Adaptive Gaussian process surrogate models for efficient uncertainty quantification of flutter boundary

George Lu*
University of Toronto, Toronto, Ontario, M3H 5T6, Canada

Amin Fereidooni† and Anant Grewal‡
National Research Council Canada, Ottawa, Ontario, K1V 2B1, Canada

Masayuki Yano§
University of Toronto, Toronto, Ontario, M3H 5T6, Canada

We present and assess non-intrusive surrogate modeling approaches to estimating the probability of flutter in the presence of model uncertainties for a range of flight conditions. Our goal is to provide a probabilistic characterization of the flutter boundary that accounts for uncertainties in model parameters and operating conditions to better inform design and testing decisions in the certification of new aircraft. We use adaptively constructed Gaussian process surrogate models and high-fidelity computational aeroelasticity models to efficiently characterize sensitive features like the transonic dip while controlling the cost of many-query uncertainty analysis. To obtain acquisition functions for adaptive sampling, we derive relationships between misclassification of flutter risk and existing sequential design strategies, propose a new sampling strategy, and use the strategies to accurately approximate flutter probability within a limited sampling budget. We assess the efficiency of these methods using synthetic test functions, a traditional aeroelasticity model with Theodorsen's aerodynamics model, and a computational aeroelasticity model based on the Euler equations.

I. Introduction

Flutter, a dynamic instability caused by interactions between aerodynamic and structural forces, poses a critical challenge in aircraft design and certification. Ensuring an aircraft's safety within its flight envelope requires rigorous testing, often involving substantial time and expense [1]. The flutter boundary is often initially estimated using analytical or computational aeroelasticity models [2], which provide boundaries about some minimal level of structural damping. However, evaluating the safety of an aircraft with respect to a failure boundary inevitably carries the notion of risk and uncertainties [3]. Previous studies have shown that variation in design parameters, such as mass and stiffness,

*Graduate student, Institute for Aerospace Studies, 4925 Dufferin St, Toronto, Ontario, M3H 5T6, Canada.

†Research Officer, Flight Research Laboratory, 1920 Research Road, Ottawa, Ontario. AIAA Senior Member.

‡Principal Research Officer, Flight Research Laboratory, 1920 Research Road, Ottawa, Ontario. AIAA Associate Fellow.

§Associate Professor, Institute for Aerospace Studies, 4925 Dufferin St, Toronto, Ontario, M3H 5T6, Canada.

can result in flutter occurring at speeds far below that predicted by deterministic analysis [4]. We hence wish to quantify this uncertainty and provide a probabilistic assessment of the safety of an aircraft.

Pettit [5], Dai and Yang [6], and Beran et al. [3] review many methods for uncertainty quantification (UQ) in flutter, including many Monte Carlo (MC) and perturbation-based approaches. However, due to the high computational cost of each computational aeroelasticity (CAE) simulation, the "brute-force" application of MC with thousands of full-order model evaluations is often infeasible. To reduce the computational cost, the original full-order model is typically replaced by a reduced-order surrogate model. One approach common in forward UQ applications in flutter is the non-intrusive polynomial chaos (NIPC) method [3, 7]. Hosder et al. [8] use NIPC to assess the impact of uncertainty in flight conditions on flutter prediction. Hantrais-Gervois and Savin [9] use polynomial surrogates to perform UQ on various quantities of interest (QoI) under structural uncertainties. Scarth et al. [10] use NIPC to propagate uncertainties in composite ply angles to the aeroelastic response of the composite wing. NIPC is non-intrusive and hence readily applicable to any CAE models and flutter configurations; however, NIPC suffers from the curse of dimensionality [11], where the number of full-order model evaluations required to construct a surrogate model increases possibly exponentially with the number of uncertain input variables.

To mitigate the effect of the curse of dimensionality, we explore in this work the use of adaptive Gaussian process (GP) surrogate models. These models use an acquisition function that targets the lower-dimensional flutter-boundary manifold to accelerate the UQ of the flutter boundary. Our goal is to provide the probability of flutter occurring for a range of flight conditions. For each flight condition, the probability of flutter is defined as the percentage of random aircraft configurations yielded by uncertain parameters that experience flutter for that flight condition. Unlike NIPC, GP models (GPMs) can be constructed from an unstructured set of parameter samples. In combination with the statistical estimates, GPMs can be constructed adaptively about the flutter boundary, greatly reducing the number of full-order model evaluations required to accurately capture the boundary.

GPs have been used to develop adaptive sampling strategies for quantifying the flutter boundary in deterministic contexts. For example, Marques et al. [12] develop a multi-fidelity approach to sampling near the flutter boundary, where GP-estimated error determines an appropriate model to evaluate at candidate sample points. Goizueta et al. [13] use GP-informed sampling to efficiently construct a surrogate representation of the flutter envelope. Allen and Camberos [4] assess the efficacy of existing GP sampling strategies for determining an aeroelastic response surface for UQ purposes, which show that GP surrogates may show good agreement with MC results, but require a judicious selection of sampling methodology. In this work, we derive relationships between existing GP sampling strategies and a flutter boundary error metric, propose a new sampling strategy, and assess the efficacy of the strategies for the UQ of the flutter boundary.

The primary contributions of this work are the following: First, we derive GP sampling strategies starting from a flutter boundary error metric, thereby characterizing relationships among various sampling strategies in terms of fidelity and computational cost in the context of the UQ of the flutter boundary. Second, we apply the sampling strategies to UQ problems with a moderate number (i.e., 4–6) of parameters, demonstrate that adaptive GPMs are significantly more efficient than widely used NIPC method for flutter boundary UQ, and compare the efficacy of the various sampling strategies.

The remainder of this paper is organized as follows. In Section II, we provide a mathematical description of the flutter UQ problem considered in this work. Section III reviews key aspects of GPMs that are relevant to the present work. Section IV introduces various acquisition functions for GPMs and relates the functions to the flutter boundary error metric. Sections V–VII assess various adaptive GPMs using a synthetic test function, a traditional aeroelasticity model based on Theodorsen's aerodynamics model, and a computational aeroelasticity model based on the Euler equations.

II. UQ for flutter prediction

A. Aeroelastic flutter

We introduce an abstract aeroelastic system, which is parameterized by a set of P input parameters $z \in \mathcal{Z} \subset \mathbb{R}^P$. The parameters may characterize the system configuration, such as structural stiffness and geometry, and/or flight condition, such as the velocity and altitude. The set of n_{dof} generalized equations of motion for the n_{dof} structural degrees of freedom in an aeroelastic system can be written as

$$M(z)\ddot{q}(t;z) + C(z)\dot{q}(t;z) + K(z)q(t;z) = \Xi(q(t;z), \dot{q}(t;z), \ddot{q}(t;z); z), \tag{1}$$

where $q(\cdot;z) : [0,\infty) \to \mathbb{R}^{n_{\text{dof}}}$ is the (time-dependent) generalized coordinates; $M(z) \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{dof}}}$, $C(z) \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{dof}}}$, and $K(z) \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{dof}}}$ are the generalized mass, damping, and stiffness matrices, respectively; and $\Xi(q(t;z), \dot{q}(t;z), \ddot{q}(t;z); z) \in \mathbb{R}^{n_{\text{dof}}}$ is the generalized aerodynamic forces. Below the flutter velocity, the solution of this system exhibits damped oscillatory motion, where perturbations decay exponentially; i.e., $\|q(\cdot;z)\|_2 \to 0$ as $t \to \infty$. Above the flutter velocity however, oscillatory modes may have negative damping factors, causing exponential growth in oscillations that lead to potential damage; i.e., $\|q(\cdot;z)\|_2 \to \infty$ as $t \to \infty$. The behaviour of these modes may be studied either through analysis of the solution in the time domain or through eigenvalue analysis in the frequency domain using the p-k method [14], among other methods. In this work, we assume that the dominant mode determining flutter is known.

B. Stochastic flutter boundary estimation

In this work, we focus on non-intrusive forward UQ that propagates parametric uncertainty in an aeroelastic model to the resulting damping coefficients, and use this to determine the probabilistic characteristics of the resulting flutter boundary. In the following discussion, we partition a subset of inputs $z \in \mathcal{Z} \subset \mathbb{R}^P$ to the CAE solver into two sets: a set of P_1 configuration parameters $x \in \mathcal{X} \subset \mathbb{R}^{P_1}$, such as parameters characterizing structural stiffness and geometry; and a set of P_2 operational parameters $y \in \mathcal{Y} \subset \mathbb{R}^{P_2}$, which are the equivalent velocity and Mach number corresponding to the domain of the aeroelastic stability margin defined in [2]. We consider a single quantity of interest in the CAE solution: the dominant damping coefficient $f(x, y)$, which is used as a flutter indicator and is a function of the input parameters $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. To evaluate $(x, y) \mapsto f(x, y)$, we first invoke the CAE solver based on OpenFOAM [15] to obtain the unsteady time series response for the configuration (x, y), and then apply the matrix pencil method [16] to the time series to obtain the damping coefficient $f(x, y)$. A negative coefficient $f(x, y) < 0$ corresponds to flutter, while a positive coefficient $f(x, y) > 0$ corresponds to damped motion free of flutter. The contour $f(x, y) = 0$ defines the crucial flutter boundary.

While the flutter boundary must be determined over a range of operating conditions $y \in \mathcal{Y}$, we also must consider uncertainty in configuration parameters $x \in \mathcal{X}$. Abstractly, we endow the configuration space $\mathcal{X}$ with a probability measure P_X and the associated density $\rho_X : \mathcal{X} \to \mathbb{R}_{\geq 0}$. In the present context of forward UQ, we assume P_x exists and is known. For any given operating condition $y \in \mathcal{Y}$, we now wish to determine the probability of flutter $P[f(\cdot, y) < 0] = P_X(\{x \in \mathcal{X} \mid \mathbb{I}[f(x, y) < 0]\})$, where $\mathbb{I}$ is the indicator function that evaluates to 1 if the argument is true and to 0 otherwise. The probability of flutter for $y \in \mathcal{Y}$ is given by

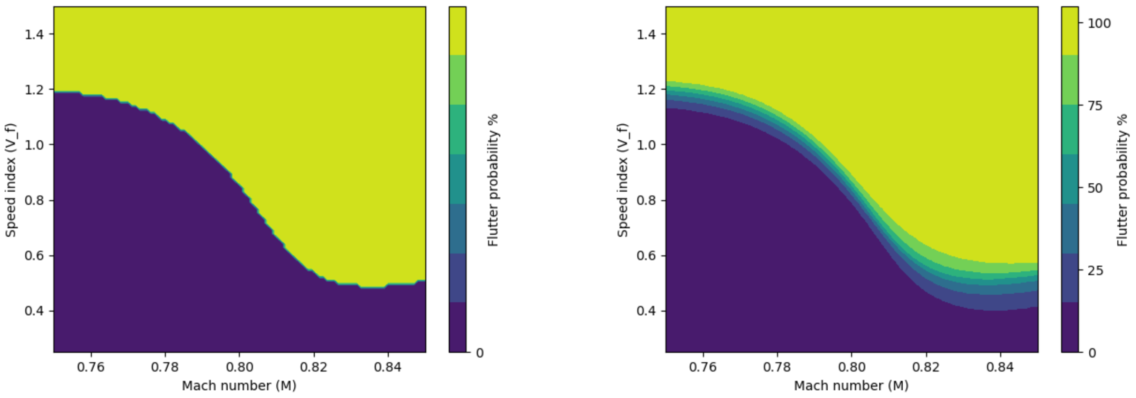
$$F(y) := \int_{\mathcal{X}} \mathbb{I}[f(x, y) \leq 0] dP_X(x). \tag{2}$$

Figure 1 shows a concrete example of deterministic and probabilistic classifications of the flutter boundary for the Isogai aeroelastic model [17] based on an Euler aerodynamics model and two degrees of freedom structural model; a more detailed description of the CAE model, including specific configuration parameter values, can be found in [15]. The deterministic classification is compared to a probabilistic classification, where uncertainties in heave and pitch stiffness k_h and k_p are considered, with the mean values $E[k_h] = 3.4 \times 10^5$ and $E[k_p] = 3.0 \times 10^5$ (which are the values in the deterministic analysis) and the variance $\text{Var}[k_h] = (0.25 \times 10^5)^2$ and $\text{Var}[k_p] = (0.3 \times 10^5)^2$. The objective of this work is to develop methods that can rapidly and reliably construct this probabilistic flutter boundary, to further inform decisions on safe operational limits.

III. Gaussian Process models

For many UQ applications involving a CAE model, it is computationally intractable to conduct UQ using "brute-force" MC simulations, which potentially require thousands of evaluations of the original expensive CAE model. To overcome this computational challenge, we first train a GP surrogate model $f_{\text{GP}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that approximates the expensive $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and then use the surrogate model to perform MC simulations. In this section, we briefly motivate and review the use of GPMs.

GPMs provide a data-driven and non-intrusive approach to surrogate modeling, yielding a probabilistic estimate



(a) Deterministic classification

(b) Probabilistic classification

Fig. 1 Failure regions (flutter boundary) computed using both deterministic and probabilistic analysis

of a response surface given a set of sampling points and the associated responses. GPMs also permit unstructured sampling, which makes them particularly suitable for constructing a surrogate model that targets localized and/or lower-dimensional features in the sample space, such as the flutter boundary.

To facilitate concise presentation of GPMs, we recall variable $z \coloneqq (x, y) \in \mathcal{X} \times \mathcal{Y} \coloneqq \mathcal{Z} \subset \mathbb{R}^{P \coloneqq P_1 + P_2}$, since the aforementioned distinction of input variables into x and y is unnecessary for the formulation or evaluation of a GPM. The model provides a probabilistic approximation of the quantity of interest $z_* \mapsto f(z_*)$ given training dataset $\{(z_i, f(z_i))\}_{i=1}^n$ of size n; e.g., evaluations of the CAE model $f(z)$ at select parameter values $\{z_i\}_{i=1}^n$. The GP approximation of the quantity of interest is a normal distribution

$$f_{\text{GP}}(z_*; \{(z_i, f(z_i))\}_{i=1}^n, k) \sim \mathcal{N}(\mu_{\text{GP}}(z_*), \sigma_{\text{GP}}(z_*)^2)$$

with the posterior mean and variance

$$\begin{aligned} \mu_{\text{GP}}(z_*) &\coloneqq E[f_{\text{GP}}(z_*; \{(z_i, f(z_i))\}_{i=1}^n, k)] = K(z_*, Z)[K(Z, Z) + \sigma_{\text{meas}}^2 I]^{-1} f(Z), \\ \sigma_{\text{GP}}(z_*)^2 &\coloneqq \text{Var}[f_{\text{GP}}(z_*; \{(z_i, f(z_i))\}_{i=1}^n, k)] = K(z_*, z_*) - K(z_*, Z)[K(Z, Z) + \sigma_{\text{meas}}^2 I]^{-1} K(Z, z_*), \end{aligned} \tag{3}$$

respectively, where $Z = (z_1, \ldots, z_n)$ is the set of n training parameter values; $k : \mathbb{R}^P \times \mathbb{R}^P \to \mathbb{R}$ is a covariance kernel used to evaluate covariance matrices $K(Z, Z) \in \mathbb{R}^{n \times n}$ such that $K(Z, Z)_{i,j} = k(z_i, z_j)$ for $i, j = 1, \ldots, n$; $K(Z, z_*) \in \mathbb{R}^{n \times 1}$ such that $K(Z, z_*)_i = k(z_i, z_*)$ for $i = 1, \ldots, n$; $K(z_*, Z) \in \mathbb{R}^{1 \times n}$ such that $K(z_*, Z)_j = k(z_*, z_j)$ for

$j = 1, \dots, n$; and $K(z_*, z_*) = k(z_*, z_*)$. In this work, we choose the anisotropic squared exponential kernel

$$k(z_1, z_2) := \sigma_{\mathrm{SE}}^2 \exp\left(-\sum_{i=1}^{P} \frac{(z_{1i} - z_{2i})^2}{2l_i^2}\right) \tag{4}$$

with hyperparameters consisting of the signal variance $\sigma_{\mathrm{SE}}^2 \in \mathbb{R}_{>0}$; dimension-dependent length scales $l_i \in \mathbb{R}_{>0}$, $i = 1, \dots, P$; and measurement variance $\sigma_{\mathrm{meas}}^2 \in \mathbb{R}_{\geq 0}$, which is included in the optimization of the other hyperparameters to avoid conditioning issues [18]. Hyperparameters are found by maximizing the log-marginal likelihood. In this work, we use SciKit-Learn's [19] GP package. To mitigate the effect of multiple local minima, we initialize the optimization with multiple sets of random hyperparameter values and take the result with the highest log-marginal likelihood.

In addition to a model estimate given by the posterior mean of the GP model, the posterior variance also informs the confidence of the model at an evaluation point and can serve as an error indicator. This variance estimate plays a crucial role in the adaptive selection of sampling points, which will be reviewed in section IV. When the (standard) dataset $\{z_i, f(z_i)\}_{i=1}^n$ of size n is used to construct a GPM, we will also use an abbreviated notation*

$$f_n(z_*) := f_{\mathrm{GP}}(z_*; \{(z_i, f(z_i))\}_{i=1}^n, k),$$

and denote the associated mean and variance by $\mu_n(z_*) := E[f_n(z_*)]$ and $\sigma_n(z_*)^2 := \mathrm{Var}[f_n(z_*)]$.

Some of the GP sampling strategies that we will consider require computationally efficient ways to update to the posterior mean $\hat{\mu}(z_*)$ and variance $\hat{\sigma}^2(z_*)$ when the training set is augmented by point z_{n+1} and the associated measurement $f(z_{n+1})$ while holding hyperparameters constant. We appeal to the linearity of the predictor to efficiently update the posterior mean and variance:

$$\mu_{n+1}(z_*) = \mu_n(z_*) + \mathrm{cov}(z_*, z_{n+1})[\sigma_n^2(z_{n+1}) + \sigma_{\mathrm{meas}}^2]^{-1}(f(z_{n+1}) - f_n(z_{n+1})), \tag{5}$$

$$\sigma_{n+1}^2(z_*) = \sigma_n^2(z_*) - \mathrm{cov}(z_*, z_{n+1})[\sigma_n^2(z_{n+1}) + \sigma_{\mathrm{meas}}^2]^{-1}\mathrm{cov}(z_{n+1}, z_*), \tag{6}$$

where $\mathrm{cov}(z_*, z_{n+1}) = K(z_*, z_{n+1}) - K(z_*, Z)[K(Z, Z) + \sigma_{\mathrm{meas}}^2 I]^{-1} K(Z, z_{n+1})$.

IV. Gaussian Process sampling methods

A. Sequential sampling strategy

Given the high evaluation cost of f in the intended setting of this work, we now consider an adaptive sequential construction of GPMs. To control the computational cost, sampling points for the GPM must be chosen efficiently to ensure sufficient accuracy with relatively few evaluations. As discussed in the introduction, in the context of deterministic

*However, a rather verbose notation $f_{\mathrm{GP}}(z_*; \{(z_i, f(z_i))\}_{i=1}^n, k)$ will be needed to precisely describe some of the sampling strategies that use a non-standard construction of GPM.

flutter boundary identification, Goizueta et al. [13] use GPM-informed sampling to efficiently construct a surrogate representation of the flutter envelope, and Marques et al. [12] demonstrate efficient sampling with multi-fidelity GPMs for the flutter boundary of the Isogai model.

Sampling strategies for sequential (as opposed to batch) sampling, for both a deterministic and stochastic model, take the general form in Algorithm 1.

Algorithm 1: Sequential sampling for GPM using an acquisition function S.

```
input: Acquisition function: $S : \mathbb{R}^P \to \mathbb{R}$
output: GPM $\tilde{f}_N$ based on dataset $\{(z_i, f(z_i))\}_{i=1}^N$
Initialize dataset $\{(z_i, f(z_i))\}_{i=1}^{n_0}$ and construct the GPM $f_{n_0} \coloneqq f_{\mathrm{GP}}(\cdot; \{(z_i, f(z_i))\}_{i=1}^{n_0})$
for $n = n_0, \ldots, N$ do
    Find $z_{n+1} = \arg\max_{z \in \mathcal{Z}} S(z)$
    Evaluate the full-order model at $z_{n+1}$ to obtain $f(z_{n+1})$
    Update the GPM and retrain hyperparameters to obtain to $f_{n+1} \coloneqq f_{\mathrm{GP}}(\cdot; \{(z_i, f(z_i))\}_{i=1}^{n+1}; k)$
end
```

The efficacy of the sequential sampling strategy depends on the choice of the acquisition function S. In the remainder of this section, we discuss eight different acquisition functions that are designed to accurately capture the stochastic flutter boundary. We roughly classify these into three groups based on their primary metric: straddle-based strategies (Sections IV.B–IV.C); misclassification-based strategies (Sections IV.D–IV.F); and entropy-based strategies (Sections IV.G–IV.I). We elaborate more on these groupings in Section IV.J.

B. Straddle sampling

The first strategy explored is the level set straddling heuristic [20], from which many similar methodologies [21–23] have originated. The sampling strategy is based on the following acquisition function:

Definition 1 (Straddle sampling (S_{st})) *The acquisition function for the straddle sampling strategy is given by*

$$S_{\mathrm{st}}(z_{n+1}) \coloneqq \beta_t \sigma_n(z_{n+1}) - |\mu_n(z_{n+1}) - T|, \tag{7}$$

where $\beta_t \in \mathbb{R}_{\geq 0}$ is a user-specified parameter that balances exploitation and exploration and T is the threshold of the level set being classified. We set $\beta_t = 1.96$ following the original straddle heuristic proposed in [20] and $T = 0$ since the flutter boundary corresponds to the zero level set.

C. Input-probability-weighted straddle sampling

For probabilistic (as opposed to deterministic) failure boundary identification, we expect sampling strategies that account for the input probability distribution $\rho_X : \mathcal{X} \to \mathbb{R}_{\geq 0}$ to provide more efficient sampling. A modification of the straddle heuristic to account for uncertainty in the input $x \in \mathcal{X}$ is given by the following acquisition function:

Definition 2 (Input-probability-weighted straddle sampling (S_{wst})) *The acquisition function for input-probability-weighted straddle sampling is*

$$S_{\text{wst}}(x_{n+1}, y_{n+1}) \coloneqq [\beta_t \sigma_n(x_{n+1}, y_{n+1}) - |\mu_n(x_{n+1}, y_{n+1})|]\, \rho_X(x_{n+1}). \tag{8}$$

This modifies S_{st} to provide a higher score for points (x_{n+1}, y_{n+1}) with a higher input probability density. Consequently, the resulting model is more accurate about input parameter values that are more likely to occur in a UQ analysis. This strategy is also referred to as contour upper confidence bound (cUCB) in [24].

D. Stepwise uncertainty reduction

Stepwise uncertainty reduction (SUR) is a powerful strategy for sequential design, with the goal of optimal reduction in error of a quantity of interest at each step. In this section, we review the integrated stepwise uncertainty reduction strategy first proposed by Bect et al. [25]. In particular, we show that an "ideal" greedy sampling strategy that selects successive sampling points (x_{n+1}, y_{n+1}) that is expected to most reduce the approximation error of the flutter probability function $F(y)$ in an $L^2(\mathcal{Y})$ sense can be reduced to a computable form equivalent to the scoring heuristic presented by Bect et al.

To begin, we introduce a flutter probability function $F : \mathcal{Y} \to [0, 1]$ given by

$$F(y) \coloneqq \int_X \mathbb{I}[f(x, y) \leq 0]\, dP_X(x).$$

The approximation of F using a GPM with a dataset of size n is given by

$$F_n(y) \coloneqq \int_X \mathbb{I}[f_n(x, y) \leq 0]\, dP_X(x).$$

Unlike $F(y)$ which is a real number, $F_n(y)$ is a random variable since $f_n(x, y)$ is a stochastic function. An "ideal" point with which to augment this training set is taken to be the minimizer of

$$S_{\text{ideal}}(x_{n+1}, y_{n+1}) = E\left[\int_{\mathcal{Y}} (F(y) - F_{n+1}(y))^2 \, \mathrm{d}y\right], \tag{9}$$

where $F_{n+1}(y)$ is based on $f_{n+1}(z \equiv (x, y)) \coloneqq f_{\text{GP}}(x, y; \{(z_i, f(z_i))\}_{i=1}^{n+1}, k)$, and we take the expectation since $F_{n+1}(y)$ is a random variable. The "ideal" acquisition function S_{ideal}, however, cannot be evaluated without prior knowledge of the true function $f(x, y)$, which is unavailable in this scenario in which surrogate modeling is necessary in the first place.

To evaluate an approximate form of the ideal acquisition using available information, we first replace the exact $n + 1$-point GPM based flutter probability function $F_{n+1}(x, y)$ with an estimated $n + 1$-point GPM formed by augmenting

f_n with the candidate point:

$$\tilde{f}_{n+1}(z \equiv (x, y)) \coloneqq f_{\mathrm{GP}}(z; \{(z_1, f(z_1)), \dots, (z_n, f(z_n)), (z_{n+1}, g_{n+1} \coloneqq f_n(z_{n+1}))\}, k),$$

where the input data value at z_{n+1} is approximated using the n-point GPM as $g_{n+1} \coloneqq f_n(z_{n+1})$ in a one step lookahead. Note that $\tilde{f}_n$ is a stochastic function for two distinct reasons: (i) it is a GPM; and (ii) the input data for the $n+1$-st point, $g_{n+1} \coloneqq f_n(z_{n+1})$, is a random variable. We also introduce the associated lookahead flutter probability

$$\widetilde{F}_{n+1}(y) \coloneqq \int_{\mathcal{X}} \mathbb{I}[\tilde{f}_{n+1}(x, y) \leq 0] dP_X(x),$$

and rewrite S_{ideal} using the estimated lookahead flutter probability

$$\widetilde{S}_{\mathrm{ideal}}(x_{n+1}, y_{n+1}) = E\left[\int_{\mathcal{Y}} (F(y) - \widetilde{F}_{n+1}(y))^2 dy\right],$$

where the expectation is now taken over both the GPM f_{n+1} and its $n+1$-st data point value $g_{n+1} \coloneqq f_n(x_{n+1}, y_{n+1})$. However, we must still approximate the acquisition function in a form that can be evaluated without knowledge of $F(y)$ for all $y \in \mathcal{Y}$.

To this end, we appeal to the probability of misclassification to construct a computable bound to $\widetilde{S}_{\mathrm{ideal}}$. Consider an approximation of the exact (deterministic) damping coefficient $f(x, y)$ by the stochastic surrogate $f_n(x, y)$. In the construction of F_n, we are concerned only with whether or not this damping coefficient corresponds to a flutter condition, indicated by the sign of f_n. We also recall that $f_n(x, y) \sim \mathcal{N}(\mu_n(x, y), \sigma_n(x, y)^2)$, with the mean $\hat{\mu}_n(x, y)$ serving as the estimate itself and standard deviation $\hat{\sigma}_n(x, y)$ serving as the confidence in the estimate. The probability that the mean estimate $\mu_n(x, y)$ lies on the wrong side of the failure boundary $f(x, y) = 0$ can be estimated using the portion of the Gaussian distribution with sign opposite to the mean, as illustrated in figure 2. Therefore,

$$P[\mathrm{sign}(f(x, y)) \neq \mathrm{sign}(f_n(x, y)] = \Phi\left(-\frac{|\mu_n(x, y)|}{\sigma_n(x, y)}\right), \tag{10}$$

where $\Phi : \mathbb{R} \to [0, 1]$ is the cumulative distribution function of the standard normal distribution.

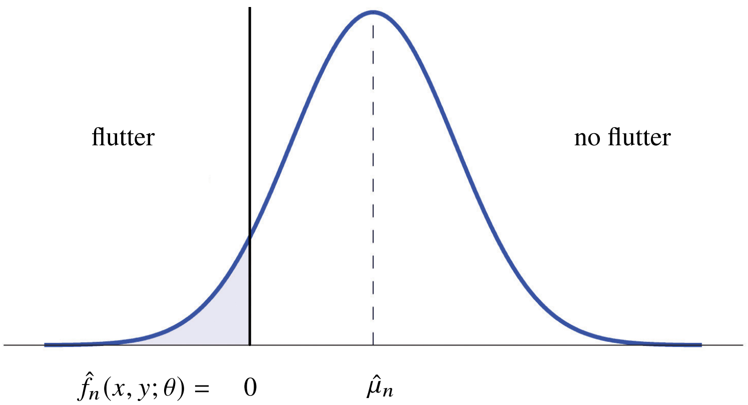


Fig. 2 Estimate of damping coefficient at (x, y) by GPM $f_n(x, y)$ with mean $\mu_n(x, y)$ and standard deviation $\sigma_n(x, y)$. The area of the shaded region of the resultant normal distribution is the probability that the mean $\mu_n(x, y)$ of the GPM has the wrong sign compared to the unknown true value $f(x, y)$: i.e., the probability of misclassification.

We now use the probability of misclassification to approximate $\widetilde{S}_{\text{ideal}}(x, y)$. To this end, we note that

$$\begin{aligned}
\tilde{S}_{\text{ideal}}(x_{n+1}, y_{n+1}) &= E\left[\int_{\mathcal{Y}}\left(\int_{\mathcal{X}}(\mathbb{I}[f(x, y) \leq 0] - \mathbb{I}[\tilde{f}_{n+1}(x, y) \leq 0])dP_X(x)\right)^2 dy\right] \\
&\leq E\left[\int_{\mathcal{Y}}\int_{\mathcal{X}}(\mathbb{I}[f(x, y) \leq 0] - \mathbb{I}[\tilde{f}_{n+1}(x, y) \leq 0])^2 dP_X(x)dy\right] \\
&= \int_{\mathbb{R}}\int_{\mathcal{F}}\int_{\mathcal{Y}}\int_{\mathcal{X}}(\mathbb{I}[f(x, y) \leq 0] - \mathbb{I}[\tilde{f}_{n+1}(x, y) \leq 0])^2 dP_X(x)dydP_{\tilde{f}_{n+1}}(\tilde{f}_{n+1})dP_{g_{n+1}}(g_{n+1}) \\
&= \int_{\mathbb{R}}\int_{\mathcal{Y}}\int_{\mathcal{X}}\int_{\mathbb{R}}(\mathbb{I}[f(x, y) \leq 0] - \mathbb{I}[\tilde{f}_{n+1}(x, y) \leq 0])^2 dP_{\tilde{f}_{n+1}(x,y)}(\tilde{f}_{n+1}(x, y))dP_X(x)dydP_{g_{n+1}}(g_{n+1}) \\
&= \int_{\mathbb{R}}\int_{\mathcal{Y}}\int_{\mathcal{X}}\Phi\left(-\frac{|\mu_{n+1}(x, y)|}{\sigma_{n+1}(x, y)}\right)dP_X(x)dydP_{g_{n+1}}(g_{n+1}),
\end{aligned}$$

where the first equality follows from the definition of $\widetilde{S}_{\text{ideal}}$, $F(y)$, and $\widetilde{F}_{n+1}(y)$; the first inequality follows from Cauchy–Schwarz; the second equality follows from taking the expectation over the GPM $\tilde{f}_{n+1}$ whose set of all realizations is denoted by $\mathcal{F}$ and its $n + 1$-st data point value $g_{n+1} \coloneqq f_n(x_{n+1}, y_{n+1})$; and the final equality follows from the interpretation of the most inner integral as the integrated misclassification probability $\int_{\mathbb{R}}(\mathbb{I}[f(x, y) \leq 0] - \mathbb{I}[\tilde{f}_{n+1}(x, y) \leq 0])^2 dP_{\tilde{f}_{n+1}(x,y)}(\tilde{f}_{n+1}(x, y)) = P[\text{sign}(f(x, y)) \neq \text{sign}(\tilde{f}_{n+1}(x, y)] = \Phi\left(-\frac{|\tilde{\mu}_{n+1}(x,y)|}{\tilde{\sigma}_{n+1}(x,y)}\right)$, a result proven in greater detail in [26]. This bound is equivalent to the SUR strategy developed by Bect et al. [25].

Definition 3 (Step-wise uncertainty reduction (SUR)) *The acquisition function for SUR sampling is given by*

$$S_{\text{SUR}}(x_{n+1}, y_{n+1}) = -\int_{\mathbb{R}}\int_{\mathcal{Y}}\int_{\mathcal{X}}\Phi\left(-\frac{|\mu_{n+1}(x, y)|}{\sigma_{n+1}(x, y)}\right)dP_X(x)dydP_{g_{n+1}}(g_{n+1}) \tag{11}$$

where $g_{n+1} \coloneqq f_n(x_{n+1}, y_{n+1})$ is the $n + 1$-st data point used in the construction of $\tilde{f}_{n+1}$ that yields $\tilde{\mu}_{n+1}$ and $\tilde{\sigma}_{n+1}$, and the sign has been flipped such that the maximum of S_{SUR} selects the minimizer of the bound of S_{ideal}.

For each candidate point, the P-dimensional integral $\int_{\mathcal{Y}} \int_{\mathcal{X}} \cdot \, dP_X(x)dy$ is approximated using importance sampling [27], and the one-dimensional integral $\int_{\mathbb{R}} \cdot \, dP_{g_{n+1}}(g_{n+1})$ is approximated using Gauss–Hermite quadrature.

E. Gradient SUR

While the SUR acquisition function S_{SUR} (unlike S_{ideal}) is computable, it is still computationally expensive to evaluate as it requires integrating over a $P+1$-dimensional space, $\int_{\mathbb{R}} \int_{\mathcal{Y}} \int_{\mathcal{X}} \cdot \, dP_X(x)dydP_{g_{n+1}}(g_{n+1})$. To further reduce the computational cost, we may neglect the integration over $\mathcal{X} \times \mathcal{Y}$ and analyze the local(ized) reduction in the misclassification error at the candidate point (x_{n+1}, y_{n+1}). This modification is proposed in [24] using the knowledge gradient strategy developed in [28].

Definition 4 (Gradient step-wise uncertainty reduction (gSUR)) *The acquisition function for the gSUR strategy is given by*

$$S_{\text{gSUR}}(x_{n+1}, y_{n+1}) = \left(\Phi\left(-\left| \frac{\mu_n(x_{n+1}, y_{n+1})}{\sigma_n(x_{n+1}, y_{n+1})} \right| \right) - \int_{\mathbb{R}} \Phi\left(-\left| \frac{\tilde{\mu}_{n+1}(x_{n+1}, y_{n+1})}{\tilde{\sigma}_{n+1}(x_{n+1}, y_{n+1})} \right| \right) dP_{g_{n+1}}(g_{n+1}) \right) \rho_X(x_{n+1}), \tag{12}$$

where $g_{n+1} := f_n(x_{n+1}, y_{n+1})$ is the $n+1$-st data point used in the construction of $\tilde{f}_{n+1}$ that yields $\tilde{\mu}_{n+1}$ and $\tilde{\sigma}_{n+1}$.

F. Local misclassification probability

These misclassification-based strategies can be further localized by directly using the probability itself as an acquisition function, selecting the candidate point z_{n+1} with the highest probability of misclassification. This behaves similarly to the heuristic proposed by Echard et al. [29] that balances between exploration and exploitation about the contour,

$$z_{n+1} = \arg\max_{z \in \mathcal{Z}} -\frac{|\mu_n(z)|}{\sigma_n(z)}, \tag{13}$$

though also weighed by the input distribution such that the score reflects the point most likely to be classified erroneously. We refer to this max-misclassification strategy as S_{MEE}, after the maximization of empirical error [24]:

Definition 5 (Maximum misclassification (S_{MEE})) *The acquisition function for the maximum misclassification metric is given by*

$$S_{\text{MEE}}(x_{n+1}, y_{n+1}) := \Phi\left(-\left| \frac{\mu_n(x_{n+1}, y_{n+1})}{\sigma_n(x_{n+1}, y_{n+1})} \right| \right) \rho_X(x_{n+1}). \tag{14}$$

We note that the MEE metric suffers from ambiguity in selection in the limit of infinite candidate points, i.e. densely sampling the parameter space: the cumulative distribution function in S_{MEE} always evaluates to 0.5 for candidate points directly on the target contour. However, since in practice we use random finite sets of candidate points, this typically does not cause problems for the algorithm's mechanics.

G. Entropy reduction

Other sampling methods developed more recently focus on reducing the (information) entropy of the uncertain failure boundary predicted by the surrogate model. These include entropy-based contour location [30], and Contour Location Via Entropy Reduction (or CLoVER) [31], which has been used to accelerate the determination of the deterministic flutter boundary in [12]. Although the methods have been developed for use with models of multiple fidelities, we will present the single-fidelity version here for comparison with other sampling strategies.

The previous three methods reviewed (SUR, gSUR, MEE) focus on reducing probabilities of point misclassification by the surrogate model. The concept of entropy allows us to extend these methods further, providing a formal measure of uncertainty regarding the classifications. First, we must define the notion of *contour entropy* in the context of the failure boundary as estimated by our uncertain surrogate model.

Given a normal random variable $V \sim \mathcal{N}(\mu(V), \sigma(V)^2)$ (e.g., $V = f_n(z)$ or $V = \tilde{f}_{n+1}(z)$), we define a discrete random variable $W(V) \mapsto \{S(V), E(V), U(V)\}$ associated with three possible events:

- S(V): $V < -\epsilon$ (configuration is stable),
- E(V): $|V| < \epsilon$ (configuration is approximately in equilibrium), and
- U(V): $V > \epsilon$ (configuration is unstable),

where $\epsilon \in \mathbb{R}_{\geq 0}$ is a threshold parameter introduced in [31].

Contour entropy is a measure of the uncertainty in the classification of a stochastic function evaluation. Choosing $\epsilon = 0$ recovers the classification definitions used previously, but choosing $\epsilon > 0$ introduces a third possible event denoting approximate equilibrium and allows the tuning of exploration vs. exploitation; the choice of $\epsilon = 2\sigma(z)$ for σ associated with the stochastic function, used in [31], encourages exploration.

Definition 6 (Contour entropy) *As $V \sim \mathcal{N}(\mu(V), \sigma(V)^2)$, we can integrate the normal probability density function (cf. Figure 2) and obtain the probability of these events associated with $W(V)$:*

$$\begin{aligned} P(S(V)) &= \Phi\left(\frac{-\epsilon(V) - \mu(V)}{\sigma(V)}\right), \\ P(U(V)) &= \Phi\left(\frac{-\epsilon(V) + \mu(V)}{\sigma(V)}\right), \\ P(E(V)) &= 1 - P(S(V)) - P(U(V)). \end{aligned}$$

The entropy of the random variable $W(V)$ is then given by

$$H(V) = -P(S(V)) \ln P(S(V)) - P(E(V)) \ln P(E(V)) - P(U(V)) \ln P(U(V)). \tag{15}$$

Figure 3 shows the probability of misclassification, compared to the entropy using those classification probabilities.

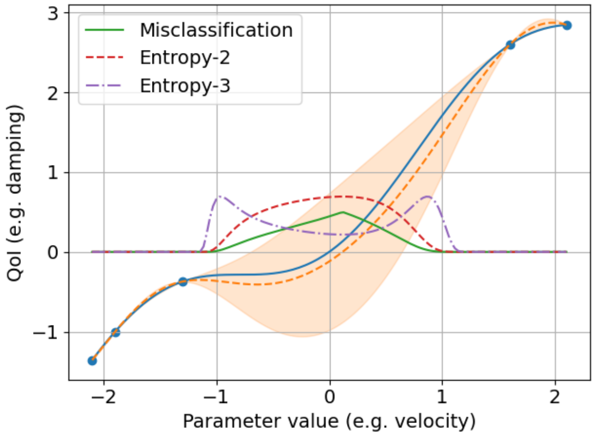


Fig. 3 Illustrative comparison of the misclassification probability and the entropy in classification. Entropy-2 refers to entropy purely for stable and unstable cases (i.e., $\epsilon(z) = 0$), while Entropy-3 includes the possibility of approximate equilibrium (i.e., $\epsilon(z) > 0$). The solid blue curve is a (one-dimensional) reference function f. The dashed orange curve and the shaded region are the mean μ_n and the 1σ region $[\mu_n - \sigma_n, \mu_n + \sigma_n]$ of the GPM f_n.

The CLoVER algorithm is designed to minimize the "global" entropy of f_n in each step. The integrated metric we use in this work is given by

$$\mathcal{H}(f_n(x, y)) \coloneqq \int_{\mathcal{Y}} \int_{\mathcal{X}} H(f_n(x, y))\, dP_X(x) dy, \tag{16}$$

where we have modified the original definition in [31] to account for the input parameter distribution. In the present single-fidelity setting, the point selected is therefore the one that reduces the entropy of the f_n-estimated flutter boundary the most in a one-step lookahead procedure.

Definition 7 (Contour Location Via Entropy Reduction (CLoVER)) *The acquisition function for CLoVER is given by*

$$S_{\text{CLoVER}}(x_{n+1}, y_{n+1}) \coloneqq \int_{\mathcal{Y}} \int_{\mathcal{X}} \left(H(f_n(x, y)) - \int_{\mathbb{R}} H(\tilde{f}_{n+1}(x, y)) dP_{g_{n+1}}(g_{n+1}) \right) dP_X(x)\, dy, \tag{17}$$

where $g_{n+1} \coloneqq f_n(x_{n+1}, y_{n+1})$ is the $n+1$-st data point used in the construction of $\tilde{f}_{n+1}$.

The integral $\int_{\mathbb{R}} \cdot\, dP_{g_{n+1}}(g_{n+1})$ can be evaluated in closed form:

$$\int_{\mathbb{R}} H(\tilde{f}_{n+1}(z)) dP_{g_{n+1}}(g_{n+1}) = \frac{\tilde{\sigma}_{n+1}(z)}{e\sigma_n(z)} \sum_{i=0}^{1} \sum_{j=0}^{1} \exp\left(-\frac{1}{2} \left(\frac{\mu_n(z) + (-1)^i \epsilon(z)}{\sigma_n(z)} + (-1)^j \frac{\tilde{\sigma}_{n+1}(z)}{\Phi(e^{-1})\sigma_n(z)} \right)^2 \right);$$

details of the approximation are provided in [31]. In this work, the integral $\int_{\mathcal{Y}} \int_{\mathcal{X}} \cdot\, dP_X(x) dy$ is approximated using importance sampling [27].

H. Gradient entropy reduction

In this section, we outline a strategy that reduces the contour entropy by comparing the current entropy at a candidate point and the expected entropy conditional on an additional sample at that point. The criterion is similar to CLoVER but considers primarily local information gains and omits the potentially expensive integral over the input space $\mathcal{X} \times \mathcal{Y}$:

Definition 8 (Gradient entropy reduction (GER)) *The acquisition function for the GER strategy is given by*

$$S_{\text{GER}}(x_{n+1}, y_{n+1}) = \left(H(f_n(x_{n+1}, y_{n+1})) - \int_{\mathbb{R}} H(\tilde{f}_{n+1}(x_{n+1}, y_{n+1})) dP_{g_{n+1}}(g_{n+1}) \right) \rho_X(x_{n+1}), \tag{18}$$

where $g_{n+1} \coloneqq f_n(x_{n+1}, y_{n+1})$ is the $n+1$-st data point used in the construction of $\tilde{f}_{n+1}$, and $H(\cdot)$ is the contour entropy.

To our knowledge, GER is a new sampling method for sequential design of experiments for failure boundary determination. The method can be regarded as a compromise between CLoVER's integrated metric and more local methods such as contour UCB, allowing for quicker prediction of information gain in a local context. In particular as we consider parameter domains reaching six dimensions or more, [30] shows that the integration schemes required by global methods begin to struggle for moderate sample sizes compared to local entropy metrics, which are still able to ensure improvement in a region about the sample point due to the regularity of GPs with appropriate kernels.

I. Local entropy

Lastly, we may also use the contour entropy itself as an acquisition function and select the maximizer of the entropy as z_{n+1}:

Definition 9 (Entropy-based contour locator (ECL)) *The acquisition function for the ECL strategy is given by*

$$S_{\text{ECL}}(x_{n+1}, y_{n+1}) = H(f_n(x_{n+1}, y_{n+1}))\rho_X(x_{n+1}). \tag{19}$$

In this work, we use the definition of contour entropy defined by Cole et al. [30], which results from choosing $\epsilon(z) = 0$, for the ECL strategy.

J. Summary of acquisition functions

Table 1 summarizes the eight acquisition functions discussed in Sections IV.B–IV.I. For ease of reading we classify the acquistion functions into three groups: straddle-based, misclassification-based, and entropy-based. All eight methods provide contour-targeting adaptive sequential construction of the surrogate model, compared to grid-based approaches used in, e.g., NIPC. Within the straddle family, the straddle and weighted-straddle are distinguished by whether the input uncertainty P_X is considered or not. The remaining six acquisition functions in the misclassification and entropy families are all countour-aware and input-uncertainty-aware (using the same score modification as weighted-straddle

	type	contour- aware	P_X- aware	one-step lookahead	global- aware	cost	section
straddle	straddle	✓	✗	✗	✗	low	IV.B
weighted-straddle		✓	✓	✗	✗	low	IV.C
SUR	misclassification	✓	✓	✓	✓	high	IV.D
gSUR		✓	✓	✓	✗	medium	IV.E
MEE		✓	✓	✗	✗	low	IV.F
CLoVER	entropy	✓	✓	✓	✓	high	IV.G
GER		✓	✓	✓	✗	medium	IV.H
ECL		✓	✓	✗	✗	low	IV.I

Table 1 Summary of acquisition functions.

if not inherently input-uncertainty aware). There are three further subclasses of acquisition functions we consider in our groupings: globally-aware one-step lookahead; local one-step lookahead, and local without a lookahead. This subgrouping provides a heirarchy of the computational cost (or complexity) of each method, which is typically small compared to the cost of a CAE model but does potentially limit integration accuracy in higher dimensions for global-aware methods [30]. Table 2 in Section VII.C provides the numerical evaluation time per candidate point of each of these methods observed in a typically flutter prediction case, which we leave in general terms in this section.

K. Batch-sequential sampling strategy

Sequential sampling methods are designed to select optimal sampling points in a greedy/sequential fashion. However, modern computers often allow parallel/simultaneous evaluation of f at multiple different points. This motivates batch sampling, where f is evaluated for multiple different points in each iteration, which may yield a less optimal sequence of points but at a lower wall-clock time. To this end, we propose a batch-sequential strategy Algorithm 2, which can be used in conjunction with any acquisition function discussed.

Although existing lookahead methods can be modified for simultaneous multi-step lookup given a candidate point batch [25, 32], the cost scales exponentially with N_p and is often computationally impractical. We propose the greedy batch selection algorithm as a compromise in such situations. We will see that accuracy of the batch selection remains comparable to sequential selection.

V. Application to a synthetic problem

A. Problem description

We now assess the GPM sampling methods using the six-dimensional Hartmann function [33] over $\mathcal{X} \times \mathcal{Y} \equiv \mathbb{R}^4 \times [0, 1]^2$ with the input parameter distribution $x_i \sim \mathcal{N}(0.5, 0.13^2)$ for $i = 1, \ldots, 4$. Each GPM sampling method is initialized with a dataset of size $n + 0 = 3$, chosen randomly in $\mathcal{X} \times \mathcal{Y}$ according to the distribution in x_i and uniformly

Algorithm 2: Batch-sequential sampling for GPM using an acquisition function S.

```
input: Acquisition function: $S : \mathbb{R}^P \to \mathbb{R}$
       Number of parallel executable processes: $N_p$
output: GPM $\tilde{f}_N$ based on dataset $\{(z_i, f(z_i))\}_{i=1}^{N}$
Initialize dataset $\{(z_i, f(z_i))\}_{i=1}^{n_0}$ and construct the GPM $f_{n_0} := f_{\mathrm{GP}}(\cdot; \{(z_i, f(z_i))\}_{i=1}^{n_0})$
while $n \leq N$ do
    Find $z_{n+1} = \arg\max_{z \in \mathcal{Z}} S(z)$
    for $i = n+1, \dots, \max\{n + N_p - 1, N - 1\}$ do
        Construct an approximate GPM
          $\tilde{\tilde{f}}_i := f_{\mathrm{GP}}(\cdot; \{(z_1, f(z_1)), \dots, (z_n, f(z_n)), (z_{n+1}, f_n(z_{n+1|})), \dots, (z_i, f_n(z_i))\}, k)$
        Find $z_{i+1} = \arg\max_{z \in \mathcal{Z}} \tilde{\tilde{S}}(z)$, where $\tilde{\tilde{S}}$ is based on $\tilde{\tilde{f}}_i$
    end
    Evaluate the full-order model at $\{z_i\}_{i=n+1}^{n+N_p}$ in parallel to obtain $\{f(z_i)\}_{i=1}^{n+N_p}$
    Update the GPM and retrain hyperparameters to obtain to $f_{n+1} := f_{\mathrm{GP}}(\cdot; \{(z_i, f(z_i))\}_{i=1}^{n+N_p}; k)$
    Set $n \leftarrow \max\{n + N_p, N\}$
end
```

in y. Then, in each iteration, the method selects the next point z_{n+1} from a candidate point set of 1000 scrambled Sobol points. Hyperparameters are updated each iteration. The integrals over $\int_{\mathcal{Y}} \int_{\mathcal{X}} \cdot \, dP_X(x) dy$ in SUR and CLoVER are approximated using importance sampling with 2500 points. The integral over $\int_{\mathbb{R}} \cdot \, dP_{g_{n+1}}(g_{n+1})$ in SUR and gSUR is approximated using five-point Gauss–Hermite quadrature. Due to the high computational cost of lookahead methods (i.e., SUR, gSUR, CLoVER, and GER), MEE is used as a preliminary filter to reduce the size of the candidate set to 500 for gSUR and GER and 100 for SUR and CLoVER.

In this work, we assess the accuracy of the sampling algorithms using the relative $L^2(\mathcal{Y})$ error of the flutter probability function $F : \mathcal{Y} \to \mathbb{R}$: i.e., $\int_{\mathcal{Y}} (F(y) - F_n(y))^2 \, dy / \int_{\mathcal{Y}} (F(y))^2 \, dy = \int_{\mathcal{Y}} \int_{\mathcal{X}} (\mathbb{I}[f(x, y) \leq 0] - \mathbb{I}[f_n(x, y) \leq 0])^2 dP_X(x) dy / \int_{\mathcal{Y}} \int_{\mathcal{X}} (\mathbb{I}[f(x, y) \leq 0])^2 dP_X(x) dy$, where we approximate the integral $\int_{\mathcal{Y}} \cdot \, dy$ using a Riemann sum with 25 points in each dimension and the integral $\int_{\mathcal{X}} \cdot \, dP_X(x)$ using MC with 10,000 points. To account for variability with respect to initial sample points, candidate point sets, and stochastic hyperparameter optimization, we run each design method 40 times for each test function and compare the statistics.

B. Results

Figures 4a and 4b summarize the error statistics in the solution obtained by all sequential design methods considered for two different full-order model evaluation sizes of 100 and 200. Given its popularity, we also provide NIPC as a reference; we use a degree-4 NIPC approximation with a fixed Gaussian quadrature sample size of 15,625. The order the strategies are presented corresponds to their groupings in Section IV.J and Table 1. First, we observe that all of these contour-targeting GPM-based methods resolve the probabilistic flutter boundary far more quickly than NIPC; these methods achieve a target median error of 0.02–0.1 using fewer than 100 function evaluations, whereas NIPC has an error of 0.16–0.17 using over 15,000 function evaluations. Second, we observe that all methods that take into account the input parameter distribution over $\mathcal{X}$ (i.e., all methods except the standard straddle) perform far better than the standard straddle heuristic which does not. Third, of these methods, we observe that local entropy-based metrics (i.e., GER and ECL) provide the most accurate results when the allocated budget is exhausted (i.e., for the evaluation size of 200), though the difference in performance is not large in relation to the next best probability-weighted methods such as SUR. Our results agree with Cole et al [30], which finds that ECL performs more reliably than CLoVER in cases where moderate- or high-dimensional integration over $\mathcal{X} \times \mathcal{Y}$ is a challenge.

Figures 4c and 4d summarize the error statistics for all sampling methods considered in this study using the proposed batch-sequential (instead of sequential) selection procedure Algorithm 2. The batch size is fixed to $N_p = 5$. For the chosen batch size, there is little, if any, drop in performance for each sampling strategy compared to sequential sampling in Figure 4a for the same number of total function evaluations, and additional evaluations show improvements for CLoVER, with smaller improvements for GER.

VI. Application to TAE model

A. Problem description

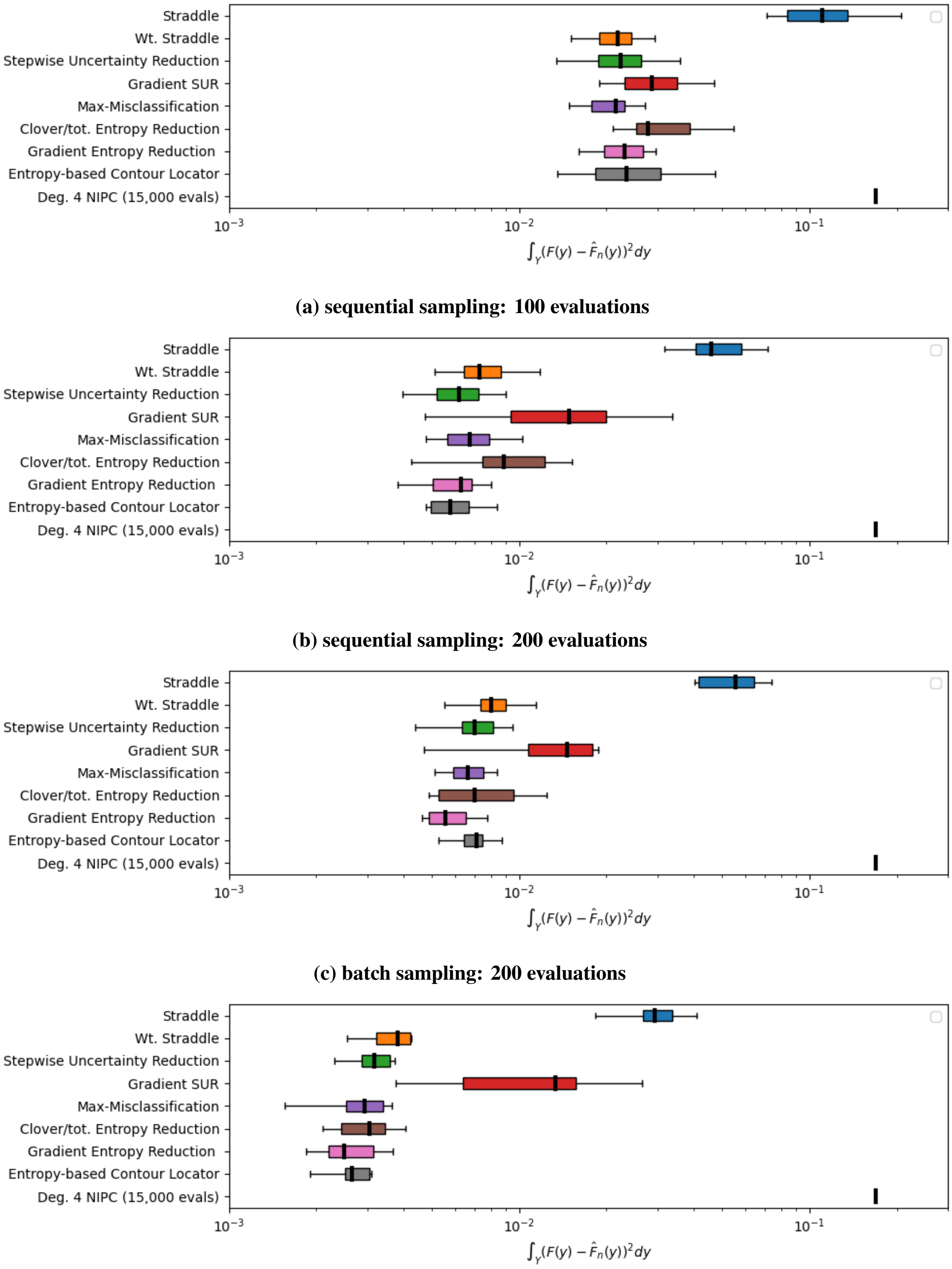
In this section, we apply sequential design methods to determine the flutter boundary of an airfoil experiencing pitch-and-plunge motion modeled by a "textbook" 2-DoF TAE model [34]. Figure 5 illustrates the two-dimensional aeroelastic system. The equations of motion for this system are

$$m(\ddot{h} + bx_\alpha \ddot{\alpha}) + m\omega_h^2 h = -L, \tag{20}$$

$$I_P \ddot{\alpha} + mbx_\alpha \ddot{h} + I_P \omega_\alpha^2 \alpha = M_{\frac{1}{4}} + b\left(\frac{1}{2} + a\right) L, \tag{21}$$

where $x_\alpha = e - a = R_x/b$ is the non-dimensionalized static imbalance between the elastic axis EA and center of gravity CG respectively, ω_α and ω_h are the uncoupled natural frequencies of the pitch and plunge modes, m is the airfoil section mass, and $I_P = I_{\text{EA}} + mb^2 x_\alpha^2$ for I_{EA} being the moment of inertia about point EA. The dynamics model is coupled with the unsteady aerodynamics model based on Theodorson's unsteady thin-airfoil theory [35]. Assuming simple harmonic motion in an incompressible flow, the lift and pitching moment for the airfoil are modeled by

$$\begin{aligned} L &= 2\pi\rho_\infty C(k)\left[\dot{h} + U\alpha + b\left(\frac{1}{2} - a\right)\dot{\alpha}\right] + \pi\rho_\infty b^2(\ddot{h} + U\dot{\alpha} - ba\ddot{\alpha}), \\ M_{\frac{1}{4}} &= -\pi\rho_\infty b^3\left[\frac{1}{2}\ddot{h} + U\dot{\alpha} + b\left(\frac{1}{8} - \frac{a}{2}\right)\ddot{\alpha}\right], \end{aligned}$$



(a) sequential sampling: 100 evaluations

(b) sequential sampling: 200 evaluations

(c) batch sampling: 200 evaluations

(d) batch sampling: 300 evaluations

Fig. 4 A summary of approximation errors for the six-dimensional Hartmann test function failure boundary obtained by each strategy with sequential and batch sampling with a batch size of 5. Each box indicates the four quartiles of errors observed over 40 stochastic runs. NIPC is provided as a reference, for a fixed evaluation size of 15,625.

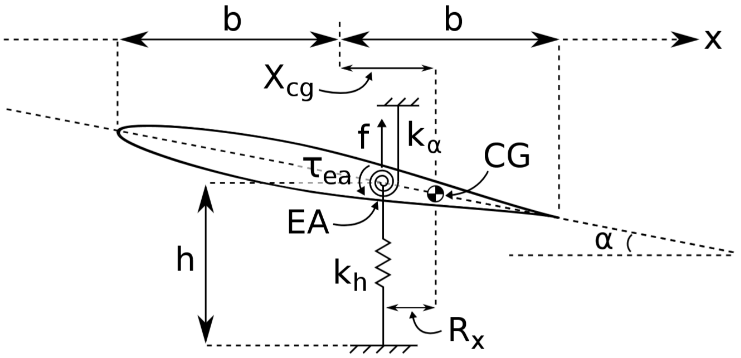


Fig. 5 Geometry of wing section with linear pitch and plunge restraints. (Source: [15]. Figure used with permission)

where ρ_∞ is the free stream density, U is the free stream velocity, and $C(k)$ is Theodorsen's function where $k := \frac{b\omega}{U}$ is the reduced frequency.

We use the p-k method [14] of flutter analysis to extract solution eigenvalues $p = \gamma k \pm ik$ for a given parameter configuration. The flutter determinant for the p-k method for this problem can be expressed in dimensionless form as

$$\begin{vmatrix} p^2 + \frac{\Omega_R^2}{V^2} - \frac{k^2}{M_R} + \frac{2ikC(k)}{M_R} & \frac{p^2 M_R x_\alpha + k(i+ak) + [2+ik(1-2a))]C(k)}{M_R} \\ \frac{p^2 M_R x_\alpha + ak^2 - ik(1+2a)C(k)}{M_R} & \frac{8M_R r^2(p^2+\frac{1}{V^2}) + 4i(1+2a)[2i-k(1-2a)]C(k) - k[k-4i+8a(i+ak)]}{8M_R} \end{vmatrix},$$

where $r^2 = \frac{I_P}{mb^2}$ is the the squared radius of gyration, $\Omega_R = \frac{\omega_h}{\omega_\alpha}$ is the frequency ratio, $M_R = \frac{m}{\rho_\infty \pi b^2}$ is the mass ratio, and $V = \frac{U}{b\omega_\alpha}$ is the reduced velocity. Our quantity of interest is thus the speed-independent representation for damping $\eta = -\frac{\Gamma}{\omega_\alpha} = -\Re(Vp)$ [34].

The following parameters and uncertainties are used in our analysis:

$$a = -1/5, \quad e = -1/10, \quad r^2 = \frac{6}{25} \times 10^{x_1}, \quad \Omega_R = \frac{2}{5} \times 10^{x_2}, \quad V = y_2, \quad M_R = 20 \times 10^{y_2}$$

for structural uncertainty $x \sim \mathcal{N}((0,0), \mathrm{diag}((0.06^2, 0.06^2)))$, and operational domain $\mathcal{Y} = [1.3, 3] \times [-0.2, 0.2]$.

B. Results

We apply the sequential design algorithms to the TAE model and test their efficacy using the same procedure applied to the test function in section V.A. Figures 6a and 6b summarize the error statistics for the GPM-based methods with adaptive sampling. We also provide a degree-4 NIPC approximation which uses 625 evaluations as a reference. Observations are similar to those made for the test function in Section V. First, we again observe that all probability-weighted GPM-based methods with adaptive sampling require significantly fewer TAE evaluations than NIPC to achieve a given error level; NIPC does not resolve the probabilistic flutter boundary to a normalized error of less than 10^{-2} using 625 function evaluations, while input probability weighted methods do achieve this target error with fewer than 100 samples. Second, the approaches that take into account the input parameter distribution over $\mathcal{X}$ perform far better than the standard straddle heuristic which does not; in fact, straddle with 100 samples is less accurate than the NIPC approximation (with 625 samples). Lastly, local entropy-based metrics ECL and GER achieve the highest accuracy at the end of the evaluation budget. Figures 6c and 6d show the errors given a batch sequential sampling strategy, with a batch size of 5. For this case as well, there is minimal if any loss in performance compared to purely sequential sampling for an equivalent number of function evaluations, with GER performing best by a marginal amount at the end of the new sampling budget.

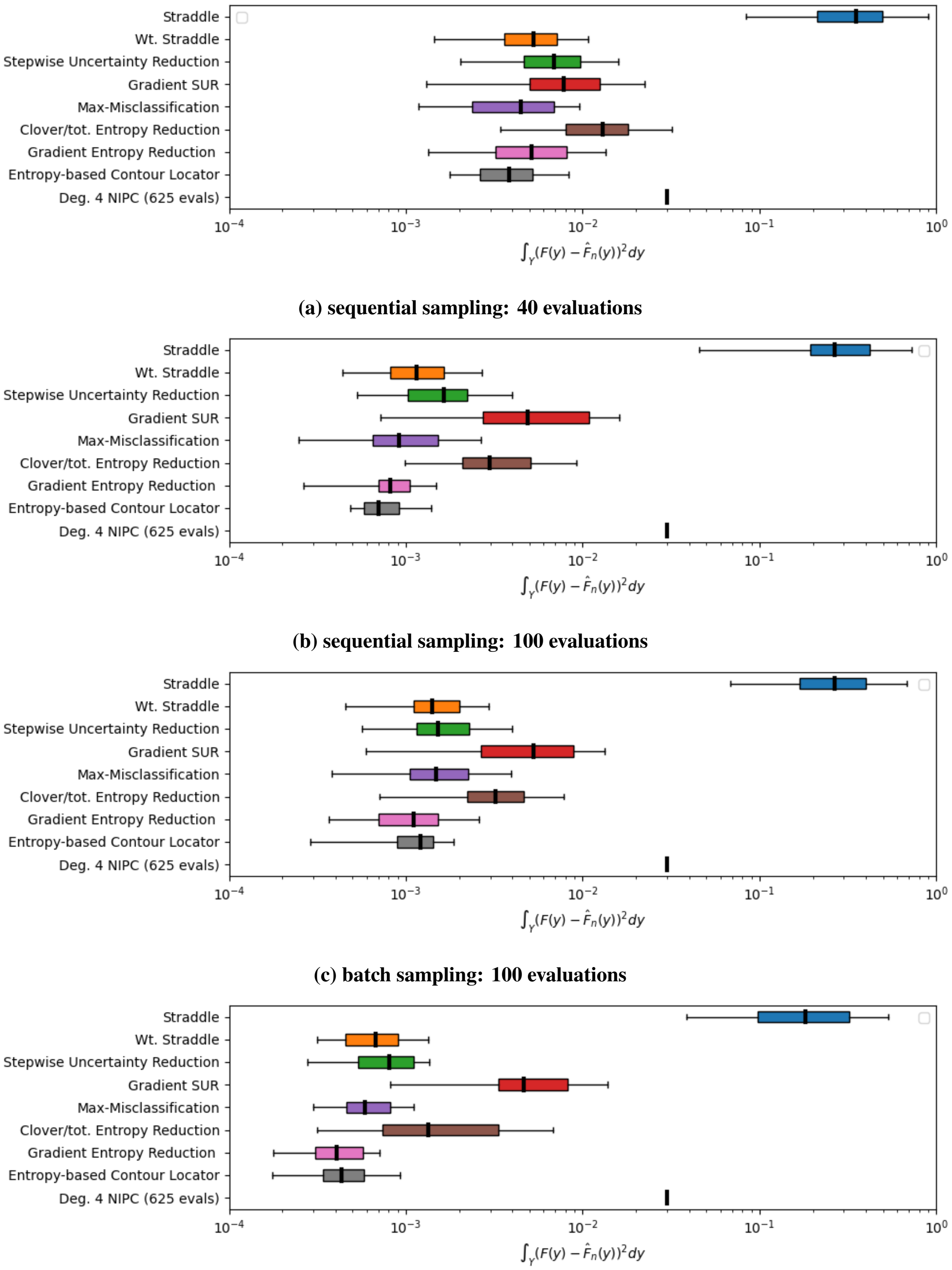
VII. Application to CAE model

A. Problem description

We now apply the sequential design methods to a two-dimensional CAE model. We consider the Isogai benchmark configuration [17], which features the NACA64A010 airfoil and a simple 2-DoF pitch-and-plunge motion, outlined in Section VI.A. The aerodynamic model is based on the Euler equations. We use a second-order finite volume method with 9313 cells to discretize in space, and second-order backward differentiation formula with 1256 time steps to discretize in time. More details of the solver are provided in [15].

The configuration parameters that define the space $\mathcal{Y}$ are the freestream Mach number $M_\infty \in [0.75, 0.9]$ and the reduced velocity $V \in [0.25, 2]$. Note that the parameter range corresponds to the region in the Mach–velocity space where we anticipate the so-called transonic dip [17]. The random structural parameters that define the space $\mathcal{X}$ are the distance between center of rotation and center of mass $R_x \sim \mathcal{N}(0.9, 0.09^2)$, the heave stiffness $k_h \sim \mathcal{N}(3.3 \times 10^5, (3.3 \times 10^4)^2)$, and the plunge damping coefficient $c_{\text{plunge}} \sim \text{Lognormal}(1.5, 0.47)$. For any given parameter $(x, y) \in \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^5$, the CAE solver yields time history of the flow field as well as the pitch and plunge data. We apply the matrix pencil analysis to the pitch and plunge data to estimate the damping parameter, which we obtain from the first mode. We then filter the damping parameter using the hyperbolic tangent function to suppress the effect of outliers (particularly those associated with the rapid initial growth of LCOs) as well as to ensure the model is most accurate near the flutter boundary. This filtered aeroelastic damping parameter defines the output of the function $f(x, y)$ considered in the abstract framework introduced in Section II.B.

In practice the CAE model would be (directly) used to evaluate $f(x, y)$ for any parameter sample (x, y); however, in the present setting where we wish to assess eight different sequential sampling strategies using error statistics that require Monte Carlo methods to compute, the direct use of the CAE model to evaluate $f(x, y)$ would be too expensive. In order to make the assessment computationally tractable, we use a "truth" GP surrogate model in place of the true CAE model. We construct the "truth" GPM for the filtered aeroelastic damping using an unscrambled Sobol point set of



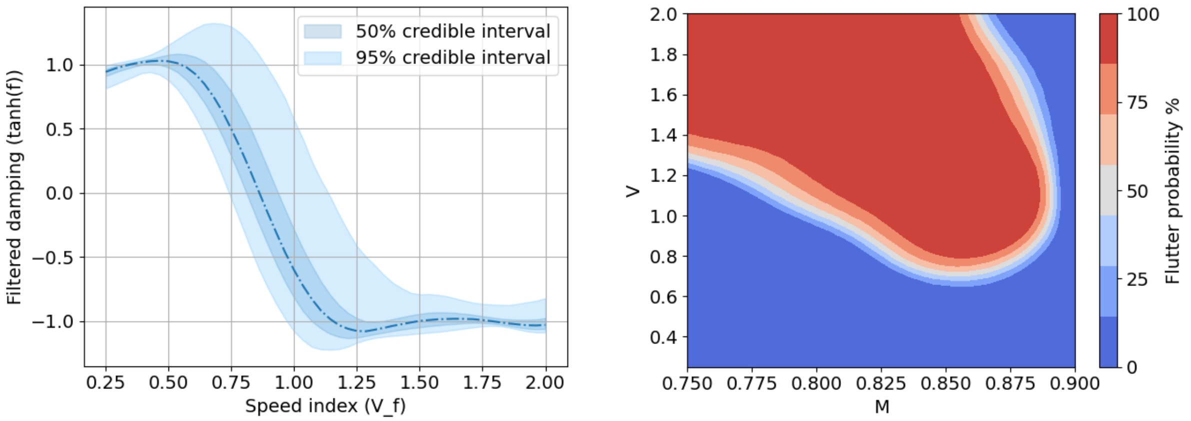
(a) sequential sampling: 40 evaluations

(b) sequential sampling: 100 evaluations

(c) batch sampling: 100 evaluations

(d) batch sampling: 200 evaluations

Fig. 6 A summary of approximation errors for the TAE flutter boundary obtained by each strategy with sequential and batch sampling with a batch size of 5. Each box indicates the four quartiles of errors observed over 40 stochastic runs. NIPC is provided as a reference, for a fixed sample size of 625.



(a) Damping (mode 1) for constant Mach number $M_\infty = 0.83$. Smaller and larger shaded areas correspond to the 50% and 95% credible regions, respectively.

(b) Probability of flutter throughout Mach number and velocity domain

Fig. 7 Probabilistic characterization of damping for failure mode used to create probabilistic flutter boundary

512 points, and 1649 points selected by the straddle heuristic with exploration parameter $\beta_t = 3$. The resulting dataset encourages both exploitation in accurately capturing the transonic dip and exploration in detecting the effects of high or low structural damping, which may result in stability or instabilities deep within regions of Mach–velocity space that are nominally unstable or stable, respectively. A validation using 10 randomly selected points near the flutter boundary suggest that the "truth" GPM has a median error of 0.0286 relative to the true CAE model.

Figure 7 illustrates the way the stochastic analysis extends the typical analysis of aeroelastic damping for flutter. Figure 7a shows the (filtered) aeroelastic damping coefficient as a function of speed index $V \in [0.25, 2]$ for a fixed Mach number of $M_\infty = 0.83$. The stochastic analysis provides not just the nominal damping coefficient curve, but also the credible region of all damping coefficient curves associated with variations in the structural and geometric variables. This damping curve can be constructed for any value of the freestream Mach number $M_\infty \in [0.75, 0.9]$. The information can be further compiled into the Figure 7b, which shows the probability of flutter over the entire Mach–velocity space under the structural and geometric uncertainties. This is the function $F : \mathcal{Y} \to \mathbb{R}$ given by (2), which we wish to characterize.

B. Results

We now assess the ability of various surrogate modeling strategies to efficiently capture the function F. We apply the sequential design algorithms to the CAE model and test their efficacy using the same procedure applied with the test function in Section V.A. Figures 8a and 8b summarize the error statistics for the GPM-based methods with adaptive sampling. We also provide a degree-5 NIPC approximation which uses 3,027 function evaluations as a reference. Using 100 sequential samples, all adaptive GPM-based methods achieve a lower median error than NIPC (using 3,027 samples), which has an error of little less than 10^{-1}. At 250 sequential samples, all lookahead-based adaptive methods achieve a median error of under 10^{-2}. Here, we see a notable hierarchy in the relative errors that corresponds to the hierarchy of computational complexity noted in Table 1. The expensive global one-step lookahead methods (i.e., SUR and CLoVER), which are derived to be most closely related to the error in Section IV.D, achieve the lowest error at little over 10^{-3}. The local one-step lookahead methods (i.e., GER and gSUR), which make use of lower cost local one step lookaheads, have the next lowest errors. The remaining methods, which are local rather than lookahead, have the highest errors. Unlike the TAE case, it appears that the CAE dataset exhibits more non-local behavior and rewards greater exploration, which results in the global lookahead methods standing out. The stronger emphasis on exploration also explains the relatively better performance of unweighted straddle.

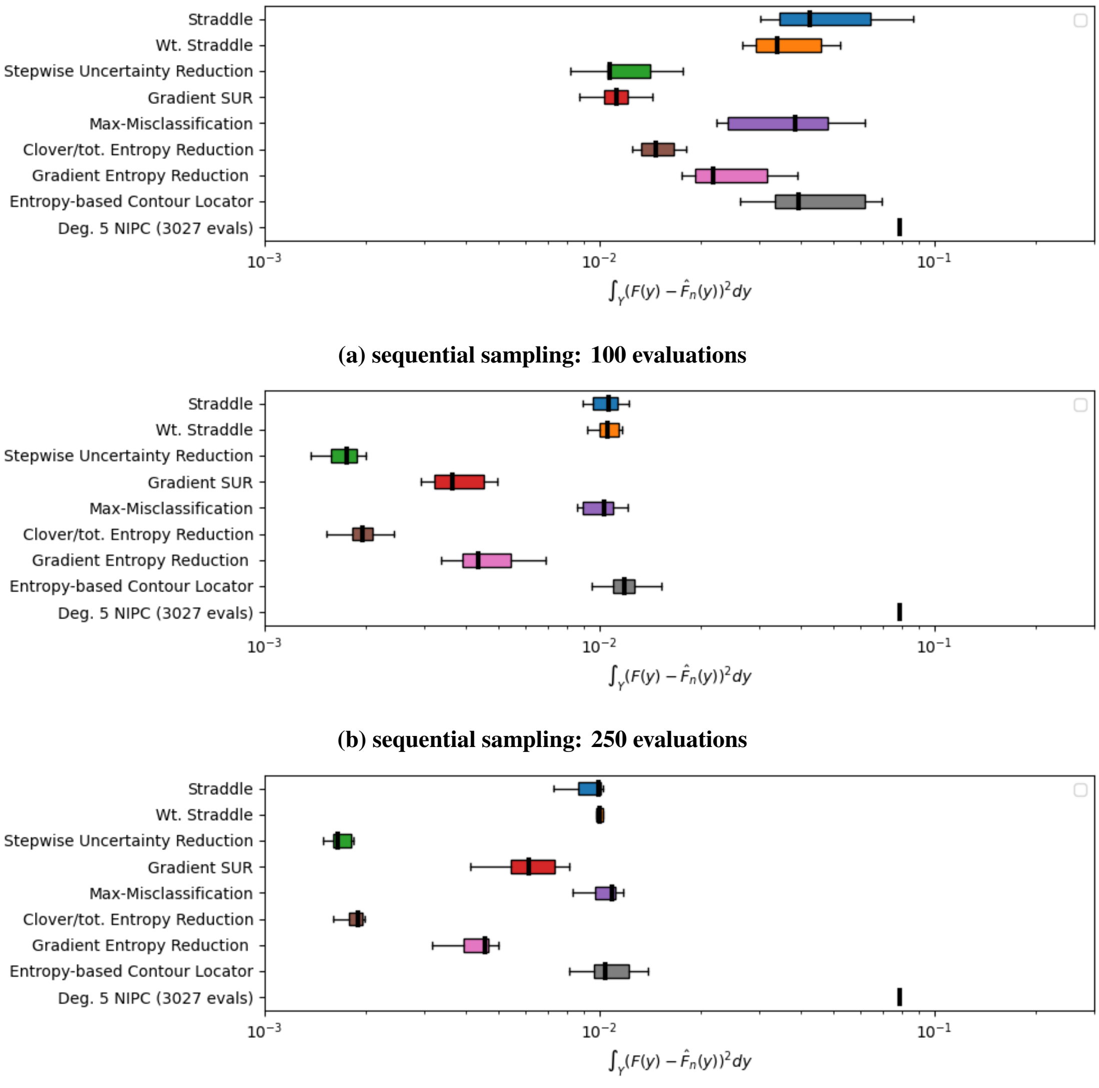
Figure 8c shows the error observed for the batch sample size of 250. Similarly to the Hartmann-6 and TAE cases, the batch sampling results are similar to the sequential sampling results. However, GER shows improvement over gSUR in this setting.

C. Computational cost

Table 2 compares the average wall clock time for 100 iterations of each sequential design strategy across each restart. We also report the time to evaluate the score of a single candidate point, as the number of candidates points are different across methods due to the use of the aforementioned filtering strategy. The non-lookahead methods (i.e., straddle and wt. straddle) are much cheaper than lookahead-based methods, although the surrogate models are not as accurate in comparison to lookahead methods. Of the lookahead methods, GER offers a reasonably cheap alternative to SUR and CLoVER. However, the total time taken to operate all these acquisition functions remains quite small compared to the time required for a single run of the CAE model used, which is on the order of 10^4 seconds.

Table 2 Evaluation time at iteration 100 of sequential design algorithms

Method	Total time (s)	Time per point (s)
Straddle	1.49	9.53×10^{-6}
Wt. Straddle	1.57	8.82×10^{-6}
SUR	152.27	2.44×10^{-2}
gSUR	159.17	7.15×10^{-3}
MEE	1.59	1.08×10^{-5}
CLoVER	117.77	2.08×10^{-2}
GER	110.87	5.4×10^{-3}
ECL	1.62	1.10×10^{-5}



(a) sequential sampling: 100 evaluations

(b) sequential sampling: 250 evaluations

(c) batch sampling: 250 evaluations

Fig. 8 A summary of approximation errors for the CAE flutter boundary obtained by each strategy with sequential and batch sampling with a batch size of 5. Each box indicates the four quartiles of errors observed over 40 stochastic runs. NIPC is provided as a reference, for a fixed sample size of 3,027.

VIII. Conclusions

We have (i) introduced a computationally efficient framework to provide a probabilistic estimate of the stochastic flutter boundary and (ii) derived relationships between existing GP sampling strategies and a reduction of error in the flutter boundary. This includes a new sampling method, GER, which offers a balance between local methods like contour UCB and globally integrated methods such as CLoVER to allow for faster prediction of information gain within a local context. We demonstrated the efficacy of adaptive GPM-based methods relative to the commonly used NIPC method. Based on the results, we recommend global one-step lookahead methods (e.g., SUR and CLoVER) for exploration-focused or low- to moderate-dimensional problems, and the local entropy methods (e.g., GER) for exploitation-focused or higher-dimensional problems.

Funding Sources

This work was supported by the National Research Council Canada under project monitor Dr. Viresh Wickramasinghe. Computations were performed on the Niagara supercomputer at the SciNet HPC Consortium. SciNet is funded by Innovation, Science and Economic Development Canada; the Digital Research Alliance of Canada; the Ontario Research Fund: Research Excellence; and the University of Toronto.

References

[1] Naval Air System Command, *Airplane strength and rigidity–vibration, flutter, and divergence*, Dept. of Defense, 1993.

[2] Elgas, D., *AC 25.629-1C - Aeroelastic stability substantiation of transport category airplanes*, Federal Aviation Administration, 2024. URL `https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentID/1043057`.

[3] Beran, P., Stanford, B., and Schrock, C., "Uncertainty quantification in aeroelasticity," *Annual Review of Fluid Mechanics*, Vol. 49, No. 1, 2017, pp. 361–386. https://doi.org/10.1146/annurev-fluid-122414-034441, URL https://doi.org/10.1146/annurev-fluid-122414-034441.

[4] Allen, M., and Camberos, J., "Comparison of uncertainty propagation / response surface techniques for two aeroelastic systems," *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA, 2009. https://doi.org/10.2514/6.2009-2269.

[5] Pettit, C. L., "Uncertainty quantification in aeroelasticity: recent results and research challenges," *Journal of Aircraft*, Vol. 41, No. 5, 2004, pp. 1217–1229. https://doi.org/10.2514/1.3961, URL https://doi.org/10.2514/1.3961.

[6] Dai, Y., and Yang, C., "Methods and advances in the study of aeroelasticity with uncertainties," *Chinese Journal of Aeronautics*, Vol. 27, No. 3, 2014, pp. 461–474. https://doi.org/https://doi.org/10.1016/j.cja.2014.04.016, URL https://www.sciencedirect.com/science/article/pii/S1000936114000867.

[7] Hosder, S., Walters, R., and Perez, R., "A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations," *44th AIAA Aerospace Sciences Meeting and Exhibit*, Vol. 891, AIAA, 2006. https://doi.org/10.2514/6.2006-891.

[8] Hosder, S., Walters, R., and Balch, M., "Efficient uncertainty quantification applied to the aeroelastic analysis of a transonic wing," *46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, 2008. https://doi.org/10.2514/6.2008-729.

[9] Hantrais-Gervois, J.-L., and Savin, E., "Sparse polynomial surrogates for non-intrusive, high-dimensional uncertainty quantification of aeroelastic computations," *Probabilistic Engineering Mechanics*, Vol. 59, 2020, p. 103027. https://doi.org/https://doi.org/10.1016/j.probengmech.2020.103027.

[10] Scarth, C., Cooper, J. E., Weaver, P. M., and Silva, G. H., "Uncertainty quantification of aeroelastic stability of composite plate wings using lamination parameters," *Composite Structures*, Vol. 116, No. Complete, 2014, pp. 84–93. https://doi.org/10.1016/j.compstruct.2014.05.007.

[11] Debusschere, B., Najm, H., Pébay, P., Knio, O., Ghanem, R., and Le Maître, O., "Numerical challenges in the use of polynomial chaos representations for stochastic processes," *SIAM Journal on Scientific Computing*, Vol. 26, 2004, pp. 698–719. https://doi.org/10.1137/S1064827503427741.

[12] Marques, A., Opgenoord, M., Lam, R., Chaudhuri, A., and Willcox, K., "Multifidelity method for locating aeroelastic flutter boundaries," *AIAA Journal*, Vol. 58, 2020, pp. 1–13. https://doi.org/10.2514/1.J058663.

[13] Goizueta, N., Wynn, A., and Palacios, R., "Fast flutter evaluation of very flexible wing using interpolation on an optimal training dataset," *AIAA SCITECH 2022 Forum*, AIAA, 2022. https://doi.org/10.2514/6.2022-1345, URL https://arc.aiaa.org/doi/abs/10.2514/6.2022-1345.

[14] Hassig, H. J., "An approximate true damping solution of the flutter equation by determinant iteration." *Journal of Aircraft*, Vol. 8, No. 11, 1971, pp. 885–889. https://doi.org/10.2514/3.44311, URL https://doi.org/10.2514/3.44311.

[15] Fereidooni, A., Grewal, A., Seraj, S., and Grzeszczyk, M., "Computational aeroelastic analysis using an enhanced openFOAM-based CFD-CSD solver," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA, 2018. https://doi.org/10.2514/6.2018-1955.

[16] Jacobson, K. E., Kiviaho, J. F., Kennedy, G. J., and Smith, M. J., "Evaluation of time-domain damping identification methods for flutter-constrained optimization," *Journal of Fluids and Structures*, Vol. 87, 2019, pp. 174–188. https://doi.org/https://doi.org/10.1016/j.jfluidstructs.2019.03.011, URL https://www.sciencedirect.com/science/article/pii/S0889974618308272.

[17] Isogai, K., "On the transonic-dip mechanism of flutter of a sweptback wing," *AIAA Journal*, Vol. 17, No. 7, 1979, pp. 793–795. https://doi.org/10.2514/3.61226, URL https://doi.org/10.2514/3.61226.

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

[19] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.

[20] Bryan, B., Schneider, J., Nichol, R., Miller, C., Genovese, C., and Wasserman, L., "Active learning for identifying function threshold boundaries." *Advances in Neural Information Processing Systems*, Vol. 18, MIT Press, 2005.

[21] Shekhar, S., and Javidi, T., "Multiscale Gaussian process level set estimation," *International Conference on Artificial Intelligence and Statistics*, 2019. URL https://api.semanticscholar.org/CorpusID:67856448.

[22] Bogunovic, I., Scarlett, J., Krause, A., and Cevher, V., "Truncated variance reduction: a unified approach to Bayesian optimization and level-set estimation," *Advances in Neural Information Processing Systems*, Vol. 29, edited by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/ce78d1da254c0843eb23951ae077ff5f-Paper.pdf.

[23] Senadeera, M., Rana, S., Gupta, S., and Venkatesh, S., "Level set estimation with search space warping," *Advances in Knowledge Discovery and Data Mining*, edited by H. W. Lauw, R. C.-W. Wong, A. Ntoulas, E.-P. Lim, S.-K. Ng, and S. J. Pan, Springer International Publishing, Cham, 2020, pp. 827–839.

[24] Lyu, X., Binois, M., and Ludkovski, M., "Evaluating Gaussian process metamodels and sequential designs for noisy level set estimation," *Statistics and Computing*, Vol. 31, No. 4, 2021, p. 43. https://doi.org/10.1007/s11222-021-10014-w, URL https://doi.org/10.1007/s11222-021-10014-w.

[25] Bect, J., Ginsbourger, D., Li, L., Picheny, V., and Vazquez, E., "Sequential design of computer experiments for the estimation of a probability of failure," *Statistics and Computing*, Vol. 22, No. 3, 2012, pp. 773–793. https://doi.org/10.1007/s11222-011-9241-4, URL https://doi.org/10.1007/s11222-011-9241-4.

[26] Vazquez, E., and Martinez, M. P., "Estimation of the volume of an excursion set of a Gaussian process using intrinsic Kriging," *arXiv: Statistics Theory*, 2006.

[27] Tokdar, S. T., and Kass, R. E., "Importance sampling: a review," *WIREs Computational Statistics*, Vol. 2, No. 1, 2010, pp. 54–60. https://doi.org/https://doi.org/10.1002/wics.56, URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.56.

[28] Frazier, P. I., Powell, W. B., and Dayanik, S., "A knowledge-gradient policy for sequential information collection," *SIAM Journal on Control and Optimization*, Vol. 47, No. 5, 2008, pp. 2410–2439. https://doi.org/10.1137/070693424, URL https://doi.org/10.1137/070693424.

[29] Echard, B., Gayton, N., and Lemaire, M., "AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation," *Structural Safety*, Vol. 33, No. 2, 2011, pp. 145–154. https://doi.org/https://doi.org/10.1016/j.strusafe.2011.01.002, URL https://www.sciencedirect.com/science/article/pii/S0167473011000038.

[30] Cole, D. A., Gramacy, R. B., Warner, J. E., Bomarito, G. F., Leser, P. E., and Leser, W. P., "Entropy-based adaptive design for contour finding and estimating reliability," *Journal of Quality Technology*, Vol. 55, No. 1, 2023, pp. 43–60. https://doi.org/10.1080/00224065.2022.2053795.

[31] Marques, A., Lam, R., and Willcox, K., "Contour location via entropy reduction leveraging multiple information sources," *Advances in Neural Information Processing Systems*, Vol. 31, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/01a0683665f38d8e5e567b3b15ca98bf-Paper.pdf.

[32] Zhang, B., Gramacy, R. B., Johnson, L. R., Rose, K. A., and Smith, E., "Batch-sequential design and heteroskedastic surrogate modeling for delta smelt conservation," *The Annals of Applied Statistics*, Vol. 16, No. 2, 2022, pp. 816 – 842. https://doi.org/10.1214/21-AOAS1521, URL https://doi.org/10.1214/21-AOAS1521.

[33] Picheny, V., Wagner, T., and Ginsbourger, D., "A benchmark of kriging-based infill criteria for noisy optimization," *Structural and Multidisciplinary Optimization*, Vol. 48, No. 3, 2013, pp. 607–626. https://doi.org/10.1007/s00158-013-0919-4, URL https://doi.org/10.1007/s00158-013-0919-4.

[34] Hodges, D. H., and Pierce, G. A., *Introduction to Structural Dynamics and Aeroelasticity*, 2nd ed., Cambridge Aerospace Series, Cambridge University Press, 2011. https://doi.org/10.1017/CBO9780511997112.

[35] Bisplinghoff, R. L., Ashley, H., and Halfman, R. L., *Aeroelasticity*, Cambridge University Press, 1956. https://doi.org/10.1017/S0001924000122213.