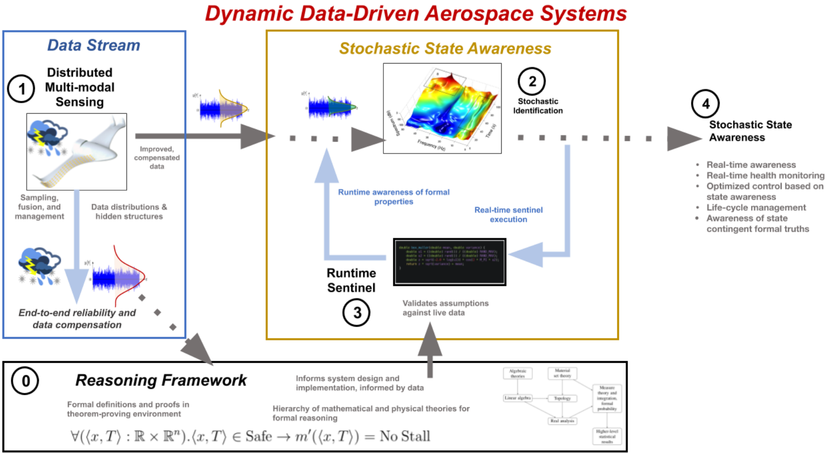
Towards Proving Runtime Properties of Data-Driven Systems Using Safety Envelopes

Samuel Breese
Fotis Kopsaftopoulos
Carlos Varela

Rensselaer Polytechnic Institute

Shonan Meeting on Programming Languages for Distributed Systems
Shonan Village Center, Kanagawa, Japan
May 27, 2019

Dynamic Data Driven Aerospace Systems



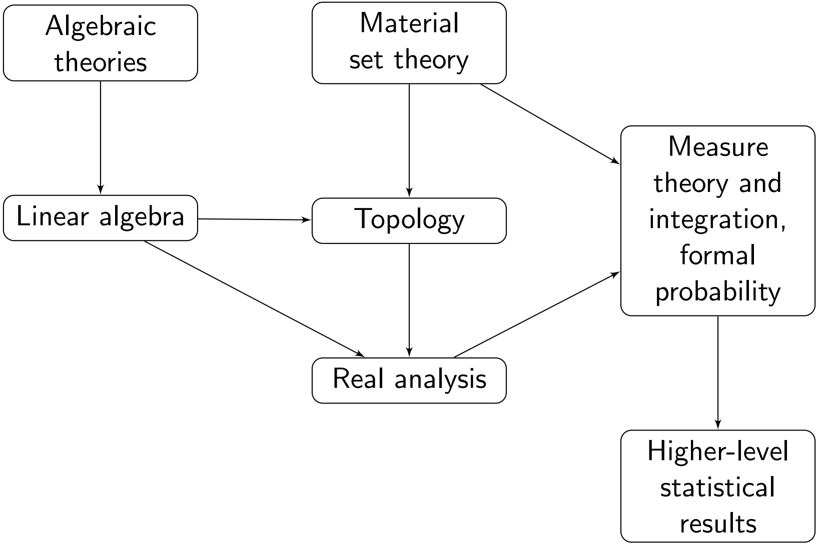
Overview

- Dynamic data-driven systems introduce complexity
- Often used in safety-critical domains (*e.g.* aerospace)
- Formal methods can yield stronger safety guarantees than testing

Formal Methods

- Computer-checked logical reasoning about a system
- Both automated and interactive approaches
- Requires a high level of rigor and detail, leading to high development costs
  - Magnified in systems involving stochastic elements
- Novel methods and techniques can help offset these costs

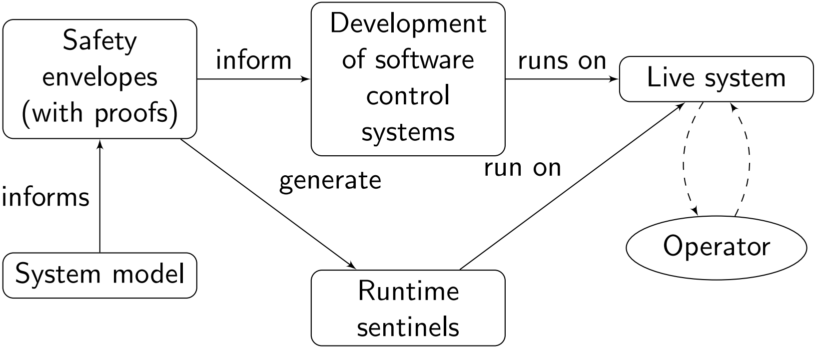
Hierarchy of Theories



Approach: Safety Envelopes

- Analogous to a flight safety envelope in an aircraft
- Describes a *safe subset* of system states
- Associates that safe subset with some correctness guarantee
- Provable formally in the proof assistant
- Checkable in live system through *runtime sentinel*

Workflow



Safety envelopes (with proofs)
inform
Development of software control systems
runs on
Live system
informs
generate
run on
System model
Runtime sentinels
Operator

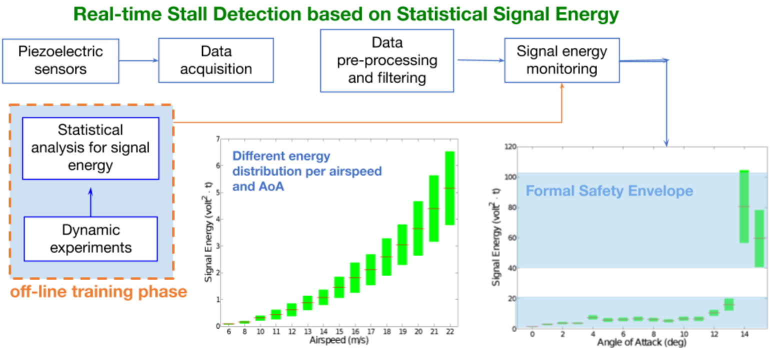
Runtime Sentinels

- Represent safe subsets as terms in some embedded domain-specific language
- Support evaluation to term in proof assistant
- Support generation of a program accessible from the runtime system
- Bring awareness of state-dependent formal properties to the system as it runs

Example: Introduction

- We study a model from Kopsaftopoulos associating a sensor reading from a wing with the likelihood that an aircraft is in a stall state
- Model is trained on experimental data from a wind tunnel - data driven
- We treat pairs of training data and runtime signal as system states
- Safe subset: intervals on runtime signal, (approximate) normality in training data

Example



Real-time Stall Detection based on Statistical Signal Energy
Piezoelectric sensors
Data acquisition
Data pre-processing and filtering
Signal energy monitoring
Statistical analysis for signal energy
Dynamic experiments
off-line training phase
Different energy distribution per airspeed and AoA
Signal Energy (volt² · t)
Airspeed (m/s)
Formal Safety Envelope
Angle of Attack (deg)

Example: Correctness

- Given definition of the model, we know that some intervals of runtime signal lead to "stall" classification
- Other intervals lead to "no stall" classification
- With appropriate definition of model, we can make this connection formal:

$$\forall(\langle x, T\rangle : \mathbb{R} \times \mathbb{R}^n).\langle x, T\rangle \in \mathrm{Safe} \rightarrow m'(\langle x, T\rangle) = \mathrm{No\ Stall}$$

Example: Sentinel

- C program testing membership in safe subset
- Using standard statistical tests for normality on training data
- Floating-point arithmetic for safe intervals of runtime signal
- Neither of these are "exact": disconnect between formal assumption and validation process
- Important area for future development