Towards Provably Correct Probabilistic Flight Systems

Elkin Cruz-Camacho, Saswata Paul, Fotis Kopsaftopoulos, and Carlos A. Varela

Rensselaer Polytechnic Institute, Troy, New York, 12180, USA
{cruzce,pauls4,kopsaf}@rpi.edu, cvarela@cs.rpi.edu

Abstract *Safety envelopes* are meant to determine under which conditions and state space regions a probabilistic property of a data-driven system can be asserted with high confidence. Dynamic data-driven applications systems (DDDAS) can make use of safety envelopes to be cognizant of the formal warranties derived from their models and assumptions. An example of safety envelopes is presented as the intersection of two simpler concepts: *z-predictability* and τ-*confidence*; which correspond to state estimation and classification, respectively. To illustrate safety envelopes, stall detection from signal energy is shown with data gathered by piezo-electric sensors in a composite wing inside a wind tunnel under varying angles of attack and airspeed configuration. A formalization of these safety envelopes is presented in the Agda proof assistant, from which formally proven sentinel code can be generated.

1 Introduction

Aerospace systems will be increasingly autonomous in terms of self-diagnosis, self-healing, and overall self-awareness. They will be capable of sensing, reasoning, and reacting in real-time to their actual operating conditions, allowing for optimal control and decision-making abilities [13]. This will be aided by access to an unprecedented amount of real-time data from onboard sensors which can be interpreted to sense the aeroelastic state, environmental conditions, and structural conditions of aerospace systems [14]. Smart aerospace systems will be capable of detecting aerodynamic conditions – *e.g.*, stall or flutter, using data from piezo-electric and other sensors placed on the wings of an aircraft [13,12]. Dynamic data-driven applications systems (DDDAS) [8] can use this data to create accurate aerodynamic models that can be updated to reflect the real-time aerodynamic performance of such systems [18].

Since the failure of safety-critical aerospace systems can cause harm to human life, the environment, or property [22], it is necessary to verify the correctness of the software used in these systems. Model checking and formal methods can be used for verification of such software [1], *e.g.*, by writing mechanically-verified proofs of correctness. However, formal proofs usually only hold under some conditions which may not necessarily be true during actual operation [19]. Breese *et al.* [6] have proposed an approach for classifying a system's state space into distinct regions with respect to a formal proof. They introduce *safety envelopes* to represent the subset of the state space where the formal proof of a probabilistic statement holds. The extent of a safety envelope depends on a data-driven model of the system and parameters to quantify the certainty of state estimation. 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 *sentinels* can analyze real-time data against a safety envelope and determine whether the system conditions fall within the envelope or not. DDDAS can use safety envelopes formalize data-driven probabilistic guarantees that hold in real-time.

The contributions of this paper are: a definition of safety envelopes as regions delimited by a model and user-definable parameters, an example of a safety envelope that warranties z-predictability and τ-confidence for state estimation and classification respectively (Section 2), and the formalization of the safety envelope concepts in Agda [17] and generation of Haskell [16] code from the formal specification (Section 3).

2 Signal Energy Safety Envelopes as Parameterized Statements

Safety envelopes are a step forward for provably robust dynamic data-driven applications systems (DDDAS). This section presents a definition for safety envelopes and exemplifies the construction of safety envelopes for the prediction of stall for a self-sensing composite wing given a single energy signal input.

A flight state can be identified as a quadruple $\langle x, \alpha, v, \mathit{stall}\rangle$, where x is the signal energy received from a sensor in a wing, α the angle of attack, v is the airspeed and *stall* is a boolean value that indicates whether the wing is stalled or not. A model $\boldsymbol{M}$ is a $\langle S_M, f_M\rangle$, where S_M is a subset of $\mathbb{R}\times\mathbb{R}$ (all possible airspeeds and angles of attack), and f_M is a map with the signature $S_M \to \mathcal{N}\times\mathbb{B}$. A map f_M receives a valid input (v, α) and returns a $\left\langle \mathcal{N}(\mu,\sigma^2), \mathit{stall}\right\rangle$ where $\mathcal{N}(\mu,\sigma^2)$ is the normal distribution that the energy signal is assumed to follow. This means, a model $\boldsymbol{M}$ is a collection of probability distributions each drawn from a partial flight state denoted by $\langle \alpha, v, \mathit{stall}\rangle$.

A model $\boldsymbol{M}$ is computed from data collected in wind tunnel experiments. The example model considered in this paper has been constructed from the experiments presented by Kopsaftopoulos & Chang in [13].

Definition 1. ***Signal Energy Safety Envelope:*** *Given a model $\boldsymbol{M}$ and parameters Π, a safety envelope for the signal energy is the region $\xi \subseteq \mathbb{R}$ under which a probabilistic statement*[1] *P with arguments $\boldsymbol{M}$ and Π holds,* i.e.*, a safety envelope is the region defined by $\xi = \{x \in \mathbb{R} : P(\boldsymbol{M}, \Pi, x) = \mathtt{true}\}$, where x is a signal energy measurement.*

For a simple and slightly contrived example of safety envelopes, suppose that all signal energy measurements follow the normal distribution with parameters $\mathcal{N}(10, 1)$ (the model $\boldsymbol{M}$) and consider the statement "the signal energy measurement falls within the 95.4% prediction interval (PI) around the mean" (the statement $\boldsymbol{P}$ with model $\boldsymbol{M}$ and at least 95.4% PI as Π), then the safety envelope defined by the statement is the region contained inside $[\mu - 2\sigma, \mu + 2\sigma] = [8, 12]$.

[1] A probabilistic statement is a statement that includes probabilistic assertions as part of its definition, *e.g.*, the expected value after flipping a fair coin (0 = heads; 1 = tails) is $\frac{1}{2}$.

2.1 Data Consistency with Model using z-predictability

Definition 2. *An energy signal x is z-predictable iff there exist $\langle d_i, b_i \rangle \in \mathrm{Im}(f_M)$ such that $x \in pred\_i(d_i, z)$, where $pred\_i$ is the prediction interval for the z score,* i.e.*, $pred\_i(\mathcal{N}(\mu, \sigma^2), z) = [\mu - z\sigma, \mu + z\sigma]$.*

In statistics z is called the z-score. The main idea of z-predictability is to determine whether a single measurement of signal energy is consistent with the model at hand. For a z score of 3, around 99.7% of the measurements are z-predictable. A value that falls outside the prediction interval is considered to be not z-predictable and it is treated as a possible error.

From the definition of z-predictability it can be proven that (see subsection 3):

Theorem 1. *An energy signal x is z-predictable iff there exist $\langle \alpha, v \rangle \in S_M$ such that $f_M(\langle \alpha, v \rangle)_1 = d_i$ and $x \in pred\_i(d_i, z)$.*

On the top row of Figures 1 and 2, a region generated by the z predictability can be seen. The z-score in Figures 1 and 2 is 2 and 4, respectively.

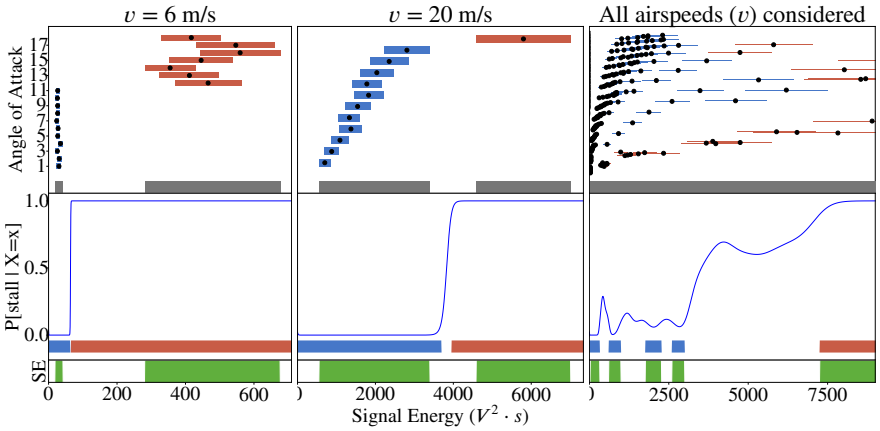

Figure 1. Rows: *Top row:* Prediction intervals for each angle of attack. The black dot is the mean, blue indicates no stall, and orange indicates stall. The gray line below is the region of z-predictability — *i.e.*, the region $\{x \in \mathbb{R} : \exists \langle \alpha, v \rangle \in S_M . f_M(\langle \alpha, v \rangle)_1 = d_i \wedge x \in pred\_i(d_i, z)\}$ — with $z = 2$. *Middle row:* Probability function $P[\texttt{stall} \mid X = x]$, which indicates the probability of the wing to be in stall given a single measurement of the signal energy. The classification regions for stall and no stall are shown below with confidence of $\tau = 90\%$, *i.e.*, the τ-confident region is the union of both colored regions, blue and red, where blue indicates no-stall and red stall. *Bottom row:* The green region indicates the safety envelopes, the region where a signal energy measurement is both z-predictable and τ-confident. Columns: *Left column:* The model M includes all flight states with an airspeed of $6m/s$. *Center column:* Only flight states with an airspeed of $20m/s$. *Right column:* All flight states recorded, all airspeeds ($v \in \{6, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22\} m/s$) and angles of attack ($\alpha \in [1, 18]$) are considered in the model M.

2.2 Stall Detection using Statistical Inference

This section presents a procedure using statistical inference to classify the stall condition of a wing, then it is shown how this classification can be parameterized to delimit a confidence region of classification.

Definition 3. ***Conditional probability of stall:*** *Given a model M and a measurement x, the probability of* `stall` *is defined as:*

$$\begin{aligned} P[stall \mid X = x] &= \frac{P[\texttt{stall}] f(x \mid \texttt{stall})}{f(x)} \\ &= \frac{\sum_{\langle \alpha, \upsilon \rangle \in S_M} \left(f_s(x) P[\langle \alpha, \upsilon \rangle] P[\texttt{stall} \mid \langle \alpha, \upsilon \rangle] \right)}{\sum_{\langle \alpha, \upsilon \rangle \in S_M} \left(f_s(x) P[\langle \alpha, \upsilon \rangle] \right)} \end{aligned} \tag{1}$$

where f_s corresponds to the probability density function for the distribution $d_s = f_M(\langle \alpha, \upsilon \rangle)_1$, the conditional probability $P[\texttt{stall} \mid \langle \alpha, \upsilon \rangle]$ is either 0 or 1 and determined by $f_M(\langle \alpha, \upsilon \rangle)_2$, and the distribution $P[\langle \alpha, \upsilon \rangle]$ uniform for all $\langle \alpha, \upsilon \rangle \in S_M$ ($P[\langle \alpha, \upsilon \rangle] = \frac{1}{|S_M|}$).

The probability of stall can be seen in the middle row of Figures 1 and 2. The following is the definition of a classification procedure from the conditional probability function:

Definition 4. ***Classification function:*** *Given a model M, an energy signal x can be classified in one of three categories as:*

$$c(M, \tau, x) = \begin{cases} \texttt{stall} & P[stall \mid X = x] \geq \tau \\ \texttt{nostall} & P[\neg stall \mid X = x] \geq \tau \\ \texttt{uncertain} & \text{in any other case.} \end{cases}$$

where τ, the threshold, is a real number in the range $(0.5, 1]$ and indicates the level of confidence wanted from the classification (alternatively, $1 - \tau$ indicates the risk that will be accepted for the classification [3]). The signature of c is $M \times \mathbb{R} \times \mathbb{R} \to \{\texttt{stall}, \texttt{nostall}, \texttt{uncertain}\}$.

The classification region can be seen at the bottom of the middle row in Figures 1 and 2, for $\tau = 90\%$ and 99.9%, respectively.

Definition 5. *A classification $c(M, \tau, x) = k$ is τ-confident iff $k \neq \texttt{uncertain}$.*

A τ-confident classification is one in which the risk of misclassification is below the threshold τ. Alternatively, τ-confidence can be defined as:

Theorem 2. *A classification k is τ-confident iff $P[k \mid x] \geq \tau$.*

2.3 Safety Envelopes as τ-confident Classifications on z-predictable Measurements

Definition 6. *A safety envelope $se(M, z, \tau)$ for stall detection is the region $x \in \mathcal{P}(\mathbb{R})$ with parameters $\Pi = \langle z, \tau \rangle$, where the following probabilistic statement holds: x is z-predictable and $c(M, \tau, x)$ is τ-confident.*

As a corollary from Theorems 1 and 2:

Theorem 3. *An energy signal x belongs to a safety envelope $se(M, z, \tau)$ iff there exist $\langle \alpha, v \rangle \in S_M$ such that $f_M(\langle \alpha, v \rangle)_1 = d_i$ and $x \in pred\_i(d_i, z)$, and the classification $k = c(M, \tau, x)$ has a confidence bigger than τ, i.e., $P[k \mid x] \geq \tau$.*

The last row of Figures 1 and 2 shows 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 $6m/s$, the safety envelope is the same as the region defined by the z-predictability; in other cases, the region defined by the τ-confidence reduces the region described by z-predictability, or viceversa.

Notice that when safety envelopes are applied to a model where all airspeeds and angles of attack 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 entails a stall condition. In Figure 2 rightmost column, safety envelopes do not include any signal with values from around 200 and until 8000.

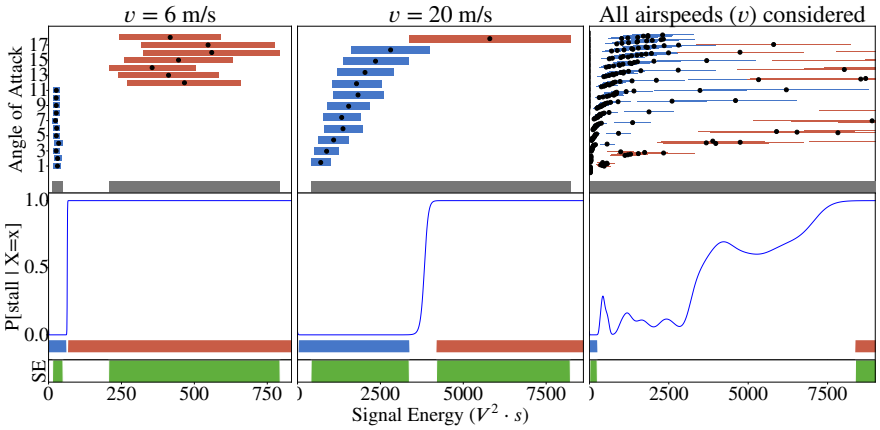

Figure 2. ROWS and COLUMNS as in Figure 1 but with parameters $z = 4$ and $\tau = 99.9\%$. Notice that compared with the τ-confident region displayed in Figure 1 with an airspeed $v = 20m/s$ (red and blue regions in the middle-center plot), the τ-confident region for a value of $\tau = 99.9\%$ is smaller, it has a wider gap, which shows that the higher the τ the smaller the safety envelope will be. Conversely, the more samples admitted as z-consistent, the bigger the safety envelope will be, as it can be seen comparing the gray regions on the top row from Figure 1. The optimum values for z and τ will depend on the application, but a meaningful range of values for z would be $[2, 6]$ (which correspond to 5% or less data thrown away) and for τ around 95% and upwards.

3 Formalization and Sentinel Generation

Formally proving properties of DDDAS using a proof assistant is a necessary step to ensure fault-free or near fault-free certified software. Signal energy safety envelopes have been implemented in Agda[2], a formal verification system, in order to prove their properties mechanically. Three procedures have been implemented: computing whether an energy signal input is z-predictable, τ-confident, and whether it falls inside the safety envelope defined by a model M with parameters $\Pi = \langle z, \tau \rangle$. The following is an excerpt of the formalization, where z-predictability is defined:

```
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)) ⟩
```

The power of formalization comes from the fact that properties can be mechanically proven, *i.e.*, it can be proven that the definition entails the implementation. Such is the case presented in the proof below, where Theorem 1 is formally proven using the proof that the implementation in Agda (above) follows from the Definition 2. In the same manner, theorems 2 and 3 have been encoded in Agda and proven formally, *i.e.*, in a mechanized manner.

```
-- 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 withing the
-- Predictable Interval
theorem1← : ∀ (M z x)
          → z-predictable M z x ≡ ⟨ x , true ⟩
          → Any (λ{⟨ ⟨α,v⟩ , ⟨ nd , p ⟩ ⟩ → x ∈ pi nd z}) (Model.fM M)
theorem1← M z x res≡x,true = any-map (proj₁ ∘ proj₂) (follows-def← M z x res≡x,true)
theorem1→ : ∀ (M z x)
          → Any (λ{⟨ ⟨α,v⟩ , ⟨ nd , p ⟩ ⟩ → x ∈ pi nd z}) (Model.fM M)
          → z-predictable M z x ≡ ⟨ x , true ⟩
theorem1→ M z x proofAny = follows-def→ M z x (any-map-rev (proj₁ ∘ proj₂) proofAny)
```

A *sentinel* is a binary, a program, whose job is to monitor for the consistency and correctness of the data received and generated in flight. Agda has the capability of generating Haskell code which can be executed and tested. From the formalization shown above, a sentinel has been built such that it monitors when a stream of floating-point numbers is z-predictable. The implementation uses floating-point numbers as an approximation to real numbers.

To write the sentinel, a wrapper was written around the generated Agda code to pass data from the standard input. The resulting binary can process a continuous stream of data and outputs to the standard output a stream of booleans representing the z-predictability. The implementation and proofs occupy a total of 760 lines in Agda and 130 lines of code in Haskell. From the Agda code, a total of 1160 lines of Haskell code were generated.

[2] Full implementation and proofs can be found at `http://wcl.cs.rpi.edu/pilots/fvdddas` (repository name: `safety-envelopes-sentinels`, version 0.1.1.0)

4 Related Work

HOL and Isabelle are interactive proof assistants with a rich history of proofs from discrete and continuous probability theory [10,9,20,5]. Agda, opposed to HOL and Isabelle, is a a programming language and proof assistant built on top of a constructive theory [15]. Copilot [19] and PILOTS [7,11] have presented strategies to find and recover from faulty data-streams due to hardware errors in airplane systems and dynamic data-driven applications systems (DDDAS), respectively. Those systems do not yet incorporate formal verification. Veridrone [21] and other Coq initiatives (*e.g.,* [4]) have incorporated formal verification into working systems to formally prove aircraft safety properties. In this work, an approach to build a formally verified monitor/sentinel from a specification was presented and applied to aircraft safety.

5 Conclusion

An extension and modularization of the concepts put forward by Breese *et al.* [6] was presented. The modularization included the separation of what it means to be consistent, *z-predictability*, and how to quantify confidence in the stall classification of an aircraft, τ-confidence. It was shown that knowing only a single energy signal measurement from a piezo-electric sensor is not enough to confidently determine the stall state of a wing. Knowing the airspeed of the aircraft significantly improves the classification confidence.

A formalization of safety envelopes in Agda was also presented. From it, formally verified Haskell code was generated, wrapped and extended to process a stream of data. Safety envelopes are an important step forward in the direction of formally correct and robust dynamic data-driven applications systems (DDDAS).

Future work includes the definition of safety envelopes for a sequence of signal energy measurements as opposed to single, isolated values, as in *Ahmed et al.* [2]; and the implementation of runnable real number arithmetic as opposed to floating-point arithmetic operations.

Acknowledgment: This research was partially supported by the National Science Foundation (NSF), Grant No. – CNS-1816307, and the Air Force Office of Scientific Research (AFOSR), DDDAS Grant No. – FA9550-19-1-0054.

References

1. Agha, G., Palmskog, K.: A Survey of Statistical Model Checking. ACM Transactions on Modeling and Computer Simulation **28**(1), 6:1–6:39 (Jan 2018)
2. Ahmed, S., Amer, A., Varela, C., Kopsaftopoulos, F.: Data-Driven State Awareness for Fly-by-Feel Aerial Vehicles via Adaptive Time Series and Gaussian Process Regression Models. In: Dynamic Data-Driven Applications Systems (InfoSymbiotics/DDDAS 2020) (Oct 2020)
3. Alpaydin, E.: Introduction to Machine Learning. The MIT Press, Cambridge, Mass, third edn. (2014)
4. Anand, A., Knepper, R.: ROSCoq: Robots Powered by Constructive Reals. In: Urban, C., Zhang, X. (eds.) Interactive Theorem Proving, vol. 9236, pp. 34–50. Springer International Publishing, Cham (2015)

5. Avigad, J., Hölzl, J., Serafin, L.: A Formally Verified Proof of the Central Limit Theorem. Journal of Automated Reasoning **59**(4), 389–423 (Dec 2017)
6. Breese, S., Kopsaftopoulos, F., Varela, C.: Towards proving runtime properties of data-driven systems using safety envelopes. In: The 12th International Workshop on Structural Health Monitoring. Stanford, CA (Sep 2019)
7. Chen, S., Imai, S., Zhu, W., Varela, C.A.: Towards Learning Spatio-Temporal Data Stream Relationships for Failure Detection in Avionics. In: Blasch, E., Ravela, S., Aved, A. (eds.) Handbook of Dynamic Data Driven Applications Systems, pp. 97–121. Springer International Publishing, Cham (2018)
8. Darema, F.: Dynamic data driven applications systems: A new paradigm for application simulations and measurements. In: International Conference on Computational Science. pp. 662–669. Springer (2004)
9. Hasan, O., Tahar, S.: Probabilistic analysis of wireless systems using theorem proving. Electronic Notes in Theoretical Computer Science **242**(2), 43–58 (2009)
10. Hurd, J.: Formal verification of probabilistic algorithms. Tech. Rep. UCAM-CL-TR-566, University of Cambridge, Computer Laboratory (May 2003)
11. Imai, S., Blasch, E., Galli, A., Zhu, W., Lee, F., Varela, C.A.: Airplane flight safety using error-tolerant data stream processing. IEEE Aerospace and Electronics Systems Magazine **32**(4), 4–17 (2017)
12. Kopsaftopoulos, F.: Data-driven stochastic identification for fly-by-feel aerospace structures: Critical assessment of non-parametric and parametric approaches. In: AIAA Scitech 2019 Forum. p. 1534 (2019)
13. Kopsaftopoulos, F., Chang, F.K.: A dynamic data-driven stochastic state-awareness framework for the next generation of bio-inspired fly-by-feel aerospace vehicles. In: Handbook of Dynamic Data Driven Applications Systems, pp. 697–721. Springer (2018)
14. Kopsaftopoulos, F., Nardari, R., Li, Y.H., Chang, F.K.: Data-driven State Awareness for Fly-by-feel Aerial Vehicles: Experimental Assessment of a Non-parametric Probabilistic Stall Detection Approach. In: Structural Health Monitoring 2017. pp. 1596–1604. DEStech Publications, Inc. (Sep 2017)
15. Luo, Z.: Computation and reasoning: a type theory for computer science. Oxford University Press, Inc., USA (1994)
16. Marlow, S., et al.: Haskell 2010 language report. Available on: https://www.haskell.org/onlinereport/haskell2010 (2010)
17. Norell, U.: Towards a practical programming language based on dependent type theory. PhD Thesis, Department of Computer Science and Engineering, Chalmers University of Technology, SE-412 96 Göteborg, Sweden (Sep 2007)
18. Paul, S., Hole, F., Zytek, A., Varela, C.A.: Flight Trajectory Planning for Fixed Wing Aircraft in Loss of Thrust Emergencies. In: Dynamic Data-Driven Application Systems (InfoSymbiotics/DDDAS 2017). Cambridge, MA (Aug 2017)
19. Pike, L., Wegmann, N., Niller, S., Goodloe, A.: Copilot: monitoring embedded systems. Innovations in Systems and Software Engineering **9**(4), 235–255 (2013)
20. Qasim, M., Hasan, O., Elleuch, M., Tahar, S.: Formalization of Normal Random Variables in HOL. In: Kohlhase, M., Johansson, M., Miller, B., de Moura, L., Tompa, F. (eds.) Intelligent Computer Mathematics. pp. 44–59. Lecture Notes in Computer Science, Springer International Publishing, Cham (2016)
21. Ricketts, D., Malecha, G., Alvarez, M.M., Gowda, V., Lerner, S.: Towards verification of hybrid systems in a foundational proof assistant. In: 2015 ACM/IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE). pp. 248–257 (Sep 2015)
22. Srivatanakul, T.: Security analysis with deviational techniques. PhD Thesis, University of York, York, UK (Apr 2005)