National Aeronautics and Space Administration

MARGInS: Model-based Analysis of Realizable Goals In Systems

Misty Davies, Tom Pressburger, Yuning He, Karen Gundy-Burlet
NASA Ames Research Center

10 June 2014

Motivation:

- NASA's Constellation & Orion Efforts:
  - *How can we analyze at complex system scales?*
  - *What are the margins to failure in a complex system?*
  - *Where should further study be directed to reduce uncertainty and increase safety margins?*
- Aviation Safety:
  - *How can we learn behaviors within hybrid systems?*
  - *How can we quantify the uncertainty in our predictions about these systems?*
  - *Can we leverage black-box and white-box testing in combination to learn more about potential system behaviors?*

Applications & Results:

- NASA's Pad Abort 1 Simulation and Experiment and the Exploration Flight Test 1:
  - *Can we learn the margins to failure?*
  - *Can we effectively analyze off-nominal conditions with hundreds of inputs and outputs over tens of thousands of runs?*
- Adaptive Flight Control:
  - *Can we automatically quantify types of behaviors from time series?*
  - *Can we predict the time series?*
  - *Can we predict a different key parameter (like time-to-failure or a failure boundary) directly from a current input state?*
- System-Level Safety Test Case Generation:
  - *For a nonlinear system in combination with a unit that can be white-boxed, can we leverage a combination of machine learning and formal techniques to exercise the unit from system-level inputs?*
- Terminal TSAFE – air traffic control:
  - *Can we predict a failure boundary in the system given two input states (for vehicles).*

Future Plans:

- Integration and release of the toolchains.
- Expansion of analysis to time series inputs.
- System-level test case generation for the aerospace domain.
- Pareto frontier generation.

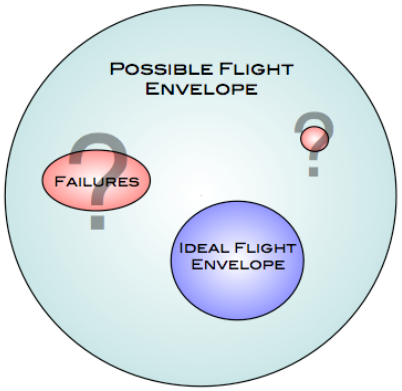
Pad Abort 1 (PA-1) was the only test of the launch abort system for Orion.
Modeling and Simulation (M&S) was expected to explore a wide range of design alternatives to identify a robust, safe Pad Abort system.
The PA-1 models and simulators were high fidelity, and provided iterative analysis for engineering the Pad Abort system and PA-1 flight test. The analysis drove redesigns.

Question: Can we characterize PA-1 behavior over the possible flight envelope (Possible), instead of just the ideal envelope (Ideal)?

Traditional "black box" validation testing typically *validates that the system has acceptable behavior for an isolated operating point.*
It ignores trends, sensitivities and emergent behavior in the dataset.

Problem: Expanding simulations over the possible flight envelope means that there are many runs (10K to 100K) over 100's of variables. Finding sensitivities in this space would require hours upon hours of expert time

Problem: Pure statistical correlation often fails to find key parameters.



Problem Domain:
Large (thousands of independent variables), complex, non-linear, with interacting modal, continuous, periodic and stochastic parameters.

The Unreasonable Effectiveness of Data

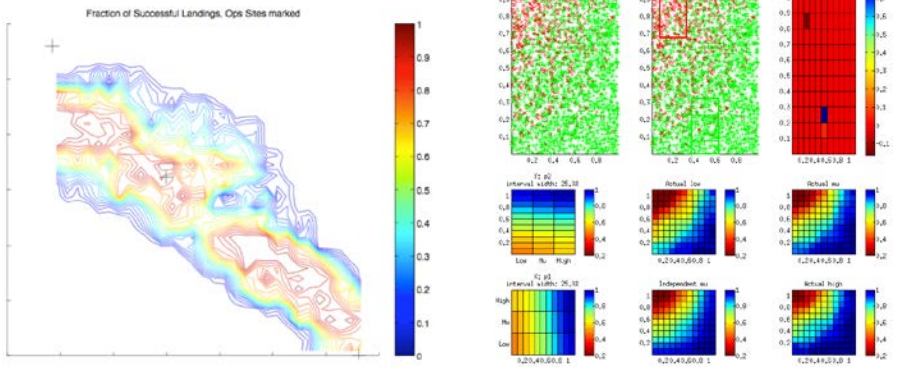
--Alon Halevy, Peter Norvig, and Fernando Pereira

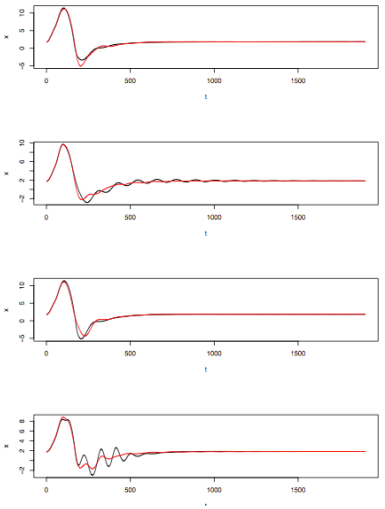
Scene Completion Using Millions of Photographs
--James Hayes, Alexei Efros (CMU)

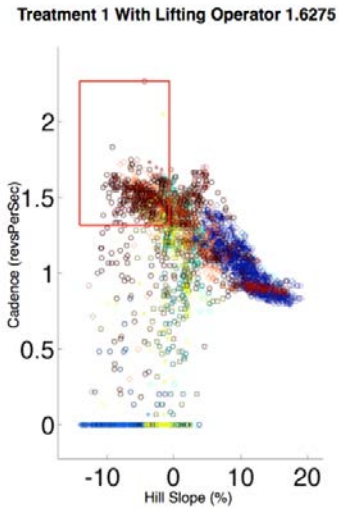
MARGInS (Model-based Analysis of Realizable Goals in Systems)

MARGInS is a set of machine learning and statistical libraries for system testing.

- Finds novel features in test suites
- Automates the finding of 'rules' that determine classes of behavior (e.g. safe/unsafe)
- Proposes new experiments to explore the boundaries between classes of behavior
- Creates visualizations for aiding analyst understanding



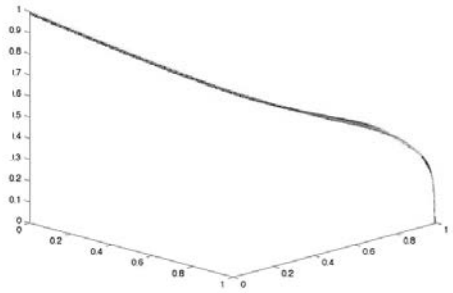




Pad Abort 1 End Result: Much larger testing space for the same level of effort.

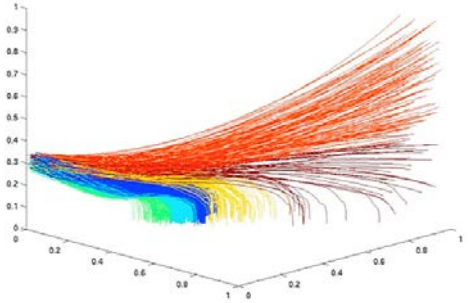
Standard testing

No automated analysis



Testing using MARGInS

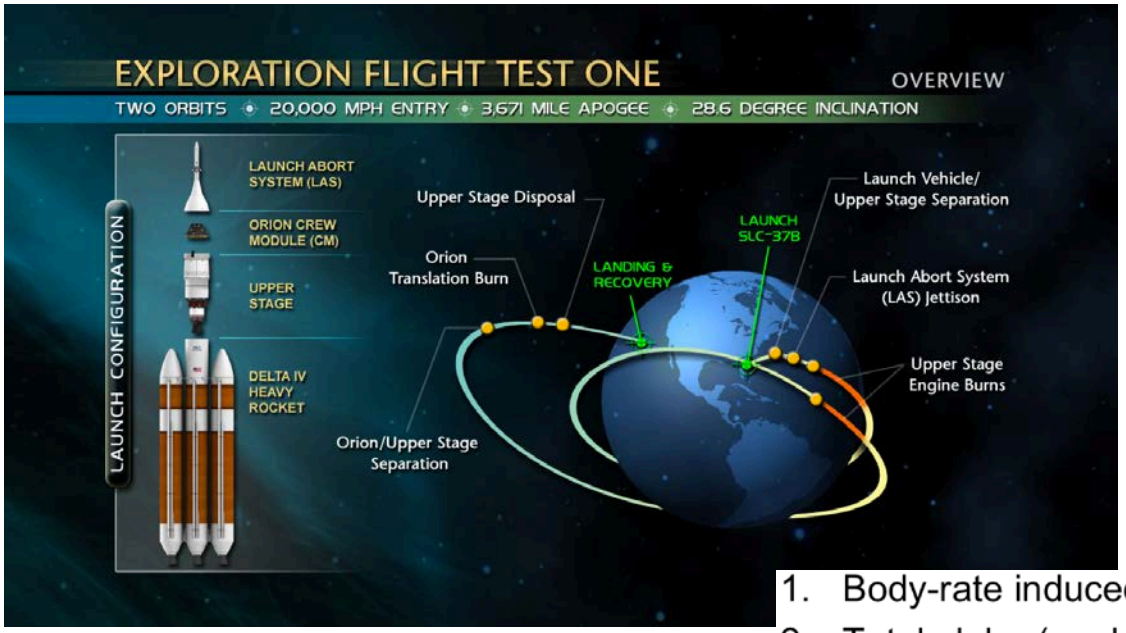
Color indicates “risk classes”



MARGInS enabled us to run massive experiments across a wide range and focus down to find the root cause of a problem. Reduced guessing and false leads.

This allowed us to troubleshoot unexpected results:

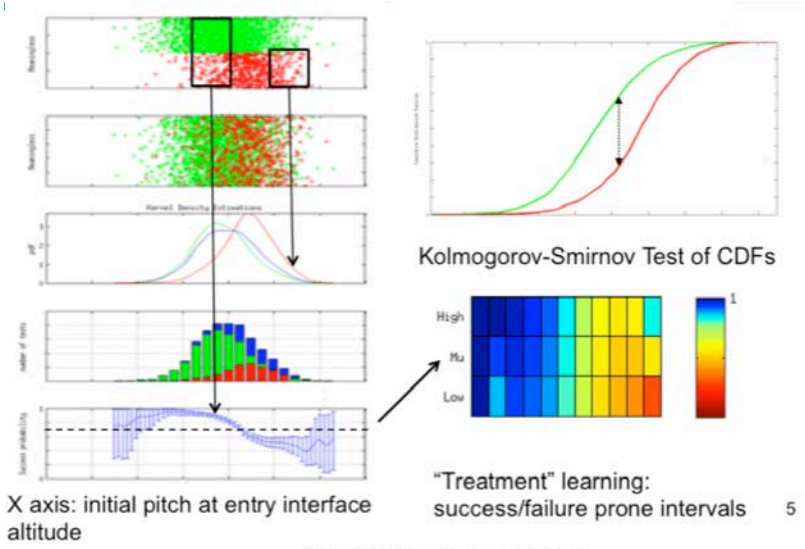
- Thrust instability – why? (A mixture of physics and model problems.)
- How do errors in the GRAM winds models (widely used environmental models) impact PA-1 experiments?

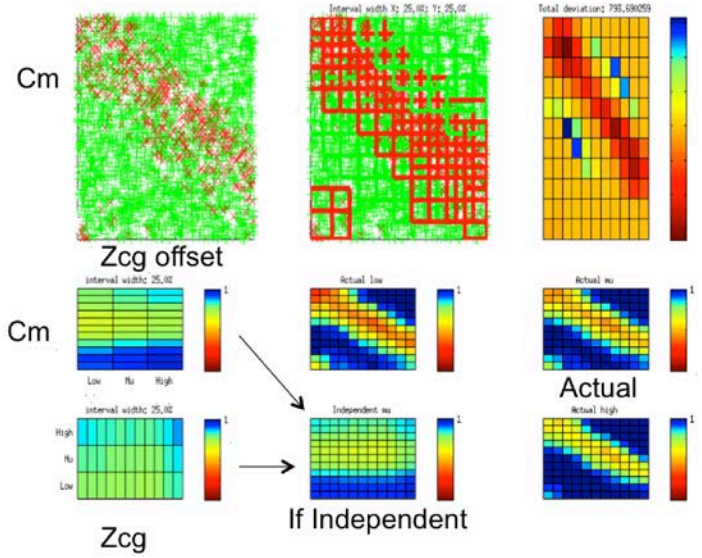


Orion EFT-1 Results: MARGInS tool extended (Critical Factors Tool – Pressburger 2013) to determine measures of influence.

Objectives and Constraints for the EFT-1 Sim:

1. Body-rate induced Forward Bay Cover Jettison
2. Total alpha (angle of attack) at FBC Jettison
3. Touchdown heading
4. Range at touchdown to target
5. Total Reaction Control System propellant used
6. Thruster pulse count
7. Number of instances of simultaneous thruster firing
8. Bank saturation
9. Maximum aerodynamic load
10. Backshell temperature, heat load, etc.





EFT-1 – the Critical Factors Tool

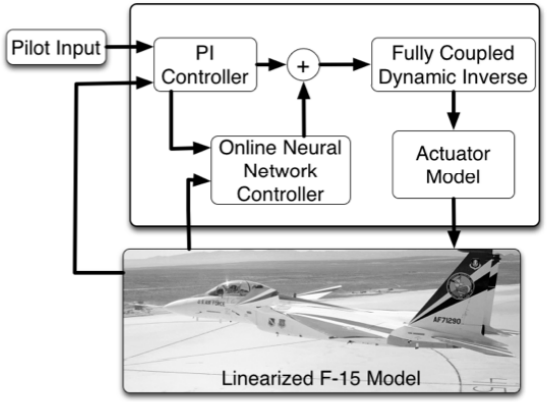
1D and 2D analyses map input conditions to risk for failure.

In turn, the critical input factors for each of the objectives and constraints are identified, and put into a table.

Domain experts examine results to understand drivers of behavior and to suggest further refinement of models.

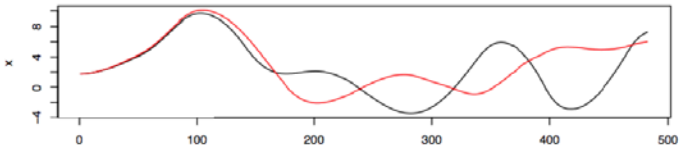
Visualizations strongly affect the domain experts ability to see and to believe the effects.

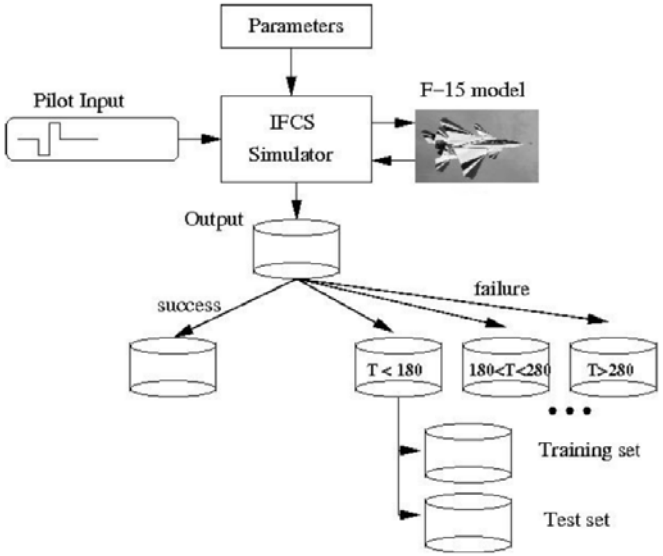
	Body-rate Induced FBC Jettison	Total Alpha at FBC Jettison	Touch-down Heading	Range to Target	RCS Prop Used	Thruster Pulse Count	Instances of Simultaneous Thruster Firing	Bank Saturation	Aero-dynamic Load	Backshell Temperature
dCm/dq	++	?	?		++		+			
Cm	++ D	++ D		+ D				++	++	++ D
Cn			+ U		++ U	++ U		+	? D	?
CD								?	?	+ D

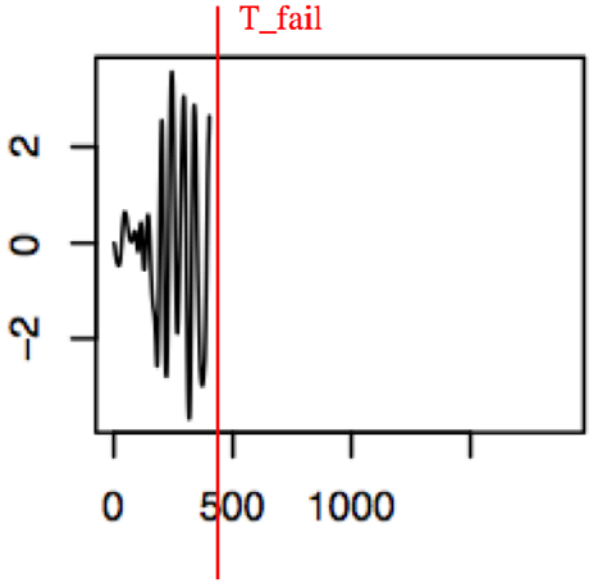


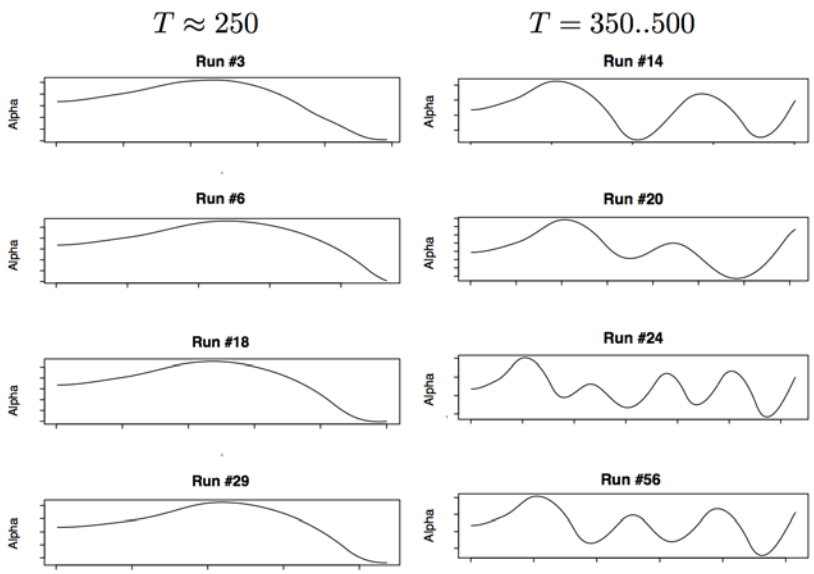
Adaptive Flight Control—Predicting Time Series

In order to determine the failure boundaries for trajectories with humans tightly-coupled in the loop, we need to be able to predict highly-nonlinear time series of varying length.





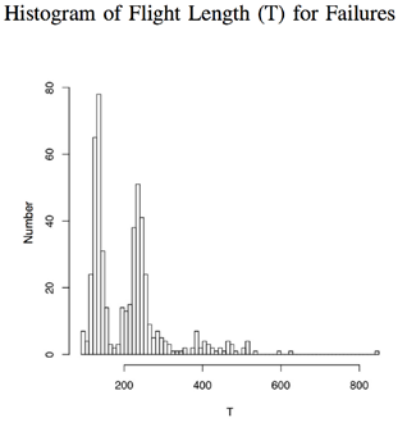


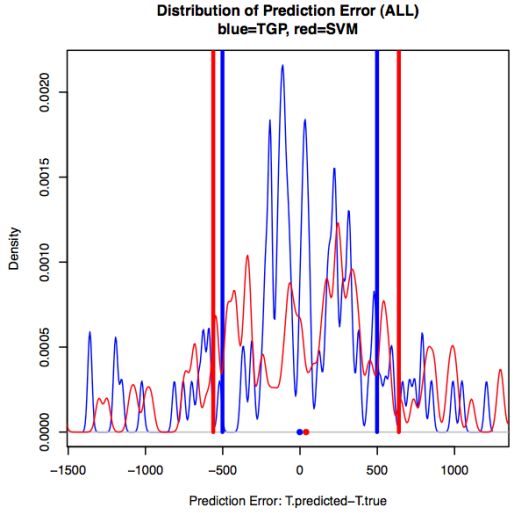


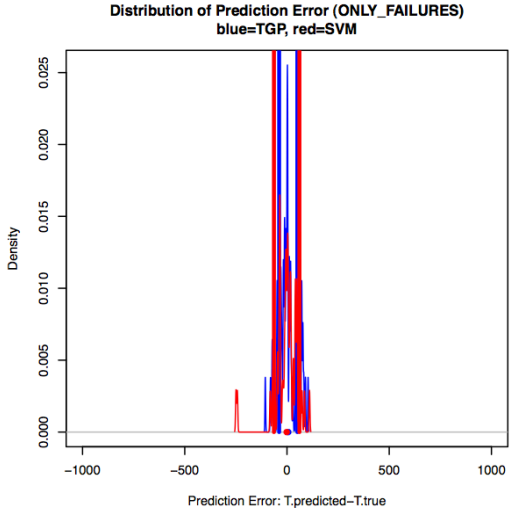
Adaptive Flight Control—Predicting Time Series

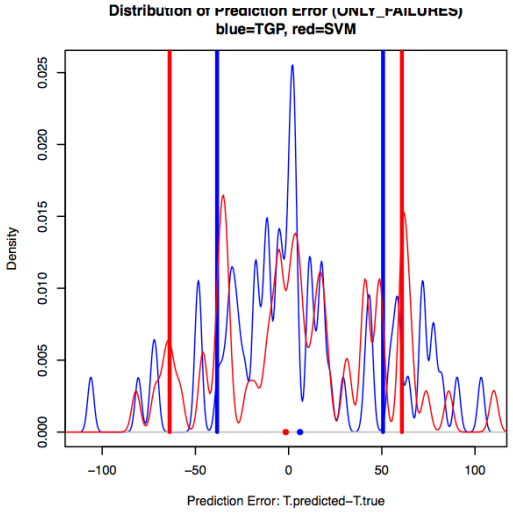
By analyzing the full trajectories, we were able to find classes of behaviors that were strongly correlated with time-to-failure.

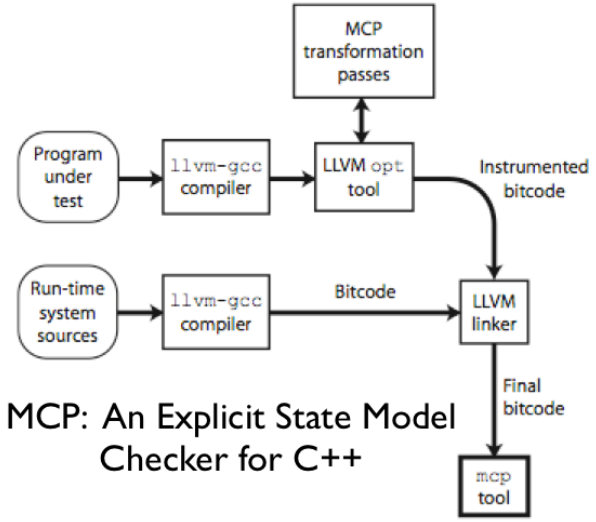
This required new statistical techniques (He, 2012) for predicting varying-length time series.











MCP: An Explicit State Model Checker for C++

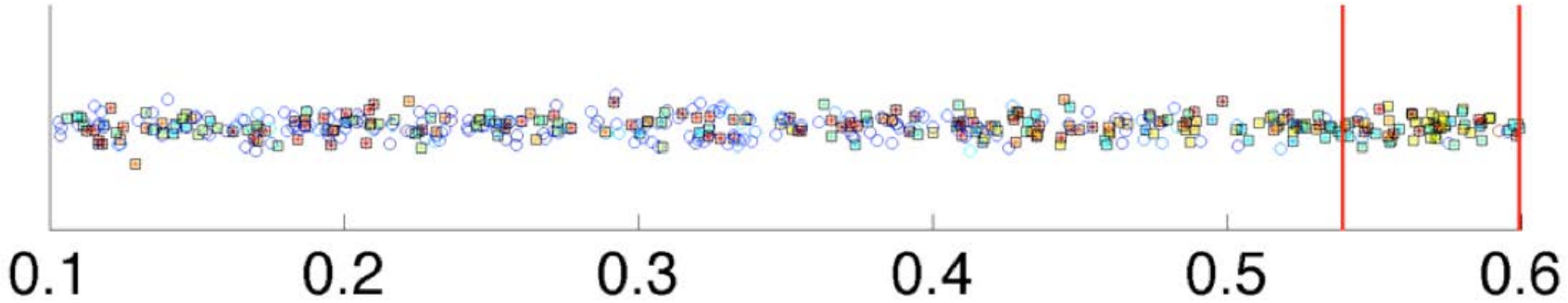
Adaptive Flight Control—Combining Testing with Model Checking

The IFCS sim also contained a computer error – occasionally would produce NANs.

Neither model checking nor MARGInS alone could find the error.

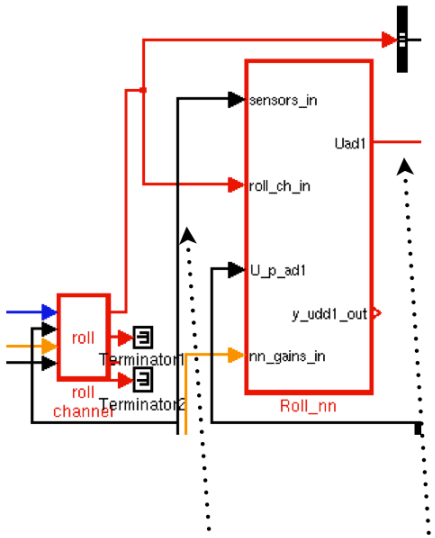
MCP ran out of memory the first time through the OLNNs

With the OLNNs removed (PID control only), MCP ran out of memory after 7 times through the loop.



NAN errors are most associated with roll gains (but the correlation isn't perfect).

Strategy: Global machine-learning based testing gave us system-level inputs that would lead to failure after some time steps. The system was already decomposed (Simulink).

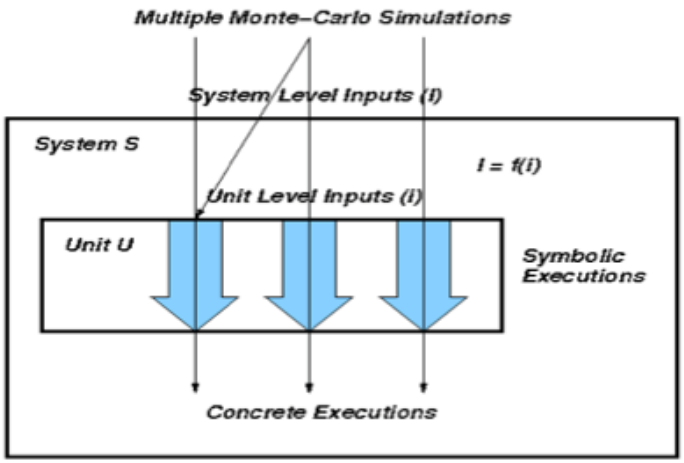


Instrumentation

Adaptive Flight Control—Combining Testing with Model Checking

Strategy: Flag the first NAN, treat the inputs to the module for that time step as the test case for the model checker.

```
---
** starting the model **
--- Step #0
--- rtu_U_p_ad1 = 5.204576e+05 (size 8)
--- mf1 = -7.478910e+01
--- mf1 = -5.205324e+05
--- mf1 = 5.205324e+05
--- mf1 = inf
--- mf1 = inf
--- mf1 = 0.000000e+00
--- mf1 = 0.000000e+00, lb->mf1 = 0.000000e+00
--- u1dd = nan
### GEN2_w_test2_Roll_nn.c(348): Assertion failure: 0
```



System-level aerodynamic equations (giving Mach number from Pitot tube sensor) highly non-linear across two regimes. *Strategy: learn an approximation to the behavior.*

Digital DATCOM provides an estimate of drag coefficient – branching is linear in the Mach number. *Strategy: explicitly solve for the Mach number and friction coefficients that will exercise each path using concolic execution.*

Airplane system test case generation—Machine learning plus concolic execution.

System Inputs (I): Pt, Ps, Alt

Subsonic: $$Ma = \sqrt{5\left[\left(\frac{P_t}{P_s}\right)^{\frac{0.4}{1.4}} - 1\right]}$$

Supersonic: $$\frac{P_t}{P_s} = \left(\frac{5.76Ma^2}{5.6Ma^2 - 0.8}\right)^{3.5} \frac{2.8Ma^2 - 0.4}{2.4}$$

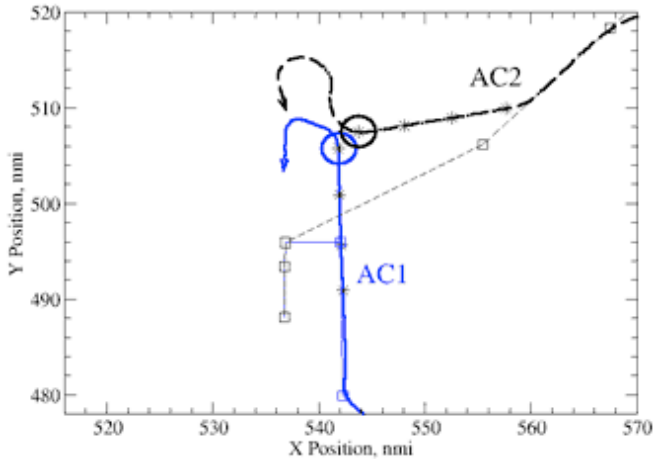
After 25 tests—

N-factor:
16 covered,
10 uncovered

Model-based:
21 covered,
12 not covered

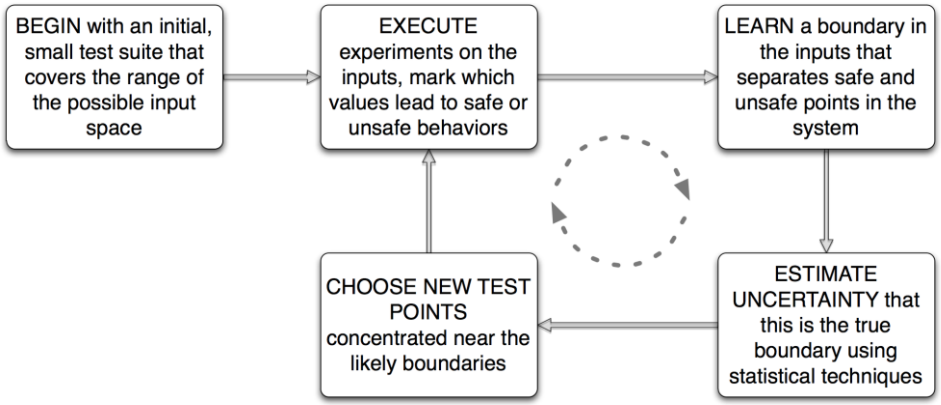
Unit Inputs (i): Ma, Cf, Cfbterm, Cfterm

```
[Tree]
9  (Cf > CfTerm) (C, ...)
10    (Ma >= (780000 / 1000000)) (C, ...)
11      (Ma > (1040000 / 1000000)) (C, ...)
12        (Ma >= (600000 / 1000000)) (C, ...)
13          (Cfb > CfbTerm) (C, ...)
14            (Ma >= 1) (C, ...)
15              (Ma <= (2000000 / 1000000)) (C, ...)
16              (Ma > (2000000 / 1000000)) (C, ...)
17            (Ma < 1) (S, ...)
18          (Cfb <= CfbTerm) (S, ...)
19        (Ma < (600000 / 1000000)) (S, ...)
20      (Ma <= (1040000 / 1000000)) (S, ...)
21    (Ma < (780000 / 1000000)) (S, ...)
22  (Cf <= CfTerm) (S, ...)
```

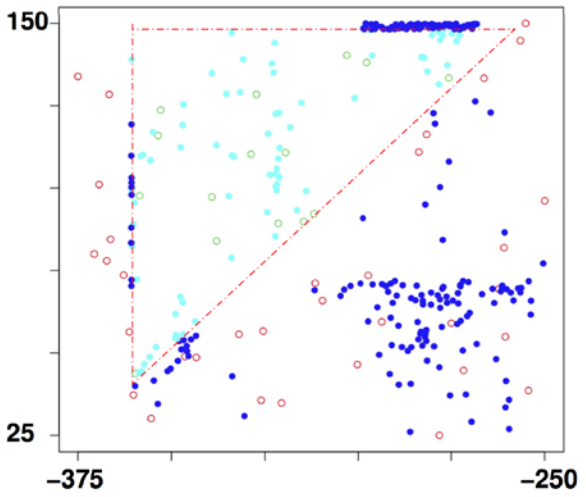


Terminal TSAFE—uncertainty quantification and failure envelope detection.

Using the algorithm in the flow chart, we automatically-detected a safety boundary for a conflict detection algorithm. The axes are the altitude offsets for two planes landing at the same airport. The triangle is a region in which the time to loss of separation is unacceptable. The solid blue and cyan points were automatically selected by our learning algorithm as it discovered the safety envelope.



Stop when the uncertainty is small enough or at maximum time (with the smallest statistical uncertainty possible in that time)



Lessons Learned:

1. For large, non-linear systems, you need a large input space to learn from.
2. The independence assumption inherent to many traditional statistical techniques is often NOT a good assumption.
3. You are likely to need to look at a combination of scalar and time-series behaviors to understand aerospace systems.
4. Analysts and domain experts need pictures in order to understand what you are telling them – wherever possible.
5. A combination of machine learning-based testing and more formal techniques can get you farther than either alone.

Current efforts:

1. Analyses that allow for time-series based inputs.
2. Pareto Frontier-based analyses.
3. Clean-up and integration of all of our current tools. (Target date: October 2015)