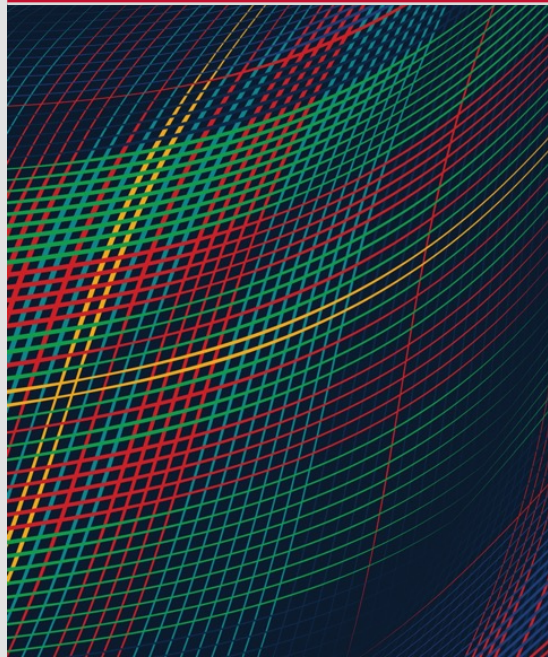
Probabilistic Verification to Support Next-Generation Certification

(ProVer-Cert)

MAY 11, 2026

Sam Procter
Dio De Niz

Carnegie Mellon University
Software Engineering Institute



Document Markings

This material is based upon work supported by the Department of War under Air Force Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM26-0477

Problem: Software Certification is often *Process*-Based

Despite *Property*-Based Certification's advantages

DO-178C:
- Standardized guidance used by FAA
- Activities: Planning, verification, tracing

Benefits:
- Predictable, schedulable
- Good safety record

Drawbacks:
- Confidence subjective and not quant.
- Supplements required for new tech

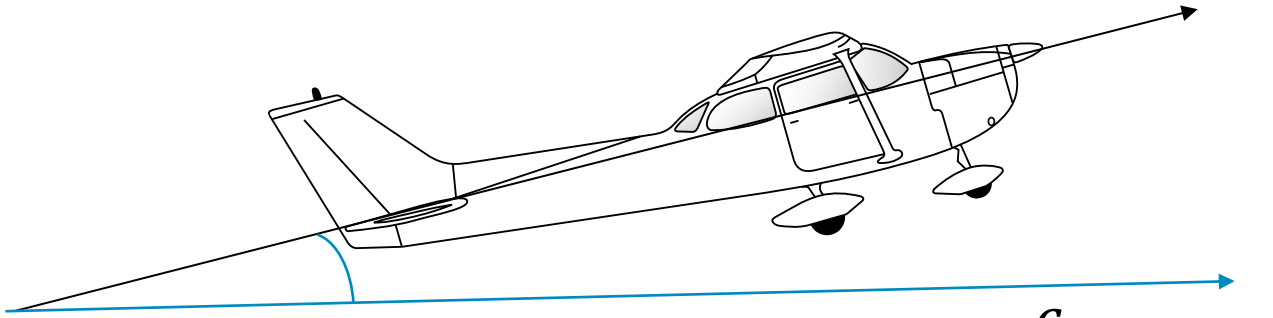
Overarching Properties (OPs):
- Developed by NASA, FAA, others
- Properties: Sufficient for certification: Intent, Correctness, and Innocuity

Benefits:
- Flexible / Powerful

Drawbacks:
- New
- Needs clarity on how to apply

Establishing Correctness:
- Tests are the most common approach.
  - A failure indicates a bug
  - A passed test gives only an *unquantifiable* increase in confidence
- Need quantitative, statistical increase in confidence from passed tests
- How? FACT approach

Approach:
1. Model the current, process-based certification process
2. Develop and model a probabilistic assessment method to support property-based certification
3. Compare the models for inputs, outputs, resources required, etc.



Technical Approach: Task 1
Define Use Case

We will define a use case to support our work in Tasks 2-4.

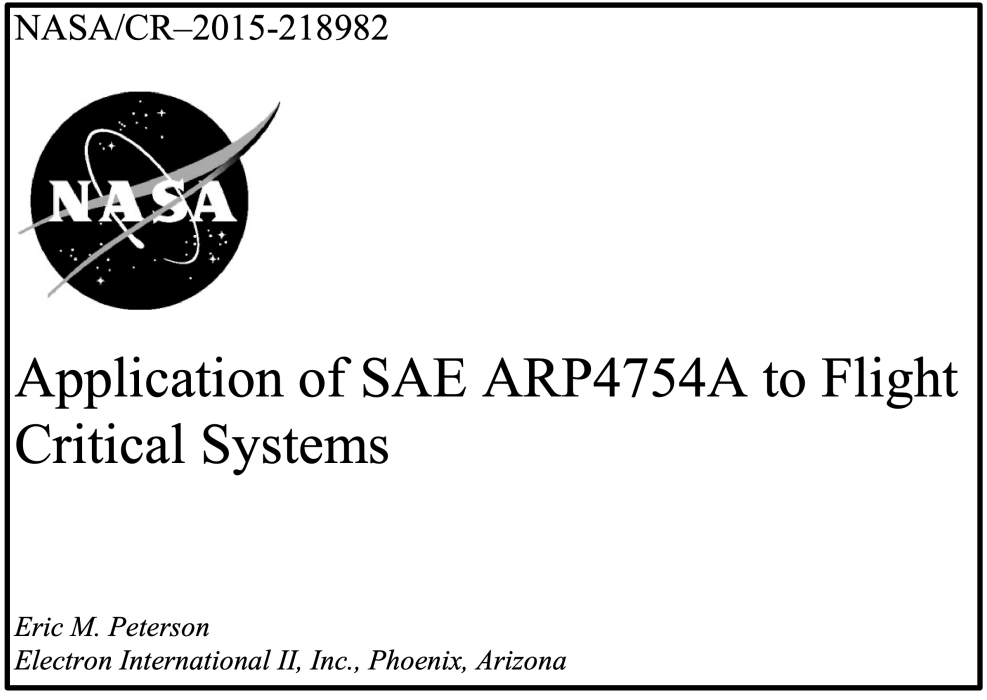
- The use case will be scoped carefully to:
  - Support (partial, mock) certification under DO-178C
  - Support (partial, mock) verification of correctness under the Overarching Properties
- The domain will be avionics, using a drone
  - This aligns us with the domain of DO-178 and many of the users of the Overarching Properties
- The structure will be narrative text augmented with models

Use Case: Source

Goal: Example certification from aviation domain

DO-178C: Software Considerations in Airborne Systems and Equipment Certification

- Coordinates with system development via, e.g., ARP4754A: “System life cycle processes can be found in other industry documents (for example, SAE ARP4754A).”


NASA/CR–2015-218982

NASA

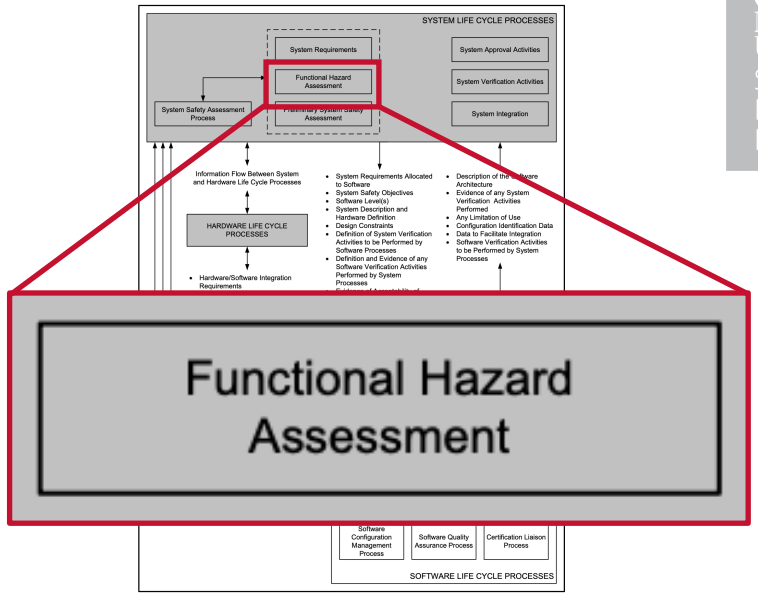
Application of SAE ARP4754A to Flight Critical Systems

Eric M. Peterson
Electron International II, Inc., Phoenix, Arizona

Use Case: Properties

Goal: Important property we can analyze quantitatively

We selected a property from the Functional Hazard Assessment in the example ARP4754A document



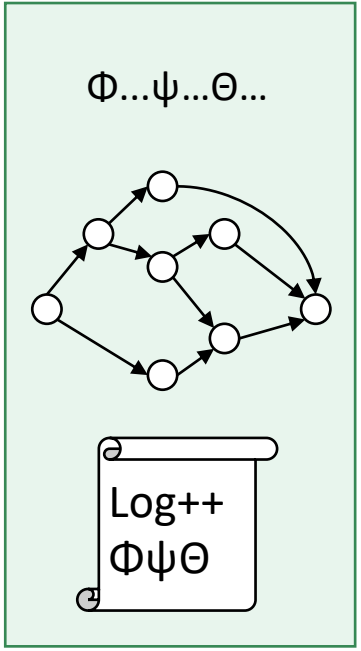
Function	FC #	Failure Condition (Hazard Description)	Flight Phase	Effect of Failure Condition on Aircraft/Crew	Classification
Provide Stability & Control: Automatic Stability & Control (2.5)	22.03	Erroneous autopilot command which exceeds authority limits	Flight	Airplane structural damage may result due to unrestricted pitch, roll or yaw commands. May result in rapid flight path responses, unsafe airplane flight paths and loss of altitude. Possible ground contact if occurs at low altitude resulting in loss of airplane.	Catastrophic

Technical Approach: Task 2
Extract and Quantify Probabilities

Currently the FACT approach verifies properties encoded in Probabilistic Computation Tree Logic (PCTL) on parameterized Markov chains.

In this task we will:

1. Encode specific safety / reliability properties from the use case developed in Task 1 in PCTL
2. Create the parameterized, abstract Markov chains which will support the establishment of confidence intervals for the the properties
3. Develop an instrumentation and observation approach (Log++) which will allow us to extract values for the parameters in the Markov chain. This will consist of two novel aspects:
    1. Determining what exactly needs to be observed to have coverage of the parameterized transitions in the abstract Markov chain
    2. Incorporating a sampling technique that ensures samples are independent and identically distributed (IID)

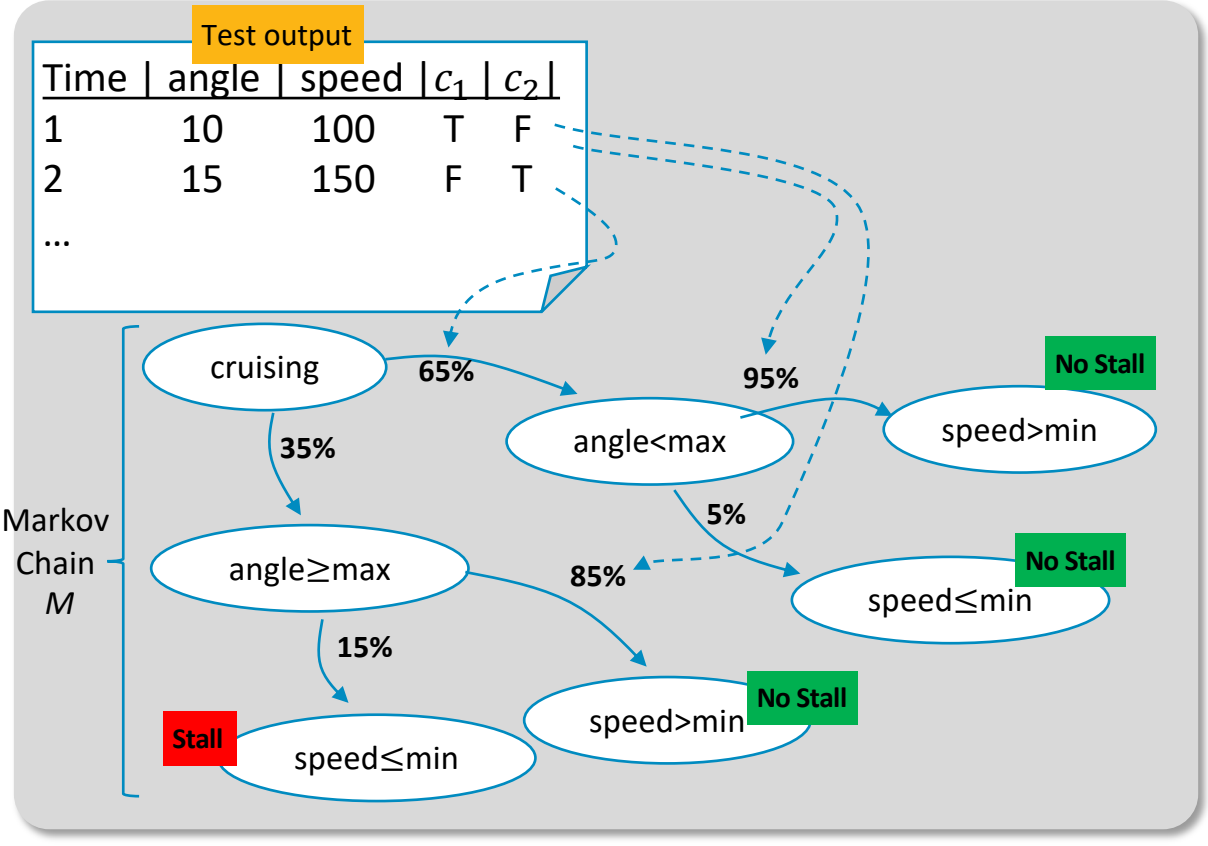


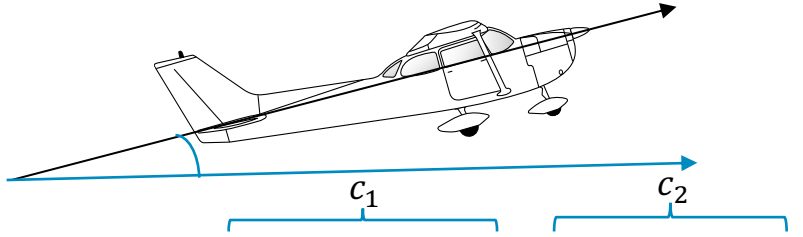
Example Safety Property

Carnegie Mellon University
Software Engineering Institute

Test output

Time	angle	speed	c_1	c_2
1	10	100	T	F
2	15	150	F	T
…				





No Stall (NS) := $G\ angleattack < maxangle\ U\ airspeed > minspeed$

More tests = Better confidence!

FACT: Probability that our property (*No Stall*) is within the confidence interval [a,b] is less than α

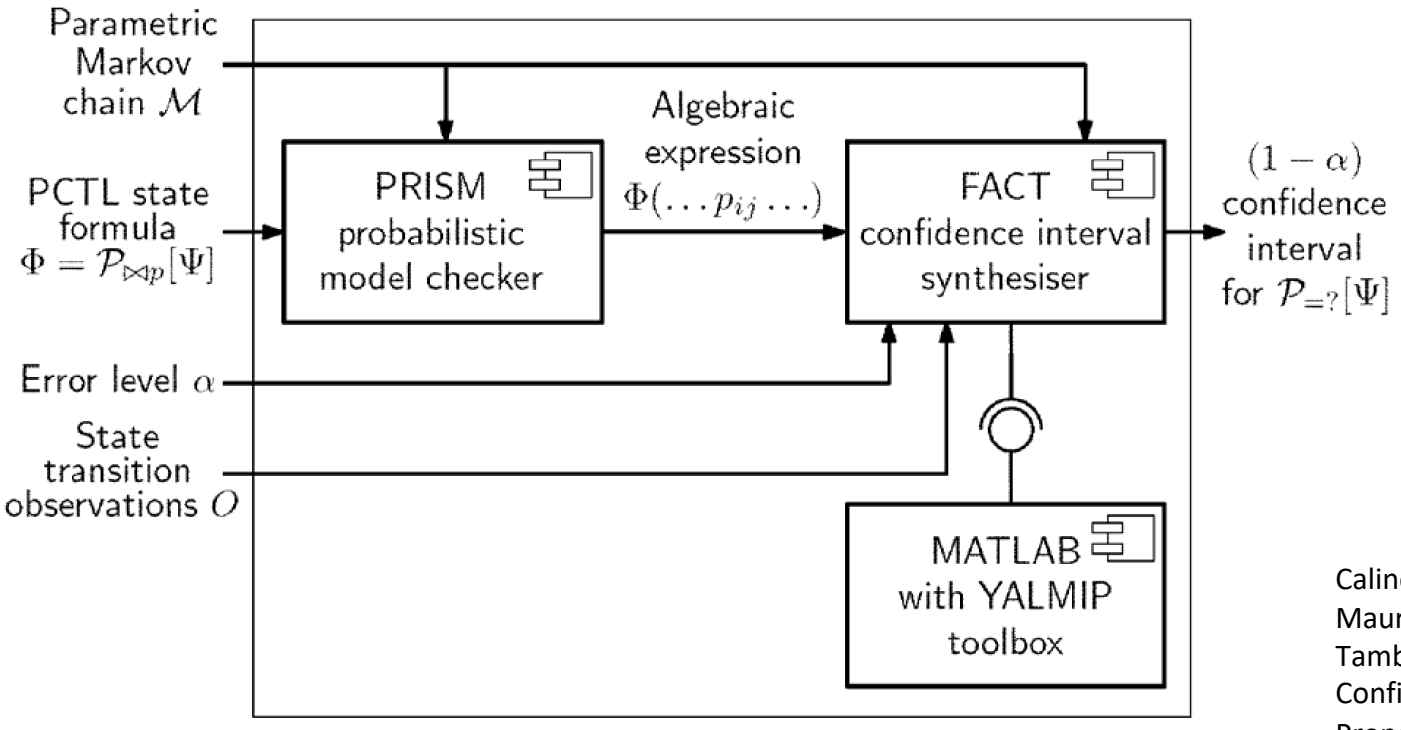
$$\text{Prob}(\text{Prop}(\pi \in paths^M(cruising) \mid \pi \vDash NoStall) \notin [a,b]) < \alpha$$

e.g.,

$$\text{Prob}(\text{Prop}(\pi \in paths^M(cruising) \mid \pi \vDash NoStall) \notin [.95,.99]) < .05$$

By Frank Murmann - Own work, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=68211518

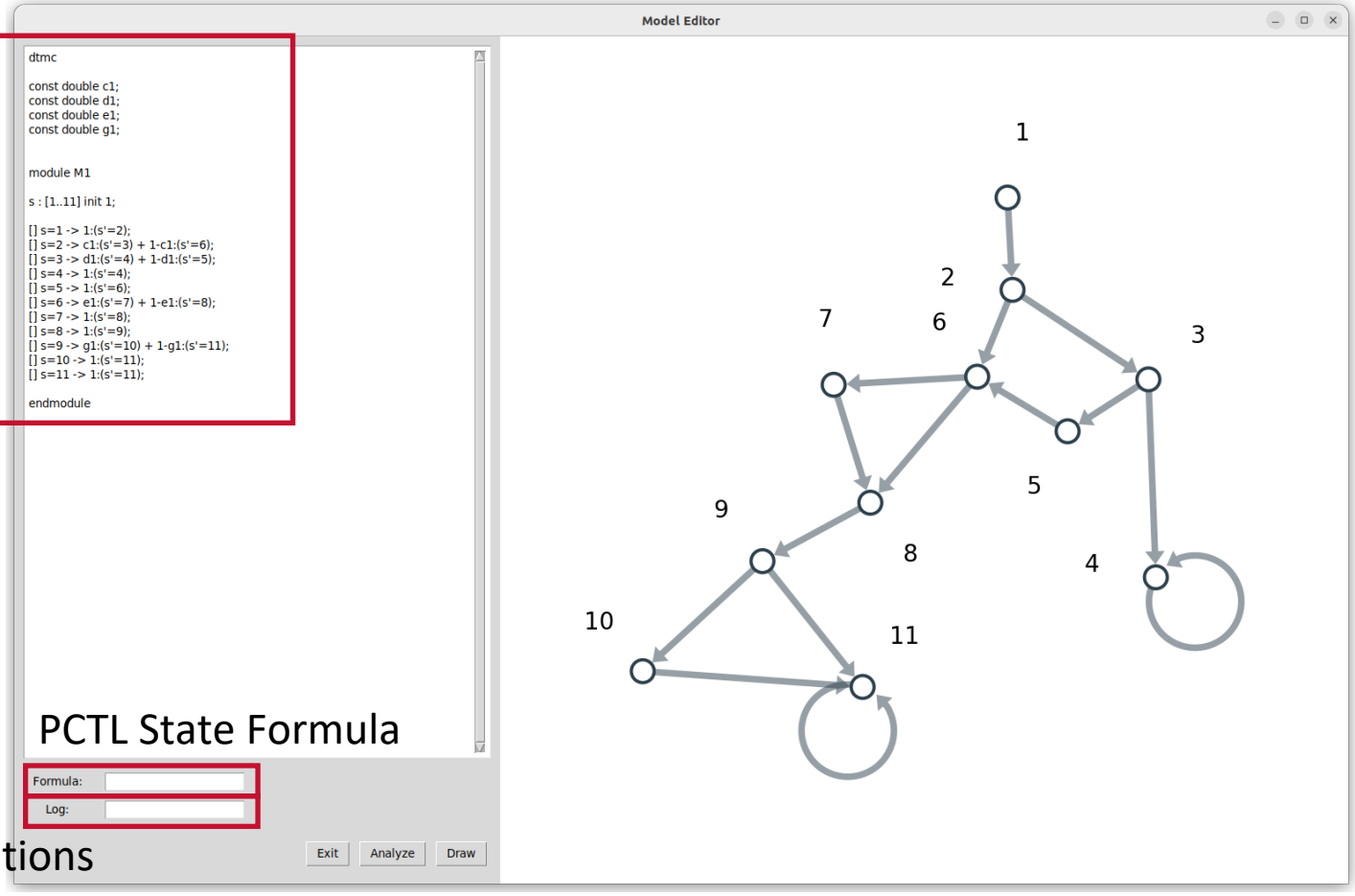
The FACT Technique: Architecture



Calinescu, Radu, Carlo Ghezzi, Kenneth Johnson, Mauro Pezzé, Yasmin Rafiq, and Giordano Tamburrelli. 2016. "Formal Verification With Confidence Intervals to Establish Quality of Service Properties of Software Systems." *IEEE Transactions on Reliability* 65 (1): 107–25. https://doi.org/10.1109/TR.2015.2452931.

Verification Tool

Parametric Markov Chain



State Observations

Instrumentable Implementation

Goal: Executable code we can instrument

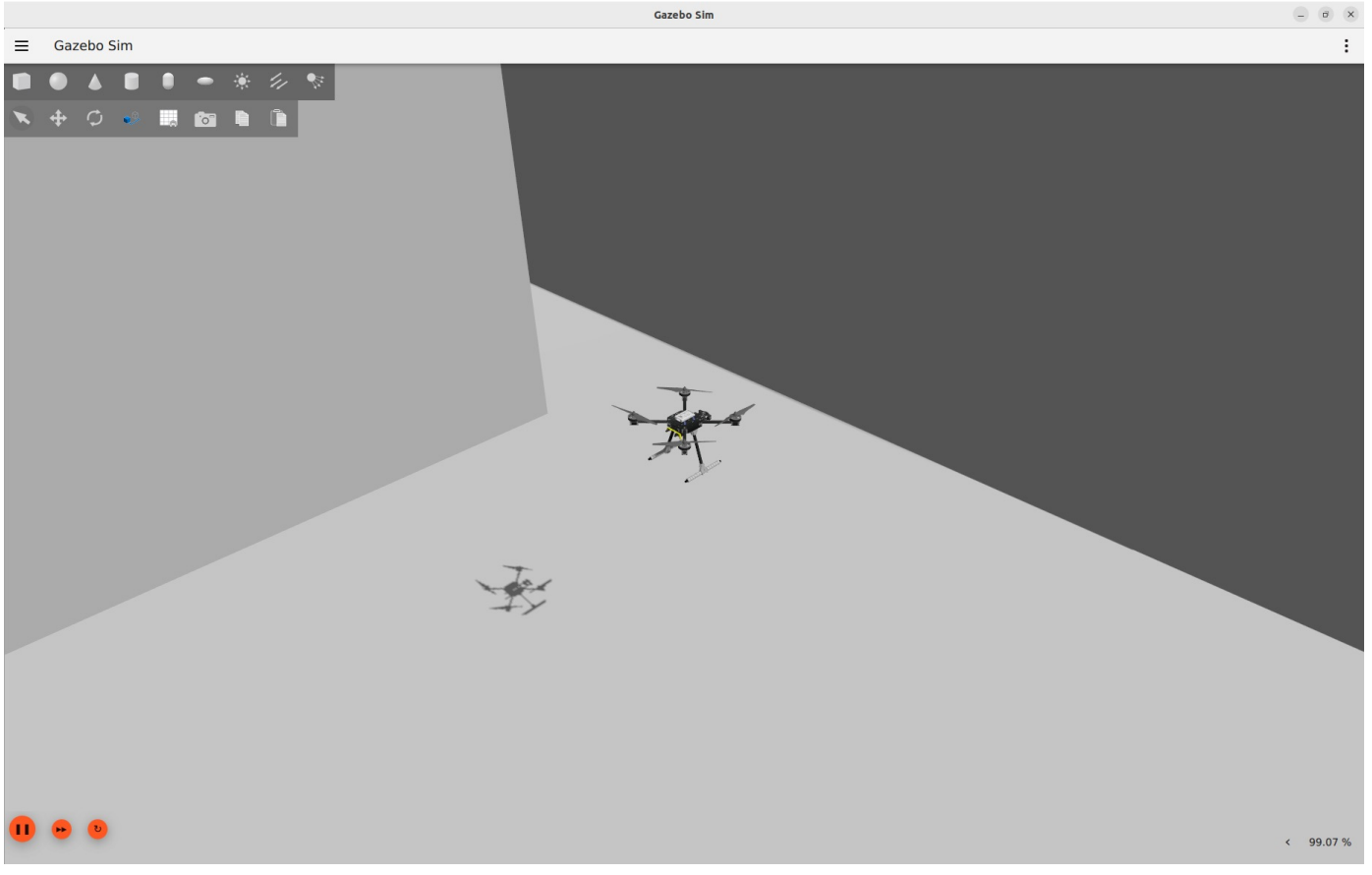
- Need to be able to observe / map to transitions in Markov Chain

Use PX4 Autopilot and the Gazebo simulator.

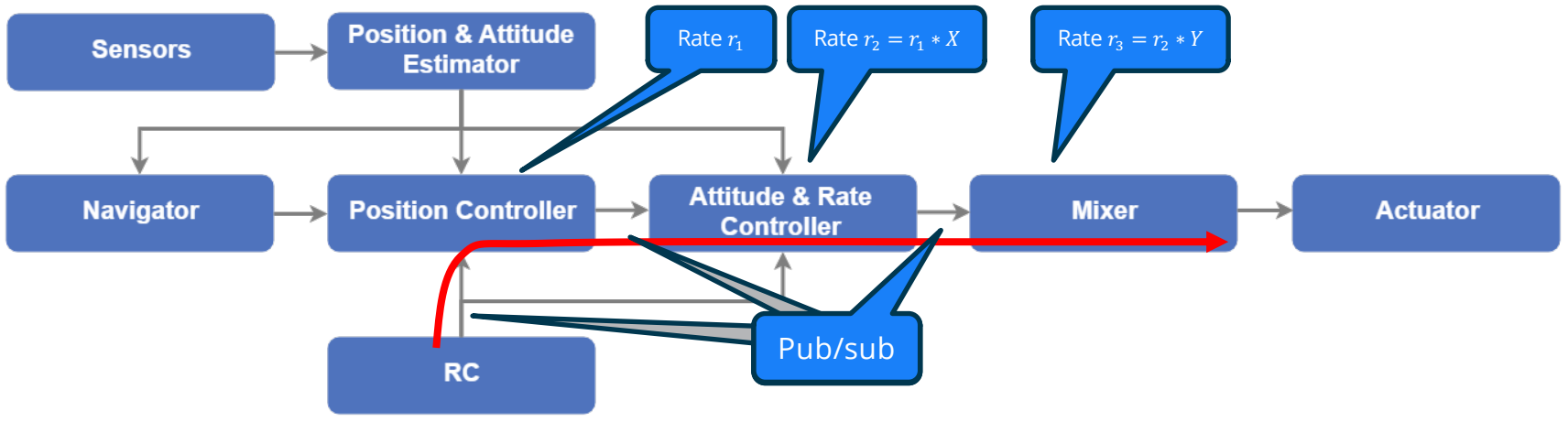
Logo from https://github.com/PX4/PX4-Autopilot

Logo from https://gazebosim.org

PX4 + Gazebo Simulator



Next Steps: Pub/Sub & PMC



Publish / subscribe architectures create asynchrony in data and control flow. This breaks assumptions of Probabilistic Model Checking!

Controller architecture uses multiple control loops at different rates

Next Steps: Collaborations

Source code

- https://github.com/dionisiodeniz/PX4-Autopilot
- https://github.com/dionisiodeniz/ProVerCert

Improvements

- Are there certification aspects we haven't considered?
- Statistical techniques that might be helpful?
- System properties you'd like to see included?

Let's talk! You can reach me at: sprocter@sei.cmu.edu