Formal Safety Envelopes for Provably Accurate State Classification by Data-Driven Flight Models

Elkin Cruz-Camacho,* Ahmad Amer,† Fotis Kopsaftopoulos,‡ and Carlos A. Varela§
Rensselaer Polytechnic Institute, Troy, New York 12180

https://doi.org/10.2514/1.I011073

Aerospace systems are inherently stochastic and increasingly data-driven, thus hard to formally verify. Data-driven statistical models can be used to estimate the state and classify potentially anomalous conditions of aerospace systems from multiple heterogeneous sensors with high accuracy. In this paper, we consider the problem of precisely bounding the regions in the sensor input space of a stochastic system in which safe state classification can be formally proven. As an archetypal application, we consider a statistical model created to detect aerodynamic stall in a prototype wing retrofitted with piezoelectric sensors and used to generate data in a wind tunnel for different flight states. We formally define *safety envelopes* as regions parameterized by z and τ, to respectively capture how model-predictable observed sensor values are, and given these values, how likely the model's accurate state classification is. Safety envelopes are formalized in the Agda proof assistant, used to also generate formally verified runtime monitors for sensor data stream analyses in the Haskell programming language. We further propose a new metric for model classification quality, evaluate it on our wing prototype model, and compare it to the model restricted to two different fixed airspeeds, and enhanced to a continuous Gaussian process regression model. Safety envelopes are an important step in formally verifying precise probabilistic properties of data-driven models used in stochastic aerospace systems and could be used by advanced control algorithms to maintain these systems well within safe operation boundaries.

I. Introduction

AEROSPACE systems are increasingly being used in societal applications, from bringing packages to customers, surveying fields of crops, and monitoring wildfires and disaster areas, to urban and advanced air mobility. Yet, to be truly autonomous as needed in many of these applications, they lack self-awareness to enable self-diagnosis and self-healing. One path forward for aerospace systems to strengthen their resilience is to make them capable of sensing, reasoning, and reacting in real time, which requires advanced control and decision-making abilities [1]. This is to be aided by access to an unprecedented amount of real-time data from onboard sensors, from which the aeroelastic state, environmental conditions, and structural conditions of aerospace systems [2,3] can be derived. Smart aerospace systems will be capable of detecting aerodynamic conditions (e.g., stall or flutter) using data from a variety of sources including piezoelectric sensors placed on the wings of an aircraft, satellite information, and accurate models of their environment [1,4]. Dynamic data-driven applications systems [5] use these data to enhance aerodynamic models updating them in real time and using them to determine the aerodynamic performance of flight systems [6].

Because the failure of safety-critical aerospace systems can cause harm to human life, the environment, or property [7], it is imperative to guarantee the correct and safe behavior of every component in these systems. As models grow and become more complex, their input space dimensionality increases and it becomes a problem for model checking because the time necessary to guarantee any property becomes intractable: thus the need to use statistical and bounded model checking. Statistical model checking uses a simulation-based approach to reasoning about stochastic systems such as aerospace systems [8]. Krishnan and Lalithambika presented an example of how to use bounded model checking for the analysis of onboard computer code in the Promela language [9]. Statistical model checking can only prove probabilistic properties typically specified in a stochastic temporal logic. Bounded model checking is typically used to find counter-example traces that illustrate when a property does not hold, so it is inherently incomplete. Formal verification using theorem proving guarantees the correctness of a system, but only if the system can be fully described. There has been recent work in the formal verification community on complex statistical aerospace systems. The VeriDrone project [10] builds upon differential dynamic logic [11] to formally verify the properties of hybrid systems [12]. Abed et al. [13] formally verified the continuous dynamics that govern the behavior of uncrewed aerial vehicles, for which they formalized the differential equations and dynamics in higher-order logic (HOL). Cohen et al. [14] formally verified the ellipsoid method used for receding horizon control written in the C language. They modified the algorithm to prevent numerical instability.

We introduce safety envelopes as boundaries in the system's input space where we can formally verify parameterized probabilistic statements on the accuracy of state estimation and classification by data-driven models. In the same manner that flight performance envelopes define a region where it is safe for an aircraft to operate, safety envelopes define regions where a data-driven systems' classification is correct according to z-predictability and τ-confidence. Note that z-predictability[¶] formalizes the intuition that "the data is captured by the model"; and τ-confidence formalizes the intuition that "the state of the system can be accurately determined from the data." The goal of safety envelopes is to reduce type I and type II errors when estimating the classification of a value by defining clear boundaries for the state of a system, thus defining safety regions where the system should be constrained to operate. Safety envelopes can only guarantee behavior for stochastic systems that follow the underlying statistical assumptions on the data, e.g., Gaussian distributions. Special runtime programs called *monitors* [15] can analyze real-time data against a safety envelope and determine whether the system is in a distinctly safe state or whether an action should be taken to steer the system away from an unsafe state.

The contributions of this paper include the following:

1) The first contribution is the definition of safety envelopes as system input regions where probabilistic statements have been formally verified for a statistical data-driven model. Since safety

Received 3 November 2021; revision received 20 May 2022; accepted for publication 10 October 2022; published online XX epubMonth XXXX.

*Ph.D. Student, Department of Computer Science; cruzce@rpi.edu.
†Research Engineer, General Electric; ahmad.amer15@gmail.com.
‡Assistant Professor, Department of Mechanical, Aerospace, and Nuclear Engineering; kopsaf@rpi.edu.
§Full Professor, Department of Computer Science; varelc@rpi.edu.

¶z is from the z score in statistics.

envelopes are parameterized by z and τ, metrics for model coverage, accuracy, error, and quality, are also introduced.

2) The second contribution is formalization (i.e., specification and proof) of four probabilistic properties as well as monitor generation using the Agda proof assistant [16] and the Haskell programming language.

3) The third contribution is the application of safety envelopes to the safety-critical aviation problem of stall detection using data from piezoelectric sensors embedded on a wing's skin. Three classes of Gaussian-based models are considered: univariate, bivariate, and univariate extended with artificial data by Gaussian process regression models (GPRMs).

The rest of this paper is organized as follows. Section II introduces our data-driven flight model, including wing piezoelectric sensor experiments, data collection, and the preprocessing strategy. Section III defines the safety envelopes and presents quality metrics for choosing different parameters used in the safety envelopes. Section IV presents proofs, an example of how the proofs are formally verified in Agda, and the code generation of runnable code from the theory. Section V contains the evaluation of safety envelopes under the three different scenarios of univariate data, bivariate data, and GPRM-generated data for the problem of stall detection. Finally, Sec. VI discusses related work, and Sec. VII concludes the paper and includes potential future work.

II. Data-Driven Flight Model

The complete experimental assessment and evaluation of this work is based on a prototype composite uncrewed aerial vehicle wing with embedded sensing capabilities. The prototype wing was designed, constructed, and tested at Stanford University (Fig. 1); for a detailed presentation of the wing, see Refs. [2,3]. The wing design is based on the cambered SG6043 high lift-to-drag ratio airfoil with a 0.86 m span, 0.235 m chord, and an aspect ratio of 7.32. The wing was outfitted with 32 distributed piezoelectric lead zirconate titanate (PZT) sensors (PZT disk was 3.175 mm in diameter) and 24 strain gauges to measure its dynamic response. The prototype composite wing was tested in the open-loop low-turbulence wind-tunnel facility at Stanford University. A series of wind-tunnel experiments was conducted for various angles of attack (AOAs) and freestream velocities U_∞. For each AOA, spanning the range from 0 up to 18 deg with an incremental step of 1 deg, data were sequentially collected for all velocities within the range of 9 to 22 m/s (with a step of 1 m/s). The aforementioned procedure resulted in a grid of flight state datasets corresponding to 266 different experiments covering the complete range of the considered flight states. For each experiment, the vibration response was recorded at different locations on the wing via the embedded piezoelectric sensors (initial sampling frequency of $f_s = 1000$ Hz, and initial signal bandwidth of 0.1–500 Hz). The signals were recorded via a National Instruments X-series 6366 data acquisition module featuring eight 16-bit simultaneously sampled analog-to-digital channels. The initial signals were low-pass filtered (Chebyshev type II 12th order; cutoff frequency of 80 Hz) and subsampled to a resulting sampling frequency of $f_s = 200$ Hz.

To investigate the response of the wing under varying AOAs and airspeeds as well as to determine its flight state, a statistical signal-energy analysis was performed for the different sensors. The initial signal of 91 s ($N = 91{,}000$ samples) was split into signal windows of 1 s each. Figure 2 presents indicative piezoelectric signals under different airspeeds and angles of attack. Then, for each signal window, the mean value and the standard deviation of the signal energy (time integration of the squared signal V^2 within the time window) were estimated. The goal was to correlate the signal energy in the time

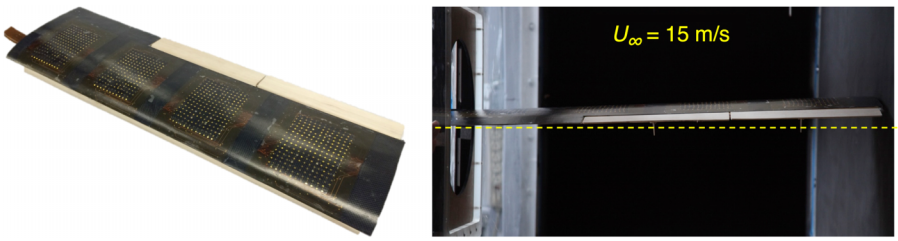

Fig. 1 The composite wing and the wind-tunnel setup used to collect data under different flight states.

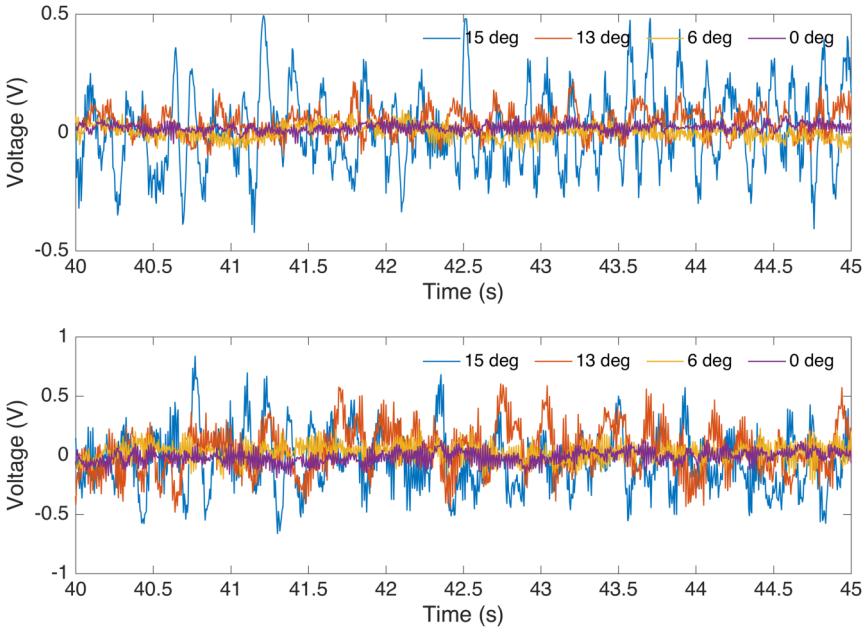

Fig. 2 Indicative signals obtained from a piezoelectric sensor under various angles of attack: a) freestream velocity of $U_\infty = 11$ m/s (top subplot), and b) freestream velocity of $U_\infty = 17$ m/s (bottom subplot).

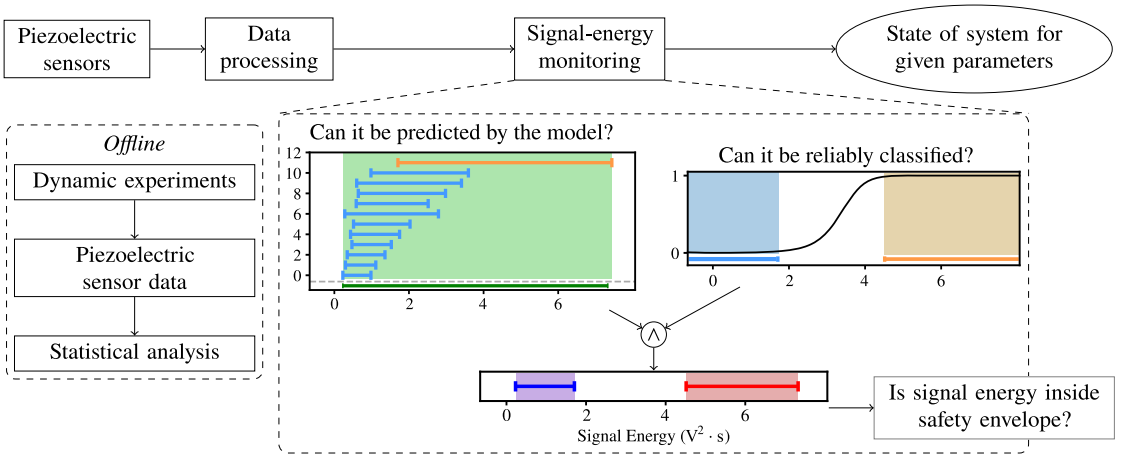

Fig. 3 Detection of stall (or its absence) using signal-energy safety envelopes.

domain with the airflow characteristics and aeroelastic properties in order to identify and track appropriate signal features that can be used for the subsequent stall detection of the wing under various flight states. Based on the results of this study [2,3], it was observed that the vibration data for all the considered states, under the aforementioned preprocessing, follow a normal distribution, and thus represent a single flight state with a normal distribution $\mathcal{N}(\mu_\theta, \sigma_\theta^2)$, where μ_θ corresponds to the mean and σ_θ corresponds to the standard deviation of the flight state θ. Therefore, it is possible to compute the multivariate joint normal distributions for the wing sensors under the considered flight states.

Under certain flight states at higher angles of attack, the lift of the wing would decrease below its weight, therefore leading to an aerodynamic stall. The ground truth for the different flight states (i.e., whether the wing exhibits stall or not) was obtained from a series of computational fluid dynamics (CFD) simulations for the same wing design and considered flight states, where the occurrence of stall or no stall was established (for details, please see Refs. [2,3]). In addition, during the experiments, the wing base was mounted on a load cell to measure the three forces and three moments (6 deg of freedom) acting on the wing. The load cell results were in agreement with the CFD-based analysis in indicating the loss of lift, and thus the stall of the wing, for the different flight states.

III. Safety Envelopes

Suppose a sensor fails midflight and, instead of sounding an alarm, the control system assumes that the sensor is producing accurate data. A human operator or logically redundant system [17] could catch such a mistake and reverse an undesirable action from the control system, but the risk of catastrophic failure is not out of the question; e.g., Air France 447, Tuninter 1153, and Boeing's 737 Max 8 accidents were initiated by such sensor failures [18]. Safety envelopes are dynamic regions of instrument measurements that can be considered correct. If a sensor failed and the data that it produced did not match a model, then the data would be outside of its safety envelope. In case the sensor is producing correct data, the question becomes whether these data should ring an alarm or be used passively by the control system. These scenarios are captured by the following two interlinked questions:

1) Is the data predictable by the model (z-predictability)?

2) What is the most probable state of the system given the data (τ-confidence)?

The goal of safety envelopes is to determine valid system input regions and their corresponding flight states, e.g., to determine whether a measurement from piezoelectric sensors corresponds to a stall state and to nothing else with very high probability. For this, the values are compared to what a model can sensibly generate (z-predictability) and from which state it is most likely produced (τ-confidence).

A. Univariate Safety Envelopes for Stall Detection

To exemplify safety envelopes, we present the case of stall detection using the signal energy of a single piezoelectric sensor. The input to signal-energy safety envelopes is a signal energy $x \in \mathbb{R}$ and its output is one of three classes: stall, no stall, or uncertain. Figure 3 shows how signal-energy safety envelopes can be used to detect stall in a live system. In the figure, z-predictability corresponds to the question "Can [the signal] be predicted by the model?" It is represented by the green-colored region. The τ-confidence corresponds to the question "Can [the signal] be reliably classified?" It is represented by the light blue and orange regions.

A signal-energy model M for stall detection consists of a triple $\langle \Theta_{\text{states}}, \{\mathcal{N}(\mu_\theta, \sigma_\theta^2)\}, C_{\text{stall}} \rangle$, where Θ_{states} are the possible states of the flight system (e.g., all states where angles of attack are natural numbers for an aircraft flying at 15 m/s); $\{\mathcal{N}(\mu_\theta, \sigma_\theta^2)\}$ are a family of Gaussian random variables for each state $\theta \in \Theta_{\text{states}}$ that encode the distributions of signal energy for each state; and $C_{\text{stall}} : \Theta_{\text{states}} \to \{\texttt{stall}, \texttt{no-stall}\}$ is a ground-truth tagging function that determines whether a given flight state is in stall or not. We assume every state is equally likely.

Note that z-predictability determines whether a signal could likely be generated by a model. A signal energy that is not likely to be generated by the model is said to be outside of the safety envelope. Assuming that the signal energy preprocessed from a piezoelectric sensor follows a normal distribution, the z-predictability is defined for a signal as follows:

Definition 1. Signal-energy z-predictability: Given a signal-energy model $M = \langle \Theta_{\text{states}}, \{\mathcal{N}(\mu_\theta, \sigma_\theta^2)\}, C_{\text{stall}} \rangle$, an energy signal x is z-predictable if, and only if, there exists a flight state $\theta \in \Theta_{\text{states}}$ in the model M such that

$$\mu_\theta - z\sigma_\theta < x < \mu_\theta + z\sigma_\theta$$

where μ_θ and σ_θ are the parameters of the normal distribution that describes the signal energy for the flight state θ.

The green regions in the top row of each plot in Fig. 4 illustrate the z-predictability given different airspeeds and z parameters.

Note that the τ-confidence determines from which state (*stall* or *no stall*) the signal energy was generated. If the state cannot be determined with enough confidence τ, then it is said that the signal energy is outside of the safety envelope (and tagged with uncertain). For this, we first define** a classification function based on a threshold parameter τ:

Helper definition 1. Signal-energy classification function: Given a signal-energy model $M = \langle \Theta_{\text{states}}, \{\mathcal{N}(\mu_\theta, \sigma_\theta^2)\}, C_{\text{stall}} \rangle$, an energy signal x can be classified in one of three categories as

**A helper definition is meant to be used for the scaffolding of important definitions (named simply "definition").

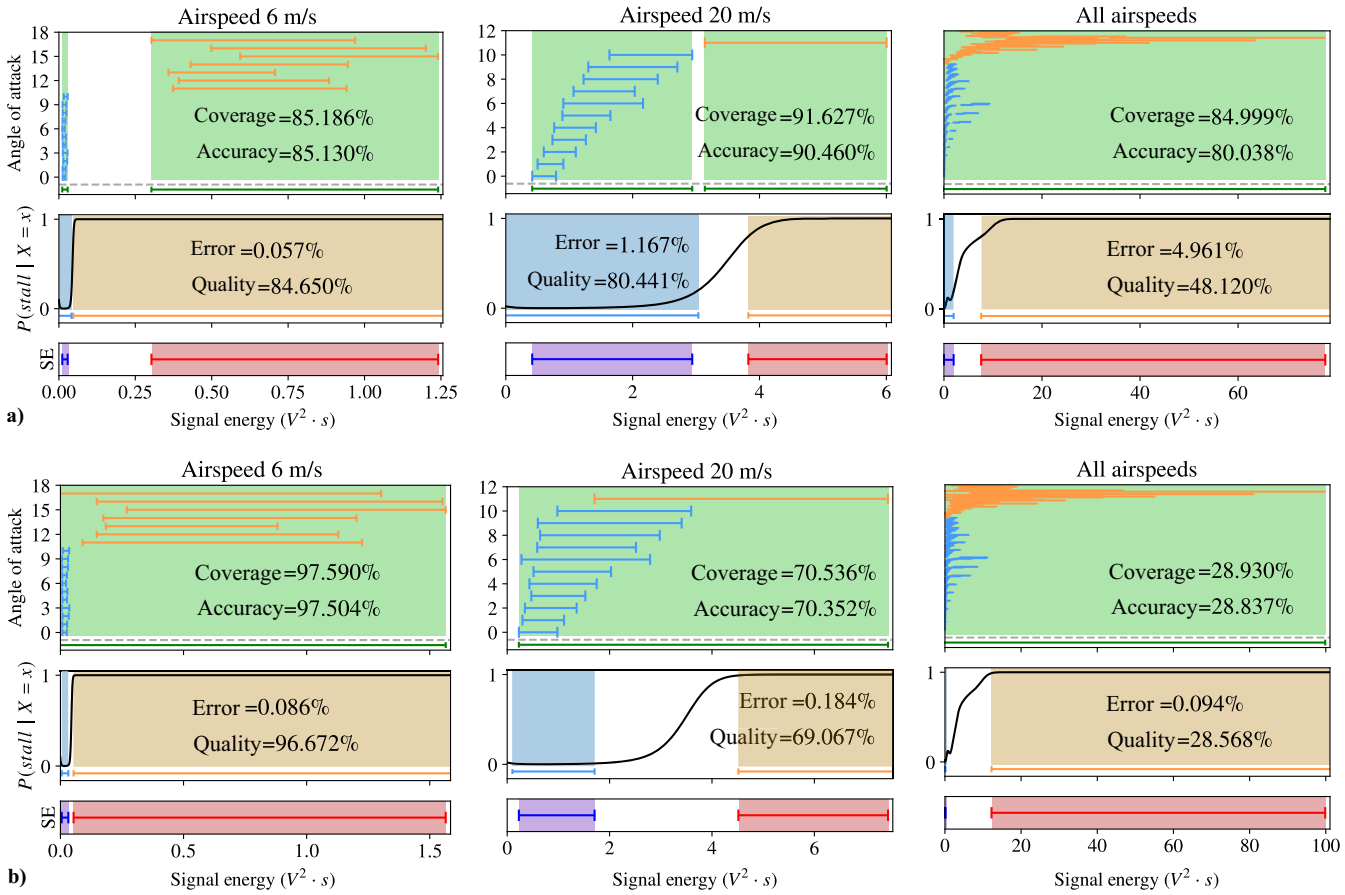

Fig. 4 Safety envelopes (bottom row) as the intersection of z-predictability (top row) and τ-confidence (middle row) for airspeeds of 6 and 20 m/s as well as all flight states: a) $z = 1$ and $\tau = 80\%$; and b) $z = 2$ and $\tau = 99\%$.

$$K_{\text{stall}}(M,\tau,x) = \begin{cases} \texttt{stall} & P(\texttt{stall}|X=x) \geq \tau \\ \texttt{nostall} & 1 - P(\texttt{stall}|X=x) \geq \tau \\ \texttt{uncertain} & \text{otherwise} \end{cases}$$

where X is the random variable for the energy signal, $\tau \in (0.5, 1]$, and $P(\text{stall}|X = x)$ is the conditional probability of stall given $X = x$.

The conditional probability of stall can be computed by the equation

$$\frac{\sum_{\theta\in\Theta} pdf_\theta(x) P(\texttt{stall} = \texttt{true}|\theta)}{\sum_{\theta\in\Theta} pdf_\theta(x)}$$

where $pdf_\theta(x)$ is the probability density function for the distribution $\mathcal{N}(\mu_\theta, \sigma_\theta)$, and the conditional probability $P(\texttt{stall}|\theta)$ is determined by the tagging function C_{stall} as one if $C_{\text{stall}}(\theta) = \texttt{stall}$ and zero otherwise.

The signature of K_{stall} is

$$M \times (0.5, 1] \times \mathbb{R} \to \{\texttt{stall}, \texttt{nostall}, \texttt{uncertain}\}$$

Note that τ is called the threshold of classification and indicates the level of confidence wanted from the classification or, alternatively, $1 - \tau$ indicates the risk associated with misclassification [19]. The conditional probability of stall is derived from Bayes's theorem. A step-by-step derivation can be found in the Appendix.

The solid, black curve in the middle row of each plot in Fig. 4 shows the probability of stall for each signal-energy value. The blue regions on the left sides correspond to the no-stall class, whereas the orange regions on the right sides correspond to the stall class; the unshaded regions in between correspond to the uncertain class. Shaded regions are the places where we are confident of the classification with τ certainty.

Definition 2. Signal-energy τ-confidence: Given a signal-energy model M and a signal energy $x \in \mathbb{R}$, a classification $K_{\text{stall}}(M, \tau, x) = k$ is called τ-confident if, and only if, $k \neq \texttt{uncertain}$.

Safety envelopes are the regions where a signal energy is both z-predictable and τ-confident, as exemplified in the following definition:

Definition 3. Signal-energy safety envelopes: Given a signal-energy model M, $z \in \mathbb{R}^+$, and $\tau \in (0.5, 1]$, a safety envelope $se(M, z, \tau)$ for stall detection is the region $X \in \mathcal{P}(\mathbb{R})$ where the following probabilistic statement holds: for all $x \in X$, x is z-predictable and $K_{\text{stall}}(M, \tau, x)$ is τ-confident.

The shaded regions on the bottom row of each plot in Fig. 4 are the safety envelopes (given the z and τ parameters) for the model derived by two different fixed airspeeds, and a third model is derived from all airspeeds.

Notice that the selection of parameters z and τ influences the size and range of the safety envelopes. A larger z increases the range of z-predictability, and thus data from a larger region of the signal-energy space are accepted. A very large z allows us to accept extremely rare events (outlier data), which include potentially unsafe data. A larger τ (closer to one) decreases the region defined by the τ-confidence. In general, the larger the safety envelopes, the weaker the formal properties associated to them; whereas the smaller the safety envelopes, the stronger the properties we can formally prove about them. The best parameters are application specific and dependent on the quality of the data and their ability to discriminate between flight classes. We present in Sec. III.C several metrics to determine the quality of data-driven models, and thus determine the best safety envelope parameters for a given application.

B. Generalized Safety Envelopes

The concepts that make up the signal-energy safety envelopes, z-predictability, and τ-confidence can be easily generalized to multiple-input data dimensions. We present one such generalization as models on multivariate-Gaussian distributions. This generalization allows us to use data from multiple correlated inputs such as the signal energy from multiple piezoelectric sensors as demonstrated in Sec. V.C. But first, we present some supporting definitions that serve as building blocks for a more rigorous definition of safety envelopes.

A collective-probability model contains all possible states a system can be in. Each state of the system is given by experimental data or by theory and follows a distribution. Each state is assumed to be independent of the others and has some non-zero probability of occurring. As in the case of stall detection, we are often not interested in determining the state of the system (flight state) but rather a condition associated with it (stall). For this, a state is associated with a condition via a tagging function.

Helper definition 2. Collective-probability model: A collective-probability model M is a tuple $\langle \Theta, \Xi, \{X_\theta\}, p_\Theta, L_\Theta, C_\Theta \rangle$ where 1) Θ is a finite set representing the possible states of the system; 2) Ξ is an arbitrary set representing the space of measurements from the system; 3) $\{X_\theta\}$ is a family of random variables, where $X_\theta : \Xi \to \mathbb{R}$ for $\theta \in \Theta$ (one per each possible state); 4) p_Θ is a probability density function, which represents the probability of the system being in a given state; 5) L_Θ is a set of labels, which correspond to the final output of the classification system; and 6) $C_\Theta : \Theta \to L_\Theta$ is the ground-truth tagging function.

The signal-energy model from Sec. III.A $\langle \Theta_{\text{states}}, \{\mathcal{N}(\mu_\theta, \sigma_\theta^2)\}, C_{\text{stall}} \rangle$ can be represented as the collective-probability model

$$M_{\text{stall}} = \langle \Theta_{\text{states}}, \mathbb{R}^+ \cup \{0\}, \{\mathcal{N}(\mu_\theta, \sigma_\theta^2)\}, p_{\text{states}}, \{\texttt{stall}, \texttt{no-stall}\}, C_{\text{stall}} \rangle$$

where 1) $\Theta = \Theta_{\text{states}}$ is a set of flight states (e.g., the flight states for angle of attack of 1 deg and airspeeds between 6 and 20 m/s), 2) $\Xi = \mathbb{R}^+ \cup \{0\}$ is the measurement space for the energy signal, 3) $\{X_\theta\} = \{\mathcal{N}(\mu_\theta, \sigma_\theta^2)\}$ is the family of Gaussian random variables that determine how the signal energy behaves at one given state, 4) $p_\Theta = p_{\text{states}} = (1/|\Theta_{\text{states}}|)$ is the uniform probability density function that encodes the probability of a state to occur, 5) $L_\Theta = \{\texttt{stall}, \texttt{no-stall}\}$ is the set of tags, and 6) $C_\Theta = C_{\text{stall}}(\theta)$ is the tagging function.

Safety envelopes are regions defined by a *probabilistic statement*, but what precisely does probabilistic statement mean?

Helper definition 3. Probabilistic statement: Given a collective-probability model $M = \langle \Theta, \Xi, \{X_\theta\}, p_\Theta, L_\Theta, C_\Theta \rangle$ and a parameter space Π, a probabilistic statement S over M is a predicate with parameters $\pi \in \Pi$ and $x \in \Xi$, i.e., $S : M \times \Pi \times \Xi \to \{\texttt{true}, \texttt{false}\}$.

Given a signal-energy model M_{stall}, let us define an example probabilistic statement S_{distinct} as

$$S_{\text{distinct}}(M_{\text{stall}}, \pi, x) = \exists \theta \in \Theta : P(\theta | X = x) \geq \pi \wedge \forall \theta \in \Theta : \theta' \neq \Theta \Rightarrow (P(\theta' | X = x) < \pi)$$

where $P(\theta | X = x)$ is the probability that the aircraft is in the state θ given a signal energy of x, and $\Pi = [0, 1]$ is the confidence threshold. S_{distinct} encodes the question of whether a signal energy x can be used to discriminate a unique flight state that generated it. For example, $S_{\text{distinct}}(M_{\text{stall}}, 0.99, 3.8)$ corresponds to the predicate of "Can the flight state that generated a signal of 3.8 be unequivocally determined with a certainty of 99%?".

Next, we define a "region of interest," which is the space under Ξ where a probabilistic statement is true: for example, the region where we can guarantee that the sensor produces adequate data (z-predictability).

Helper definition 4. Region of interest: Given a collective-probability model $M = \langle \Theta, \Xi, \{X_\theta\}, p_\Theta, L_\Theta, C_\Theta \rangle$, a parameter space Π, and a probabilistic statement S over M, a region of interest is the region in Ξ under which S holds with parameters π; i.e., a region of interest (RI) is the region defined by

$$RI(M, S, \pi) = \{x \in \Xi : S(M, \pi, x) = \texttt{true}\}$$

with one per tag.

The region of interest for S_{distinct} with parameter $\pi = 0.99$ is a subset of $\mathbb{R}$ for which $S_{\text{distinct}}(M_{\text{stall}}, 0.99, x)$ is true, i.e., $RI(M_{\text{stall}}, S_{\text{distinct}}, 0.99) \in \mathcal{P}(\mathbb{R})$.

Figure 5 presents two regions of interest for a collective-probability model with two states (Gaussian distributions). The region on the left (blue line at the bottom) corresponds to the probabilistic statement of "The value falls in the interval $(-1.6, 1)$." The region on the right (red line) corresponds to "The value falls in the interval $(1.5, 4.6)$." Notice that these example probabilistic statements lack any parameters.

With everything in place, let us define a multidimensional generalization for the signal-energy model from Sec. III.A:

Helper definition 5. Collective-Gaussian stall model: A collective-Gaussian stall model is the collective-probability model for stall classification

$$M_{\text{stall}} = \langle \Theta_{\text{states}}, \mathbb{R}, \{\mathcal{N}(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)\}, p_{\text{states}}, \{\texttt{stall}, \texttt{no-stall}\}, C_{\text{stall}} \rangle$$

where 1) $\Theta = \Theta_{\text{states}}$ is the set of flight configurations for which there are data, 2) $\Xi = \mathbb{R}$ is the energy signal space, 3) $X_\theta = \mathcal{N}(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)$ is a multivariate-normally distributed random variable for the flight configuration $\theta \in \Theta_{\text{states}}$, 4) $p_\Theta = p_{\text{states}}$ is the probability density function that determines the probability $p_{\text{states}}(\theta)$ for a flight configuration $\theta \in \Theta_{\text{states}}$ to occur, 5) $L_\Theta = \{\texttt{stall}, \texttt{no-stall}\}$ is the set of labels, and 6) $C_\Theta = C_{\text{stall}} : \Theta \to L$ is a tag function for each flight state.

Now, we can formally define the z-predictability, which encodes how well the data being given adjust to the (flight) model.

Definition 4. z-predictability: Given a collective-Gaussian stall model

$$M_{\text{stall}} = \langle \Theta_{\text{states}}, \mathbb{R}, \{\mathcal{N}(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)\}, p_{\text{states}}, \{\texttt{stall}, \texttt{no-stall}\}, C_{\text{stall}} \rangle$$

and a parameter $z \in \Pi = \mathbb{R}^+$, z-predictability is defined as the probabilistic statement:

$$S_{z\text{-pred}}(M_{\text{stall}}, z, \boldsymbol{x}) = \exists \theta \in \Theta_{\text{states}} : D_M(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta, \boldsymbol{x}) < z$$

where

$$D_M(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta, \boldsymbol{x})$$

corresponds to the Mahalanobis distance, and it is equal to

$$\sqrt{(\boldsymbol{\mu}_\theta - \boldsymbol{x})^T \boldsymbol{\Sigma}_\theta^{-1} (\boldsymbol{\mu}_\theta - \boldsymbol{x})}$$

The Mahalanobis distance for univariate Gaussian distributions reduces to the z-score region of the distribution, which is proven in Theorem 1.

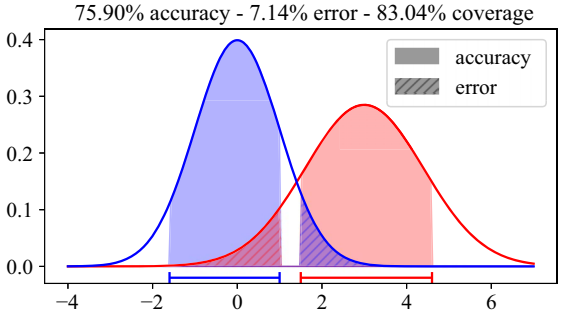

Fig. 5 Representation of accuracy, error, and coverage where the regions of interest are given by the blue and red intervals.

The soft green region on each plot in Fig. 4 shows the z-predictability region for three different models and two values of z. The first two models contain all flight states corresponding to airspeeds of 6 and 20 m/s, respectively, and angles of attack in the ranges of $\alpha \in [1, 18]$ and $\alpha \in [1, 12]$, respectively.

Next, we define a generalization for the τ-confidence using as input a collective-probability model. We introduce a τ (threshold) dependent classification function:

Helper definition 6. Conditional classification function with regard to a model: Given a collective-probability model $M = \langle \Theta, \Xi, \{X_\theta\}, p_\Theta, L_\Theta, C_\Theta \rangle$, the conditional classification function is defined as

$$K_{\text{cond}}(M, \tau, x) = \begin{cases} \texttt{tag} & P(\texttt{tag}|X = x) \geq \tau \\ \texttt{uncertain} & \text{otherwise} \end{cases}$$

where $\tau \in (0.5, 1]$, $x \in \Xi$, X is the random variable for the values measured on Ξ, and $P(\texttt{tag}|X = x)$ is the conditional probability for the class tag given x. The signature of K_{cond} is $M \times (0.5, 1] \times \Xi \to L_\Theta \cup \{\texttt{uncertain}\}$.

The conditional probability $P(\texttt{tag}|X = x)$ can be computed by the equation

$$\frac{\sum_{\theta \in \Theta} pdf_\theta(x) p_\Theta(\theta) P(\texttt{tag}|\theta)}{\sum_{\theta \in \Theta} pdf_\theta(x) p_\Theta(\theta)}$$

where $pdf_\theta(x)$ is the probability density function for the distribution X_θ, and the conditional probability $P(\texttt{tag}|\theta)$ is determined by the tagging function C_Θ as one if $C_\Theta(\theta) = \texttt{tag}$ and zero otherwise.

The probability of stall (for a univariate collective-Gaussian stall model) can be seen in the middle row of Figs. 4a and 4b. The black curve corresponds to the probability function $P(\texttt{stall}|X = x)$, which indicates the probability of the wing being in a stall condition given a single measurement of the signal energy. A derivation of $P(\texttt{stall}|X = x)$ can be found in the Appendix. The classification region can be seen at the bottom of the middle row in Fig. 4, for $\tau =$ 80 and 99%. The τ-confident region is the union of both colored regions (light blue and orange), where light blue indicates no stall and orange indicates stall.

We have all that is needed for a multivariate signal-energy safety envelope definition.

Definition 5. Given a collective-probability model M and a measurement $x \in \Xi$, a classification $K(M, \tau, x) = k$ is called τ-confident if, and only if, $k \neq \texttt{uncertain}$.

Safety envelopes encode two properties at the same time: whether a value is captured by a model or can be predicted by it, and whether it is likely correctly classified:

Definition 6. Safety envelopes: Given a collective-Gaussian stall model

$$M_{\text{stall}} = \langle \Theta_{\text{states}}, \mathbb{R}, \{\mathcal{N}(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)\}, p_{\text{states}}, \{\texttt{stall}, \texttt{no-stall}\}, C_{\text{stall}} \rangle$$

and $(z, \tau) \in \mathbb{R}^+ \times (0.5, 1]$, a safety envelope (SE) is the region of interest defined by the probabilistic statements:

1) The no-stall probabilistic statement: "A signal-energy value x is z-predictable, and the no-stall classification is τ-confident":

$$S_{\text{nostall}}(M_{\text{stall}}, (z, \tau), x) = S_{z-\text{pred}}(M_{\text{stall}}, z, x) \wedge K(M_{\text{stall}}, \tau, x) = \texttt{no-stall}$$

2) The stall probabilistic statement: "A signal-energy value x is z-predictable and the stall classification is τ-confident":

$$S_{\text{stall}}(M_{\text{stall}}, (z, \tau), x) = S_{z-\text{pred}}(M_{\text{stall}}, z, x) \wedge (K(M_{\text{stall}}, \tau, x) = \texttt{stall})$$

The last row of Figs. 4a and 4b show the safety envelopes derived from three different data-driven models with varying z scores and τ thresholds. For easily separable stall/no-stall conditions, such as 6 m/s, the safety envelope is the same as the region defined by z-predictability; in other cases, the region defined by the τ-confidence reduces the region described by z-predictability or vice versa. Notice that when safety envelopes are applied to a model where all airspeeds and AOAs have been taken into account, the safety envelopes become significantly smaller. This means that it is not possible to assert with high confidence whether a signal-energy value entails a stall condition. In the right column of Fig. 4b, the safety envelopes do not include any signal with values from around one until 12. In contrast, if we know the airspeed to be 6 m/s, a signal of one likely corresponds to a stall; whereas for an airspeed of 20 m/s, a signal of one likely corresponds to no stall.

Safety envelopes can be generalized along other dimensions such as using a sample of measurements instead of a single measurement; allowing the probability of a state to occur p_Θ to change depending on the input (which can be accomplished by defining a p_Θ as a prior, as well as using $P(\Theta|x)$ inside the τ-confidence instead); or replacing the assumption of normality by defining a custom τ-confident process. It is left to the designer of the model to determine the best z-predictability and τ-confidence definitions for their problem.

C. Metrics for Safety Envelopes

In this section, we explore a variety of metrics for safety envelopes that can be used to find the best parameters for a given model and to determine their quality. Metrics allow us to determine the qualities of different models, and thus compare them. A simple metric is to determine how many data points fall within a region of interest:

Definition 7. Coverage: Given a collective-probability model $M = \langle \Theta, \Xi, \{X_\theta\}, p_\Theta, L_\Theta, C_\Theta \rangle$, a parameter space Π, and a probabilistic statement S over M, coverage is the cumulative probability that falls within the region of interest; i.e.,

$$\text{coverage}(M, S, \pi) = \sum_{\theta \in \Theta} p_\Theta(\theta) P(x \in RI(M, S, \pi))$$

with parameters $\pi \in \Pi$.

The simplest useful model that can be expressed as a collective-probability model is one with two classes. Figure 5 shows an example model with two states and two tags with the following arguments:

$$M_{eg} = \langle \{\texttt{blue}, \texttt{red}\}, \mathbb{R}, \{X_{\texttt{blue}}, X_{\texttt{red}}\}, p_{eg}, \{\texttt{blue}, \texttt{red}\}, C_{eg} \rangle$$

where $X_{\texttt{blue}} = \mathcal{N}(0, 1^2)$, $X_{\texttt{red}} = \mathcal{N}(3, 1.4^2)$, $p_{eg}(x) = 1/2$ is the discrete uniform probability density function representing that either blue or red can occur with equal probability, and $C_{eg}(\theta) = \theta$ is the identity function. Two regions of interest are presented for two probabilistic statements:

$$S_{\texttt{blue}}(M, (y1, y2), x) = y1 < x < y2$$

and

$$S_{\texttt{red}}(M, (z1, z2), x) = z1 < x < z2$$

Notice that we need four parameters to define the probabilistic statements and are independent of the data. The example in Fig. 5 showcases the regions for $RI(M, S_{\texttt{blue}}, (-1.6, 1))$ and $RI(M, S_{\texttt{red}}, (1.5, 4.6))$. The coverage of the combined regions

$$[RI(M, S_{\texttt{blue}}, (-1.6, 1)) \cup RI(M, S_{\texttt{red}}, (1.5, 4.6))]$$

is 83.04%.

Coverage does not take into account the correct or incorrect classification of a data point. We want a metric that can tell us the quality of the classification. For this, let us analyze from the first

Table 1 Extended binary confusion matrix for safety envelopes

	Positive (P)	Negative (N)
Positive estimation	TP	FP
Negative estimation	FN	TN
Not in SE	Missed P	Missed N

principles what the possible metrics for safety envelopes are. A metric, in the area of statistical classification, is a value that relates a classification procedure to its performance given some data. A metric is the combination of four possible classification outcomes, namely, false positives (FPs), false negatives (FNs), true positives (TPs), and true negatives (TNs). (A confusion matrix is an $n \times n$ matrix that encodes all possible classification outcomes for a classification problem with n classes. Predicted values are assigned to rows, and actual values correspond to columns.) These outcomes come from the fact that there are two possible classes and two possible estimations. Unfortunately, safety envelopes do not partition the space in only two regions (positive and negative regions) but, instead, they partition the space into three regions: positive, negative and "not inside safety envelope." An extended confusion matrix[††] showing all possible six classification outcomes is presented in Table 1.

We would like to find a safety envelope that accepts as few mistakes as possible while covering as large a safe region as possible. In other words, we expect that safety envelopes take 1) as few unsafe points as possible (FNs + FPs)/total, i.e., error; 2) as many safe points as possible (TPs + TNs)/total, i.e., accuracy; and 3) as many points as possible (TPs + TNs + FNs + FPs)/total, i.e., coverage.

Notice that in contrast to machine learning practice, where metrics are computed given a dataset, we are interested in computing the metrics from a model: a collective-probability model.

The expected proportion of safe points, (TN + TP)/total, is captured by the following:

Definition 8. Accuracy: Given a collective-probability model $M = \langle \Theta, \Xi, \{X_\theta\}, p_\Theta, L_\Theta, C_\Theta \rangle$, a parameter space Π, and a set of probabilistic statements $\{S_l\}$ over M for each $l \in L_\Theta$, accuracy is the cumulative probability that falls within the region of interest and is correctly classified, i.e.,

$$\text{accuracy}(M, \{S_l\}, \pi) = \sum_{l \in L_\Theta} \left(\sum_{\theta \in C_\Theta(\Theta) = l} p_\Theta(\theta) P(x \in RI(M, S_l, \pi)) \right)$$

where $C_\Theta(\Theta) = l$ corresponds to the set $\{\theta \in \Theta : C_\Theta(\theta) = l\}$, which is the set containing all states that are tagged with l.

In Fig. 5, we can see that accuracy corresponds only to the regions correctly classified: red with red and blue with blue. Notice that we can have the same accuracy for different regions of interest (compare Fig. 5 with Fig. 7).

The expected proportion of unsafe points, (FN + FP)/total, is captured by the following:

Definition 9. Error: Given a collective-probability model $M = \langle \Theta, \Xi, \{X_\theta\}, p_\Theta, L_\Theta, C_\Theta \rangle$, a parameter space Π, and a set of probabilistic statements $\{S_l\}$ over M for each $l \in L_\Theta$, the error is the cumulative probability that falls within the region of interest and its incorrectly classified; i.e.,

$$\text{error}(M, \{S_l\}, \pi) = \sum_{l \in L_\Theta} \left(\sum_{\theta \in C_\Theta(\Theta) \neq l} p_\Theta(\theta) P(x \in RI(M, S_l, \pi)) \right)$$

where $C_\Theta(\Theta) \neq l$ corresponds to the set $\{\theta \in \Theta : C_\Theta(\theta) \neq l\}$, which is the set containing all states that are not tagged with l.

[††]A false positive is also known as a type I error. A false negative is a type II error.

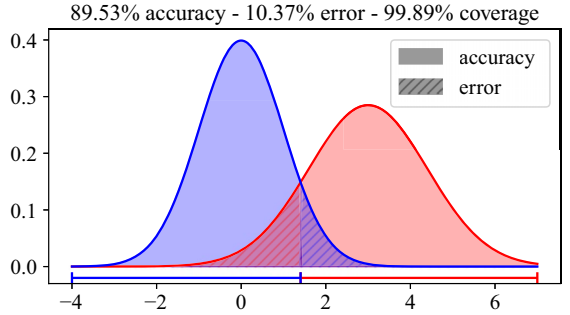

Fig. 6 Representation of accuracy, error, and coverage as in Fig. 5. Larger intervals mean higher accuracy and coverage but also larger error.

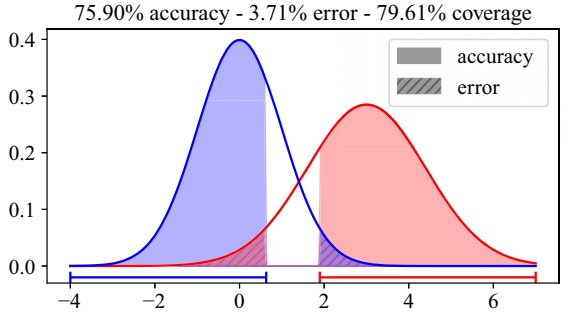

Fig. 7 Representation of accuracy, error, and coverage as in Fig. 5. Accuracy can be kept as in Fig. 5 while the error is reduced, thanks to a careful tuning of the intervals.

In Figs. 5–7, the error corresponds to the stripped areas. It is clear that

$$\text{coverage}(M, \{S_l\}, \pi) = \text{accuracy}(M, \{S_l\}, \pi) + \text{error}(M, \{S_l\}, \pi)$$

This means that we can have two different regions of interest with the same accuracy but different errors or different combinations of accuracy and errors that give rise to the same coverage.

Notice that high accuracy does not mean small errors. It depends on how many points are being discarded by the classification. For this reason, we have to define a metric that encodes our desire for a small error and high accuracy, namely, the following:

Definition 10. Model quality: Given a collective-probability model $M = \langle \Theta, \Xi, \{X_\theta\}, p_\Theta, L_\Theta, C_\Theta \rangle$, a parameter space Π, a set of probabilistic statements $\{S_l\}$ over M for each $l \in L_\Theta$, and a weight $w \in \mathbb{R}$, the combination of the accuracy and error is

$$\text{quality}(M, \{S_l\}, \pi, w, x) = \text{accuracy}(M, \{S_l\}, \pi, x) \times (1 - \text{error}(M, \{S_l\}, \pi, x))^w$$

The model quality increases as accuracy does, and it decreases as the error increases (1 − error). The w parameter is the weight given to the error. The larger the weight, the costlier the error becomes. From observations that can be found in the Supplemental Material, we have found that an error of $w = 10$ discourages safety envelopes with "too much" error while enforcing a good accuracy.

IV. Theorems and Formal Proofs

Safety envelopes are to be deployed as an external module to a control system in order to guarantee safe input data. This external module is called a monitor (in runtime verification [15]), and it is to be generated from the safety envelopes in an automatic manner. It is of integral importance that safety envelopes are correctly implemented and guarantee what they are designed for; thus, formal proofs of their correct behavior must accompany them. Figure 8 presents safety

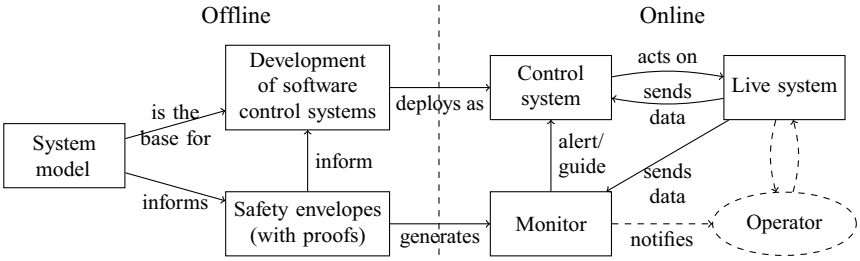

Fig. 8 Safety envelopes and monitors to check for the correct behavior of a system. Adapted from Ref. [20].

```
inside : NormalDist → ℝ → ℝ → Bool
inside nd z x = ((μ - z * σ) <ᵇ x) ∧ (x <ᵇ (μ + z * σ))  where open NormalDist nd using (μ; σ)

z-predictable : Model → ℝ → ℝ → ℝ × Bool
z-predictable M z x = ⟨ x , any (λ nd → inside nd z x) (map (proj₁ ∘ proj₂) (Model.fM M)) ⟩
```

Fig. 9 Excerpt of the formalization of z-predictable as Agda code.

```
-- In words: Given a Model `M` and parameter `z`, `x` is z-predictable iff
-- there exists a pair ⟨α,v⟩ (angle of attack and velocity) such that they are
-- associated to a `nd` (Normal Distribution) and `x` falls within the
-- Predictable Interval
theorem2← : ∀ (M z x)
          → z-predictable M z x ≡ ⟨ x , true ⟩
          → Any (λ{⟨ ⟨α,v⟩ , ⟨ nd , p ⟩ ⟩ → x ∈ pi nd z}) (Model.fM M)
theorem2← M z x res≡x,true = any-map (proj₁ ∘ proj₂) (follows-def← M z x res≡x,true)
theorem2→ : ∀ (M z x)
          → Any (λ{⟨ ⟨α,v⟩ , ⟨ nd , p ⟩ ⟩ → x ∈ pi nd z}) (Model.fM M)
          → z-predictable M z x ≡ ⟨ x , true ⟩
theorem2→ M z x proofAny = follows-def→ M z x (any-map-rev (proj₁ ∘ proj₂) proofAny)
```

Fig. 10 Agda code for proof of Theorem 2. *Any* represents existential quantification, i.e. "there exists".

envelopes and monitor placement within the production chain and the control loop of a control system [20].

A. Theorems

The following are theorems that we proved mechanically in Agda[‡‡] to guarantee the expected behavior of safety envelopes. (Full implementation and the proofs can be found in the Supplemental Material.) The first property to be mechanically proven corresponds to the relationship between the z-predictability for univariate normal distributions and the z score:

Theorem 1: In the case of univariate normal distributions, the z-predictability condition $D_M(\mu_\theta, \sigma_\theta^2, x) < z$, where D_M is the Mahalanobis distance, reduces to $\mu_\theta - z\sigma_\theta < x < \mu_\theta + z\sigma_\theta$, which is the prediction interval with a z score of z.

Given a signal-energy safety envelope, we can determine the connection between z-predictability and the signal input as follows:

Theorem 2: Given a signal-energy model $M_{\text{stall}} = \langle \Theta_{\text{states}}, \{\mathcal{N}(\mu_\theta, \sigma_\theta^2)\}, C_{\text{stall}} \rangle$, an energy signal $x \in \mathbb{R}$ is z-predictable if, and only if, there exist $\theta \in \Theta_{\text{states}}$ such that $x \in (\mu_\theta - z\sigma_\theta, \mu_\theta + z\sigma_\theta)$; i.e., x falls within one of the prediction intervals.

We can prove that a (univariate) signal-energy τ-confidence is a special case of τ-confidence:

Theorem 3: Given a signal-energy model $M_{\text{stall}} = \langle \Theta_{\text{states}}, \{\mathcal{N}(\mu_\theta, \sigma_\theta^2)\}, C_{\text{stall}} \rangle$ and $\tau \in (0.5, 1]$, a classification $K_{\text{stall}}(M, \tau, x) = k$ for an observation x is τ-confident if, and only if, $P(k|x) \geq \tau$.

[‡‡]The repository titled "safety-envelopes-sentinels" (version 0.1.2.0) can be found online at http://wcl.cs.rpi.edu/pilots/fvdddas [retrieved 11 November 2022].

As a consequence of Theorems 2 and 3, we can guarantee that signal-energy safety envelopes are in fact (general) safety envelopes:

Theorem 4: Given a signal-energy model $M_{\text{stall}} = \langle \Theta_{\text{states}}, \{\mathcal{N}(\mu_\theta, \sigma_\theta^2)\}, C_{\text{stall}} \rangle$, $\tau \in (0.5, 1]$, and $z \in \mathbb{R}^+$, an energy signal x belongs to safety envelope $RI(M_{\text{stall}}, S_{\text{nostall}} \wedge S_{\text{stall}}, (z, \tau))$ if, and only if, x is z-predictable and τ-confident.

B. Formal Proofs and Monitor Generation

We have used the Agda proof assistant to guarantee that the implementation of safety envelopes follows its expected behavior and the theorems presented before. Additionally, from the Agda code, we can generate verified Haskell code, which can be compiled into binary and run separately. Thus, we show a path to implement monitors (see Fig. 8). Figure 9 displays an excerpt of the formalization where signal-energy z-predictability is defined (see Definition 1):

Once safety envelopes are formally defined, we can prove properties on them. Such is the case of Theorem 2, which is proven by the Agda code shown in Fig. 10.

From the Agda formalization, we have generated a monitor. A monitor is a computer program to observe a stream of data to evaluate its consistency and correctness. The interested reader can check the Supplemental Material, where the full proofs and an extended explanation of the code found in Fig. 10 can be found. The generated monitor checks when a stream of signal-energy measurement, encoded as a floating-point number, is z-predictable. The resulting executable can process a continuous stream of floating-point numbers and outputs a stream of Booleans determining the z-predictability of each value. Figure 11 displays a

```
name72 = "Avionics.SafetyEnvelopes.z-predictable"
d72 ::
  T24 ->  -- This corresponds to the Model M
  MAlonzo.Code.Avionics.Real.T4 ->  -- real number z
  MAlonzo.Code.Avionics.Real.T4 ->  -- real number x
  MAlonzo.Code.Agda.Builtin.Sigma.T14  -- a tuple <x, is x z-confident?>
d72 v0 v1 v2

  = coe (d62 (coe (MAlonzo.Code.Data.List.Base.du20 (coe
          (\ v3 -> MAlonzo.Code.Agda.Builtin.Sigma.d28
           (coe (MAlonzo.Code.Agda.Builtin.Sigma.d30 (coe v3)))))
              (coe (d46 (coe v0)))))
         (coe v1) (coe v2))
```

Fig. 11 Haskell autogenerated code from z-predictable Agda code shown in Fig. 9.

(frankly obscure) piece of Haskell code generated from z-predictable on Agda.

With the help of some wrapping functions and code, the function can be called like any other function in Haskell. The implementation and proofs occupy a total of 980 lines in Agda. From the Agda code, a total of 1160 lines of Haskell code were generated.

V. Experimental Results

A. Signal-Energy Safety Envelopes

As explained in Sec. II, to evaluate safety envelopes, we have constructed multiple models from wind-tunnel experiments. Each model is composed of different flight state distributions. We consider three test cases: an easily separable case where only flight states for an airspeed of 6 m/s are considered, a slightly less separable case with an airspeed of 20 m/s, and a case where we assume no knowledge of airspeed (all airspeeds and AOA flight states are taken into account). The exploration of the optimal τ and z for each of the three cases can be seen in Fig. 12. In all cases, the value of the metrics plateau as z increases outside, further out to the right. Note that z is restricted to the range of $[0, 0.4]$ in Fig. 12c in order to display a readable plot.

As it can be seen in Fig. 12, increasing τ reduces accuracy and error irrespectively of z. Because accuracy and error change at slighly different rates, quality does not consistently decrease as τ grows, as clearly seen in Figs. 12b and 12c. For a given z, there is a τ that maximizes quality. If we analyze z by fixing τ, we will notice that accuracy and error grow as z does. But in contrast to τ, there is generally no value of z that maximizes accuracy or quality, or minimizes error. This means that the user has to choose a value of z, from which an effective τ can be found. As for what value of z to choose, we recommend a value as small as possible that maintains a high model quality. That is because larger z values (eg, $> 3\sigma$) imply that uncommon events will be treated as predictable when their chance of being so is small.

Choosing z and τ carelessly could lead us to safety envelope which are potentially unsafe. For example, make $\tau = 0.5001$ and $z = 9$. In this case, we would capture in the safety envelope values that are consistent to either stall or no stall distributions and have low confidence of being correctly classified. This might not be a big problem if the distributions are many σ apart, but it is a big problem if the model is not highly separable. In general, it makes sense to evade very low values of τ (smaller than 0.65) as well as very high values for z (larger than 3).

In Fig. 13, we can see the safety envelopes defined with the optimum z and τ obtained from the exploration shown in Fig. 12. As mentioned before, larger w values penalize errors higher in the model quality metric. For easily separable data, there are smaller error rates, which lessen the impact of w. For non-easily separable data, as in the case of the model that contains all possible flight states at once, the model quality suffers drastically, since its error is relatively high. This shows that the quality of the signal-energy model for stall detection decreases significantly when no airspeed information is given.

B. GPRMs Applied to Safety Envelopes

GPRMs excel at generalizing a reduced number of observed points into an infinite number of interlinked Gaussian distributions. They are heavily used in aerospace applications, as demonstrated by Ahmed et al. [21]. Ahmed et al. made use of variational heteroscedastic GPRMs to extend the wind-tunnel data, and thus created a larger more refined model. We sample 100 Gaussian distributions from these variational heteroscedastic GPRMs (VHGPRMs), increasing with artificial data the size of the collective-Gaussian model used to define a safety envelope.

In Figs. 14 and 15, we show a comparison between safety envelopes defined using the distributions computed from wind-tunnel experiments and safety envelopes from artificially generated data from GPRMs. With a sufficiently high sample resolution, the z-predictability region becomes a single interval with no gaps, even with small values of z. This means that z takes a step back in its influence on the metrics while τ takes full control. There is a significant improvement in the *coverage* and *quality* of the GPRM extended safety envelopes. For a speed of 14 m/s, the coverage improved from 65.513 to 81.520 and the quality improved from 63.997 to 77.181; see Fig. 14. A similar improvement can be seen for an airspeed of 17 m/s; see Fig. 14. Preliminary results for other airspeeds, including 6 and 20 m/s, indicated the same trend of improvements for the metrics. The lack of figures for other airspeeds is due to the nature of GPRM training and the ease with which they overfit, thus producing unnaturalistic results.

Even with these disadvantages, VHGPRMs' results are hard to match with other techniques. Assuming that VHGPRMs generate artificial distributions as if they were produced by a wind tunnel, any sample we take from them will define a well-behaved safety envelope. With this assumption in place, Theorems 1, 2, and 4 apply equally to GPRM synthetic data, where the mean and standard deviation in those expressions can be replaced by the predictive moments of a GPRM. Thus, VHGPRMs add some flexibility in defining safety envelopes, where the moments used for defining them can either come from experimental data or from properly trained data-driven VHGPRMs.

In a similar framework, Gaussian process classification models (GPCMs; see Ref. [22]), which produce predictive probabilities instead of predictive moments, can be used to "interpolate" the conditional probability for stall in order to allow for "higher-resolution" probabilities across the different angles of attack. This

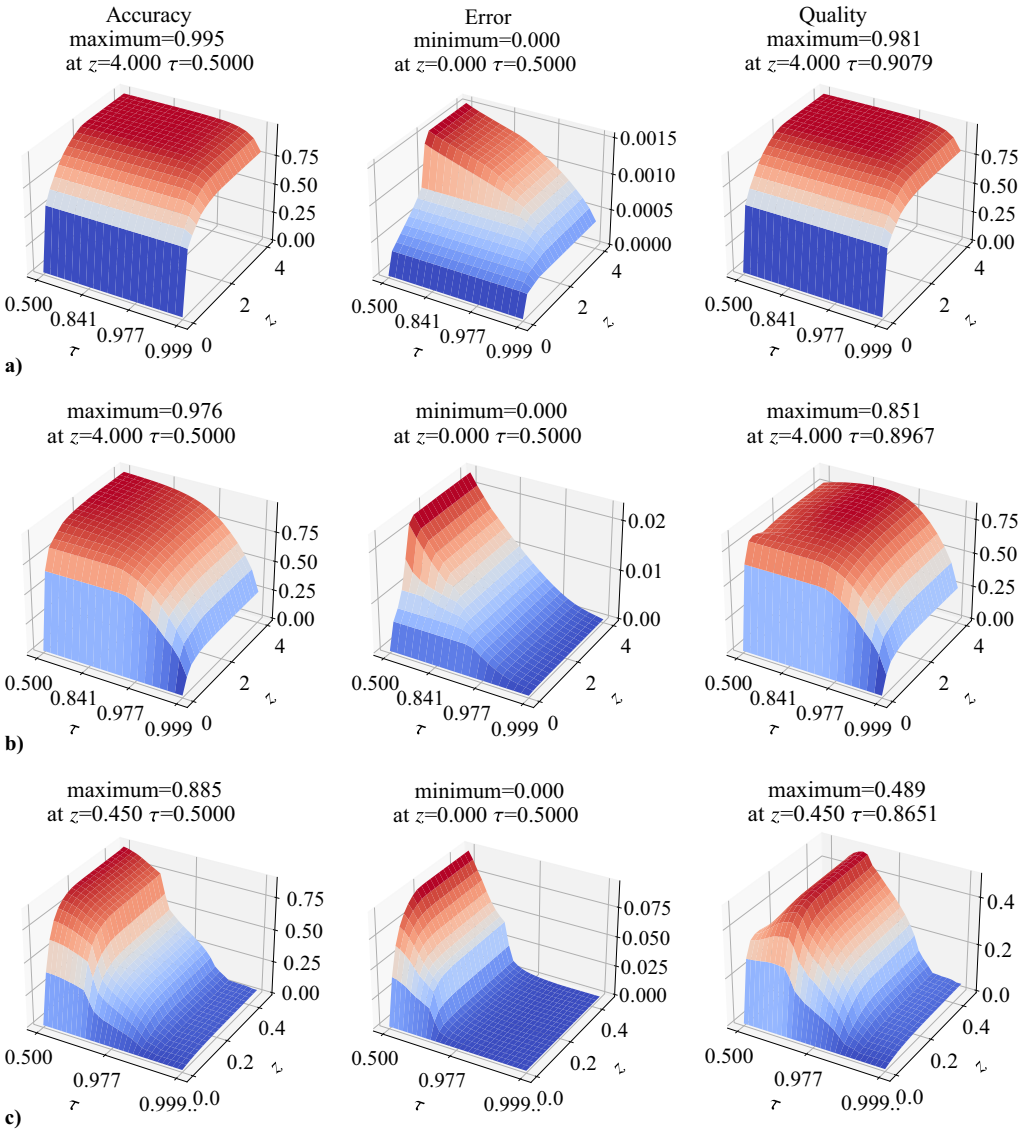

Fig. 12 Optimizing model quality. Exploration of z and τ for the airspeed of a) 6 m/s; b) 20 m/s; and c) all airspeeds, with $w = 10$.

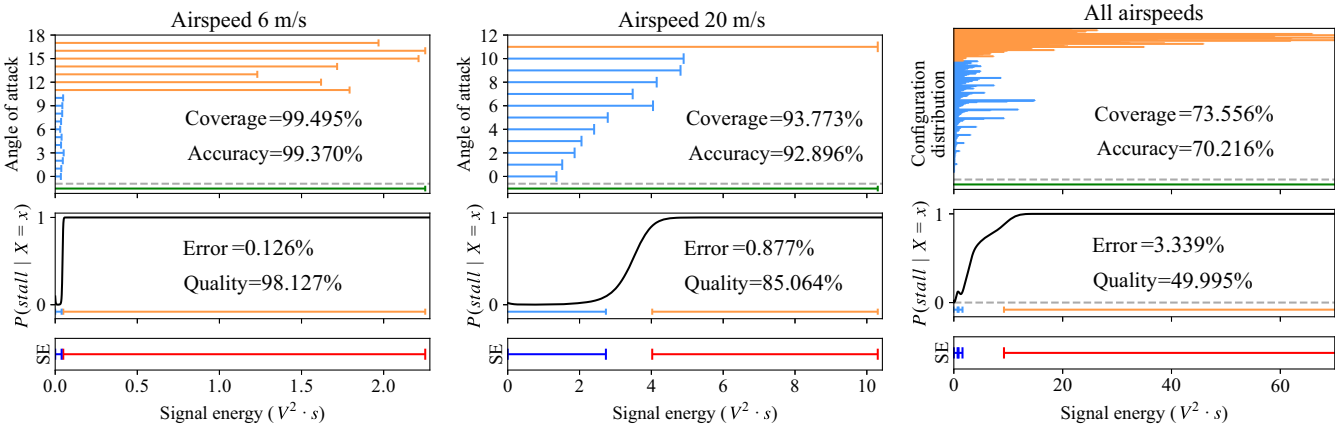

Fig. 13 Safety envelopes with optimal parameters z and τ that maximize quality with $w = 10$.

approach elegantly allows for the application of Theorem 3 onto GPCMs. Thus, with properly trained Gaussian process models (for regression and classification), the concept of safety envelopes can be expanded beyond experimental data using data-driven model-based predictive moments and probabilities.

C. Multivariate Safety Envelopes

The advantage of safety envelopes is their generalizability to multiple dimensions of input data. From the eight available sensors, we chose two sensors with low correlation between them: sensors 1 and 7. We proceeded to compute the mean and covariance matrix for

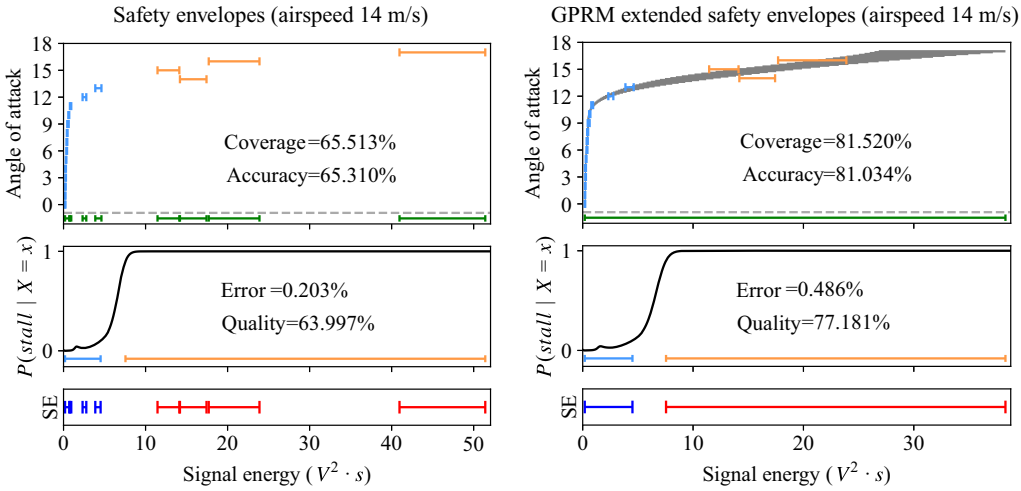

Fig. 14 Safety envelopes at 14 m/s with parameters $z = 0.3$ and $\tau = 0.9$: a) original data, and b) GPRM-generated data.

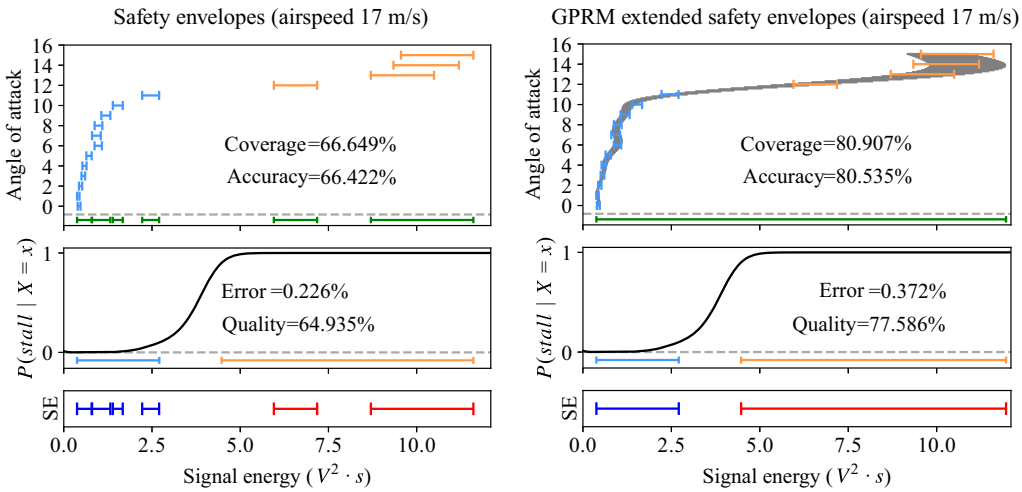

Fig. 15 Safety envelopes at 17 m/s with parameters $z = 0.3$ and $\tau = 0.9$: a) original data, and b) GPRM-generated data.

each flight state given the sensors' data. Choosing lowly correlated sensors allows us to show safety envelopes more clearly because highly correlated sensors show up as lines on the plots. Additional tests revealed no significant difference in the particular selection of sensors concerning the metrics.

Figure 16 shows the safety envelope defined for a bivariate normal distribution-based model. Computing the error and accuracy for multivariate-normal distributions required the use of a Monte Carlo simulation as opposed to straightforward computing of the regions from the cumulative distribution function, as in the univariate models.

Error and quality improve as we increase the number of sensors used to define safety envelopes. In fact, for the univariate case (see Fig. 13) error is 0.877% and quality 85.064%, while for the bivariate case (see Fig. 16b) error is 0.047% and quality 91.361%. Safety envelopes perform better (with less error and higher model quality) for the case of 20 m/s, even at nonoptimized values of z and τ. The difference is heightened when all flight states are considered and where all metrics improve: accuracy = 78.959%, error = 0.994, and quality = 71.449% for the multivariate case against accuracy = 70.216%, error = 3.339, and quality = 49.995% for the univariate case.

VI. Related Work

Jackson et al. [23] defined a novel concept to determine when a Markov process is safe. From a dataset, they constructed a Markov process that they restricted given a parameter ϵ. Note that ϵ is used similarly for safety envelopes z; namely, it restricts safe behavior to a region where the behavior is likely to occur and discards anything else outside this region as possibly problematic. Specifically, they checked whether the current state of a Markov model was within the range of the possible things the model should do parameterized by ϵ. Safety envelopes take a step further and include a classification step, which is τ-confidence, which is firmly grounded on Bayes's rule.

HOL and Isabelle are interactive proof assistants with a rich history of proofs from discrete and continuous probability theory [24–27]. These libraries implement measure theory; discrete, continuous, and normal random variables; and many other fundamental theorems on probability theory like the central limit theorem: all of them built from the bottom up in a robust verifiable environment. In our work, we followed a top-down methodology in which we assume the correct formalization of well-known real number theory and probability theory. In this way, we differ from previous work by implementing more complex structures than what could be done in a limited time with a bottom-up approach. Agda, as opposed to HOL and Isabelle, is a programming language and proof assistant built on top of a constructive theory [28]. It is possible to write code and create an executable with very little extra work in Agda as opposed to HOL and Isabelle. We were able to produce a simple monitor from the formalization on about 90 lines of Haskell code and an additional 130 lines of Agda. Copilot [29] and PILOTS [17,30] have presented strategies to detect and recover from faulty data streams due to hardware errors in airplane systems and dynamic data-driven applications systems, respectively. Those systems do

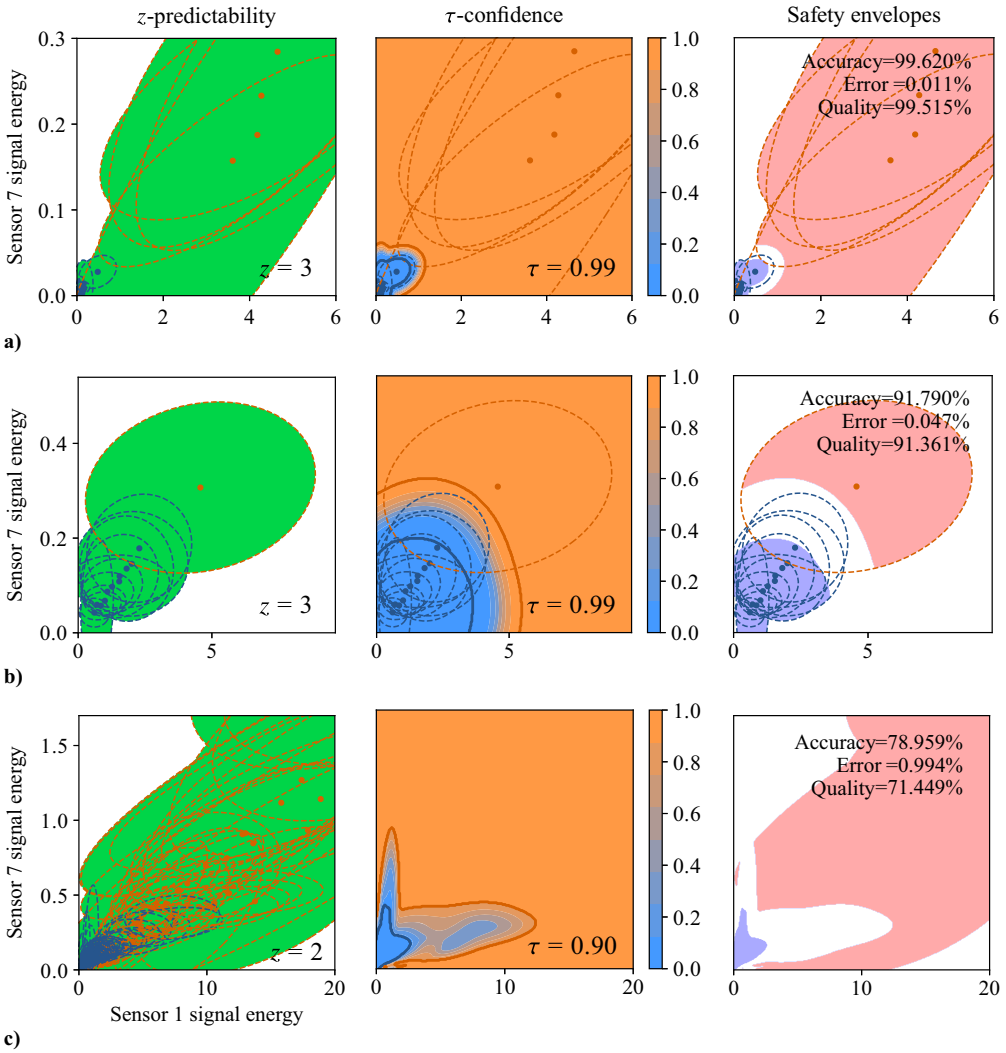

Fig. 16 Multivariate safety envelopes, where distribution means are dots, and orange lines (ellipses) are the confidence regions: a) 6 m/s, b) 20 m/s, and c) all flight states.

not yet incorporate formal verification, and therefore depend on the quality of the software implementation, the testing environment, and the robustness of the programming language.

VeriDrone [31] and other Coq initiatives (e.g., Ref. [32]) have incorporated formal verification into working systems to formally prove properties like maximum speed restrictions and correct behavior according to a specification. These approaches follow a similar framework to ours (namely, the use of software verification techniques for the creation of verified pieces of code) but are different in their end goal: the implementation of verified control systems. Safety envelopes do not steer a system in a specific direction; rather, they are meant as a warning system and input for the control system. Another approach for the verification of control systems is DryVR [33], in which a system defined as a labeled, directed acyclic graph is determined to behave safely (or not) with the help of simulation trajectories and a blacklist of unsafe states. Although safety envelopes are not an approach to building control systems, they can be used to find unsafe states, which can be later used as blacklists in systems like DryVR.

The concept of a region restricted by parameters where an aircraft can operate safely appears in the literature time and time again. Such is the case of Jeannin et al. [34], who defined "safe regions" for an aircraft to operate where no collisions are expected, assuming correct behavior; or Paul et al. [35] with the concept of "correctness envelopes," which determine the conditions for a system to satisfy different correctness properties. Safety envelopes can be used on their own, but they can shine when incorporated into larger frameworks of control systems such as the simplex architecture [36]. The simplex architecture's goal is to allow for safe control upgrades of complex control systems. A safety region in the simplex architecture delimits the region where a system can be controlled. If the experimental controller gets close to the boundary of the safe region, a robust simpler controller takes over. Safety envelopes could be used to define a tighter region within the handwritten safety region of simplex architecture or as a replacement.

Breese et al. [20] presented the idea of *formal* safety envelopes from which this work sprung into life. They proposed a first-order logic-based definition for safety envelopes in which only one parameter is necessary (z) and not two (z and τ). Cruz-Camacho et al. [37] extended safety envelopes to encompass both predictability and classification, which result in higher tunability, thanks to the extra parameter τ. Based on the work of Breese et al. [20], Paul et al. [35] proposed a metric for safety envelopes using preprocessed data as input. This paper combines and extends all of these prior works with a detailed, justified, and generalized definition of safety envelopes; in particular, we extended Cruz-Camacho et al.'s [37] safety envelopes to multivariate distributions and GPRM-generated data from Ahmed et al. [21].

VII. Conclusions

A novel, formally verified concept was presented for classification given a statistical model for one or multiple real-numbered data inputs. Safety envelopes encode two conditions a safe classification must have: z-predictability, whether an input value is consistent with a model; and τ-confidence to quantify confidence in a classification.

Four metrics to compare different models and parameters of safety envelopes were given: coverage, accuracy, error, and quality. Metrics are fundamental to finding the proper parameters a safety envelope should have. A formalization of safety envelopes in Agda was presented; and with it, four formal proofs that tie z-predictability and τ-confidence with any input value. Formally verified Haskell code was generated from the Agda formalization; and from it, an executable was produced to process a stream of data. How to integrate GPRMs into safety envelopes was explored, and their results were showcased as the extensibility of safety envelopes to use synthetic data. Safety envelopes were shown to work seamlessly with one as well as two input value dimensions, i.e., with models constructed out of univariate or bivariate normal distributions.

Future work will include extending safety envelopes to correct faulty inputs where physical or logical redundancy is available, as in the case of multiple or heterogeneous sensor inputs; finding the minimum number of flight states needed to construct a good model to reduce the number of physical experiments necessary to perform; studying the impact of better-informed priors in the quality of the models; and finding all possible sources of numerical instability that would make floating-point numbers a bad fit as approximations for real numbers.

Appendix: Conditional Probability Deduction

In this Appendix, we present a derivation for the equation to compute the conditional probability $P(\texttt{tag}|X = x)$, which was given as

$$P(\texttt{tag}|X = x) = \frac{\sum_{\theta\in\Theta} pdf_\theta(x) p_\Theta(\theta) P(\texttt{tag}|\theta)}{\sum_{\theta\in\Theta} pdf_\theta(x) p_\Theta(\theta)} \tag{A1}$$

This is possible thanks to Bayes' rule for classification. Namely, we can rewrite the expression as

$$P(\texttt{tag}|X = x) = \frac{f_{X|\texttt{tag}}(x) P(\texttt{tag})}{f_X(x)} \tag{A2}$$

The marginal probability $f_X(x)$ can be computed as

$$f_X(x) = \sum_{\theta\in\Theta} f_{X,\theta}(x) = \sum_{\theta\in\Theta} f_{X|\Theta=\theta}(x) p_\Theta(\theta) \tag{A3}$$

where $f_{X|\Theta=\theta}(x)$ is the probability density function for the state θ, i.e., $f_{X|\Theta=\theta}(x) = pdf_\theta(x)$ with parameters from X_θ. Note that X is the same as the space Ξ where x lies, whereas X_θ is the random variable associated with the state θ.

The conditional probability $f_{X|\texttt{tag}}(x)$ is computed in a similar manner as the marginal probability. First, we apply the law of total probability and then Bayes's rule again:

$$\begin{aligned} f_{X|\texttt{tag}}(x) &= \sum_{\theta\in\Theta} pdf_\theta(x) P(\Theta = \theta|\texttt{tag}) \\ &= \sum_{\theta\in\Theta} pdf_\theta(x) \frac{p_\Theta(\theta) P(\texttt{tag}|\Theta = \theta)}{P(\texttt{tag})} \end{aligned} \tag{A4}$$

where $P(\texttt{tag}|\Theta = \theta)$ is the probability that a specific configuration (flight state) is tagged with tag (e.g., to produce stall). This probability is either zero or one, and it is given by expert judgment.

With the marginal [Eq. (A3)] and conditional probabilities [Eq. (A4)] in place, we can rewrite Eq. (A2) as

$$P(\texttt{stall}|X = x) = \frac{\sum_{\theta\in\Theta} pdf_\theta(x) p_\Theta(\theta) P(\texttt{stall}|\theta)}{\sum_{\theta\in\Theta} pdf_\theta(x) p_\Theta(\theta)} \tag{A5}$$

which is the expression shown previously in Eq. (A1).

Acknowledgments

This research was partially supported by the National Science Foundation (grant no. CNS-1816307) and the U.S. Air Force Office of Scientific Research (dynamic data-driven application systems grant no. FA9550-19-1-0054). We thank Saswata Paul for his input on the metrics for safety envelopes.

References

[1] Kopsaftopoulos, F., and Chang, F.-K., "A Dynamic Data-Driven Stochastic State-Awareness Framework for the Next Generation of Bio-Inspired Fly-by-Feel Aerospace Vehicles," *Handbook of Dynamic Data Driven Applications Systems*, Springer, Berlin, 2018, pp. 697–721.

[2] Kopsaftopoulos, F., Nardari, R., Li, Y.-H., and Chang, F.-K., "Data-Driven State Awareness for Fly-by-Feel Aerial Vehicles: Experimental Assessment of a Non-Parametric Probabilistic Stall Detection Approach," *Structural Health Monitoring 2017*, DEStech Publ., New York, 2017, pp. 1596–1604.

[3] Kopsaftopoulos, F., Nardari, R., Li, Y.-H., and Chang, F.-K., "A Stochastic Global Identification Framework for Aerospace Structures Operating Under Varying Flight States," *Mechanical Systems and Signal Processing*, Vol. 98, Jan. 2018, pp. 425–447. https://doi.org/10.1016/j.ymssp.2017.05.001

[4] Kopsaftopoulos, F., "Data-Driven Stochastic Identification for Fly-by-Feel Aerospace Structures: Critical Assessment of Non-Parametric and Parametric Approaches," *AIAA SciTech 2019 Forum*, AIAA Paper 2019-1534, 2019.

[5] Darema, F., "Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements," *International Conference on Computational Science*, Springer, Berlin, 2004, pp. 662–669.

[6] Paul, S., "Emergency Trajectory Generation for Fixed-Wing Aircraft," MS Thesis, Rensselaer Polytechnic Inst., Dec. 2018, http://wcl.cs.rpi.edu/theses/paul_ms.pdf.

[7] Srivatanakul, T., "Security Analysis with Deviational Techniques," Ph.D. Thesis, Univ. of York, York, England, U.K., April 2005.

[8] Agha, G., and Palmskog, K., "A Survey of Statistical Model Checking," *ACM Transactions on Modeling and Computer Simulation*, Vol. 28, No. 1, 2018, pp. 6:1–6:39.

[9] Krishnan, R., and Lalithambika, V. R., "Modeling and Validating Launch Vehicle Onboard Software Using the SPIN Model Checker," *Journal of Aerospace Information Systems*, Vol. 17, No. 12, 2020, pp. 695–699. https://doi.org/10.2514/1.I010876

[10] Malecha, G., Ricketts, D., Alvarez, M. M., and Lerner, S., "Towards Foundational Verification of Cyber-Physical Systems," *2016 Science of Security for Cyber-Physical Systems Workshop (SOSCYPS)*, Inst. of Electrical and Electronics Engineers, New York, 2016, pp. 1–5.

[11] Platzer, A., "Differential Dynamic Logic for Hybrid Systems," *Journal of Automated Reasoning*, Vol. 41, No. 2, 2008, pp. 143–189.

[12] Ghorbal, K., Jeannin, J.-B., Zawadzki, E., Platzer, A., Gordon, G. J., and Capell, P., "Hybrid Theorem Proving of Aerospace Systems: Applications and Challenges," *Journal of Aerospace Information Systems*, Vol. 11, No. 10, 2014, pp. 702–713.

[13] Abed, S., Rashid, A., and Hasan, O., "Formal Analysis of Unmanned Aerial Vehicles Using Higher-Order-Logic Theorem Proving," *Journal of Aerospace Information Systems*, Vol. 17, No. 9, 2020, pp. 481–495. https://doi.org/10.2514/1.I010730

[14] Cohen, R., Feron, E., and Garoche, P.-L., "Verification and Validation of Convex Optimization Algorithms for Model Predictive Control," *Journal of Aerospace Information Systems*, Vol. 17, No. 5, 2020, pp. 257–270. https://doi.org/10.2514/1.I010686

[15] Falcone, Y., Havelund, K., and Reger, G., "A Tutorial on Runtime Verification," *Engineering Dependable Software Systems*, IOS Press, Amsterdam, 2013, pp. 141–175.

[16] Norell, U., "Towards a Practical Programming Language Based on Dependent Type Theory," Ph.D. Thesis, Dept. of Computer Science and Engineering, Chalmers Univ. of Technology, Göteborg, Sweden, Sept. 2007.

[17] Imai, S., Blasch, E., Galli, A., Zhu, W., Lee, F., and Varela, C. A., "Airplane Flight Safety Using Error-Tolerant Data Stream Processing," *IEEE Aerospace and Electronics Systems Magazine*, Vol. 32, No. 4, 2017, pp. 4–17.

[18] Imai, S., Hole, F., and Varela, C. A., "Self-Healing Data Streams Using Multiple Models of Analytical Redundancy," *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC 2019)*, Inst. of Electrical and Electronics Engineers, New York, 2019, pp. 1–10. https://doi.org/10.1109/DASC43569.2019.9081716

[19] Alpaydin, E., *Introduction to Machine Learning*, 3rd ed., MIT Press, Cambridge, MA, 2014, pp. 51–52.

[20] Breese, S., Kopsaftopoulos, F., and Varela, C., "Towards Proving Runtime Properties of Data-Driven Systems Using Safety Envelopes," *Proceedings of the 12th International Workshop on Structural Health Monitoring*, Vol. 2, DESTech Publ., Lancaster, PA, Sept.–Nov. 2019, pp. 1748–1757.
https://doi.org/10.12783/shm2019/32302

[21] Ahmed, S., Amer, A., Varela, C., and Kopsaftopoulos, F., "Data-Driven State Awareness for Fly-by-Feel Aerial Vehicles via Adaptive Time Series and Gaussian Process Regression Models," *Dynamic Data Driven Applications Systems*, edited by F. Darema, E. Blasch, S. Ravela, and A. Aved, Springer International Publ., Cham, 2020, pp. 57–65.
https://doi.org/10.1007/978-3-030-61725-7_9

[22] Amer, A., and Kopsaftopoulos, F. P., "Towards Unified Probabilistic Rotorcraft Damage Detection and Quantification via Non-Parametric Time Series and Gaussian Process Regression Models," *Proceedings of the Vertical Flight Society 76th Annual Forum and Technology Display*, The Vertical Flight Soc., Fairfax, VA, Oct. 2020.

[23] Jackson, J., Laurenti, L., Frew, E., and Lahijanian, M., "Safety Verification of Unknown Dynamical Systems via Gaussian Process Regression," *2020 59th IEEE Conference on Decision and Control (CDC)*, Inst. of Electrical and Electronics Engineers, New York, 2020, pp. 860–866.
https://doi.org/10.1109/CDC42340.2020.9303814.

[24] Hurd, J., "Formal Verification of Probabilistic Algorithms," Univ. of Cambridge, Computer Lab. TR UCAM-CL-TR-566, Cambridge, England, U.K., May 2003.

[25] Hasan, O., and Tahar, S., "Probabilistic Analysis of Wireless Systems Using Theorem Proving," *Electronic Notes in Theoretical Computer Science*, Vol. 242, No. 2, 2009, pp. 43–58.

[26] Qasim, M., Hasan, O., Elleuch, M., and Tahar, S., "Formalization of Normal Random Variables in HOL," *Intelligent Computer Mathematics*, edited by M. Kohlhase, M. Johansson, B. Miller, L. de Moura, and F. Tompa, Springer International, Cham, Switzerland, 2016, pp. 44–59.

[27] Avigad, J., Hölzl, J., and Serafin, L., "A Formally Verified Proof of the Central Limit Theorem," *Journal of Automated Reasoning*, Vol. 59, No. 4, 2017, pp. 389–423.

[28] Luo, Z., *Computation and Reasoning: A Type Theory for Computer Science*, Oxford Univ. Press, Oxford, England, U.K., 1994.

[29] Pike, L., Wegmann, N., Niller, S., and Goodloe, A., "Copilot: Monitoring Embedded Systems," *Innovations in Systems and Software Engineering*, Vol. 9, No. 4, 2013, pp. 235–255.

[30] Chen, S., Imai, S., Zhu, W., and Varela, C. A., "Towards Learning Spatio-Temporal Data Stream Relationships for Failure Detection in Avionics," *Handbook of Dynamic Data Driven Applications Systems*, edited by E. Blasch, S. Ravela, and A. Aved, Springer International, Cham, Switzerland, 2018, pp. 97–121.

[31] Ricketts, D., Malecha, G., Alvarez, M. M., Gowda, V., and Lerner, S., "Towards Verification of Hybrid Systems in a Foundational Proof Assistant," *2015 ACM/IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE)*, Inst. of Electrical and Electronics Engineers, New York, 2015, pp. 248–257.

[32] Anand, A., and Knepper, R., "ROSCoq: Robots Powered by Constructive Reals," *Interactive Theorem Proving*, Vol. 9236, edited by C. Urban and X. Zhang, Springer International, Cham, Switzerland, 2015, pp. 34–50.

[33] Fan, C., Qi, B., Mitra, S., and Viswanathan, M., "DryVR: Data-Driven Verification and Compositional Reasoning for Automotive Systems," *Computer Aided Verification*, Vol. 10426, edited by R. Majumdar and V. Kunčak, Springer International, Cham, Switzerland, 2017, pp. 441–461.
https://doi.org/10.1007/978-3-319-63387-9_22

[34] Jeannin, J.-B., Ghorbal, K., Kouskoulas, Y., Gardner, R., Schmidt, A., Zawadzki, E., and Platzer, A., "A Formally Verified Hybrid System for the Next-Generation Airborne Collision Avoidance System," *Tools and Algorithms for the Construction and Analysis of Systems*, Vol. 9035, edited by C. Baier and C. Tinelli, Springer, Berlin, 2015, pp. 21–36.
https://doi.org/10.1007/978-3-662-46681-0_2

[35] Paul, S., Kopsaftopoulos, F., Patterson, S., and Varela, C. A., "Towards Formal Correctness Envelopes for Dynamic Data-Driven Aerospace Systems," *Handbook of Dynamic Data-Driven Application Systems*, edited by F. Darema and E. Blasch, Springer, Berlin, Dec. 2020, https://wcl.cs.rpi.edu/bib/Year/2020.complete.html#paul-hbdddas-2020.

[36] Seto, D., Krogh, B., Sha, L., and Chutinan, A., "The Simplex Architecture for Safe Online Control System Upgrades," *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*, Vol. 6, Inst. of Electrical and Electronics Engineers, New York, 1998, pp. 3504–3508.
https://doi.org/10.1109/ACC.1998.703255.

[37] Cruz-Camacho, E., Paul, S., Kopsaftopoulos, F., and Varela, C. A., "Towards Provably Correct Probabilistic Flight Systems," *Dynamic Data Driven Application Systems*, edited by F. Darema, E. Blasch, S. Ravela, and A. Aved, Springer International, Cham, Switzerland, 2020, pp. 236–244.

D. Allaire
Associate Editor