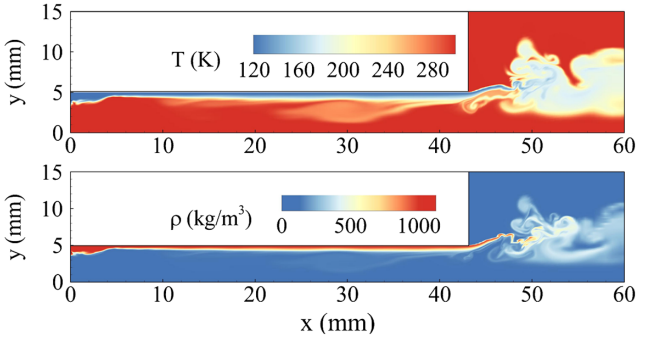

Fig. 3 Instantaneous distributions of temperature and density for design C.

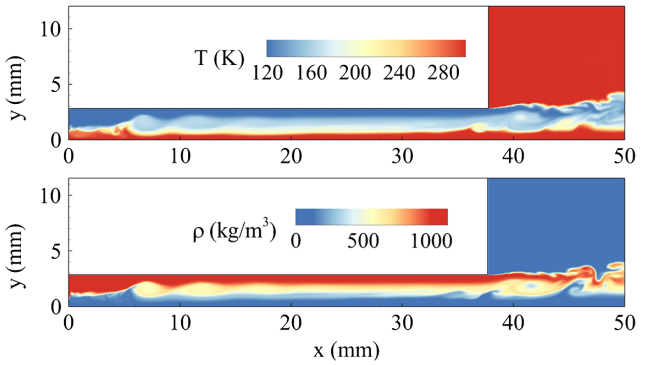

Fig. 4 Instantaneous distributions of temperature and density for design D.

how the design space is partitioned between the identified flow behaviors. Implicitly, the extracted coherent structures change slightly, depending on the established criteria separating swirling and jetlike flows.

III. Data-Driven Framework

Design points may display similar or significantly different flow structures. In this section, the collected dataset from all 30 high-fidelity simulations is used to perform a data-driven analysis of the design space using machine-learning tools. The work consists of two components: 1) a sensitivity analysis for identifying important design parameters with respect to the quantities of interest, where the Sobol indices [46] are used and b) a decision-tree learning process with respect to the jet-swirl dichotomy, and the incorporation of this information into the emulator model. After the sensitivity analysis and decision-tree learning, a technique called common POD is then implemented to extract the flow characteristics over the design space. Lastly, the time coefficients for the obtained basis functions are employed as training data for the kriging model. This methodology allows us to make accurate flow predictions at any new design setting. A flowchart of the overall data-driven emulator framework is provided in Fig. 5.

A. Sensitivity Analysis

The first component of this emulator framework is a sensitivity analysis using Sobol indices [46] to identify which design parameters contribute more to changes in responses of interest, such as liquid-film thickness or spreading angle. The analysis is also a valuable tool for parameter reduction. The idea is to decompose the variations of certain desired output variables into the partial variations attributable to each input parameter and the effects of interactions between parameters. Such a method of analyzing sensitivity has close connections to the classical analysis of variance employed in linear regression models [47].

To put it in mathematical terms, let $f(\boldsymbol{c})$ be the desired response output at design setting $\boldsymbol{c}$, where $\boldsymbol{c} = (c_1, c_2, \ldots, c_p)$ corresponds to the input parameters over a unit hypercube $[0, 1]^p$. Specifically, for the current study, $p = 5$, $c_1 = L$, $c_2 = R_n$, $c_3 = \theta$, $c_4 = \delta$, and $c_5 = \Delta L$, with the design range for all parameters normalized to the interval [0,1]. Define the random variable $\boldsymbol{X}$ as a uniform distribution over $[0, 1]^p$, and let $f_0 = \mathbb{E}[f(\boldsymbol{X})]$ be the response mean and $D = \mathrm{Var}[f(\boldsymbol{X})]$ be the response variance over the design range. The goal is to decompose the response variance D into the contributions for each design parameter $c_1, \ldots, c_p$, as well as the effects of interactions between parameters. Consider the following decomposition:

$$f(\boldsymbol{c}) = f_0 + \sum_{i=1}^{p} f_i(c_i) + \sum_{1\le i<j\le p} f_{i,j}(c_i, c_j) + \ldots + f_{1,2,\ldots,p}(c_1, \ldots, c_p) \quad (1)$$

where each summand satisfies

$$\int_0^1 f_{i_1,\ldots,i_t}(c_{i_1}, \ldots, c_{i_t})\,\mathrm{d}c_k = 0 \quad (2)$$

for any $k = i_1, \ldots, i_t$ and has orthogonal components. In Eq. (1), the main effect index of input i is

$$f_i(c_i) = \int (f(c) - f_0)\,\mathrm{d}c_{-i}, \qquad c_{-i} = \{c_1, \ldots, c_p\} \setminus \{c_i\} \quad (3)$$

and the twoway interaction index of inputs i and j is

$$f_{i,j}(c_i, c_j) = \int \{f(c) - f_0 - f_i(c_i) - f_j(c_j)\}\,\mathrm{d}c_{-i,j}, \qquad c_{-i,j} = \{c_1, \ldots, c_p\} \setminus \{c_i, c_j\} \quad (4)$$

Squaring both sides of Eq. (1) and taking the integral over $[0, 1]^p$, we get the following:

$$D = \sum_{i=1}^{p} D_i + \sum_{1\le i<j\le p} D_{ij} + \sum_{1\le i<j<l\le p} D_{ijl} + \ldots + D_{1,2,\ldots,p} \quad (5)$$

where D_u is the partial variance corresponding to a subset of parameters $u \subseteq \{1, \ldots, p\}$:

$$D_u = \int f_u^2(c_u)\,\mathrm{d}c_u \quad (6)$$

The Sobol sensitivity indices for parameter subset u can be defined as follows [46]:

$$S_u = \frac{\mathrm{D}_u}{\mathrm{D}} \in [0, 1] \quad (7)$$

with larger values of S_u indicating greater importance of the interaction effect for u.

In practice, Sobol indices can be estimated as follows. First, a pseudorandom parameter sequence is generated using a low discrepancy Sobol point set [48]. Second, this sequence is used to approximate the aforementioned integrals, which can then provide estimates for the corresponding Sobol indices. The quantification of the response sensitivity for each parameter serves two purposes:

1) It provides a preliminary analysis of important effects in the system, which can guide further physical investigations.

2) It allows for a reduction of the number of parameters that must be considered in the emulator, thereby providing a computationally efficient way to survey flow properties within the design space.

A detailed discussion of the sensitivity analysis is presented in Sec. IV for the current physical model.

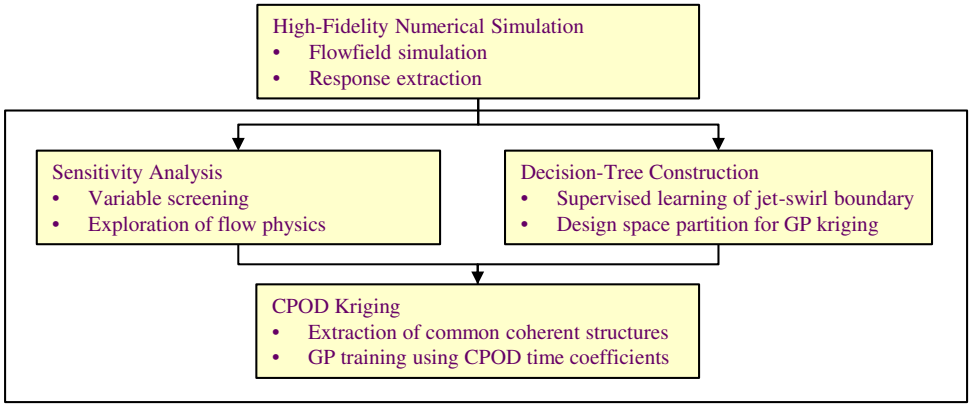

Fig. 5 Flowchart for data-driven analysis and emulator construction.

B. Decision Tree

As mentioned in Sec. II.D, there exists a jetlike/swirling flow dichotomy within the design space. For simulated design points, it is easy to classify whether such a parameter combination results in a jetlike or swirling flow because the flowfield data are readily available. For design settings that have not been simulated, a data-driven technique is needed to make such a classification. There are two reasons why such a classification tool may be of interest. First, a boundary between jetlike and swirling cases can be established over the design space of interest, which can then be used to gain physical insight into the design space and to guide additional experiments. Second, the classification information can be used to train separate surrogate models within the jetlike and swirling domains. This partitioning of the emulator training dataset allows the model to extract different flow characteristics associated with jetlike and swirling behaviors separately, and it can thereby improve its predictive accuracy. A powerful machine-learning tool "decision tree" is employed for the classification process.

A decision tree is a decision support tool that models decisions and their possible consequences. Decision trees are one of the most popular predictive models in data mining and machine learning [49,50]. Such methods are a part of a larger class of learning methods called supervised learning [51], which aims to predict an objective function from labeled training data. A classification tree, a special type of decision tree, is used here. It specializes in predicting classification outcomes, such as whether a parameter set has a jetlike or swirling flow. The trained model can be summarized by a binary tree, separating the design space into two subgroups. Each node of this tree represents a parameter decision, and each leaf of the tree indicates the class of outcomes, following the chain of decisions made from the tree root.

A classification tree can be trained using the following algorithm (see [52] for details). First, the simulated flowfields of each sampled design point are examined and classified as either jetlike or swirling flow, depending on the radial penetration of the propellant in the downstream region. Next, a search is conducted over all the design parameters and possible split points, finding the parameter constraints that minimize misclassification. A branch is then made in the classification tree corresponding to the parameter constraint. The same branching procedure is repeated for each of the resulting child nodes. For the analysis in Sec. IV, the Gini impurity index [52] is selected as the misclassification measure:

$$\hat{p}_j(1-\hat{p}_j)+\hat{p}_s(1-\hat{p}_s) \tag{8}$$

where $\hat{p}_j$ and $\hat{p}_s$ are the proportions of jetlike and swirl cases in a split. The index measures how often a randomly chosen sample is incorrectly labeled when such a label is randomly assigned from the dataset. Notice that a Gini index of zero indicates that 1) $\hat{p}_j = 1$ and $\hat{p}_s = 0$, or 2) $\hat{p}_j = 0$ and $\hat{p}_s = 1$, both of which suggest perfect classification. When the Gini impurity index is around 0.5, jetlike and swirl cases are equally distributed. If more than two groups are considered in the injector dynamics, the Gini impurity index can be generalized for other numbers of groups, which can be seen in [52].

This decision-tree learning technique not only provides a means for partitioning the training dataset for the model into jetlike and swirling flows but also reveals physical insights on the important design parameter constraints causing this jet-swirl dichotomy. The quantification of this split is achieved through the calculation of the Gini impurity. The Gini index is a criterion to minimize for classification. Note that 0.5 is the worst classification possible, but the optimization procedure aims to find the best classification possible (i.e., one with the smallest Gini index). If this optimal classification with two categories is not good enough, then the approach should be generalized for classification trees with more than two categories. The interpretability of these constraints is elaborated on in Sec. IV.

C. Kriging Surrogate Model (Emulator)

The primary objective of this work is to develop an emulator model that uses data from 30 simulation runs to predict the flowfield of a new design point within a practical turnaround time. With the tools described above (the sensitivity analysis for parameter screening and the decision tree for partitioning the design space into jet-swirl cases), a surrogate model for flowfield emulation is proposed. The kriging surrogate model, also known as an emulator, combines machine-learning techniques, statistical modeling, and a physics-driven data reduction method. A brief description of each part of this model is provided before delving into the specific mathematical details. A complete description of the model development from the statistical perspective is given in [17].

First, the proposed model is constructed through a POD analysis of the simulation dataset used for training. For a given flow property f, the POD analysis determines a set of orthogonal basis functions ϕ_j such that the projection of the property onto these basis functions has the smallest error, defined as $E(\|f-\hat{f}\|^2)$, where $E(\cdot)$ and $\|\cdot\|$ denote the time average and norm in the L^2 space, respectively [37]:

$$\hat{f}(\boldsymbol{x},t)=\sum_{j=0}^{n}\beta_j(t)\phi_j(\boldsymbol{x}) \tag{9}$$

The basis functions, or mode shapes, are spatial distributions of the fluctuating fields of flow properties, which can be closely linked to physical phenomena and coherent structures. The basis functions are ordered in such a way that the lowest modes have the highest "energy," as defined by the inner product of f. The flow properties for POD analysis include pressure, density, temperature, and velocity components. POD decomposition yields not only the eigenfunction modes ϕ_j, but also their corresponding time-varying coefficients β_j, which are referred to as POD coefficients. It should be noted that this process is not completed for the entire dataset, with physical variables being processed separately. To treat the data together, the scaling and dimensions need to be carefully formulated to obtain interpretable mode shapes.

Although the usage of POD simplifies the complex nature of a spatiotemporal model, a common set of basis functions is required for the emulator in order to accommodate different injector geometries. Physically, this means that a common set of coherent structures needs to be extracted over the design space. One option is to select a computational region of interest that is unaffected by any design changes [53]. Taking advantage of the basis functions generated by the POD analysis, an emulator can be obtained as long as a set of common basis functions exists. One of the challenges for the current study is the wide disparity of geometries in the design space, as illustrated in Fig. 6. The present work uses a common grid for the 30 grid systems to find a set of common basis functions. To achieve this, the densest grid system among all cases is identified and split into four sections covering the effects of design parameters on the simulated grid. This partitioned grid is used for interpolation and rescaling of each simulated case to obtain a common grid. Then, an inverse distance weighting interpolation method with 10 nearest-neighborhood points is used to map the original raw data onto the common grid [54]. Algorithmically, the CPOD expansion is obtained by first rescaling the different cases to the common grid, then computing the POD expansion, and finally rescaling the resulting modes back to the original grid [17].

Because of the limited variation of the Reynolds number among the different injector geometries, the scaling of the data to the common grid is appropriate in the present study. The smallest injector diameter of concern is 4 mm, with a corresponding exit velocity of 27.5 m/s. With the liquid oxygen density of 1000 kg/m^3 and the viscosity of 0.114 cP, the Reynolds number based on the injector diameter is about 9.6×10^5. The largest injector diameter in the design space has a value of 10 mm, and the corresponding exit velocity is 11 m/s. At the same operating condition, the Reynolds number is about 9×10^5. For some geometries where the liquid film does not produce a noticeable spreading angle, the Reynolds number is reduced to about 9×10^5. Despite this difference, the model is

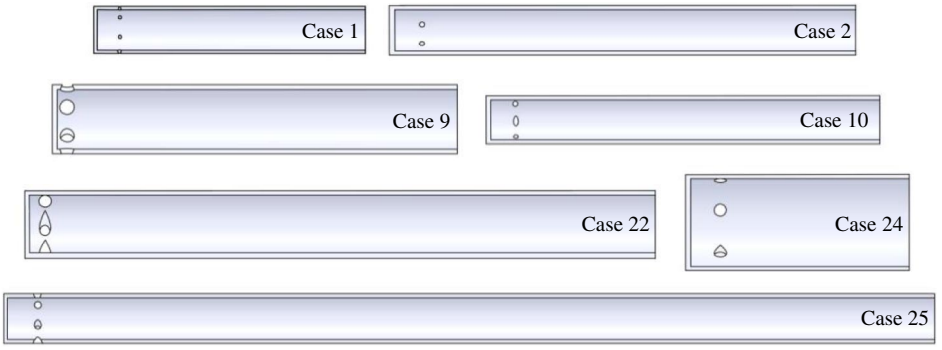

Fig. 6 Schematics of different injector geometries in the design space.

capable of avoiding excessive smoothing, provided the correlation function is bounded correctly. Such scaling of POD modes to establish common basis functions is vital to the emulator. It should be noted that the scaling is only appropriate for flow simulations that do not exhibit distinctively different physical phenomena, such as those of reacting-flow simulations, where the mode shapes change drastically. Additional similarity parameters may be necessary when different physics and chemical reactions are incorporated, as noted by Dexter et al. [55].

The mathematical details for CPOD are provided in the following [17]. Suppose n simulations are conducted at various design geometries $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n$ and let $f(\boldsymbol{x}, t; \boldsymbol{c}_i)$ be the simulated flowfield at design $\boldsymbol{c}_i$ for a given time t and spatial coordinate $\boldsymbol{x}$. The kth CPOD mode is defined as

$$\phi_k(\boldsymbol{x}) = \underset{\psi:\,\|\psi\|_2=1}{\operatorname{argmax}} \sum_{i=1}^{n} \int \left[\int \mathcal{M}_i[\psi(\boldsymbol{x})] f(\boldsymbol{x}, t; \boldsymbol{c}_i)\, \mathrm{d}\boldsymbol{x} \right]^2 \mathrm{d}t,$$
$$\text{s.t.} \int \psi(\boldsymbol{x}) \phi_l(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = 0, \quad \forall\ l < k \tag{10}$$

Here, the map $\mathcal{M}_i: \mathbb{R}^2 \to \mathbb{R}^2$ is the transformation that linearly scales spatial features from the common geometry $\boldsymbol{c}$ to the ith geometry $\boldsymbol{c}_i$. The sequence of POD coefficients is defined as follows:

$$\beta_k(\boldsymbol{c}_i, t) = \int \mathcal{M}_i\{\phi_k(\boldsymbol{x})\} f(\boldsymbol{x}, t; \boldsymbol{c}_i)\, \mathrm{d}\boldsymbol{x} \tag{11}$$

with the corresponding POD expansion using K modes given by the following:

$$f^{(K)}(\boldsymbol{x}, t; \boldsymbol{c}_i) = \sum_{k=1}^{K} \beta_k(\boldsymbol{c}_i, t) \mathcal{M}_i\{\phi_k(\boldsymbol{x})\} \tag{12}$$

The transformation allows for the extraction of common basis functions. In addition, the obtained modes can be used to identify key mechanisms of flow dynamics. It should be noted that reacting-flow simulations are characterized by additional dimensionless parameters and linear mapping may not perform well when combustion is involved.

Two computational challenges need to be addressed to implement this methodology. As previously mentioned, to calculate the inner product of the snapshots from different simulation cases, a common set of spatial grid points is needed. Not only does the calculation of the inner product become a computational bottleneck because the covariance matrix consists of snapshots from each simulation, the number of modes required to capture a certain energy level is significantly increased relative to an individual simulation, which needs approximately six modes to capture more than 95% of the total energy [12]. The computation of CPOD modes and associated time-varying coefficients requires eigendecomposition of a $nT \times nT$ matrix, where n is the number of simulation cases and T the number of snapshots. This usually requires $\mathcal{O}(n^3T^3)$ computation work. A typical value for T is 1000 snapshots spanning 10 ms, which achieves a frequency resolution of 100 Hz. An iterative method of eigendecomposition based on periodic restarts of Arnoldi decompositions is then used to quickly calculate the first few eigenvectors with the largest eigenvalues. These eigenvalues can also be interpreted as the amount of the energy as defined by the inner product used to calculate the covariance matrix. For a particular reconstruction using a linear combination of POD modes and associated time-varying coefficients, there is reconstruction error, which decreases when more eigenvectors (the POD modes) are included.

Next, a kriging model is applied to the CPOD time-varying coefficients $\beta_k(\boldsymbol{c}_i, t)$. With the mean and variance computable in closed form, uncertainty quantification and confidence intervals can be calculated easily. Kriging (and, more generally, GP-based learning) has been applied to great success in a variety of fields [56]. The mathematical approach of kriging is described here. For notational simplicity, let $\beta(\boldsymbol{c})$ denote $\beta_k(\boldsymbol{c}, t)$, which is the k-th CPOD coefficient at setting $\boldsymbol{c}$ and time step t. As the temporal resolution is fine, there is no practical need to estimate temporal correlations, especially because predictions will not be made in between time steps. This time-independent emulator uses independent kriging models at each instant of time, assuming the following GP model:

$$\beta(\boldsymbol{c}) = \mu + Z(\boldsymbol{c}), \qquad Z(\boldsymbol{c}) \sim N\{0, \sigma^2 R(\cdot,\cdot)\} \tag{13}$$

Here, μ is the mean, $Z(\boldsymbol{c})$ is a zero-mean GP with variance σ^2, and $R(\cdot,\cdot)$ is a prespecified correlation function governed by unknown parameters η. A typical choice for $R(\cdot,\cdot)$ is the so-called Gaussian correlation function:

$$R(\boldsymbol{c}_i, \boldsymbol{c}_j) = \exp\left[-\sum_{k=1}^{p} \eta_k (c_{ik} - c_{jk})^2 \right] \tag{14}$$

where p is the number of input parameters.

Now, suppose the function values $\boldsymbol{\beta}^{(n)} = [\beta(\boldsymbol{c}_i)]_{i=1}^n$ are observed at input settings $\{\boldsymbol{c}_i\}_{i=1}^n$, and let $\boldsymbol{c}_{\text{new}}$ be a new setting for which prediction is desired. Conditional on the observed values $\boldsymbol{\beta}^{(n)}$, the best linear unbiased estimator of $\beta(\boldsymbol{c}_{\text{new}})$ can be shown to be [14]

$$\hat{\beta}(\boldsymbol{c}_{\text{new}}) = \mu + \boldsymbol{r}_{\text{new}}^T \boldsymbol{R}^{-1} (\boldsymbol{\beta}^{(n)} - \mu \mathbf{1}) \tag{15}$$

Here, $\mathbf{1}$ is the $n \times 1$ vector of ones,

$$r_{\text{new}} = [R(\boldsymbol{c}_i, \boldsymbol{c}_{\text{new}})]_{i=1}^n$$

is the $n \times 1$ vector of correlations between the new point and sampled points, and

$$\boldsymbol{R} = [(\boldsymbol{c}_i, \boldsymbol{c}_j)]_{i=1\,j=1}^{n\quad n}$$

is the covariance matrix for the sampled points. Such a predictor minimizes the mean-squared prediction error (MSPE), a which is

commonly used criterion for prediction error. In the context of flowfield prediction, employing this kriging estimator allows us to obtain accurate flow predictions from the CPOD coefficients. It can also be shown [14] that this best MSPE predictor is unbiased, matching the expected and true function values.

To close the formulations, the model parameters μ, σ^2, and η need to be trained using data. A technique called maximum likelihood estimation (MLE), which is a ubiquitous estimation technique in statistical literature [57], is employed. The key idea in MLE is to search for the optimal parameter setting that minimizes the likelihood function of the GP model. In the present work, optimization is achieved by means of the L-BFGS algorithm [58], which is a method employed for many training algorithms. A more detailed explanation can be found in the work of Santner et al. [14].

The kriging models are trained independently over each time step, due to the inherent fine-scale temporal resolution of the simulation. This time-independence assumption is made for two reasons. First, the fully developed flow is treated as statistically stationary and has high-frequency resolution, so there is no practical value for estimating temporal correlations. Second, as in the high-fidelity simulation procedure, the assumption of time independence allows exploitation of parallel computation in training the emulator model. Once the model is trained, the predictor is used with the CPOD expansion to predict the flow evolution at a new design point, that is,

$$\hat{f}(\boldsymbol{x}, t; \boldsymbol{c}_{\text{new}}) = \sum_{k=1}^{K} \hat{\beta}_k(\boldsymbol{c}_{\text{new}}, t)\mathcal{M}_i\{\phi_k(\boldsymbol{x})\} \tag{16}$$

It is worth noting that the computation time of the proposed model is orders of magnitude smaller than that of LES. Simulation data that typically take a week, or around 30,000 CPU hours, to acquire can be predicted by the model with an associated uncertainty in a manner of tens of minutes. The full emulator model and algorithm are provided in the statistical paper [17], which considered the statistical properties of a broader class of models. The current paper focuses on applying new machine-learning techniques and investigates the practical performance of the emulator with respect to flow physics.

IV. Results and Discussion

A. Sensitivity of Injector Geometrical Parameters

The liquid-film thickness and spreading angle are two important injector characteristics. An inviscid incompressible-flow theory predicts the spreading angle as a function of solely the geometric constant [9,10], and it increases with increasing geometric constant. For real fluids at supercritical conditions as treated in the present study, the fluid density varies continuously [12,13]. The spreading angle can be determined based on the slope of the maximum density gradient near the injector exit in a time-averaged sense. As the maximum density gradient is used as the boundary for liquid film, the spreading angle and film thickness have variances related to how prominent the maximum density peak appears in the radial direction.

To gauge the importance of each injector parameter on the liquid-film thickness and spreading angle, a sensitivity analysis using a Monte Carlo estimate of Sobol indices was performed [46]. Figure 7 shows the primary effects from this sensitivity analysis. The points indicate the Sobol index estimate for each design parameter, with lines indicating the Monte Carlo integration error for each index estimate. The lines were calculated based on a 95% confidence interval of the estimate. The slot width (δ) was found to be the parameter with the largest Sobol index, and thus the strongest influence on the spreading angle. Physically, this could be explained by how geometric parameters govern the inlet flow properties. Assuming a constant mass flow rate, the incoming velocity was inversely proportional to the slot width, and a decrease in slot width increased liquid-film momentum, increasing the momentum of the liquid film.

Similarly, the tangential inlet angle (θ) and the slot width significantly affect the liquid-film thickness, whereas the length (L) and radius (R_n) of the injector have minor effects. The tangential inlet angle controls the direction of momentum. As the injector angle increases, more azimuthal momentum is imparted to the liquid film, thereby increasing the spreading angle at the injector exit. The length and radius can dictate how much viscous loss is experienced by the propellant as it travels in both the axial and azimuthal directions. The present study, however, has shown viscous losses to be a minor effect. Referring to Eq. (7), larger values of S_u indicate greater importance of the interaction effect for u. When $|u| = 1$, the sensitivity is called the mean effect index. Suppose $|u| = 1$, another measure of sensitivity often considered is the total effect index, which measures the contribution to the output of a given input $\boldsymbol{X}_u$, including all interactions of $\boldsymbol{X}_u$ with other inputs. That is,

$$T_u = S_u + \sum_{l \in u^c} S_{l \cup u} \in [0, 1]$$

where u^c is the complementary set of u. Similarly, larger values of T_u indicate greater importance of the effect for u.

Figure 8 shows the two-factor interaction effects. It further demonstrates that the main design parameters are the slot width and the tangential inlet angle (interaction effect circled in blue), which couple to affect the liquid-film response. This is hardly surprising because the slot width and inlet angle govern the flow area and direction of momentum, respectively. The mass and momentum conservation equations are inherently coupled to govern the flowfield.

As previously mentioned, the empirical geometric constant for a swirl injector can be employed to estimate the film thickness and spreading angle, using the hydrodynamics theories described by Bazarov and Yang [9] and Basarov et al. [10]. These theories, however, are based on the assumption of incompressible, inviscid flows and can only be used as a preliminary guide. In real injectors, viscous and compressibility effects must be considered. The liquid viscosity results in boundary-layer formation along the walls, which

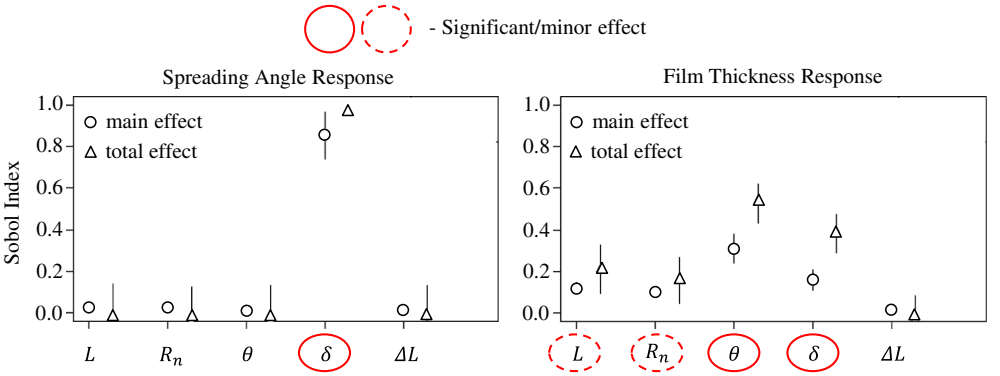

Fig. 7 Sensitivity analysis of liquid-film thickness and spreading angle.

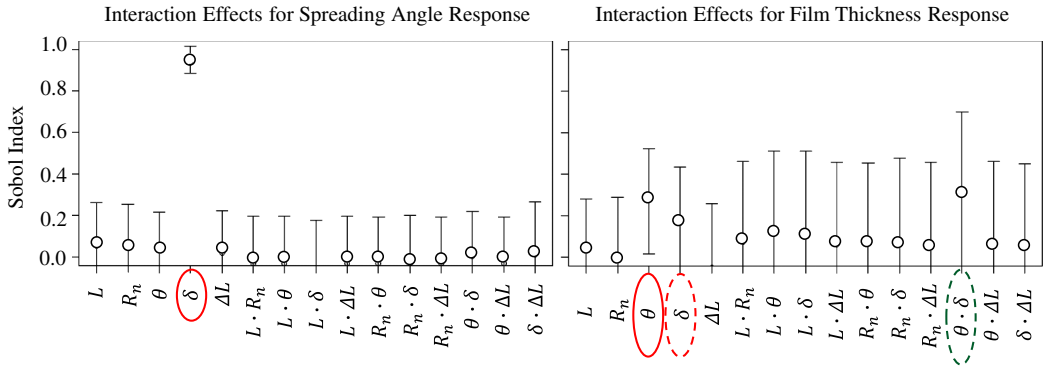

Fig. 8 Two-factor interaction of liquid-film thickness and spreading angle.

causes spatially nonuniform velocity profiles. A primary effect of compressibility lies in the existence of acoustic waves [12,13]. The supercritical conditions within high-pressure systems make these effects even more pronounced. High-fidelity simulations taking into account real-fluid effects are required to address these issues [12,13].

B. Decision-Tree Exploration of Injector Design Space

With further examination of simulated design points, a clear distinction exists between two different types of underlying physics. One type is the expected swirling film that noticeably spreads radially upon exiting the injector. The other is a jetlike behavior of the liquid film where the radial spreading is weak. The DOE methodology uses space-filling properties such that design points in both regimes are simulated. This section explores how to efficiently incorporate this information into the CPOD methodology to refine prediction results.

Designs A and B (geometric parameters are listed in Table 2) are each arbitrarily chosen, from among the simulated design points, as baseline geometry for determining offdesign points. By offsetting injector parameters, two benchmark design points are obtained (denoted as red points in Fig. 2). Design A is classified with swirling behavior. Although design B is classified with jetlike behavior in its developing stage, the flowfield transitions to a swirling flow in its stationary state. This trend may be an indicator that design B is near the jet-swirl regime boundary. Its stationary state was used to classify this hybrid physics case.

A full design tradeoff study requires quantifying how every parameter affects key performance metrics. Hence, all injector variables are retained for the first benchmark: E. The second benchmark, F, only varies design parameters with significant effects on the liquid-film response. The corresponding geometries are shown in Table 3. For benchmark E, each design parameter deviates +10% from that of design A. With normalized parameters, the distance traversed in the design space is estimated to be about 18.1%, as calculated in the L_2 linear sense.

The sensitivity study showed that the injector radius and the injection location had less effect than the slot width and tangential inlet angle on the film thickness and spreading angle. They are thus fixed, and the other three parameters are offset from design B by −10% to explore the design space at benchmark F. The closest two simulation points are designs C and D. The neighboring points are provided because design B seems to be near the jet-swirl dichotomy.

The second component of the data-driven framework for the design survey is a decision tree [51,52]. Figure 9 shows the decision-tree splitting process, indicating how the algorithm decides the way an injector parameter dictates whether the flow is jetlike or swirling. The initial decision between the two behaviors is achieved by assessing the extent to which the liquid film spreads radially from the

Table 3 Injector geometries for benchmark cases

Benchmark	L, mm	R_n, mm	θ, deg	δ, mm	ΔL, mm
E	22.0	3.22	58.2	0.576	3.42
F	37.7	3.06	59.0	1.417	1.00

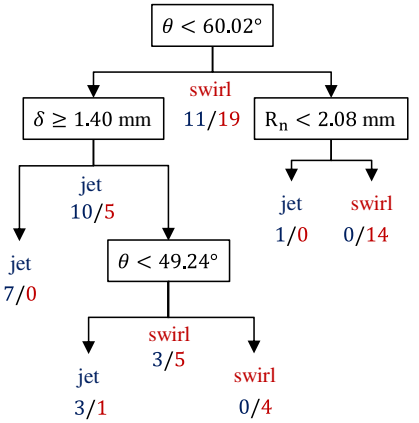

Fig. 9 Decision-tree splitting process with numeric classifiers.

injector exit. The numeric outputs are essentially binary flags between the two subgroup classifications. For example, the first numeric output of $\theta < 60.02$ deg, splits the dataset into 11 jetlike and 19 swirl cases. The decision tree then further classifies the data according to the injector inlet and radius. Intuitively, when the tangential inlet angle θ, is smaller, there is less azimuthal momentum in the liquid film to cause radial spreading. When the injector inlet δ, becomes large, the decreased momentum results in jetlike behavior. The decision tree quantifies these effects and predicts a jetlike injector with $\theta < 60.02$ deg and $\delta > 1.40$ mm. Following the previous two criteria, if the tangential inlet angle is large enough (that is, $\theta > 49.24$ deg), the injector retains swirling behavior.

The two benchmark cases are used to verify the decision tree. With such an algorithm, simulation results can be predicted using the model with proper training data. As the next section will further detail, the emulator relies upon the set of common basis functions extracted from the dataset. With two different types of underlying coherent structures, the two datasets should be trained separately to predict design parameter sets that lead to their corresponding flow behavior.

C. Surrogate Model

To train an emulator and make predictions, a set of common basis functions must be used as previously mentioned. Figure 10 shows the process of the common grid generation. The red lines partition the axisymmetric domain for each case into five regions: injector headend region, injector interior, and three subregions downstream of the injector. The densest grid system among the 30 training cases is selected as the common grid, upon which the partitioned regions for all other cases are then scaled to the corresponding regions in the common grid. This scaling is designed such that the ensuing model is able to leverage common basis functions, without significantly changing the flow features of interests. It should be noted that the scaling has a marginal impact upon liquid-film development visualization within the injector, which has the broadest range among

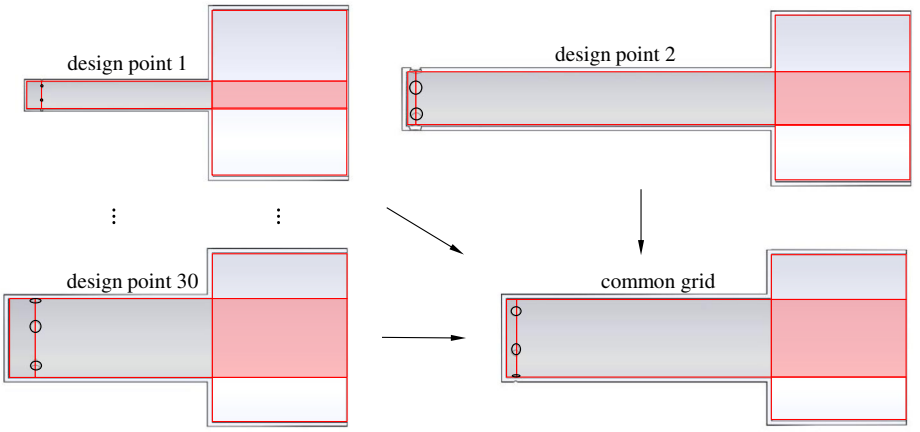

Fig. 10 Schematic of common grid generation process.

the design parameters. The original data are interpolated with an inverse distance weighting interpolation method using the 10 nearest neighborhood points to retain the fine points in regions of interest, specifically near the liquid film. The results on the common grid are used for the POD analysis.

Figure 11 shows the energy spectrum of the azimuthal velocity captured by the CPOD analysis. This spectrum is chosen as a demonstrative example because the overall behavior is shared by all other physical variables. Forty-five CPOD modes are required to retain 99% of the energy and limit the corresponding truncation error for the reconstruction. The leading two modes are presented in Fig. 12, with both indicating swirling flow structures with dominant fluctuations near the injector wall. The flow evolution within the injector and subsequent liquid-film development downstream of the exit are clearly observed.

The kriging of time-varying coefficients combined with the CPOD modes allows for emulation of the spatiotemporal evolving flow at a new design point. The CPOD modes represent the common physics extracted from the training dataset. A new injector geometry is assumed to produce similar flow physics including a hollow gas core, a swirling liquid film attached to the wall, and a conical liquid sheet spreading outward at the injector exit. Figure 13 shows snapshots of the temperature field for the simulation and emulations of benchmark E ($L = 22.0$ mm, $R_n = 3.22$ mm, $\theta = 58.2$ deg, $\delta = 0.576$ mm, and $\Delta L = 3.42$ mm). For the temperature CPOD analysis, 2000 modes, out of the 30,000 modes that can be extracted, are required to capture 90% of the energy and are used for the prediction. Good agreement is obtained, illustrating the same qualitative trends for the flow structures, with a liquid film along the injector wall and a center recirculating flow downstream of injector. The POD analysis can be interpreted as a spatial averaging technique using the covariance matrix of the flow variable of interest. Some flow details, such as the surface wave propagation of the liquid film, may be smoothed out due to averaging. This concern, however, can be addressed effectively using the aforementioned statistical and optimization algorithms to tune GP model parameters. The resultant emulator model thus mitigates the smoothing effects and captures the flow structures well.

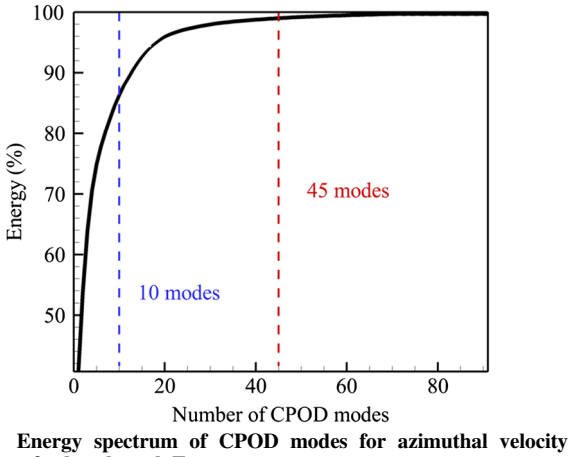

Fig. 11 Energy spectrum of CPOD modes for azimuthal velocity component for benchmark E.

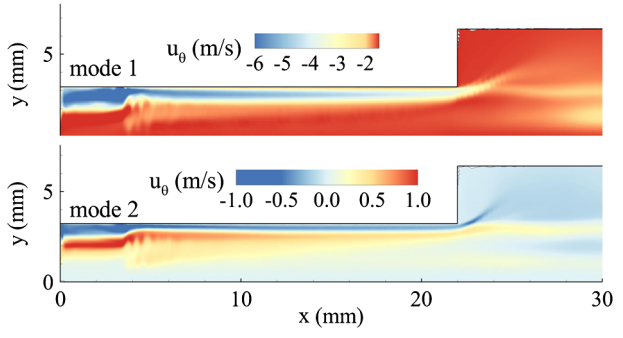

Fig. 12 First two CPOD modes of azimuthal velocity for benchmark E.

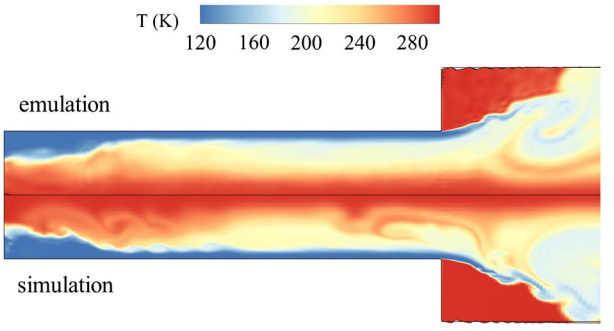

Fig. 13 Comparison of instantaneous temperature distributions for benchmark E.

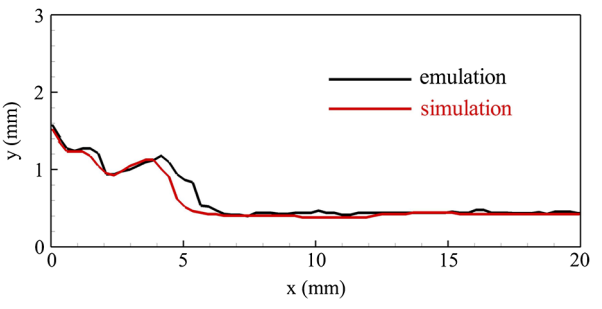

Fig. 14 Comparison of mean liquid-film thickness along axial distance.

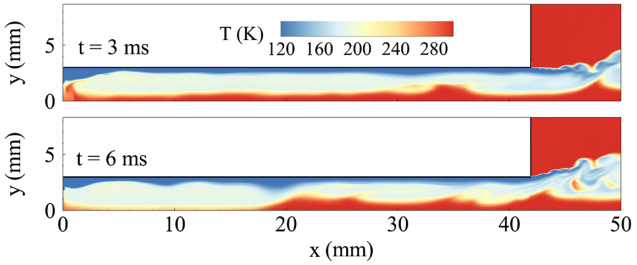

Fig. 15 Time evolution of the temperature for baseline case for benchmark F.

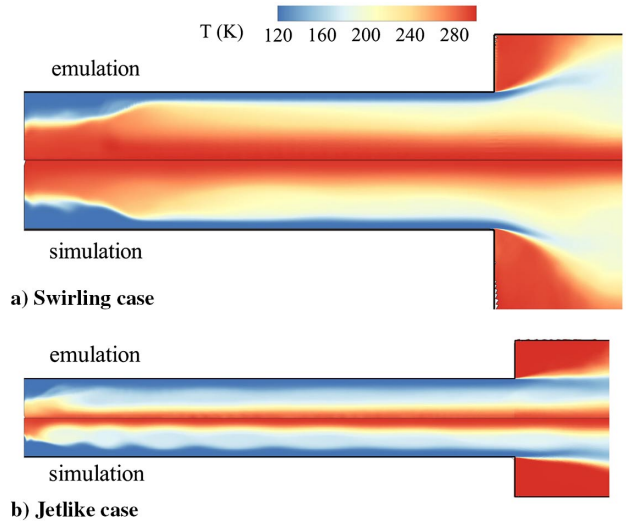

Fig. 16 Mean temperature distributions for benchmark cases: a) swirl-like case and b) jetlike case.

Table 4 RMSRE of temperature distribution

Benchmark	Overall, %	Upstream, %	Downstream, %
E (swirl)	5.18	6.62	3.10
F (jetlike)	8.65	8.30	9.03

1. Response Performance Metrics

As a preliminary comparison, a kriging surrogate model was applied to the extracted liquid-film thickness and spreading angle at the injector exit. The training process was implemented for the 30-case dataset. The following discussion is based on benchmark E: a swirl case. The liquid-film thickness is estimated, based on hydrodynamics theories, to be 0.618 mm; and the spreading angle 91.8 deg. The single-point emulator predicts a liquid-film thickness of 0.520 mm and a spreading angle of 99.0 deg. The data are compared with the simulation results of 0.430 mm and 103 deg, respectively. Figure 14 shows the variation of the film thickness along the injector wall.

At the injector exit, the time-averaged film thickness and spreading angle predicted by the kriging surrogate model are 0.420 mm and 107 deg, corresponding to percentage errors of 2.38 and 3.88%, respectively. The model matches the simulation in terms of key features such as the liquid-film distribution and spreading angle, which are performance measures needed for assessing injector design.

For benchmark F, the baseline case (design B) develops from jetlike to swirling behavior, as shown in Fig. 15. The design parameters are near a critical hyperplane separating different flow features.

Figure 16 shows the time-mean temperature distributions for the two benchmark cases. The accumulation of liquid propellant at the injector headend is observed in both results. The liquid-film thickness and spreading angle match well. For benchmark case F, which produces a jetlike flow, a standing wave appears in the upstream portion of the injector. The emulation result captures the wavy structure only to some extent. In the downstream region, the liquid-film thickness and spreading angle are better predicted. In the region where the film breaks apart, less propellant appears in the simulation result.

2. Root-Mean-Square Relative Error

The root-mean-square-relative error (RMSRE) is defined by

$$\mathrm{RMSRE}(t;S)=\frac{\left[\int_S\{f(\boldsymbol{x},t;\boldsymbol{c}_{\mathrm{new}})-\hat{f}(\boldsymbol{x},t;\boldsymbol{c}_{\mathrm{new}})\}^2\,\mathrm{d}\boldsymbol{x}/|S|\right]^{1/2}}{\max(f(\boldsymbol{x},t;\boldsymbol{c}_{\mathrm{new}}))-\min(f(\boldsymbol{x},t;\boldsymbol{c}_{\mathrm{new}}))}\times 100\% \tag{17}$$

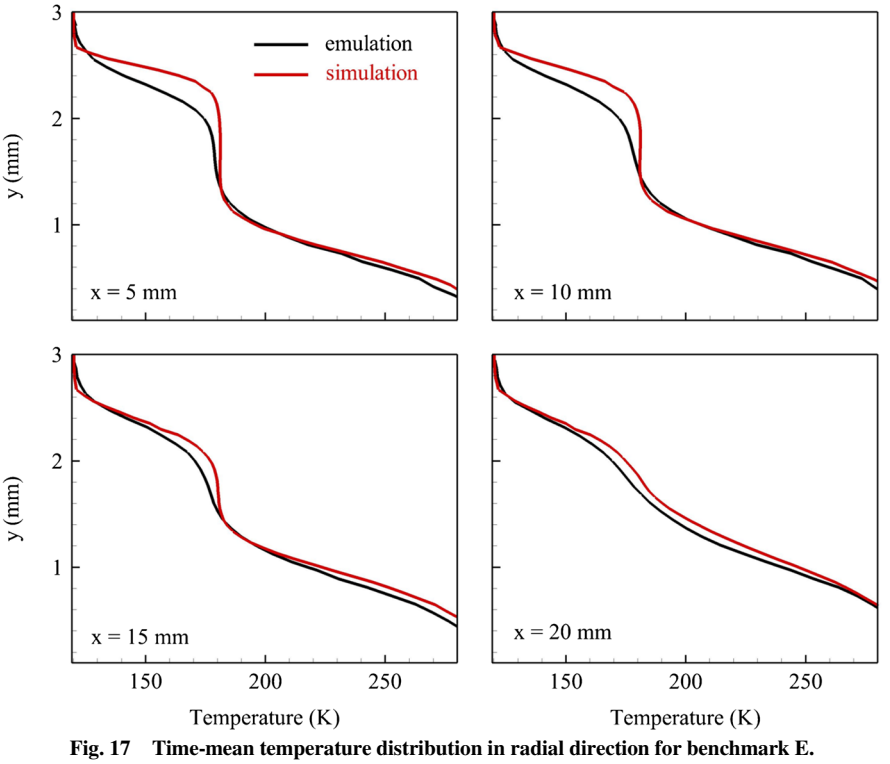

Fig. 17 Time-mean temperature distribution in radial direction for benchmark E.

where S is the desired region, $|S|$ is the number of grid points under S, $f(\boldsymbol{x}, t; \boldsymbol{c}_{\text{new}})$ is the simulated flowfield at geometry $\boldsymbol{c}_{\text{new}}$, $\hat{f}(\boldsymbol{x}, t; \boldsymbol{c}_{\text{new}})$ is the emulated flowfield, and $\max(f(\boldsymbol{x}, t; \boldsymbol{c}_{\text{new}}))$ and $\min(f(\boldsymbol{x}, t; \boldsymbol{c}_{\text{new}}))$ are the maximum and minimum values of $f(\boldsymbol{x}, t; \boldsymbol{c}_{\text{new}})$ over $\boldsymbol{x}$, respectively.

Table 4 lists the RMSRE for the two benchmark cases. This quantitatively compares the simulation and emulation shown in Fig. 16, illustrating minor discrepancies near the injector wall. For the jetlike case, the error is reduced if only upstream results (that is, upstream of the injector exit) are considered.

Table 5 RMSRE temperature distribution results (without dataset classification)

Benchmark case	Overall, %	Upstream, %	Downstream, %
E (swirl)	5.93	6.70	5.09
F (jet)	13.2	7.43	17.7

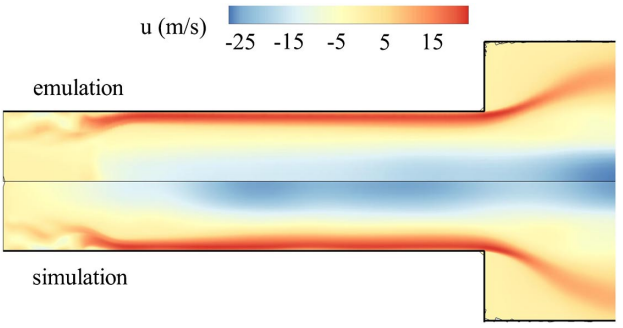

Fig. 18 Mean axial velocity distribution for benchmark E.

Table 6 RMSRE velocity distribution results

Benchmark	Overall, %	Upstream, %	Downstream, %
E (swirl)	4.12	4.58	3.64
F (jet)	3.97	4.71	2.85

Figure 17 shows the time-mean temperature distribution in the radial direction at various axial locations for benchmark E. The high gradient region represents the transition between the liquid film and the gas core. There is a slight deviation in this transition region, where the simulated temperature gradient is sharper than that of the emulation. Similar results, not shown, are also obtained for benchmark F.

To illustrate the importance of incorporating the decision tree within the framework, a comparison is made with the prediction from an emulator without dataset classification [17]. Table 5 lists the RMSRE for the two benchmark cases using the emulator trained with the entire dataset. Benchmark E results are slightly worse, and benchmark F's prediction is significantly off.

Next, the axial velocity is used as training data, demonstrating the capability of modeling other flowfield variables. Figure 18 shows the time-mean distribution comparison between the simulation and emulation for benchmark E. The key flow features, such as the gaseous core and swirling film, are predicted well. The RMSREs listed in Table 6 numerically outperform temperature results. This improvement can be explained by the broader range, which leads to higher errors.

Figure 19 shows the time-mean axial velocity distribution in the radial direction for various axial locations in the injector for benchmark E. The transition region is matched, with a deviation near the injector centerline where the gradient is smoother in the emulator prediction. Similar results, not shown, were seen for benchmark F.

3. *Injector Dynamics*

Injector dynamics cover a wide range of time scales, which can be quantified using a power spectral density analysis. Figure 20 shows the position of the pressure probes in the fluid transition region.

Injector dynamics involve downstream pressure fluctuations causing pressure drop oscillations across the liquid film. These changes, in turn, trigger mass flow rate variations across the tangential inlets [9,10], over a wide range of time scales. The probes are located near the film surface to measure dynamics like this mechanism. A spectral analysis can quantify these oscillations and capture the periodicity of flow features. Mathematically, the PSD can be interpreted as the Fourier transform of the autocorrelation function for a signal. The pressure PSDs are calculated for both the simulation

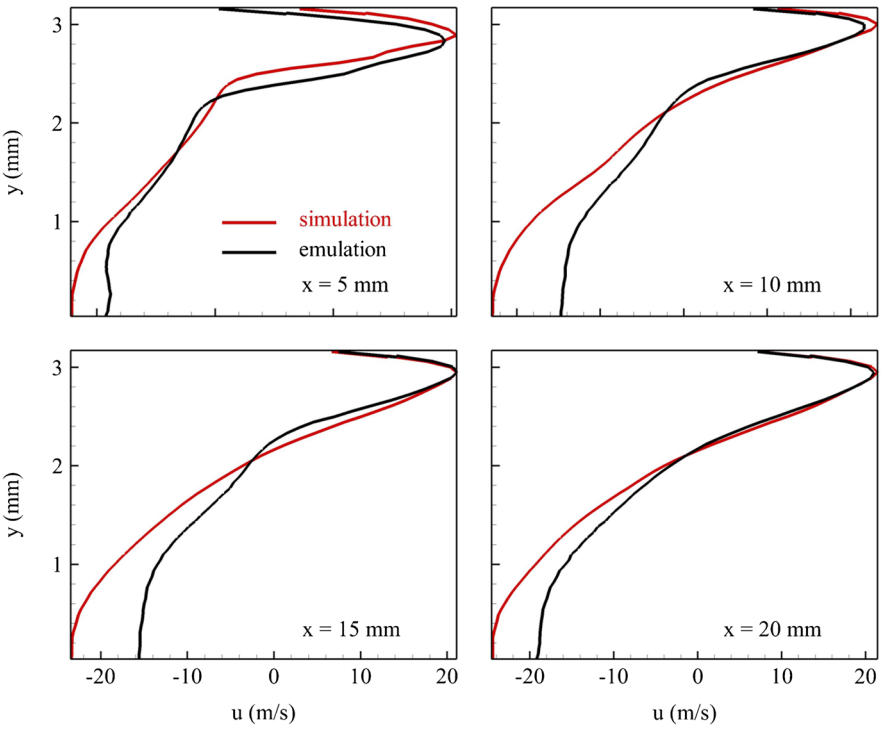

Fig. 19 Axial variation of velocity distribution in the radial direction for benchmark E.

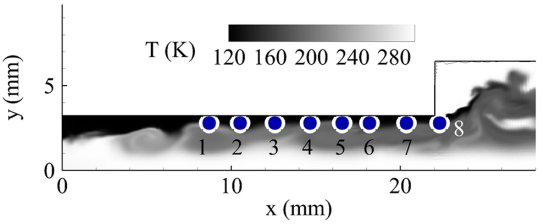

Fig. 20 Probe positions along liquid-film surface.

and emulation results. Figure 21 shows the PSDs of probes 1, 3, 5 and 7; the frequency content is observed to be well quantified.

The high-frequency oscillations that are typically present in swirl injectors that have a vortex chamber are not prominent. Most of the signal comprises low-frequency content, representing surface wave propagations along the film. In addition, acoustic waves propagate, couple, and interact with hydrodynamic waves, appearing as several different frequencies. The simulated and emulated probes show similar dynamics, such that the peak frequencies of the simulation and emulation results match. However, the emulator amplifies the dominant frequencies because the kriging model may be overfitting slightly due to insufficient data. Despite this signal strengthening phenomenon, the analysis displays the modeling capability for flow dynamics, properly capturing the simulated periodic oscillations. Downstream of the injector exit, the dynamics become more broadband and no dominant oscillations appear, because there exist strong interactions between the shear layer and recirculation zone generated from vortex breakdown.

4. *Uncertainty Quantification*

In addition to the aforementioned validation methods, the emulator model also allows for quantification of predictive uncertainty that can be used to define confidence intervals for model fit. Moreover, these uncertainties can be linked to dynamic flow physics. As an example, the spatial uncertainty quantification is shown in Fig. 22, displaying the one-sided width of the 80% confidence interval for the pressure and temperature (a derivation of this interval is found in [17]). The uncertain areas, in the time-mean temperature distribution,

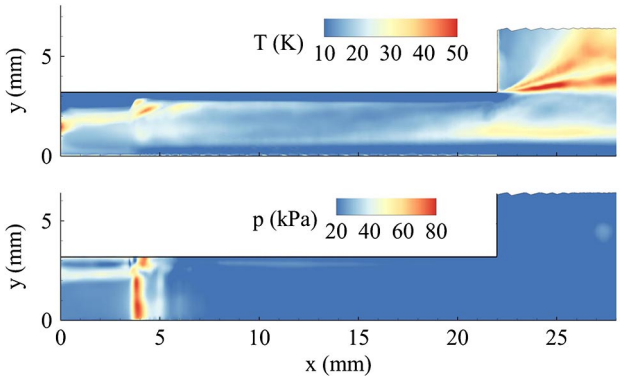

Fig. 22 One-sided width of the 80% confidence interval for benchmark E: temperature and pressure predictions.

correspond to the most dynamic sections of the liquid transition region. The downstream uncertainty is caused by the recirculation induced through vortex breakdown.

5. *Computation Time*

Figure 23 presents the simulation and emulation timeline. The computation times are calculated based on performance for a parallelized system of 200 Intel Xeon E5-2603 1.80 GHz processors. A total of 900,000 CPU hours is required for the 30 GB dataset. CPOD extraction and parameter estimation for the model takes about 45 min.

The parallelized predictions from the developed model only need around 30 CPU hours, significantly reducing the turnaround time as compared with LES requiring 30,000 CPU hours. This improved computational efficiency is crucial because it enables quick design iterations. The existing spatiotemporal emulators mentioned in the Introduction (Sec. I) require much more computation time to fit the underlying statistical model because the training dataset of each simulation is too large to directly manipulate [22]. By carefully using physical knowledge to make informed model assumptions, state-of-the-art machine-learning techniques have been leveraged to develop a methodology offering an efficient strategy to survey the design space.

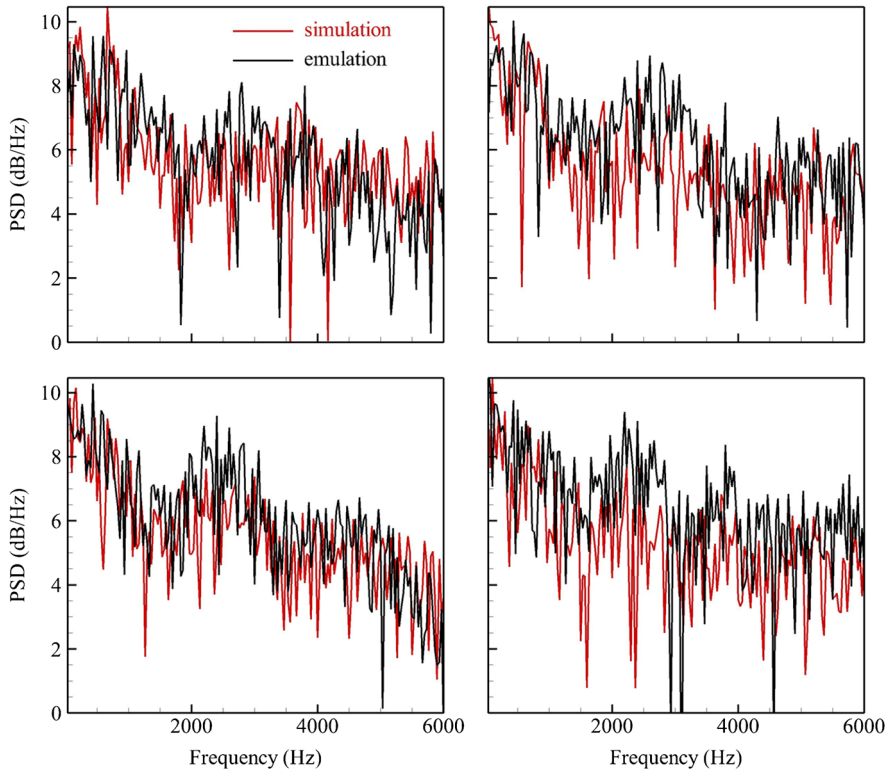

Fig. 21 PSD results of pressure fluctuations for probes 1, 3, 5, and 7.

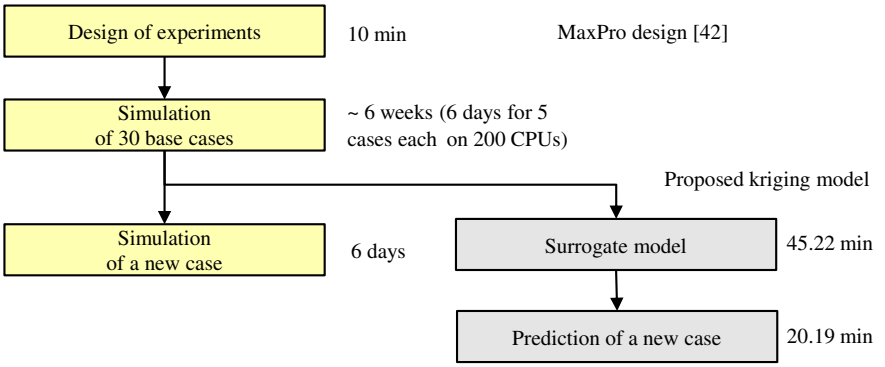

Fig. 23 Simulation and emulation timeline.

V. Conclusions

The present work develops an integrated framework that incorporates state-of-the-art statistical methods, machine-learning algorithms, and a physics-driven data reduction method to obtain a surrogate model for a broad-range design space. Taking a swirl injector as an example, the common POD (CPOD)-based emulation framework is used to extract the flow physics, reduce the data, and build an efficient physics-driven emulator.

The key contributions are twofold: the use of statistical and machine-learning techniques to quantify the impact of design parameters on important flow physics, and the incorporation of such methods with physics-guided model assumptions to build an efficient surrogate model for flowfield prediction. A vital model assumption is that the CPOD, which is the common basis, accurately retains the rich set of physics over varying geometries. This model successfully captures the simulation results and fares better than analytical estimations for performance measures. The emulated flowfield is validated against the LES-simulated flowfield to demonstrate how the flow features and injector characteristics are preserved by the model. Moreover, this methodology significantly reduces the computational time required for assessing a design based on spatiotemporal information. While the focus of the present study is on a data-driven analysis and emulation of flow physics, the principle of applying machine-learning techniques with physics-guided assumptions can be applied to any type of engineering application.

For future work, additional investigation should be carried out in dynamic regions of the flowfield, where the surrogate model has higher predictive uncertainties. One potential cause is the extreme range of the design points; this can be addressed by setting a smaller range. The uncertainty quantification and propagation of underlying flow couplings are also important research directions, which can perhaps be tackled using techniques such as support points.

Acknowledgments

This work was sponsored partly by the William R. T. Oakes Endowment and partly by the Coca-Cola Endowment of the Georgia Institute of Technology.

References

[1] Lieuwen, T. C., and Yang, V., *Combustion Instabilities in Gas Turbine Engines: Operational Experience, Fundamental Mechanisms and Modeling*, Progress in Astronautics and Aeronautics, Vol. 210, AIAA, Reston, VA, 2005, pp. 481–529. doi:10.2514/4.866807

[2] Lieuwen, T. C., and Yang, V., *Gas Turbine Emissions*, Cambridge Univ. Press, Cambridge, England, U.K., 2013, pp. 3–23. doi:10.2514/1.J053061

[3] Yang, V., Brill, T. B., and Ren, W.-Z., *Solid Propellant Chemistry Combustion and Motor Interior Ballistics*, Progress in Astronautics and Aeronautics, Vol. 185, AIAA, Reston, VA, 2000, pp. 381–412. doi:10.2514/4.866562

[4] Yang, V., Habiballah, M., Hulka, J., and Popp, M., *Liquid Rocket Thrust Chambers: Aspects of Modeling, Analysis, and Design*, Progress in Astronautics and Aeronautics, Vol. 200, AIAA, Reston, VA, 2004, pp. 251–322. doi:10.2514/4.866760

[5] O'Connor, J., Acharya, V., and Lieuwen, T. C., "Transverse Combustion Instabilities: Acoustic, Fluid Mechanic, and Flame Processes," *Progress in Energy and Combustion Science*, Vol. 49, Aug. 2015, pp. 1–39. doi:10.1016/j.pecs.2015.01.001

[6] Gröning, S., Hardi, J. S., Suslov, D., and Oschwald, M., "Injector-Driven Combustion Instabilities in a Hydrogen/Oxygen Rocket Combustor," *Journal of Propulsion and Power*, Vol. 32, No. 3, 2016, pp. 560–573. doi:10.2514/1.B35768

[7] Poinsot, T., and Veynante, D., *Theoretical and Numerical Combustion*, RT Edwards, Inc., Philadelphia, PA, 2005, pp. 424–430.

[8] Huang, Y., and Yang, V., "Dynamics and Stability of Lean-Premixed Swirl-Stabilized Combustion," *Progress in Energy and Combustion Science*, Vol. 35, No. 4, 2009, pp. 293–364. doi:10.1016/j.pecs.2009.01.002

[9] Bazarov, V. G., and Yang, V., "Liquid-Propellant Rocket Engine Injector Dynamics," *Journal of Propulsion and Power*, Vol. 14, No. 5, 1998, pp. 797–806. doi:10.2514/2.5343

[10] Bazarov, V., Yang, V., and Puri, P., "Design and Dynamics of Jet and Swirl Injectors," *Liquid Rocket Thrust Chambers: Aspects of Modeling, Analysis, and Design*, Progress in Astronautics and Aeronautics, Vol. 200, AIAA, Reston, VA, 2004, pp. 19–103, Chap. 2. doi:10.2514/5.9781600866760.0019.0103

[11] Wang, X., and Yang, V., "Supercritical Mixing and Combustion of Liquid-Oxygen/Kerosene Bi-Swirl Injectors," *Journal of Propulsion and Power*, Vol. 33, No. 2, 2017, pp. 316–322. doi:10.2514/1.B36262

[12] Zong, N., and Yang, V., "Cryogenic Fluid Dynamics of Pressure Swirl Injectors at Supercritical Conditions," *Physics of Fluids*, Vol. 20, No. 5, 2008, Paper 056103. doi:10.1063/1.2905287

[13] Wang, X., Huo, H., Wang, Y., and Yang, V., "Comprehensive Study of Cryogenic Fluid Dynamics of Swirl Injectors at Supercritical Conditions," *AIAA Journal*, Vol. 55, No. 9, 2017, pp. 3109–3119. doi:10.2514/1.J055868

[14] Santner, T. J., Williams, B. J., and Notz, W. I., *The Design and Analysis of Computer Experiments*, Springer Science and Business Media, New York, 2013, pp. 121–162.

[15] Loeppky, J. L., Sacks, J., and Welch, W. J., "Choosing the Sample Size of a Computer Experiment: A Practical Guide," *Technometrics*, Vol. 51, No. 4, 2009, pp. 366–376. doi:10.1198/TECH.2009.08040

[16] Matheron, G., "Principles of Geostatistics," *Economic Geology*, Vol. 58, No. 8, 1963, pp. 1246–1266. doi:10.2113/gsecongeo.58.8.1246

[17] Mak, S., Sung, C.-L., Wang, X., Yeh, S.-T., Chang, Y.-H., Joseph, V. R., Yang, V., and Wu, C. F. J., "An Efficient Surrogate Model for Emulation and Physics Extraction of Large Eddy Simulations," *Journal of American Statistical Association*, Jan. 2017. doi:10.1080/01621459.2017.1409123

[18] Williams, B., Higdon, D., Gattiker, J., Moore, L., McKay, M., and Keller-McNulty, S., "Combining Experimental Data and Computer Simulations, with an Application to Flyer Plate Experiments," *Bayesian Analysis*, Vol. 1, No. 4, 2006, pp. 765–792. doi:10.1214/06-BA125

[19] Rougier, J., "Efficient Emulators for Multivariate Deterministic Functions," *Journal of Computational and Graphical Statistics*, Vol. 17, No. 4, 2008, pp. 827–843. doi:10.1198/106186008X384032

[20] Hung, Y., Joseph, V. R., and Melkote, S. N., "Analysis of Computer Experiments with Functional Response," *Technometrics*, Vol. 57, No. 1, 2015, pp. 35–44. doi:10.1080/00401706.2013.869263
[21] Qian, P. Z. G., Wu, H., and Wu, C. J., "Gaussian Process Models for Computer Experiments with Qualitative and Quantitative Factors," *Technometrics*, Vol. 50, No. 3, 2008, pp. 383–396. doi:10.1198/004017008000000262
[22] Zhou, Q., Qian, P. Z., and Zhou, S., "A Simple Approach to Emulation for Computer Models with Qualitative and Quantitative Factors," *Technometrics*, Vol. 53, No. 3, 2011, pp. 266–273. doi:10.1198/TECH.2011.10025
[23] Bayarri, M., Berger, J., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R., Paulo, R., Sacks, J., and Walsh, D., "Computer Model Validation with Functional Output," *Annals of Statistics*, Vol. 35, No. 5, 2007, pp. 1874–1906. doi:10.1214/009053607000000163
[24] Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H., "Gaussian Predictive Process Models for Large Spatial Data Sets," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 70, No. 4, 2008, pp. 825–848. doi:10.1111/rssb.2008.70.issue-4
[25] Nguyen, V. B., Buffoni, M., Willcox, K., and Khoo, B. C., "Model Reduction for Reacting Flow Applications," *International Journal of Computational Fluid Dynamics*, Vol. 28, Nos. 3–4, March 2014, pp. 91–105. doi:10.1080/10618562.2014.911848
[26] Ling, J., and Templeton, J., "Evaluation of Machine Learning Algorithms for Prediction of Regions of High Reynolds Averaged Navier Stokes Uncertainty," *Physics of Fluids*, Vol. 27, No. 8, Aug. 2015, Paper 085103. doi:10.1063/1.4927765
[27] Singh, A. P., and Duraisamy, K., "Using Field Inversion to Quantify Functional Errors in Turbulence Closures," *Physics of Fluids*, Vol. 28, No. 4, April 2016, Paper 045110. doi:10.1063/1.4947045
[28] Parish, E. J., and Duraisamy, K., "A Paradigm for Data-Driven Predictive Modeling Using Field Inversion and Machine Learning," *Journal of Computational Physics*, Vol. 305, Jan. 2016, pp. 758–774. doi:10.1016/j.jcp.2015.11.012
[29] Ling, J., Kurzawski, A., and Templeton, J., "Reynolds Averaged Turbulence Modelling Using Deep Neural Networks with Embedded Invariance," *Journal of Fluid Mechanics*, Vol. 807, Nov. 2016, pp. 155–166. doi:10.1017/jfm.2016.615
[30] Wang, J.-X., Wu, J.-L., and Xiao, H., "Physics-Informed Machine Learning Approach for Reconstructing Reynolds Stress Modeling Discrepancies Based on DNS Data," *Physical Review Fluids*, Vol. 2, No. 3, 2017, Paper 034603. doi:10.1103/PhysRevFluids.2.034603
[31] Qian, Z., Seepersad, C. C., Joseph, V. R., Allen, J. K., and Wu, C. F. J., "Building Surrogate Models Based on Detailed and Approximate Simulations," *Journal of Mechanical Design*, Vol. 128, No. 4, June 2005, pp. 668–677. doi:10.1115/1.2179459
[32] Willcox, K., and Peraire, J., "Balanced Model Reduction via the Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 40, No. 11, 2002, pp. 2323–2330. doi:10.2514/2.1570
[33] Lucia, D. J., and Beran, P. S., "Projection Methods for Reduced Order Models of Compressible Flows," *Journal of Computational Physics*, Vol. 188, No. 1, June 2003, pp. 252–280. doi:10.1016/S0021-9991(03)00166-9
[34] Rowley, C. W., Colonius, T., and Murray, R. M., "Model Reduction for Fluids, Using Balanced Proper Orthogonal Decomposition," *International Journal of Bifurcation and Chaos*, Vol. 15, No. 3, 2005, pp. 997–1013. doi:10.1142/S0218127405012429
[35] Walton, S., Hassan, O., and Morgan, K., "Reduced Order Modelling for Unsteady Fluid Flow Using Proper Orthogonal Decomposition and Radial Basis Functions," *Applied Mathematical Modelling*, Vol. 37, Nos. 20–21, Nov. 2013, pp. 8930–8945. doi:10.1016/j.apm.2013.04.025
[36] Benner, P., Gugercin, S., and Willcox, K., "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems," *SIAM Review*, Vol. 57, No. 4, Jan. 2015, pp. 483–531. doi:10.1137/130932715
[37] Lumley, J. L., "The Structure of Inhomogeneous Turbulent Flows," *Atmospheric Turbulence and Radio Wave Propagation*, Vol. 790, 1967, pp. 166–178.
[38] Loève, M., *Probability Theory: Foundations, Random Sequences*, van Nostrand, Princeton, NJ, 1955.
[39] Higdon, D., Gattiker, J., Williams, B., and Rightley, M., "Computer Model Calibration Using High-Dimensional Output," *Journal of the American Statistical Association*, Vol. 103, No. 482, 2008, pp. 570–583. doi:10.1198/016214507000000888.
[40] Wang, X., Wang, Y., and Yang, V., "Geometric Effects on Liquid Oxygen/Kerosene Bi-Swirl Injector Flow Dynamics at Supercritical Conditions," *AIAA Journal*, Vol. 55, No. 10, Oct. 2017, pp. 3467–3475. doi:10.2514/1.J055952
[41] Dranovsky, M. L., *Combustion Instabilities in Liquid Rocket Engines: Testing and Development Practices in Russia*, Progress in Astronautics and Aeronautics, AIAA, Reston, VA, 2007, pp. 253–282, http://arc.aiaa.org/doi/book/10.2514/4.866906 [retrieved 2018].
[42] Joseph, V. R., Gul, E., and Ba, S., "Maximum Projection Designs for Computer Experiments," *Biometrika*, Vol. 102, No. 2, 2015, pp. 371–380. doi:10.1093/biomet/asv002
[43] Yang, V., "Modeling of Supercritical Vaporization, Mixing, and Combustion Processes in Liquid-Fueled Propulsion Systems," *Proceedings of the Combustion Institute*, Vol. 28, No. 1, 2000, pp. 925–942. doi:10.1016/S0082-0784(00)80299-4
[44] Meng, H., and Yang, V., "A Unified Treatment of General Fluid Thermodynamics and Its Application to a Preconditioning Scheme," *Journal of Computational Physics*, Vol. 189, No. 1, 2003, pp. 277–304. doi:10.1016/S0021-9991(03)00211-0
[45] Zong, N., and Yang, V., "An Efficient Preconditioning Scheme for Real-Fluid Mixtures Using Primitive Pressure–Temperature Variables," *International Journal of Computational Fluid Dynamics*, Vol. 21, Nos. 5–6, 2007, pp. 217–230. doi:10.1080/10618560701584373
[46] Sobol, I. M., "Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates," *Mathematics and Computers in Simulation*, Vol. 55, Nos. 1–3, 2001, pp. 271–280. doi:10.1016/S0378-4754(00)00270-6
[47] Wu, C. J., and Hamada, M. S., *Experiments: Planning, Analysis, and Optimization*, Wiley, New York, 2011, pp. 1–44.
[48] Sobol, I. M., "Distribution of Points in a Cube and Integration Nets," *Uspekhi Matematicheskikh Nauk*, Vol. 21, No. 5, 1966, pp. 271–272.
[49] Geurts, P., Irrthum, A., and Wehenkel, L., "Supervised Learning with Decision Tree-Based Methods in Computational and Systems Biology," *Molecular Biosystems*, Vol. 5, No. 12, 2009, pp. 1593–1605. doi:10.1039/b907946g
[50] Blockeel, H., Schietgat, L., Struyf, J., Džeroski, S., and Clare, A., "Decision Trees for Hierarchical Multilabel Classification: A Case Study in Functional Genomics," *European Conference on Principles of Data Mining and Knowledge Discovery*, Springer, Berlin, 2006, pp. 18–29. doi:10.1007/11871637_7
[51] Friedman, J., Hastie, T., and Tibshirani, R., *The Elements of Statistical Learning*, Springer Series in Statistics, Springer, Berlin, 2001, pp. 1–42.
[52] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A., *Classification and Regression Trees*, CRC Press, London, 1984, pp. 246–280.
[53] Xiao, M., Breitkopf, P., Coelho, R. F., Knopf-Lenoir, C., Sidorkiewicz, M., and Villon, P., "Model Reduction by CPOD and Kriging," *Structural and Multidisciplinary Optimization*, Vol. 41, No. 4, 2010, pp. 555–574. doi:10.1007/s00158-009-0434-9
[54] Joseph, V. R., and Kang, L., "Regression-Based Inverse Distance Weighting with Applications to Computer Experiments," *Technometrics*, Vol. 53, No. 3, 2011, pp. 254–265. doi:10.1198/TECH.2011.09154
[55] Dexter, C. E., Fisher, M. F., Hulka, J., Denisov, K. P., Shibanov, A. A., and Agarkov, A. F., "Scaling Techniques for Design, Development, and Test," *Liquid Rocket Thrust Chambers: Aspects of Modeling, Analysis, and Design*, Progress in Astronautics and Aeronautics, Vol. 200, AIAA, Reston, VA, 2004, pp. 553–600, Chap. 16. doi:10.2514/5.9781600866760.0553.0600
[56] Rasmussen, C. E., and Williams, C. K. I., *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006, pp. 1–6.
[57] Casella, G., and Berger, R. L., *Statistical Inference*, Duxbury Press, Pacific Grove, CA, 2002, pp. 373–416.
[58] Liu, D. C., and Nocedal, J., "On the Limited Memory BFGS Method for Large Scale Optimization," *Mathematical Programming*, Vol. 45, Nos. 1–3, 1989, pp. 503–528. doi:10.1007/BF01589116

K. E. Willcox
Associate Editor