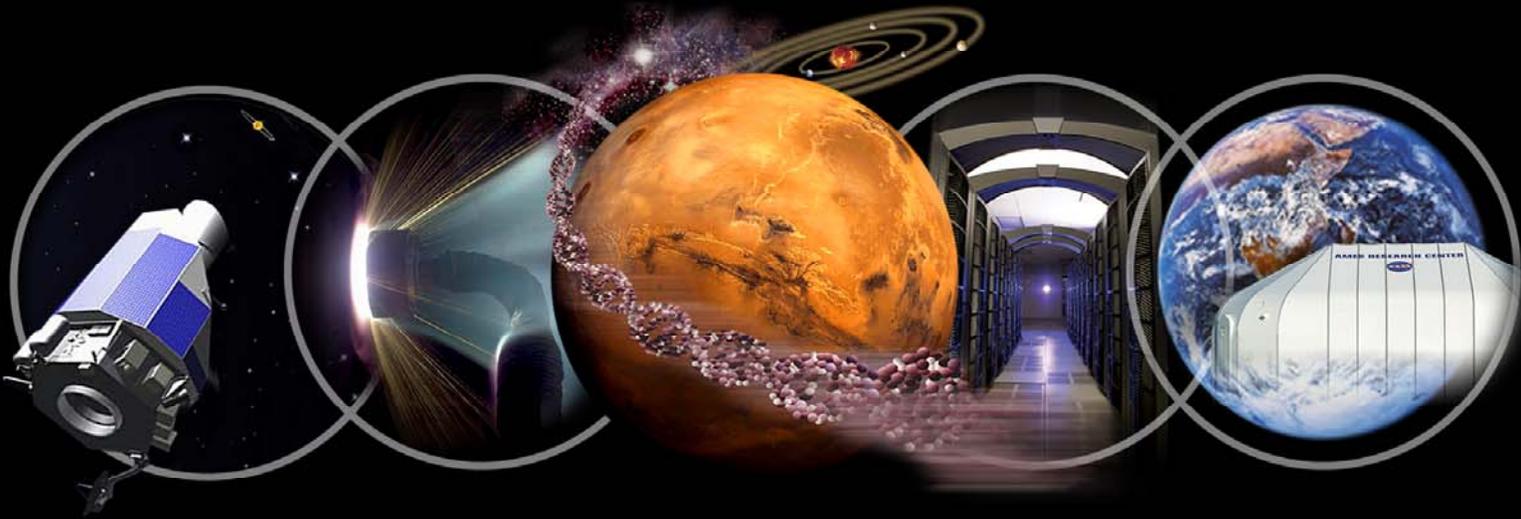


Discovery ➡ Innovation ➡ Solutions



Robust Software Engineering

Dr. Michael R. Lowry

Area Lead: Automated Software Engineering

Ames Exploration Systems Technology Partnerships Forum

July 22-23, 2004



Visibility ➡ Excellence ➡ Impact





Exploration Needs in Software Engineering

Robust, capable systems : more system capabilities embedded in software.

Reliability : systems certified for manned flight, assuring humans are as safe as is reasonably achievable.

Affordable: achieve ambitious goals for exploration of Mars and beyond within a limited NASA budget.

Reconfigurable: deploying systems that can be reconfigured following initial deployment, to enable adaptation to new circumstances.

All within the context of sustained engineering spanning multiple decades.



Software Failures in Launch

Ariane 5 Failure



Background

European Space Agency's reusable launch vehicle

Ariane-4 a major success

Ariane-5 developed for larger payloads

Launched

4 June 1996

Mission

\$500 million payload to be delivered to orbit

Fate:

Veered off course during launch

Self-destructed 40 seconds after launch

Cause:

Failure to adequately V&V software from Ariane-4 for reuse in Ariane-5.

Unhandled floating point exception in Ada code - floating point to fixed point conversion overflow.

This type of error can now be caught with static analysis tools.



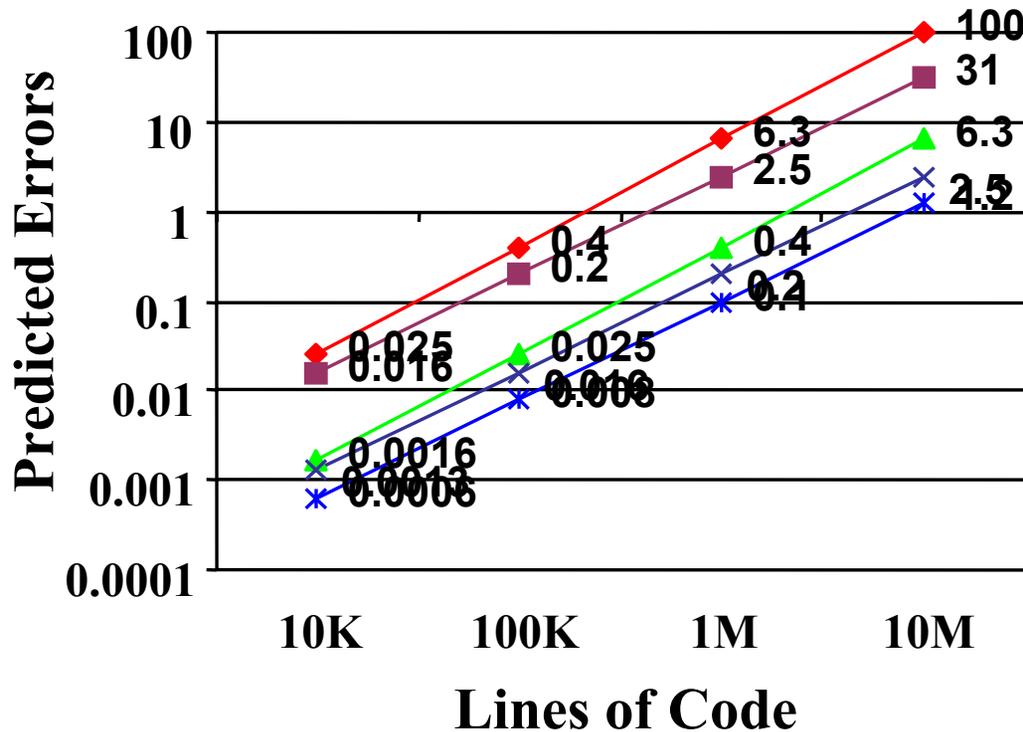
NASA's Unmanned Mars Program

Mission	Launch Date	Arrival Date	Outcome
Viking I Viking II	20 Aug 1975 9 Sept 1975	Landed 20 Jul 1976 Landed 3 Sept 1976	Operated until 1982 Operated until 1980
Mars Observer	25 Sept 1992	Last contact: 22 Aug 1993	Contact lost just before orbit insertion
Pathfinder	4 Dec 1996	Landed 4 July 1997	Operated until 27 Sept 1997
Global Surveyor	7 Nov 1996	Orbit attained 12 Sept 1997	Still operational
Climate Orbiter	11 Dec 1998	Last contact: 23 Sept 1999	Contact lost just before orbit insertion
Polar Lander	3 Jan 1999	Last contact: 3 Dec 1999	Contact lost before descent
Deep Space 2	3 Jan 1999	Last contact: 3 Dec 1999	No data was ever retrieved



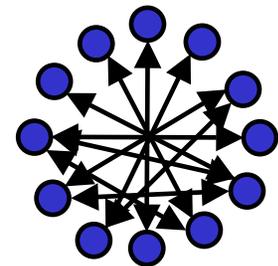


Predicted Number of Errors and Mitigations



Software size, S , increasing exponentially (doubling every three or four years).

Errors, cost over-runs, schedule slip due primarily to non-local dependencies during integration.



Errors increase faster than software size.

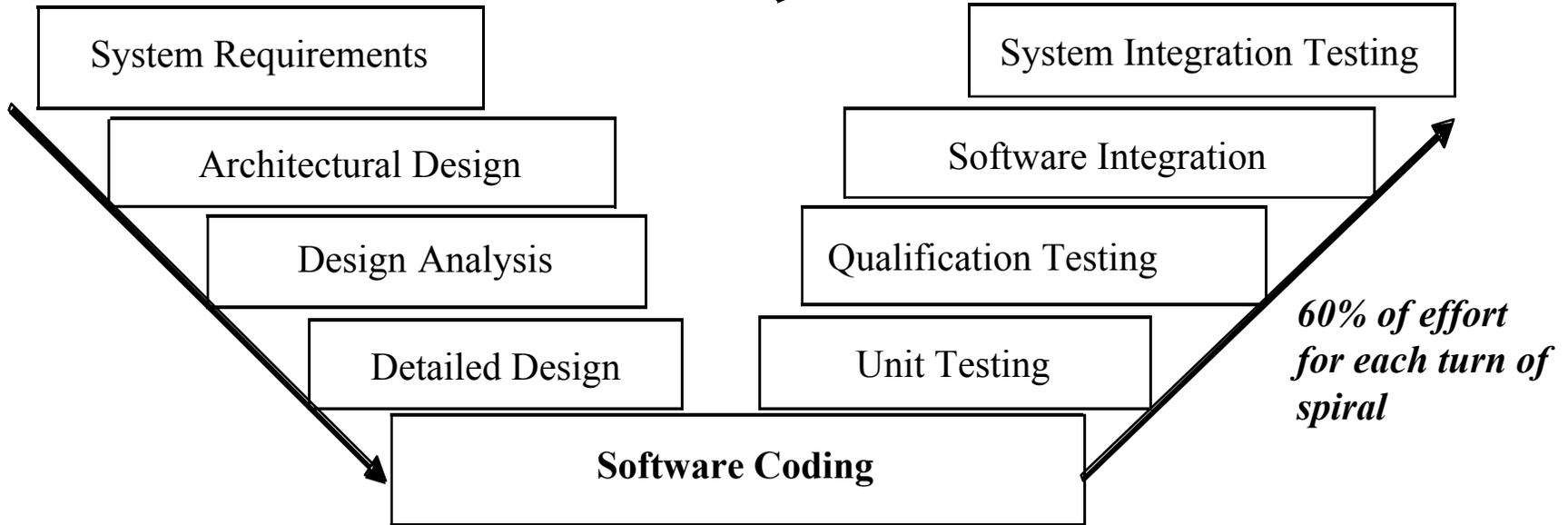
$$\text{Errors} = e \times S^N;$$

where S is the number of modules (LOC/M), and error rate $e = 1/10,000$



Cost Factors for Sustained Engineering

80% effort comes after initial deployment in subsequent spirals





Selected ARC Capabilities in Robust Software Engineering

Scaling mathematical approaches for software verification, validation, and integration to mission software.

Combining technologies for synergy and automation.

Hierarchical 'divide and conquer' approaches to verification, validation, and integration.

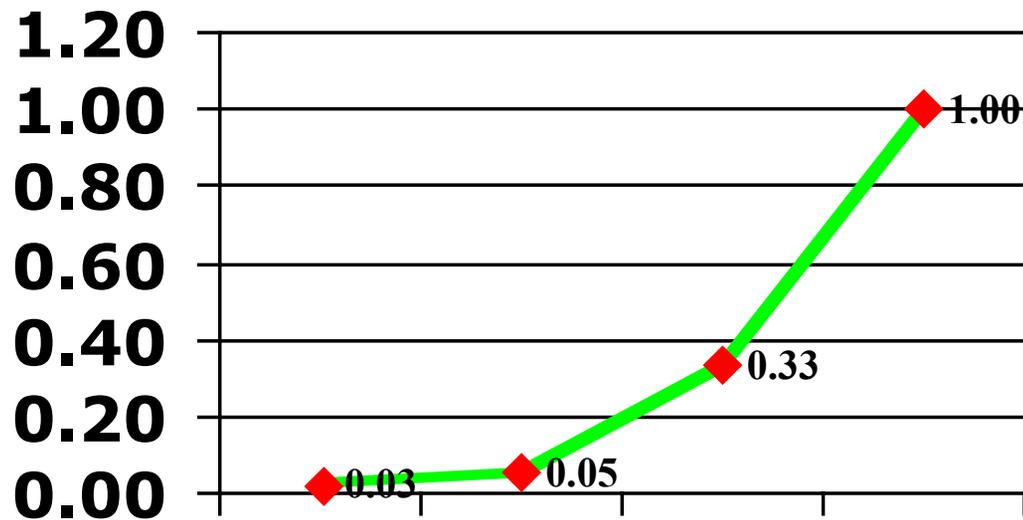
Automated software generation with built-in hooks for V&V.

Autonomy software V&V

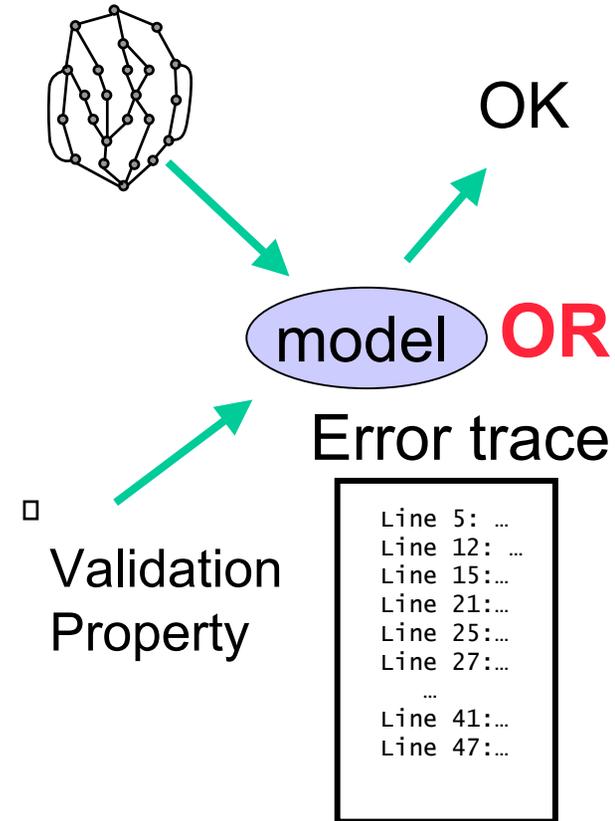
Adaptive software V&V



Scaling Model Checking Technology



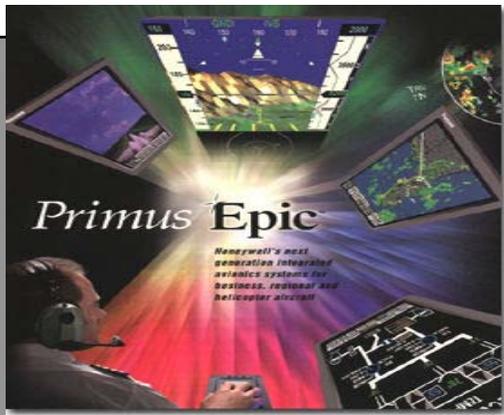
1997





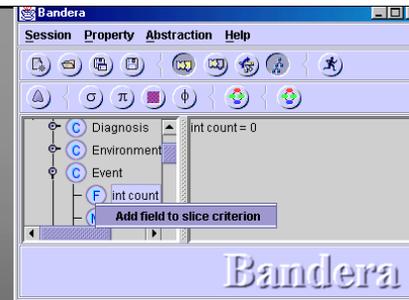
Synergy of Analysis Technologies

Bandera code-level debugging of error-path



← **Repair** →

Combined techniques allows $O(10^2)$ source line and $O(10^6)$ state-space increase over state of practice



Spurious error elimination during abstraction

2x Heuristic search
10x Focused search for errors

JPF

Model Checker

State compression
2x 15x

Partial-order reduction
2x 10x

DEOS
10000 lines to 1500

3x **Slicing** 30x

Property preserving

```
Case 0: new();
Case 1: Stop();
Case 2: Remove();
Case 3: Wait();
```

5x **Abstraction** 100x

DEOS
Infinite state to 1,000,000 states

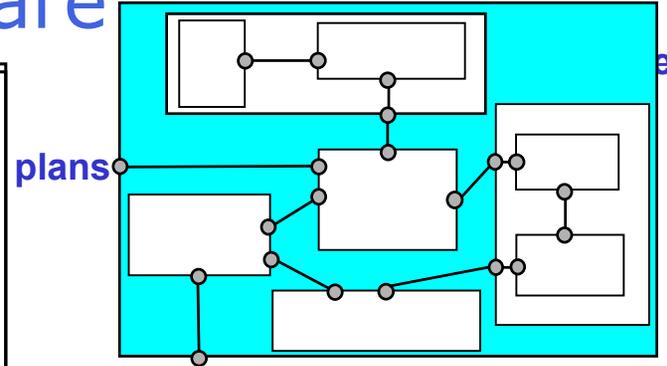
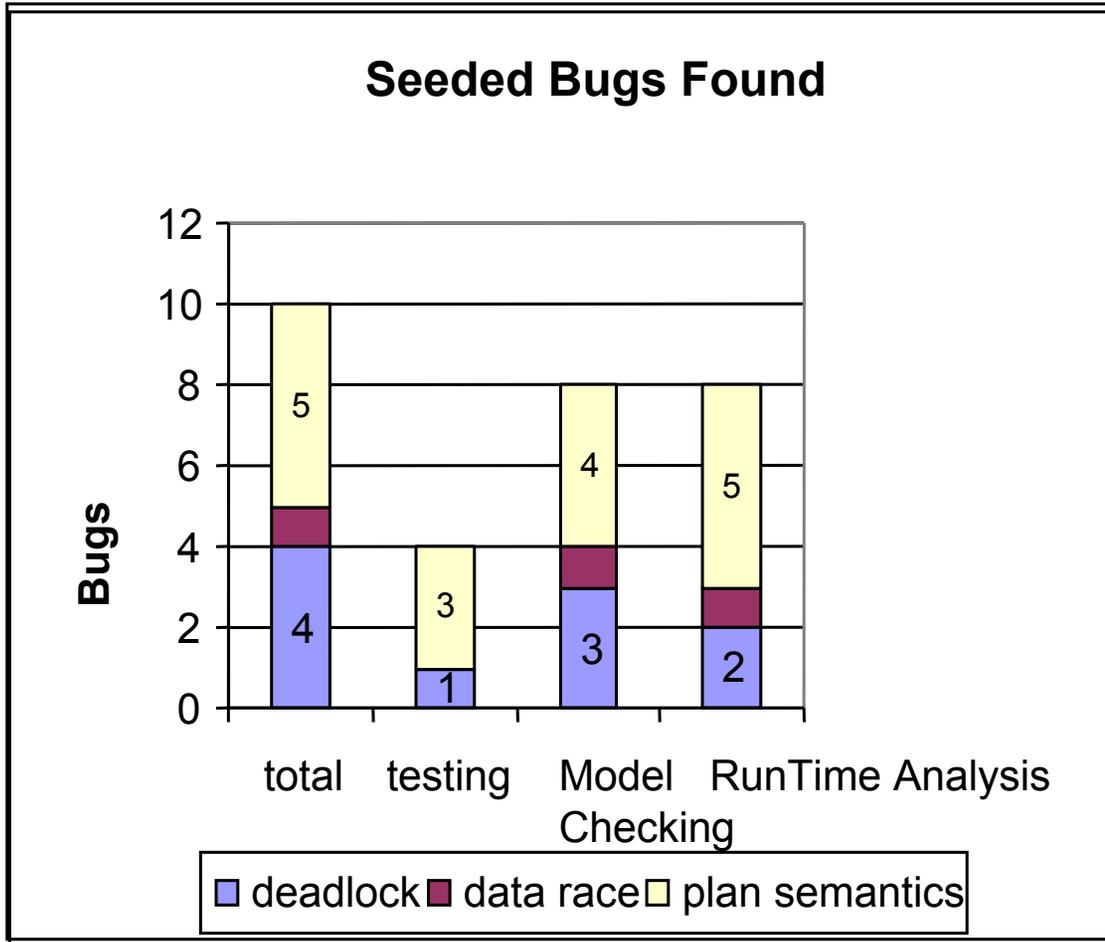
Environment Generation

Semi-automated and requires domain knowledge

```
Case 0: new();
Case 2: Remove();
```



Benchmarking V&V Technologies for Autonomy Software





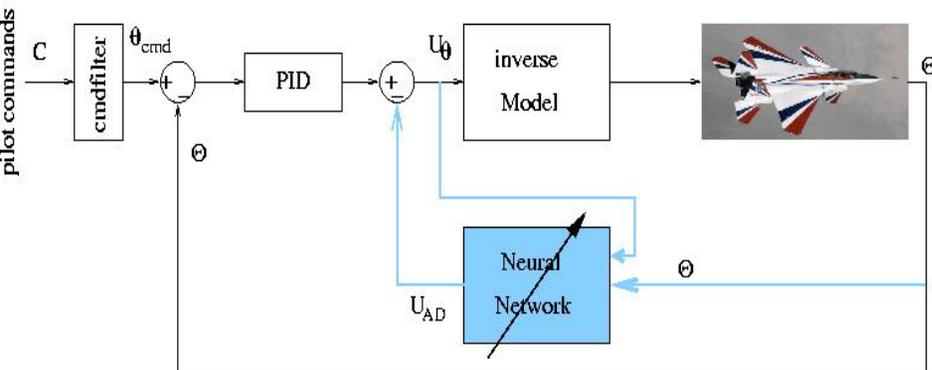
Adaptive Control V&V



Fixed gain controllers cannot deal with *catastrophic changes* or *degradation* in plant
Adaptive systems (e.g., NN) can react to unexpected situations through learning

Relevance and potential:

- IFCS NN controlled aircraft (F-15, C-17)
- Space exploration
- Any safety-critical application of NN control



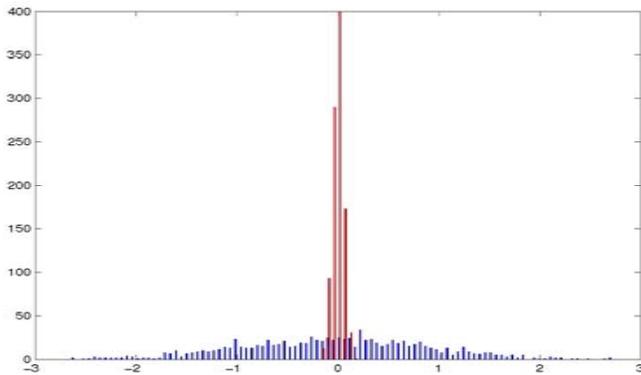
Network “learns” to compensate for deviations between plant and model

The major obstacle to the deployment of adaptive and autonomous systems is being able to verify their correct operation – *In Flight*



Bayesian Approach to Adaptive Control Verification and Validation

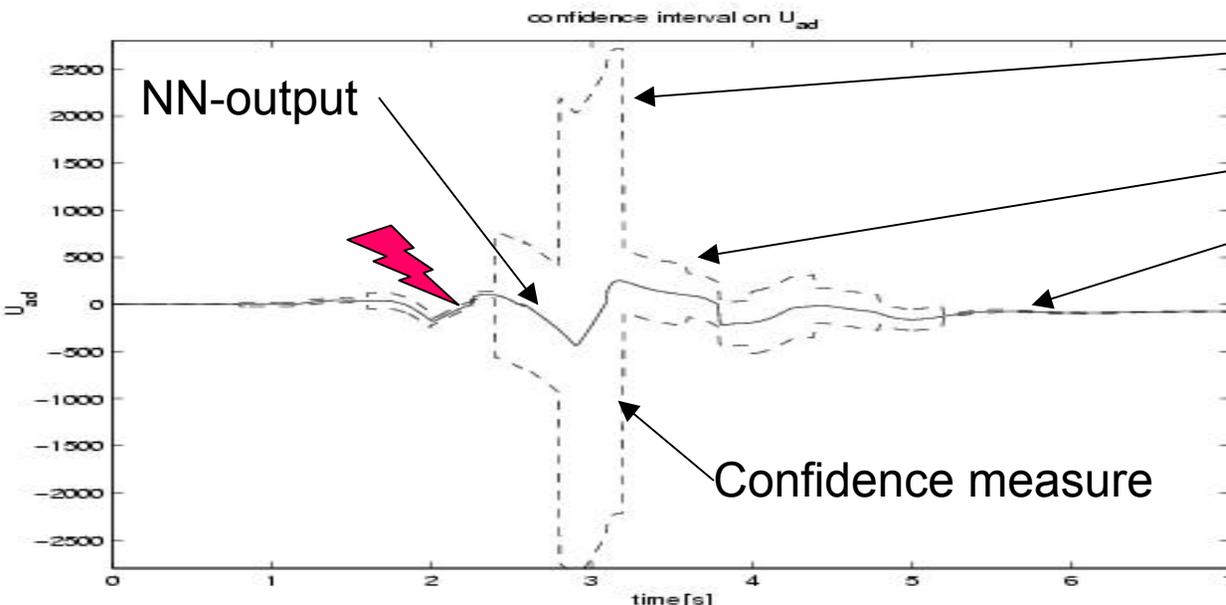
Look at the *probability distribution* of the NN output



- Small variance = good estimate
- Large variance = bad estimate
no reliable result, just a guess

Variance (confidence measure) depends on:

- How well is the network trained?
- How close are we to “well-known” areas?



- Bad confidence: controller still has to adapt
- Adaptation in progress
- Successfully adapted



Teaming Opportunities

Team to develop & mature new approaches to Exploration Software Engineering

Reliability for software-based capabilities for manned flight

Affordable sustained engineering

Rapid reconfiguration

Modular systems

Multi-disciplinary teams to enable new capabilities to be verified, validated, and certified

Autonomy systems

- On-board decision making
- Model-based autonomy
- In-space and extra-terrestrial robotic surface operations

Human-computer interactions

Adaptive systems