

Pritthi Chattopadhyay

Department of Mechanical and
Nuclear Engineering,
Pennsylvania State University,
University Park, PA 16802
e-mail: prithichatterjee@gmail.com

Sudepta Mondal

Department of Mechanical and
Nuclear Engineering,
Pennsylvania State University,
University Park, PA 16802
e-mail: sudepta979@gmail.com

Chandrachur Bhattacharya

Department of Mechanical and
Nuclear Engineering,
Pennsylvania State University,
University Park, PA 16802
e-mail: chandrachur.bhattacharya@gmail.com

Achintya Mukhopadhyay

Department of Mechanical Engineering,
Jadavpur University,
Kolkata 700 032, India
e-mail: achintya.mukho@gmail.com

Asok Ray¹

Fellow ASME
Department of Mechanical and
Nuclear Engineering,
Pennsylvania State University,
University Park, PA 16802
e-mail: axr2@psu.edu

Dynamic Data-Driven Design of Lean Premixed Combustors for Thermoacoustically Stable Operations

Prediction of thermoacoustic instabilities is a critical issue for both design and operation of combustion systems. Sustained high-amplitude pressure and temperature oscillations may cause stresses in structural components of the combustor, leading to thermomechanical damage. Therefore, the design of combustion systems must take into account the dynamic characteristics of thermoacoustic instabilities in the combustor. From this perspective, there needs to be a procedure, in the design process, to recognize the operating conditions (or parameters) that could lead to such thermoacoustic instabilities. However, often the available experimental data are limited and may not provide a complete map of the stability region(s) over the entire range of operations. To address this issue, a Bayesian nonparametric method has been adopted in this paper. By making use of limited experimental data, the proposed design method determines a mapping from a set of operating conditions to that of stability regions in the combustion system. This map is designed to be capable of (i) predicting the system response of the combustor at operating conditions at which experimental data are unavailable and (ii) statistically quantifying the uncertainties in the estimated parameters. With the ensemble of information thus gained about the system response at different operating points, the key design parameters of the combustor system can be identified; such a design would be statistically significant for satisfying the system specifications. The proposed method has been validated with experimental data of pressure time-series from a laboratory-scale lean-premixed swirl-stabilized combustor apparatus. [DOI: 10.1115/1.4037307]

Keywords: dynamic data-driven application, symbolic dynamics, combustion instability, uncertainty quantification

1 Introduction

Thermoacoustic instabilities result from the coupling between unsteady heat release rate and acoustic pressure fluctuations inside the combustion chamber [1,2]. Design optimization of combustors is a challenging problem due to the difficulties in modeling the nonlinear dynamics involved in thermoacoustic instabilities. These difficulties limit the application of model-based design optimization to combustion systems that involve several input parameters (e.g., inlet velocity of air, air-fuel ratio, premixing level, and combustor geometry) [3–5]; these parameters potentially affect the combustion dynamics. Examples are existence of bifurcations in the dynamic behavior of combustors and extremely high sensitivity of the combustor behavior to even small changes in some of the design parameters. On the other hand, it is economically infeasible and unrealistic to have sensors for online measurements of all dynamic variables involved in combustion and to conduct experiments at a sufficiently dense set of operating points. Traditional design of combustion systems has focused on issues like efficiency, power generation, and emission [3,4]. However, with the implementation of low emission technologies like lean premixed combustion, combustors have to operate in regimes where they are prone to thermoacoustic instabilities. The instability problems have been aggravated by use of newer grades of fuels like biofuels and hydrogen-based fuels like synthetic gas, because

these fuels have energy content and heat release pattern, which are widely different from those of conventional hydrocarbon fuels. Consequently, the behavior of the system becomes significantly altered in terms of phenomena like occurrence of instabilities, blowout, and flashback. Thermoacoustic instability is also encountered in rocket engines.

The current state-of-the-art of mitigating combustion instabilities at the design stage itself mostly involves introduction of passive devices, such as quarter wave tube arrangement [6] and perforated liners [7]. These devices improve the stability of the system by damping the oscillations in the combustion chamber. However, the use of passive devices can be only partially successful in mitigating different types of anomalous behavior as these passive devices are not designed on the basis of actual performance of the combustor. An alternative approach that has been gaining popularity is the implementation of active control devices/mechanisms, where appropriate actions are initiated like injection of secondary fuel [8–10]. These strategies are primarily designed to alter the phase difference between the pressure and heat-release-rate oscillations. Due to the complexity of the physics involved, control algorithms often use data-driven methods. However, implementation of these strategies in real time is challenging as the underlying dynamics is extremely fast (e.g., in the order of kilohertz for some of the circumferential modes of instability in gas turbine afterburners). Another method of avoiding combustion instabilities is identification of the stable operating zone at the design stage itself and limiting the parameter space of the design variables to the stable operating zone only. The most commonly followed approach for such predictive modeling is the network model [11], where the combustor is resolved into a network of

¹Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received February 20, 2017; final manuscript received June 19, 2017; published online October 2, 2017. Assoc. Editor: Yan Wang.

interconnected simpler elements and the response of each component to specific units is studied. To account for the complex flame dynamics, which is highly nonlinear, the linear transfer function approach is generalized to development of “flame describing function” [12]. The flame describing function, which describes an input–output relation for the flame element, is usually derived from experiments or computational fluid dynamics analysis. However, the network model and the flame describing function, in particular, have limitations in predicting different dynamic regimes of a combustor. With the availability of increasing computing power, high fidelity computer simulations have also been used for predicting the dynamic behavior [13]. Nevertheless, these simulations can be too time-consuming and computationally expensive to use as a design tool.

An alternative approach for combustor design has been proposed in this paper. Data generated from limited number of runs under diverse operating conditions have been used to generate the knowledge base about the system dynamics, which has then been used for designing the combustor system. Although such a data-driven approach has been used quite extensively for characterizing and controlling the combustor dynamics [14,15], dynamic data-driven approach for designing combustors from the viewpoint of thermoacoustic stability is rather uncommon.

Recently, the concept of dynamic data-driven application systems (DDDAS) [16] has found its way into design methodologies due to the advent of fast sensing and computational technologies as well as due to the inherent flexibility of DDDAS. Both quantitative and qualitative data have been used in various fields of design, where a given software or program analyzes the collected data to produce a decision that would aid in the design of the system under consideration. Especially the dynamic characterization of the system evolution allows for continual optimization of the design space as new data become available, thus enhancing the overall design quality. In particular, the notion of DDDAS has been used in the field of combustion monitoring and control. In a recent work, DDDAS has been used for the prediction of instability and flame lean-blow-out in combustors [14,15] using symbolic time series analysis (STSA) [17]. More recently, such an imaging-based analysis has also been reported using neural networks, where flame images have been used to detect the onset of combustion instability [18] also using STSA.

From the previously mentioned perspectives, this paper proposes a novel method of combustor design that is built upon the concept of DDDAS [16], instead of completely relying on model-based design tools. The proposed design method only needs limited amounts of process data in the form of time series and does not require any detailed knowledge of the underlying combustion dynamics. Given the information in the form of time series data at certain operating conditions, a Bayesian nonparametric statistical method has been adopted to predict the system behavior for operating conditions at which data may not be available. In addition, the algorithm also quantifies the confidence in the estimate of the system response. The design algorithm produces a mapping from a set of operating conditions to that of stability regions in the combustion system. Once this map is generated, combustor designers can use it for predicting the system response by statistically quantifying the uncertainties at operating conditions for which experimental data may not be available. This information facilitates the identification of the combustion system parameters, which will allow the design to be statistically significant in terms of satisfying the system specifications.

In the present work, experimental data from a laboratory-scale swirl-premixed combustor apparatus have been used to generate a stability map in the parameter space. To do this, time series data of pressure oscillations at different combinations of combustor length, equivalence ratio (ϕ), and inlet flow velocities have been considered. Using the limited data at hand, the devised algorithm creates a stability map, which can then predict the system response for an unknown set of parameters of combustor length, equivalence ratio, and inlet flow velocities. This provides the

designer with an estimate of the probability of the system at new parameters to become unstable, without the need for actual experimentation.

The objective here is prediction of combustion instabilities in order to prevent serious structural damage in the combustor. Hence, the user must design a combustor with a smaller probability of becoming unstable, which translates to the following two requirements:

- (1) *Extraction of a feature* that is highly sensitive to deviations in the underlying state of the system from the nominal state. Then, even a small deviation in the system behavior from the nominal state would be manifested as a large feature divergence.
- (2) *Quantification of the uncertainty* in the estimate of the system stability for unknown parameters (e.g., combustor length). This would help the user to make the parameter choice, which would have a significantly reduced probability of resulting in an unstable system.

From the previously mentioned perspectives, major contributions and innovations of the paper are summarized below:

- (1) Development of a dynamic data-driven method for combustor design, based on STSA [17,19], which satisfies the above two requirements.
- (2) Validation of the above method on experimental data from a swirl-stabilized combustor apparatus [20].

The paper is organized in six sections including the abstract and introduction. Section 2 describes the combustor apparatus that has been used to generate time series data. Section 3 develops the proposed dynamic data-driven algorithm for combustor design. Section 4 describes the procedure and results of validation of the proposed algorithm on the experimental data. Section 5 summarizes the work and delineates possible directions for future research.

2 Description of the Experimental Apparatus

A swirl-stabilized, lean-premixed, laboratory-scale combustor has been used for validation of the proposed algorithm with experimental data. Figure 1 depicts a schematic diagram of the variable-length combustor apparatus [20], consisting of an inlet section, an injector, a combustion chamber, and an exhaust section. There is an optically accessible quartz section followed by a variable-length steel section.

High-pressure air is delivered to the apparatus from a compressor system after passing through filters to remove any liquid or particles that might be present. The air supply pressure is set to 180 psig (1.338 MPa) using a dome pressure regulator. The air is preheated to a maximum temperature of 250 °C by an 88 kW electric heater. The fuel for this study is natural gas (approximately 95% methane), which is supplied to the system at a pressure of 200 psig (1.475 MPa). The flow rates of the air and natural gas are measured by thermal mass flow meters. The desired equivalence ratio and mean inlet velocity are set by adjusting the flow rates.

Synchronized pressure time series have been collected under different operating conditions, by varying the inlet velocity, equivalence ratio, and combustor length. Figure 2 displays typical profiles of pressure oscillations over a time window of 30 ms to unambiguously present the distinctive characteristics of stable and unstable signals around their respective mean values. Further details on the analysis of experimental data are provided in Sec. 4.

3 Description of the Proposed Design Algorithm

The first step in the proposed dynamic data-driven algorithm for combustor design involves feature extraction from the pressure time series collected under different operating conditions. The features encode the temporal dynamics of the combustion system. The second step of the algorithm involves inferring the relation

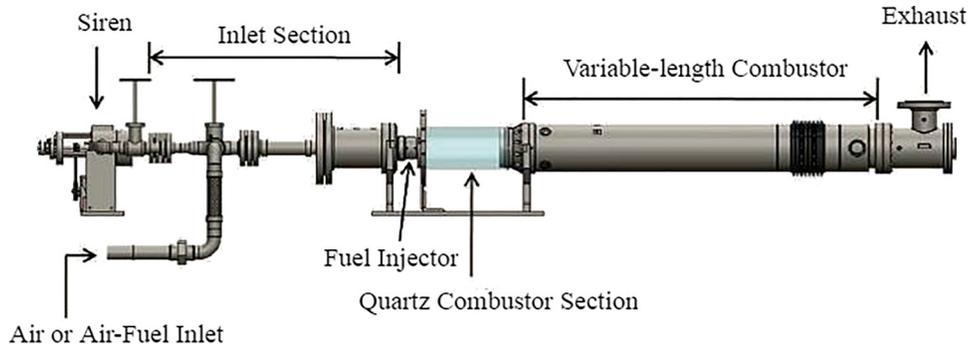


Fig. 1 Schematic diagram of the combustion apparatus

between the operating conditions and the corresponding system response, represented in the form of a function of features extracted from the corresponding pressure time series. Then, using the inferred relation, the distribution of the system response is predicted for each operating condition for which experiments have not been conducted. Therefore, the user now has at his/her disposal the ensemble of information, representing the system response at every point in the space of operating conditions. With this information, the user can then determine the parameters of the combustor to be constructed, which would exhibit the desired response. Section 3.1 describes the feature extraction procedure in detail using the concept of probabilistic finite state automata

(PFSA). Section 3.2 describes Bayesian nonparametric regression (i.e., Gaussian process (GP) regression) used for determining the relation between operating conditions and system response in detail. Section 3.3 develops the combustor design methodology.

3.1 Feature Extraction From Time Series. This subsection describes the procedure of feature extraction from time series data by using the concepts of symbolization and finite state automata [21]. First, each time series in an ensemble is discretized into a symbol string [22]. Then, a D -Markov machine [23,24], which belongs to a special class of finite-state automata (PFSA), is constructed from each of these symbol strings to extract the feature that encodes the dynamics of the time series under consideration.

3.1.1 Symbolization of Time Series. Symbolization requires partitioning (also known as quantization) of the time series [17,19,25], where the signal space is partitioned into a finite number of cells that are labeled as symbols, i.e., the number of cells is identically equal to the cardinality $|\Sigma|$ of the (symbol) alphabet Σ . If the value of time series at a given instant is located within a particular cell, then it is coded with the symbol associated with that cell. In this way, a symbol string is generated from the (finite-length) time series. Details are reported in Refs. [23] and [24]. The ensemble of time series data is partitioned by using a partitioning tool, called maximum entropy partitioning [26], that maximizes the entropy of the generated symbols; therefore, the information-rich cells of a data set are partitioned finer and those with sparse information are partitioned coarser (i.e., each cell contains approximately equal number of data points). The choice of alphabet size $|\Sigma|$ largely depends on the specific data set and the allowable loss of information [24].

3.1.2 Symbolic Time Series Analysis (STSA). This subsection briefly describes the underlying concept of STSA upon which the proposed dynamic-data-driven tool is constructed for identification of combustor parameters; STSA encodes the behavior of (possibly nonlinear) dynamical systems from the observed time series by symbolization and construction of state machines (i.e., PFSA) [23]. This is followed by computation of the state emission matrices that are representatives of the evolving statistical characteristics of the dynamical system.

The core assumption in the STSA analysis for construction of PFSA from symbol strings is that the symbolic process under both nominal and off-nominal conditions can be approximated as a Markov chain of order D , called the D -Markov machine, where D is a positive integer. While the details of the D -Markov machine construction are reported in Refs. [23] and [24], the pertinent definitions and their implications are succinctly presented later.

DEFINITION 4.1. (Symbol Block) A symbol block, also called a word, is a finite-length string of symbols s_i belonging to the alphabet Σ , where the length of a word $w \triangleq s_1 s_2 \dots s_\ell$ with $s_i \in \Sigma$ is $|w| = \ell$, and the length of the empty word ϵ is $|\epsilon| = 0$. The parameters of deterministic finite-state automata (DFA) are extended as:

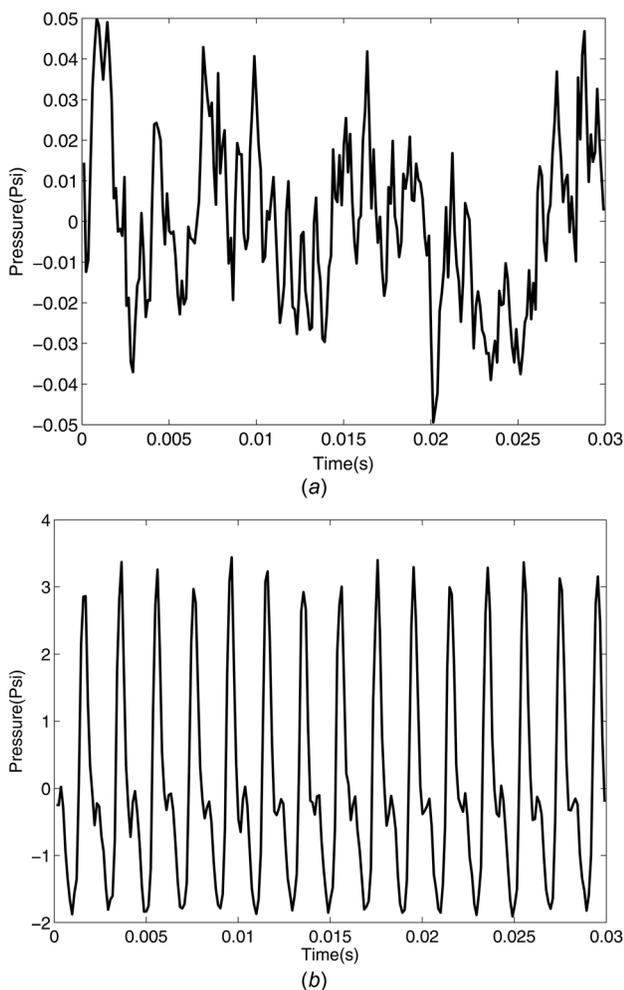


Fig. 2 Examples of pressure signals in the combustor: (a) stable signal oscillations and (b) unstable signal oscillations

- (1) The set of all words constructed from symbols in Σ , including the empty word ϵ , is denoted as Σ^* ,
- (2) The set of all words, whose suffix (respectively, prefix) is the word w , is denoted as Σ^*w (respectively, $w\Sigma^*$).
- (3) The set of all words of (finite) length ℓ , where $\ell > 0$, is denoted as Σ^ℓ .

DEFINITION 4.2. A deterministic finite-state automaton (DFSA) G is a triple $(\Sigma, \mathcal{Q}, \delta)$ [27], where:

- (1) Σ is a (nonempty) finite alphabet with cardinality $|\Sigma|$;
- (2) \mathcal{Q} is a (nonempty) finite set of states with cardinality $|\mathcal{Q}|$;
- (3) $\delta: \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$ is the state transition map.

DEFINITION 4.3. A probabilistic finite-state automaton (PFSA) is constructed on the algebraic structure of deterministic finite state automata (DFA) $G = (\Sigma, \mathcal{Q}, \delta)$ as a pair $K = (G, \Pi)$, i.e., the PFSA K is a four-tuple $K = (\Sigma, \mathcal{Q}, \delta, \Pi)$ [23,24], where:

- (1) Σ is a nonempty finite set, called the symbol alphabet, with cardinality $|\Sigma| < \infty$;
- (2) $\mathcal{Q} = \{q_1, q_2, \dots, q_{|\mathcal{Q}|}\}$ is the state set with cardinality $|\mathcal{Q}| \leq |\Sigma|^D$, i.e., the states are represented by equivalence classes of symbol blocks of maximum length D corresponding to a symbol sequence \mathbb{S} .
- (3) $\delta: \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$ is the state transition mapping, which generates the symbol sequences;
- (4) $\Pi: \mathcal{Q} \times \Sigma \rightarrow [0, 1]$ is the morph matrix of size $|\mathcal{Q}| \times |\Sigma|$; the ij th element $\Pi(i, j)$ of the matrix Π denotes the probability of finding the symbol σ_j at next time step while making a transition from the state q_i .

3.1.3 *D-Markov Modeling.* This subsection introduces a special class of PFSA, called *D-Markov machine*, which has a simple algebraic structure and is computationally efficient for construction and implementation [23,24].

DEFINITION 4.4. (*D-Markov*) A *D-Markov machine* [23] is a PFSA in which each state is represented by a (nonempty) finite string of D symbols where

- (1) D , a positive integer, is the depth of the Markov machine;
- (2) \mathcal{Q} is the finite set of states with cardinality $|\mathcal{Q}| \leq |\Sigma|^D$. The states are represented by equivalence classes of symbol strings of maximum length D , and each symbol in the string belongs to the alphabet Σ ;
- (3) $\delta: \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$ is the state transition map that satisfies the following condition if $|\mathcal{Q}| = |\Sigma|^D$: There exist $\alpha, \beta \in \Sigma$ and $s \in \Sigma^*$ such that $\delta(\alpha s, \beta) = s\beta$ and $\alpha s, s\beta \in \mathcal{Q}$.

Remark 4.1. It follows from Definition 4.4 that a *D-Markov chain* is treated as a statistically stationary stochastic process $S = \dots s_{-1}s_0s_1\dots$, where the probability of occurrence of a new symbol depends only on the last D symbols, i.e., $P[s_n | \dots s_{n-D} \dots s_{n-1}] = P[s_n | s_{n-D} \dots s_{n-1}]$.

The construction of a *D-Markov machine* is based on (i) state splitting that generates symbol blocks of different lengths according to their relative importance and (ii) state merging that assimilates histories from symbol blocks leading to the same symbolic behavior. Words of length D on a symbol string are treated as the states of the *D-Markov machine* before any state-merging is executed. Thus, on an alphabet Σ , the total number of possible states becomes less than or equal to $|\Sigma|^D$; and operations of state merging may significantly reduce the number of states [24]. However, no state splitting or state merging is required for $D = 1$, which is the simplest configuration of a *D-Markov machine*.

The PFSA states represent different combinations of blocks of symbols on the symbol string. In the graph of a PFSA, the directional edge (i.e., the emitted event) that interconnects a state (i.e., a node) to another state represents the transition probability between these states. Therefore, the “states” denote all possible symbol blocks (i.e., words) within a window of certain length, and the set of all states is denoted as $\mathcal{Q} = \{q_1, q_2, \dots, q_{|\mathcal{Q}|}\}$ and $|\mathcal{Q}|$ is

the number of (finitely many) states. The procedure for estimation of the emission probabilities is presented next.

Given a (finite length) symbol string S over a (finite) alphabet Σ , there exist several PFSA construction algorithms to discover the underlying irreducible PFSA model K of S . These algorithms start with identifying the structure of the PFSA $K \triangleq (\mathcal{Q}, \Sigma, \delta, \pi)$. To estimate the state emission matrix, a $|\mathcal{Q}| \times |\Sigma|$ count matrix C is constructed and each element c_{kj} of C is computed as: $c_{kj} \triangleq 1 + N_{kj}$, where N_{kj} denotes the number of times that a symbol σ_j is generated from the state q_k upon observing the symbol string S . The maximum a posteriori probability estimates of emission probabilities for PFSA K are computed by frequency counting as

$$\pi(\sigma_j | q_k) \triangleq \frac{c_{kj}}{\sum_{\ell} c_{k\ell}} = \frac{1 + N_{kj}}{|\Sigma| + \sum_{\ell} N_{kj}} \quad (1)$$

The rationale for initializing each element of the count matrix C to 1 is that if no event is generated at a state $q \in \mathcal{Q}$, then there should be no preference to any particular symbol and it is logical to have $\pi(\sigma | q) = (1/|\Sigma|) \forall \sigma \in \Sigma$, i.e., the uniform distribution of event generation at the state q . The above procedure guarantees that the PFSA, constructed from a (finite-length) symbol string, must have an (elementwise) strictly positive emissivity map Π .

Having computed the emission probabilities $\pi(\sigma_j | q_k)$ for $j \in \{1, 2, \dots, |\Sigma|\}$ and $k \in \{1, 2, \dots, |\mathcal{Q}|\}$, the estimated $(|\mathcal{Q}| \times |\Sigma|)$ emission probability matrix of the PFSA is obtained as

$$\Pi \triangleq \begin{bmatrix} \pi(\sigma_1 | q_1) & \dots & \pi(\sigma_{|\Sigma|} | q_1) \\ \vdots & \ddots & \vdots \\ \pi(\sigma_1 | q_{|\mathcal{Q}|}) & \dots & \pi(\sigma_{|\Sigma|} | q_{|\mathcal{Q}|}) \end{bmatrix} \quad (2)$$

Bahrampour et al. [25] presented a comparative evaluation of Cepstrum, principal component analysis (PCA) and STSA as feature extractors for target detection and classification. The underlying algorithms of feature extraction were executed in conjunction with three different pattern classification algorithms, namely, support vector machines (SVM), k -nearest neighbor, and sparse representation classifier. The results of comparison show consistently superior performance of STSA-based feature extraction over both Cepstrum-based and PCA-based feature extraction in terms of successful detection, false alarm, and wrong detection and classification decisions. Similar results on superior performance of STSA over PCA have been reported by Mallapragada et al. [28] for robotic applications. Rao et al. [29] reported a review of STSA and its performance evaluation relative to other classes of pattern recognition tools, such as Bayesian filters and artificial neural networks.

Section 3.2 describes the methodology for determining the mapping between operating conditions and system response as a function of the estimated emission probability matrix Π that is taken to be the extracted feature.

3.2 Gaussian Process Regression. This section describes the technique used for determining the relation between operating conditions and the system responses as a function of the features extracted in Sec. 3.1. This relation is then used for predicting the responses for operating conditions for which experiments have not been conducted.

Given a set of operating conditions and the corresponding continuous-valued system responses, there exist several regression algorithms in the machine learning literature (e.g., see Ref. [30]), which infer the underlying relation between the conditions and response, under different assumptions on the characteristics of the relation. The inferred relation can then be used to predict the response of an unseen system condition (i.e., whose response is unknown). In contrast, Gaussian process (GP) regression [31] is a nonparametric method that can model arbitrary relations between the condition and response without making any specific

assumptions on the relation. In addition, most regression algorithms only provide point estimates of the response, but they do not quantify the confidence in that estimate. Being a Bayesian method, GP also quantifies the uncertainties in the predictions resulting from possible measurement noise and errors in the parameter estimation procedure. Hence, GP regression has been adopted in this paper to infer the relation between the operating conditions and the corresponding system response.

3.2.1 Theory of Gaussian Process Regression. This subsection succinctly presents the underlying theory of Gaussian process (GP) regression. Further details are available in standard literature (e.g., see Ref. [31]).

A stochastic process is a collection of random variables, $\{\xi(t) : t \in T\}$, where T is an index set. A Gaussian process is a stochastic process such that any finite collection of random variables has a multivariate jointly Gaussian distribution. In particular, a collection of random variables $\{\xi(t) : t \in T\}$ is said to be drawn from a Gaussian process with mean function $m(\cdot)$ and covariance function $k(\cdot, \cdot)$ if, for any finite set of elements $t_1, \dots, t_l \in T$, the corresponding random variables $\xi(t_1), \dots, \xi(t_l)$ have multivariate jointly Gaussian distribution

$$\begin{bmatrix} \xi(t_1) \\ \vdots \\ \xi(t_l) \end{bmatrix} \sim N \left(\begin{bmatrix} m(t_1) \\ \vdots \\ m(t_l) \end{bmatrix}, \begin{bmatrix} k(t_1, t_1) & \dots & k(t_1, t_l) \\ \vdots & \ddots & \vdots \\ k(t_l, t_1) & \dots & k(t_l, t_l) \end{bmatrix} \right) \quad (3)$$

where $m(t) \triangleq E[\xi(t)]$ is the mean function and $k(t, t') \triangleq E[(\xi(t) - m(t))(\xi(t') - m(t'))]$ is the covariance function. Equation (3) is denoted in vector notation as: $\underline{\xi} \sim \text{GP}(\underline{m}, \underline{K})$. The GP regression algorithm is now described below.

Let $X = \{x_i\}$ and $Y = \{y_i\}$, $i = 1, \dots, n$, be the training data set, where X denotes a set of operating conditions and Y denotes the corresponding set of system responses. The objective here is to determine the relation between X and Y so that, given an unknown operating condition x , the corresponding system response y can be predicted. In the GP regression algorithm, it is assumed that $y = \xi(x) + \varepsilon$, where ε is independent and identically distributed (iid) (additive) noise, $N(0, \sigma^2)$. That is, the response y is assumed to be a stochastic process that is a function of the operating condition x with additive noise. Then, a zero-mean Gaussian process prior $\text{GP}(0, \underline{K})$ is assumed for the function ξ . By the property of GP in Eq. (3), the marginal distribution over any set of operating conditions belonging to X must be multivariate jointly Gaussian. Thus, given a set $\{x_{k1}, x_{k2}, \dots, x_{km}\}$, the corresponding $\{\xi(x_{k1}), \xi(x_{k2}), \dots, \xi(x_{km})\}$ has a multivariate jointly Gaussian distribution. Hence, by concatenating the training and testing sets of operating conditions as: $[X, X^{\text{test}}]$, the marginal distribution of their respective system responses $[\xi(X), \xi(X^{\text{test}})]$ is also multivariate jointly Gaussian. Thus, the training and testing responses are jointly distributed as

$$\begin{bmatrix} \xi(X) \\ \xi(X^{\text{test}}) \end{bmatrix} \sim N \left(0, \begin{bmatrix} K(X, X) & K(X, X^{\text{test}}) \\ K(X^{\text{test}}, X) & K(X^{\text{test}}, X^{\text{test}}) \end{bmatrix} \right) \quad (4)$$

where $\xi(X) = [\xi(x_1), \dots, \xi(x_n)]'$; $\xi(X^{\text{test}}) = [\xi(x_1^{\text{test}}), \dots, \xi(x_m^{\text{test}})]'$; $K(X, X^{\text{test}}) \in R^{n \times m}$ such that $(K(X, X^{\text{test}}))_{ij} = k(x_i, x_j^{\text{test}})$; $K(X, X) \in R^{n \times n}$ such that $(K(X, X))_{ij} = k(x_i, x_j)$; $K(X^{\text{test}}, X^{\text{test}}) \in R^{m \times m}$ such that $(K(X^{\text{test}}, X^{\text{test}}))_{ij} = k(x_i^{\text{test}}, x_j^{\text{test}})$; $K(X^{\text{test}}, X) \in R^{m \times n}$ such that $(K(X^{\text{test}}, X))_{ij} = k(x_i^{\text{test}}, x_j)$.

Since the system responses Y and Y^{test} are contaminated with additive iid Gaussian noise, i.e., $\begin{bmatrix} \varepsilon \\ \varepsilon^{\text{test}} \end{bmatrix} \sim N \left(0, \begin{bmatrix} \sigma^2 I & 0 \\ 0 & \sigma^2 I \end{bmatrix} \right)$ where $\varepsilon = [\varepsilon^1, \dots, \varepsilon^n]'$; and $\varepsilon^{\text{test}} = [\varepsilon_{\text{test}}^1, \dots, \varepsilon_{\text{test}}^m]'$, it follows that

$$\begin{bmatrix} Y \\ Y^{\text{test}} \end{bmatrix} = \begin{bmatrix} \xi \\ \xi^{\text{test}} \end{bmatrix} + \begin{bmatrix} \varepsilon \\ \varepsilon^{\text{test}} \end{bmatrix} \sim N \left(0, \begin{bmatrix} K(X, X) + \sigma^2 I & K(X, X^{\text{test}}) \\ K(X^{\text{test}}, X) & K(X^{\text{test}}, X^{\text{test}}) + \sigma^2 I \end{bmatrix} \right) \quad (5)$$

where $Y^{\text{test}} = [y_1^{\text{test}}, \dots, y_m^{\text{test}}]'$. The rules for conditioning on Gaussians, $Y^{\text{test}} | Y \sim N(\mu^{\text{test}}, \Sigma^{\text{test}})$ yield

$$\mu^{\text{test}} = K(X^{\text{test}}, X)(K(X, X) + \sigma^2 I)^{-1} Y \quad (6)$$

$$\Sigma^{\text{test}} = K(X^{\text{test}}, X^{\text{test}}) + \sigma^2 I - K(X^{\text{test}}, X)(K(X, X) + \sigma^2 I)^{-1} K(X, X^{\text{test}}) \quad (7)$$

Thus, the algorithm predicts the mean and variance of the system response for every test condition. Instead of a zero-mean prior (i.e., $E[\xi(x)] = 0$), a mean function $m(x)$ could also be incorporated into the prior. In such a case

$$\mu^{\text{test}} = m(X^{\text{test}}) + K(X^{\text{test}}, X)(K(X, X) + \sigma^2 I)^{-1}(Y - m(X)) \quad (8)$$

instead of Eq. (6), and Σ^{test} remains unchanged in Eq. (7).

3.3 Development of the Design Methodology. This subsection develops the combustor design methodology by combining the algorithms, described in Secs. 3.1 and 3.2. Figure 3 presents a flowchart of the proposed combustor design algorithm, while Sec. 4 explains the details along with a discussion on experimental results.

4 Experimental Validation of the Design Method

This section validates the algorithms, developed in Sec. 3, on the data collected from an experimental apparatus that is described in Sec. 2. Synchronized time series data of pressure oscillations have been collected under different operating conditions, by varying the following parameters:

- (1) Inlet velocity from 25 to 50 m/s in steps of 5 m/s.
- (2) Equivalence ratio (ϕ) as 0.525, 0.550, 0.600, and 0.650.
- (3) Combustor length from 25 to 59 in in steps of 1 in.

Time series data of pressure oscillations have been collected at a sampling rate of 8192 Hz for each of the above $6 \times 4 \times 35 = 840$ distinct operating conditions. The time span of data collection has been 8 s (i.e., 65,536 measurement data per channel) for each time series, which is within the safe limit of operation of the combustor apparatus and which is long enough to provide sufficient information for statistical analysis. The root-mean-square (rms) value, $Prms$, of pressure has been calculated for each time series. For this data set, an observed ground truth is that all systems with $Prms \geq 0.07$ psi are unstable, while all those with $Prms < 0.07$ psi are stable. Although $Prms$ appears to serve as a good indicator of stability in this set of experiments, a natural question arises whether $Prms$ could be universally adopted as a feature. Since $Prms$ is the standard deviation of the signal in the statistical sense, it is sensitive to noise. Therefore, $Prms$ may not be an ideal choice as a feature, because of its lack of robustness to measurement noise.

In the design of a real-life combustor, sufficiently long durations of data acquisition might not be feasible due to various challenges inflicted on the system performance and operability (e.g., high-amplitude pressure and temperature oscillations, and local air-fuel ratio variations leading to flame blowout) specifically during combustion instabilities. Figure 4 shows representative plots of $Prms$ calculated for different durations of time series in increments of 0.1 s for (a) a stable signal and (b) an unstable signal. It

is shown in Fig. 4(a) that the threshold $Prms$ of 0.07 psi is exceeded at time series window lengths ranging from about 2.5 s to 4 s. Although $Prms$ over the entire 8 s window is less than the threshold of 0.07 psi, an online stability criterion based on solely a $Prms$ threshold may generate false alarms for those durations, where the threshold is exceeded. Similar conclusions can be drawn from Fig. 4(b) where, for an actually unstable signal, $Prms$ drops below 0.07 psi within several time windows between 1 s to 3 s and close to 4 s. In both cases, a hard $Prms$ threshold is likely to yield misclassifications.

Remark 5.1. In general, stability decisions made on $Prms$ -based hard thresholding would not be robust relative to measurement noise. In contrast, symbolic dynamics-based decisions are expected to be significantly more robust as established by Biem Garben [19], which is the approach taken for feature extraction in this design algorithm (see Sec. 3).

Next, it is demonstrated how the features extracted from time series can be used with confidence at relatively shorter time windows provided that they are sufficiently long to capture the system dynamics.

Different durations of time series data (i.e., 2 s, 4 s, 6 s, and 8 s) have been considered for feature extraction. Approximately 80% of the feature vectors that are extracted from the corresponding pressure time series (see Sec. 3.1) along with their true stability labels have been used for training a binary classifier in the setting of SVM [30,32]. The trained classifier is tested to predict the stability labels with the remaining 20% of the data. The D -Markov machine which yields the best classification accuracy by using a radial basis function kernel-based SVM classifier has been chosen for each duration of data under consideration.

The classification error is defined in terms of the percent of misclassified samples among the test data. Table 1 lists the classification accuracy using D -Markov machines, with $D = 1$, for time series data of different window lengths at a fixed sampling rate. It is shown in Table 1 that the classification accuracy is very high and they are comparable for all different durations under consideration.

4.1 Results and Discussion. The objective in this analysis is to determine the degree of stability/instability of the combustion system at different combustor lengths for a given inlet velocity u and equivalence ratio ϕ , which would then be used for designing the combustor. Accordingly, the entire data set has been divided into subsets of constant inlet velocity and equivalence ratio. Hence, each subset consists of a set of combustor lengths and the corresponding pressure time series. The data in each subset are first randomly divided into training and the testing sets in the proportion of 80% and 20%, respectively. In the training phase, sufficiently long (e.g., 8 s duration) time-series data have been collected for each operating condition. Therefore, the most stable condition is determined based on $Prms$, which is calculated for each time series. The combustor length (in the training set) with the corresponding lowest $Prms$ is taken to be the one representing the nominal state of the combustion system under the given input of u and ϕ .

Each time series in a subset is first partitioned by maximum entropy partitioning [26]. The resulting symbol string is then compressed as a PFSA by assigning the states as symbol strings of finite length. The features chosen in this paper are the morph matrices of the PFSA (see Definition 4.3). This feature encodes the dynamics of the time series in the form of a morph matrix as described in Sec. 3.1, where the feature extraction procedure is represented by the “STSA” block in the flowchart of Fig. 3. The morph matrix feature, corresponding to the nominal combustor length, is taken to be the nominal feature for this subset. For every case (i.e., each combustor length) in the training set, the divergence of its corresponding feature from the nominal feature is calculated and is denoted as F_{div} , the Frobenius matrix norm of the difference between the two morph matrices, which is an indication of how far away the system is from the most stable operating condition; this is represented by the “computation of feature divergence block” in the flowchart of Fig. 3. This metric is introduced in order to have an estimate of the behavior of the predicted state of the system at different lengths of the combustor. A higher value of F_{div} indicates that the system is likely to be more unstable, while a lower value of F_{div} indicates that the system response is

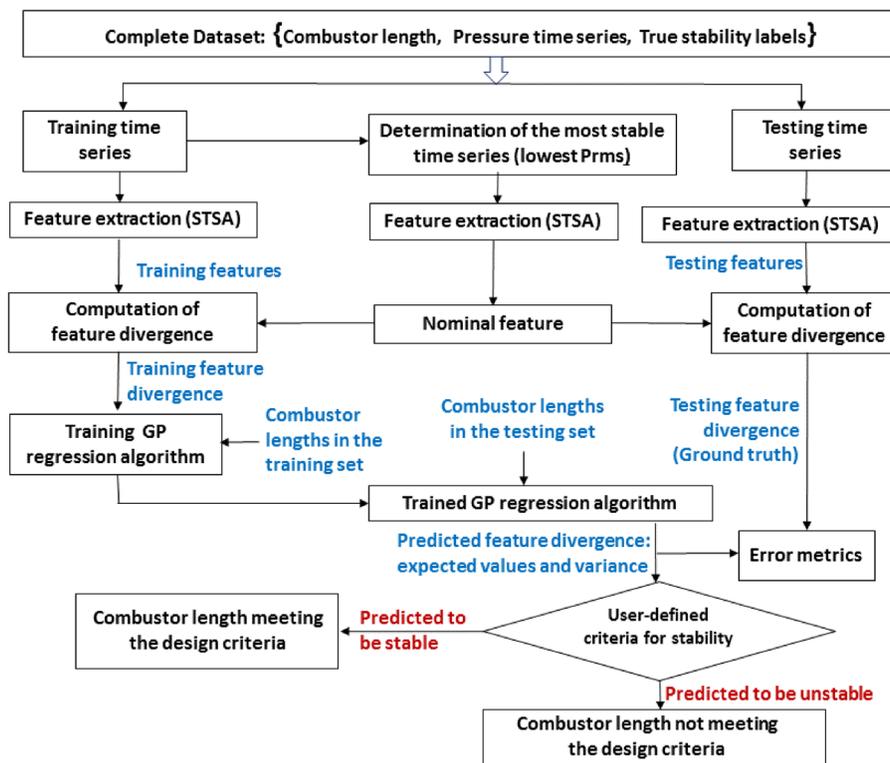
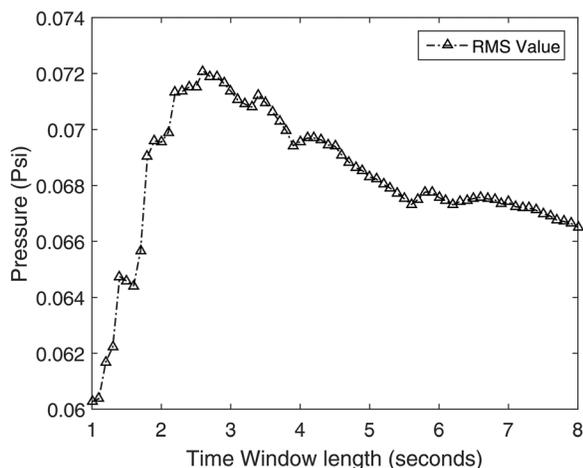


Fig. 3 Flowchart of the combustor design algorithm

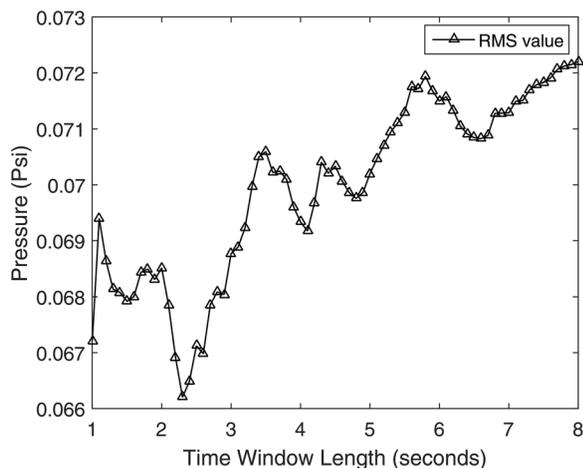
closer to the most stable operating condition. The inputs to the GP regression algorithm (see Sec. 3.2.1) thus consist of (combustor length, feature divergence) pairs (as represented in the inputs to the block “training GP regression algorithm” in the flowchart of Fig. 3). For the GP regression algorithm, a variety of mean and covariance functions can be used. Here, the following mean and covariance functions are considered.

- (1) Mean function: (i) constant $m(x) = c$, (ii) linear $m(x) = \sum_{i=1}^J a^i x^i$, and (iii) sum of the constant and linear terms yields: $m(x) = c + \sum_{i=1}^J a^i x^i$, where J is the dimension of the input space.
- (2) Covariance function: (i) linear $k(x^p, x^q) = x^p * (x^q)'$, where $*$ is matrix multiplication operation, and (ii) squared exponential automatic relevance determination $k(x^p, x^q) = s^2 * \exp\left(- (x^p - x^q)' * P^{-1} * (x^p - x^q) / 2\right)$, where the P matrix is diagonal with automatic relevance determination parameters $\ell_1^2, \dots, \ell_j^2$, and s^2 is the signal variance [31].

The log likelihood of the training data for all combinations of mean and covariance functions has been compared. It is observed that the combination of constant mean function and squared exponential automatic relevance determination covariance function resulted in highest likelihood; hence, these mean and covariance functions have been used for all subsequent analysis in the paper.



(a)



(b)

Fig. 4 Effects of time series length on P_{rms} profiles: (a) profile of P_{rms} for a typical stable pressure signal and (b) profile of P_{rms} for a typical unstable pressure signal

In addition, GP regression being a Bayesian algorithm, it is not necessary to know the optimal values of the hyperparameters (i.e., c , $\{\ell_j^2\}$ and s^2) in the mean and covariance function a priori. The algorithm identifies the optimal values of these hyperparameters by maximizing the log likelihood of training data.

For each subset of constant inlet velocity and equivalence ratio, GP regression algorithm is used to determine the mapping from combustor length to the system response (i.e., divergence F_{div} from the nominal condition), represented by the block trained GP regression algorithm in the flowchart of Fig. 3. Using this mapping, the algorithm then predicts the distribution of F_{div} for the combustor lengths in the testing set. For each test case (i.e., combustor length), the GP algorithm predicts the mean μ and variance σ^2 of the distribution of the system response at that point using Eqs. (7) and (8) under the Gaussian assumption. In other words, for a given combustor length, the GP regression algorithm predicts how different the system response is expected to be from the nominal state. One of the reasons for using GP for system response prediction is its ability to quantify the uncertainty in the estimate of the response. The proposed algorithm thus estimates the most likely response (i.e., mean) together with the variations about the mean, which may accrue from possible sources of uncertainties in the estimation (e.g., measurement noise and insufficient training data).

The difference between the predicted mean and true response value for each test case is noted. In addition, the number of test cases for which the true test value does not fall in the predicted 95% confidence intervals (i.e., $[\mu - 2\sigma, \mu + 2\sigma]$) is recorded, which is represented by the block “error metric” in the flowchart of Fig. 3. The entire procedure has been repeated for 20 random combinations of training and testing sets, and their average performance is computed. For every combination of training and testing sets of combustor lengths under a given inlet velocity and equivalence ratio, four different sampling durations of the corresponding time series have been considered: 2 s, 4 s, 6 s, and 8 s.

Figure 5 shows the mean (μ) and the 95% confidence intervals (i.e., $[\mu - 2\sigma, \mu + 2\sigma]$) of F_{div} , predicted by the GP algorithm for all test cases, and the corresponding true values of F_{div} superimposed on it, for a single run (i.e., a specific combination of training and testing data sets) for four different window lengths of time series: 2 s, 4 s, 6 s, and 8 s. The inlet velocity and equivalence ratio for this subset are 40 m/s and 0.55, respectively. For this run, it is observed that, for all test cases, the true value of F_{div} lies in the predicted 95% confidence interval, for all four window lengths of time series. In addition, for each window length of time series, the difference between predicted mean and true response value has been noted for each test case, and the average is taken over all test cases in the testing set. The average errors over the testing set in this run for the four different window lengths of time series data are listed in Table 2.

Figure 6 shows the mean (μ) and the 95% confidence intervals (i.e., $[\mu - 2\sigma, \mu + 2\sigma]$) of F_{div} , predicted by the GP algorithm for all test cases, and the corresponding true values of F_{div} superimposed on it, for a single run (i.e., a specific combination of training and testing data sets) for four different window lengths of time series data (2 s, 4 s, 6 s, and 8 s), for another data set. The inlet velocity and equivalence ratio for this subset are 25 m/s and 0.525, respectively. Similar to the results reported on the previous subset, it is observed in this run that the true value of F_{div} lies in

Table 1 Classification accuracy for D -Markov machines ($D = 1$) using SVM classifiers for different data lengths

Time series window length (s)	2	4	6	8
D -Markov parameters	$ \Sigma = 9$	$ \Sigma = 9$	$ \Sigma = 8$	$ \Sigma = 8$
Accuracy for linear kernel (%)	94	95	96	94
Accuracy for radial basis function	97	97	98	98
Kernel ($\sigma = 1$) (%)				

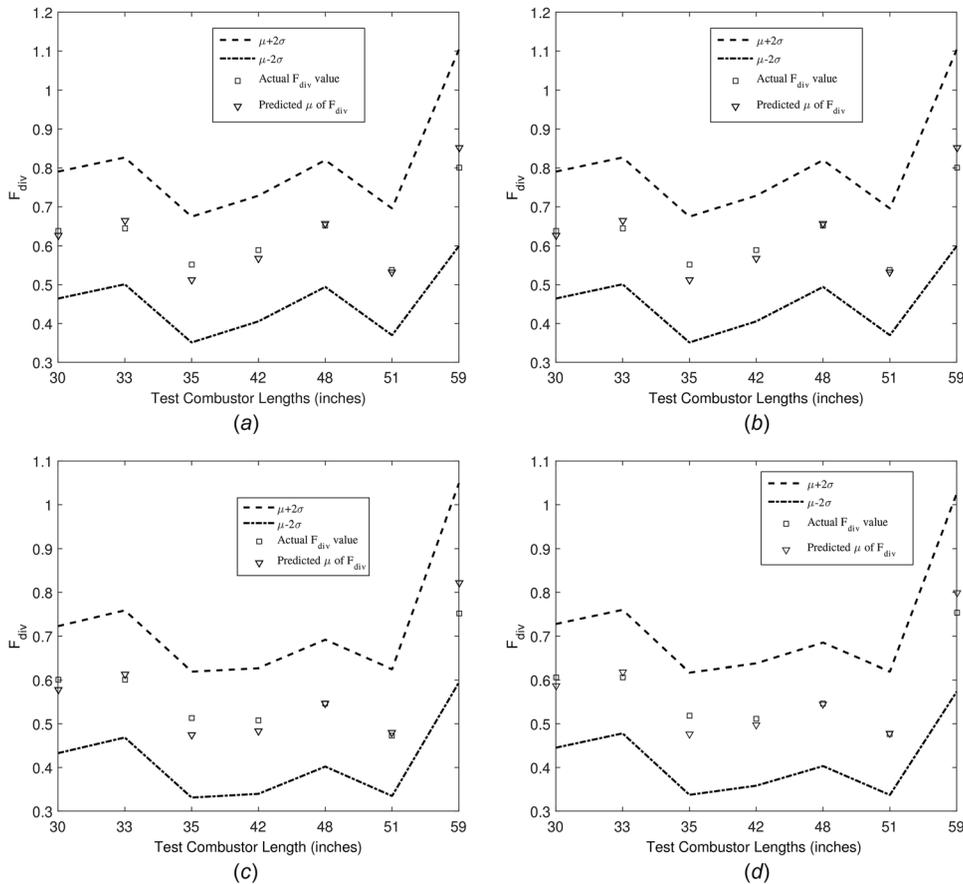


Fig. 5 Feature divergence F_{div} predicted by GP regression algorithm for inlet velocity = 40 m/s and $\phi = 0.55$: (a) F_{div} using 2 s of data, (b) F_{div} using 4 s of data, (c) F_{div} using 6 s of data, and (d) F_{div} using 8 s of data

the predicted 95% confidence interval for all window lengths of time series. The implication of the above results for a design exercise is that these figures provide the information on how close the system is to the stability boundary. Considering the uncertainties involved, the designer would try to select a set of parameters, within the constraint of permissible range at which F_{div} is lowest, as represented by the block “user defined criteria for stability” in the flowchart of Fig. 3. The average errors over the testing set in this run for four different window lengths of time series data are listed in Table 3. It is also shown in Tables 2 and 3 that the window length of time series data does not significantly affect the accuracy of the estimated mapping between combustor length and system response, provided that the data are sufficiently long to capture the system dynamics.

The plots in Fig. 7 compare the profiles of normalized F_{div} and normalized $Prms_{div}$ for the test run corresponding to inlet velocity = 40 m/s and equivalence ratio = 0.55, for 2 s and 8 s window lengths of time series data (whose prediction results are displayed in Fig. 5). Normalized F_{div} follows a trend similar to that of normalized $Prms_{div}$, thus showing that the PFSA feature is consistent with $Prms_{div}$ in terms of quantifying the system stability. Since F_{div} is significantly larger in magnitude than $Prms_{div}$ for the majority of cases, the PFSA feature is apparently more sensitive (and hence more discriminative) to changes in system dynamics. In other words, for the same change in system dynamics, the divergence of the PFSA feature from the original feature would be much higher than that corresponding to the $Prms$ feature. For applications to combustor design, this increased sensitivity is preferable because of the capability of correctly detecting smaller anomalies with the same threshold. In other words, the user would like to design a combustor with a smaller chance of becoming

unstable. In such a scenario, if a more sensitive feature is used, even a small deviation in the system behavior from the nominal state would be manifested as a large feature divergence; this would dissuade the user from choosing the combustor parameters that could potentially lead to instabilities. Under this train of logic, the combustor parameters, chosen corresponding to least feature divergence, would be more conservative and thus have a smaller chance of combustion instability.

5 Conclusions and Future Work

A dynamic data-driven method, based on a Bayesian nonparametric technique of Gaussian process (GP) regression, has been proposed in this paper as a tool for combustor design. The underlying assumption of GP regression is that, for a finite set of operating conditions, the corresponding system responses are jointly Gaussian; no assumptions are made on the nature of the relation between operating conditions and the resulting system response. However, the normal assumptions may not strictly hold, because of the introduction of estimated parameters, which is reflected in the uncertainties of the estimated response.

Table 2 Mean error for a single run on four different window lengths (sampled at 8192 Hz) of time series at inlet velocity = 40 m/s and $\phi = 0.55$

Window length (s)	2	4	6	8
Mean error	0.0003	-0.0004	0.0008	-0.0025

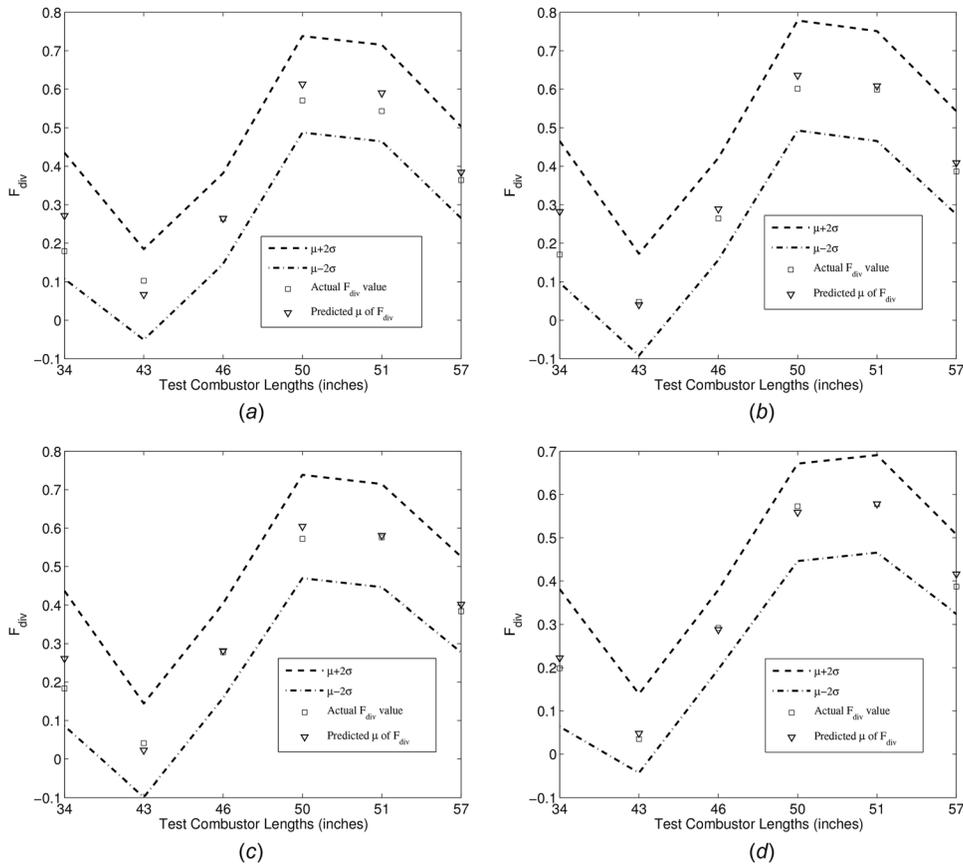


Fig. 6 Feature divergence F_{div} predicted by GP regression algorithm for inlet velocity = 25 m/s and $\phi = 0.525$: (a) F_{div} using 2 s of data, (b) F_{div} using 4 s of data, (c) F_{div} using 6 s of data, and (d) F_{div} using 8 s of data

Table 3 Mean error for a single run on four different window lengths (sampled at 8192 Hz) of time series at inlet velocity = 25 m/s and $\phi = 0.525$

Window length (s)	2	4	6	8
Mean error	0.0276	0.0326	0.0194	0.0081

The method has been validated on experimental data of pressure time-series from a lean-premixed swirl-stabilized combustor. Given an ensemble of training data for a set of constant equivalence ratio and inlet velocity, the algorithms determine the

mapping between the combustor length and the system response. The algorithms can then predict the distribution of the system response for every other point in the space of combustor lengths, for which experimental data may not be available. This information can then be used for identifying the length of the combustor that will yield the desired system response. While the proposed design methodology is more discriminative with respect to small deviations (e.g., those resulting from evolving anomalies) than that based on *Prms*, it is apparently more robust (i.e., less fragile) to measurement noise that is inherent in the time series data, because of symbolization [19].

Although the work presented here is validated for design of a laboratory-scale combustor as a proof of concept study, it is

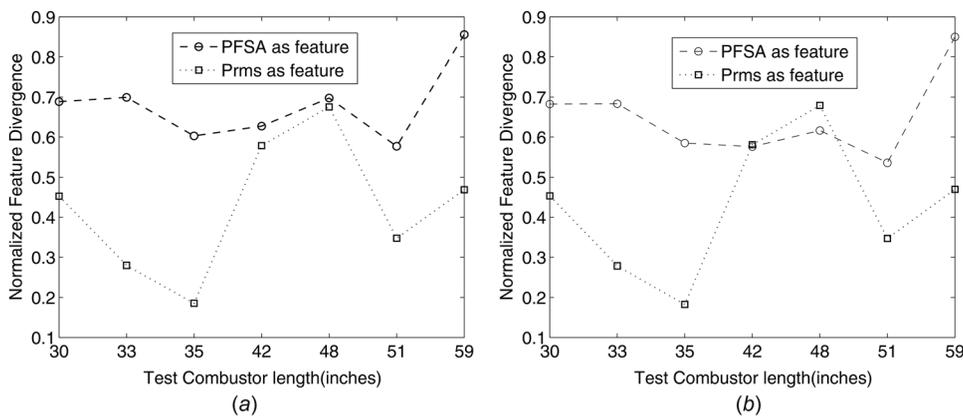


Fig. 7 Sensitivity comparison of PFSA and *Prms* features: (a) PFSA and *Prms* feature divergence for 2 s data and (b) PFSA and *Prms* feature divergence for 8 s data

envisioned that this method can be extended to more complex industrial-scale combustors, because the input to this dynamic data-driven approach is the pressure time series, which can be easily generated in such combustors. In the initial stage of design, the actual experiments used to generate data in the present work may be replaced by a limited number of high-fidelity simulations, involving unsteady Reynolds-averaged Navier–Stokes equation or large eddy simulations and state-of-the-art models for turbulent combustion appropriate to the combustion mode at hand (e.g., premixed, nonpremixed, or partially premixed). On the other hand, for modifications of existing combustors, knowledge acquired from experiments can be used in the design. From these perspectives, topics of future research on the proposed design method are delineated below.

- (1) Theoretical and experimental research on how the proposed dynamic data-driven method can be gainfully integrated with the current state-of-the-art (including the model-based) tools of combustor design.
- (2) Performance comparison of the design algorithm with state-of-the-art design methodologies not including usage of *Prms*.
- (3) Evaluation of the design algorithm for different parameters (e.g., depth $D > 1$ instead of $D = 1$ (see Definition 4.4)).
- (4) Testing of the design algorithm on combustors of different geometries and input parameters.
- (5) Extension of the proposed methodology for design of combustors under active control.

Acknowledgment

The work reported in this paper has been supported in part by the U.S. Air Force Office of Scientific Research (AFOSR) under Grant No. FA9550-15-1-0400. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies. The authors are thankful to Professor Domenic Santavicca at Pennsylvania State University, who has provided the experimental data.

References

- [1] Laera, D., 2015, "Nonlinear Combustion Instabilities Analysis of Azimuthal Mode in Annular Chamber," *Energy Procedia*, **82**, pp. 921–928.
- [2] Lieuwen, T. C., and Yang, V., 2005, *Combustion Instabilities in Gas Turbine Engines: Operational Experience, Fundamental Mechanisms, and Modeling*, American Institute of Aeronautics and Astronautics, Reston, VA.
- [3] Lefebvre, A. H., 1999, *Gas Turbine Combustion*, 2nd ed., Taylor & Francis, Philadelphia, PA.
- [4] Mattingly, J., Heiser, W., and Pratt, D., 2002, *Aircraft Engine Design* (AIAA Education Series), 2nd ed., American Institute of Aeronautics and Astronautics, Reston, VA.
- [5] O'Connor, J., Acharya, V., and Lieuwen, T., 2015, "Transverse Combustion Instabilities: Acoustic, Fluid Mechanic, and Flame Processes," *Prog. Energy Combust. Sci.*, **49**, pp. 1–39.
- [6] Zahn, M., Schulze, M., Hirsch, C., and Sattelmayer, T., 2016, "Impact of Quarter Wave Tube Arrangement on Damping of Azimuthal Modes," *ASME Paper No. GT2016-56450*.
- [7] Lei, L., Zhihui, G., Chengyu, Z., and Xiaofeng, S., 2010, "A Passive Method to Control Combustion Instabilities With Perforated Liner," *Chin. J. Aeronaut.*, **23**(6), pp. 623–630.

- [8] Ćosić, B., Bobusch, B. C., Moeck, J. P., and Paschereit, C. O., 2011, "Open-Loop Control of Combustion Instabilities and the Role of the Flame Response to Two-Frequency Forcing," *ASME Paper No. GT2011-46503*.
- [9] Jones, C., Lee, J., and Santavicca, D., 1999, "Closed-Loop Active Control of Combustion Instabilities Using Subharmonic Secondary Fuel Injection," *J. Propul. Power*, **15**(4), pp. 584–590.
- [10] Paschereit, C. O., Gutmark, E. J., and Weisenstein, W., 1998, "Control of Thermoacoustic Instabilities and Emissions in an Industrial Type Gas-Turbine Combustor," *Symp. Combust.*, **27**(2), pp. 1817–1824.
- [11] Bauerheim, M., Parmentier, J.-F., Salas, P., Nicoud, F., and Poinso, T., 2014, "An Analytical Model for Azimuthal Thermo-Acoustic Modes in Annular Chamber Fed by an Annular Plenum," *Combust. Flame*, **161**(5), pp. 1374–1389.
- [12] Bourgoin, J.-F., Durox, D., Moeck, J. P., Schuller, T., and Candel, S., 2015, "Characterization and Modeling of a Spinning Thermoacoustic Instability in an Annular Combustor Equipped With Multiple Matrix Injectors," *ASME J. Eng. Gas Turbines Power*, **137**(2), p. 021503.
- [13] Innocenti, A., Andreini, A., Facchini, B., and Cerutti, M., 2016, "Numerical Analysis of the Dynamic Flame Response and Thermo-Acoustic Stability of a Full-Annular Lean Partially-Premixed Combustor," *ASME Paper No. GT2016-57182*.
- [14] Sarkar, S., Chakravarthy, S. R., Ramanan, V., and Ray, A., 2016, "Dynamic Data-Driven Prediction of Instability in a Swirl-Stabilized Combustor," *Int. J. Spray Combust. Dyn.*, **8**(4), pp. 235–253.
- [15] Sen, S., Sarkar, S., Chaudhari, R. R., Mukhopadhyay, A., and Ray, A., 2017, "Lean Blowout (LBO) Prediction Through Symbolic Time Series Analysis," *Combustion for Power Generation and Transportation: Technology, Challenges and Prospects*, A. K. Agarwal, S. De, A. Pandey, and A. P. Singh, eds., Springer, Singapore, pp. 153–167.
- [16] Darema, F., 2004, "Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements," International Conference on Computational Science (ICCS), Kraków, Poland, June 6–9, pp. 662–669.
- [17] Daw, C., Finney, C., and Tracy, E., 2003, "A Review of Symbolic Analysis of Experimental Data," *Rev. Sci. Instrum.*, **74**(2), pp. 915–930.
- [18] Hauser, M., Li, Y., Li, J., and Ray, A., 2016, "Real-Time Combustion State Identification Via Image Processing: A Dynamic Data-Driven Approach," American Control Conference (ACC), Boston, MA, July 6–8, pp. 3316–3321.
- [19] Beim Graben, P., 2001, "Estimating and Improving the Signal-to-Noise Ratio of Time Series by Symbolic Dynamics," *Phys. Rev. E*, **64**(5), p. 051104.
- [20] Kim, K. T., Lee, J. G., Quay, B. D., and Santavicca, D. A., 2010, "Response of Partially Premixed Flames to Acoustic Velocity and Equivalence Ratio Perturbations," *Combust. Flame*, **157**(9), pp. 1731–1744.
- [21] Sipser, M., 2012, *Introduction to the Theory of Computation*, Cengage Learning, Boston, MA.
- [22] Lind, D., and Marcus, B., 1995, *An Introduction to Symbolic Dynamics and Coding*, Cambridge University Press, Cambridge, UK.
- [23] Ray, A., 2004, "Symbolic Dynamic Analysis of Complex Systems for Anomaly Detection," *Signal Process.*, **84**(7), pp. 1115–1130.
- [24] Mukherjee, K., and Ray, A., 2014, "State Splitting and Merging in Probabilistic Finite State Automata for Signal Representation and Analysis," *Signal Process.*, **104**, pp. 105–119.
- [25] Bahrapour, S., Ray, A., Sarkar, S., Damarla, T., and Nasrabadi, N. M., 2013, "Performance Comparison of Feature Extraction Algorithms for Target Detection and Classification," *Pattern Recognit. Lett.*, **34**(16), pp. 2126–2134.
- [26] Rajagopalan, V., and Ray, A., 2006, "Symbolic Time Series Analysis Via Wavelet-Based Partitioning," *Signal Process.*, **86**(11), pp. 3309–3320.
- [27] Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., and Carrasco, R. C., 2005, "Probabilistic Finite-State Machines—Part I," *IEEE Trans. Pattern Anal. Mach. Intell.*, **27**(7), pp. 1013–1025.
- [28] Mallapragada, G., Ray, A., and Jin, X., 2012, "Symbolic Dynamic Filtering and Language Measure for Behavior Identification of Mobile Robots," *IEEE Trans. Sys. Man Cybern., Part B*, **42**(3), pp. 647–659.
- [29] Rao, C., Ray, A., Sarkar, S., and Yasar, M., 2009, "Review and Comparative Evaluation of Symbolic Dynamic Filtering for Detection of Anomaly Patterns," *Signal Image Video Process.*, **3**(2), pp. 101–114.
- [30] Bishop, C., 2006, *Pattern Recognition and Machine Learning* (Information Science and Statistics), Springer-Verlag, New York.
- [31] Rasmussen, C., and Williams, C., 2005, *Gaussian Processes for Machine Learning* (Adaptive Computation and Machine Learning), MIT Press, Cambridge, MA.
- [32] Cortes, C., and Vapnik, V., 1995, "Support-Vector Networks," *Mach. Learn.*, **20**(3), pp. 273–297.