Model Checking of Aerospace Domain Models in an Industrial Context

Michael Dierkes
Rockwell Collins France

Forum Méthodes Formelles
16 Octobre 2014

Rockwell Collins

Agenda

1. Presentation of Rockwell Collins
2. The RC formal analysis framework
3. Case studies
   - Adaptive Display & Guidance System
   - UAV Flight Control System
   - Effector Blender
   - Triplex Sensor Voter

} US

France

Presentation

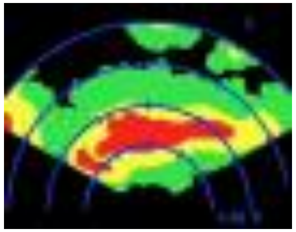
ROCKWELL COLLINS

Who Are We?

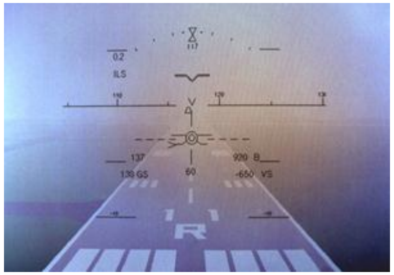
A World Leader In Aviation Electronics And Airborne/ Mobile Communications Systems For Commercial And Military Applications



- Communications
- Navigation



- Automated Flight Control
- Displays / Surveillance

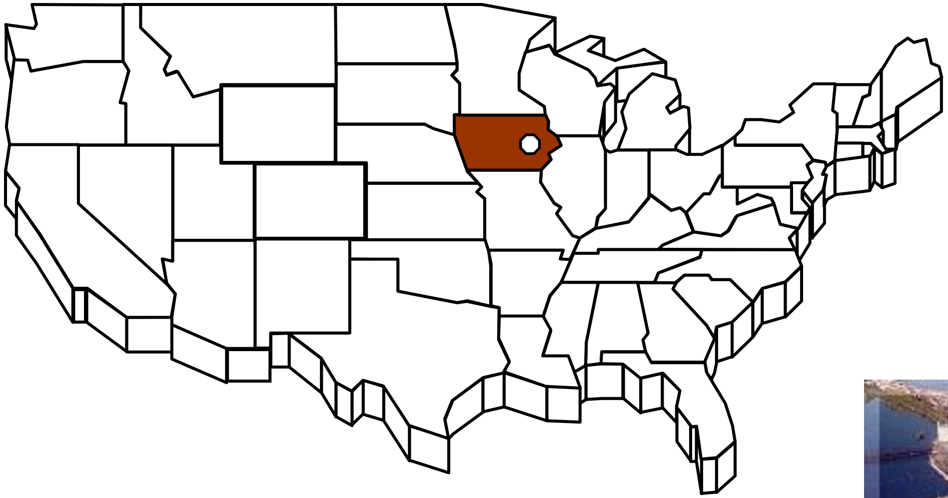



- Aviation Services

- In-Flight Entertainment
- Integrated Aviation Electronics
- Information Management Systems

Rockwell Collins

Headquartered in Cedar Rapids, Iowa
~20.000 Employees Worldwide
Present in 27 countries

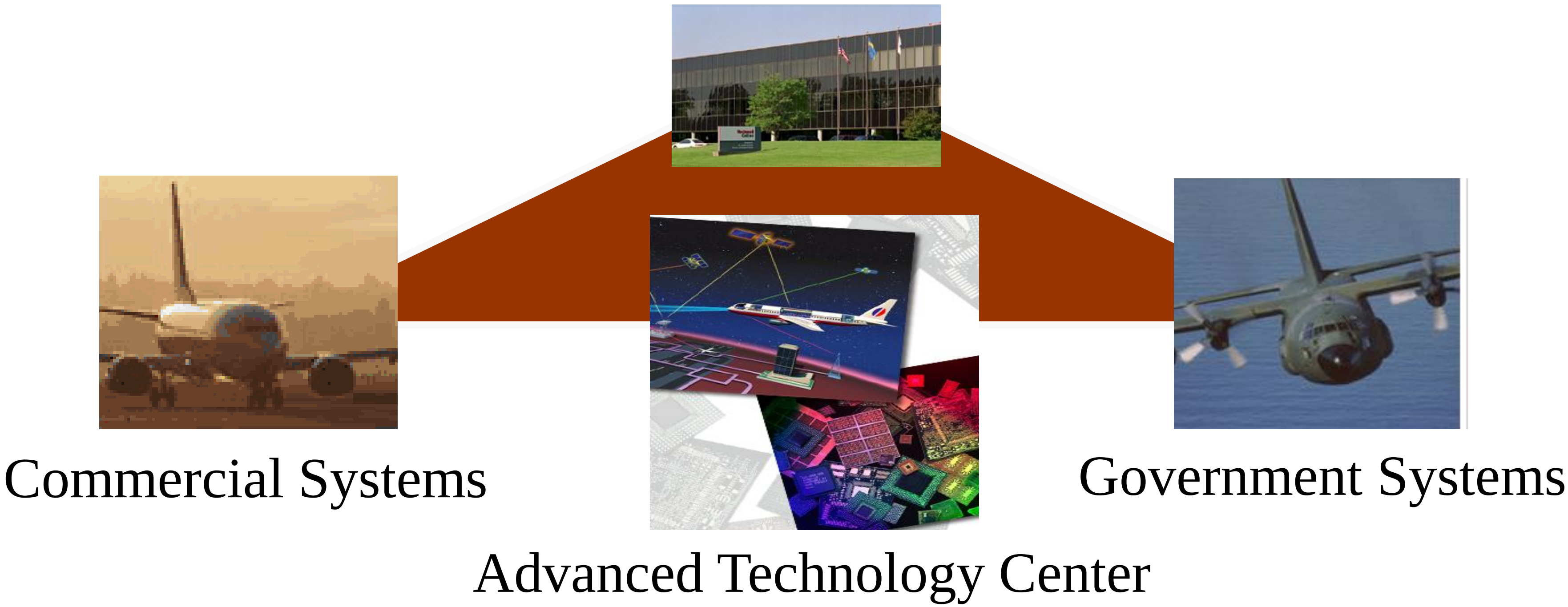


Rockwell Collins France

- 700+ employees, mainly located in Toulouse, France
- R&D, development of own products and technologies (direction finder, …)
- Systems and equipments for aircraft and rotary wing manufacturers (Airbus, Eurocopter, Augusta,…)
  - Communication, Navigation, Radar, Surveillance, Cockpit equipments
- We provide communication systems for European MODs (radio, networks)
  - Software define radio, Data Links (Link11, Link 16,…), Localization and SAR (Search And Rescue) equipments

Rockwell Collins

RCI Advanced Technology Center



- **The Advanced Technology Center (ATC) identifies, acquires, develops and transitions value-driven technologies**

- **The Automated Analysis section of ATC applies mathematical tools and reasoning**

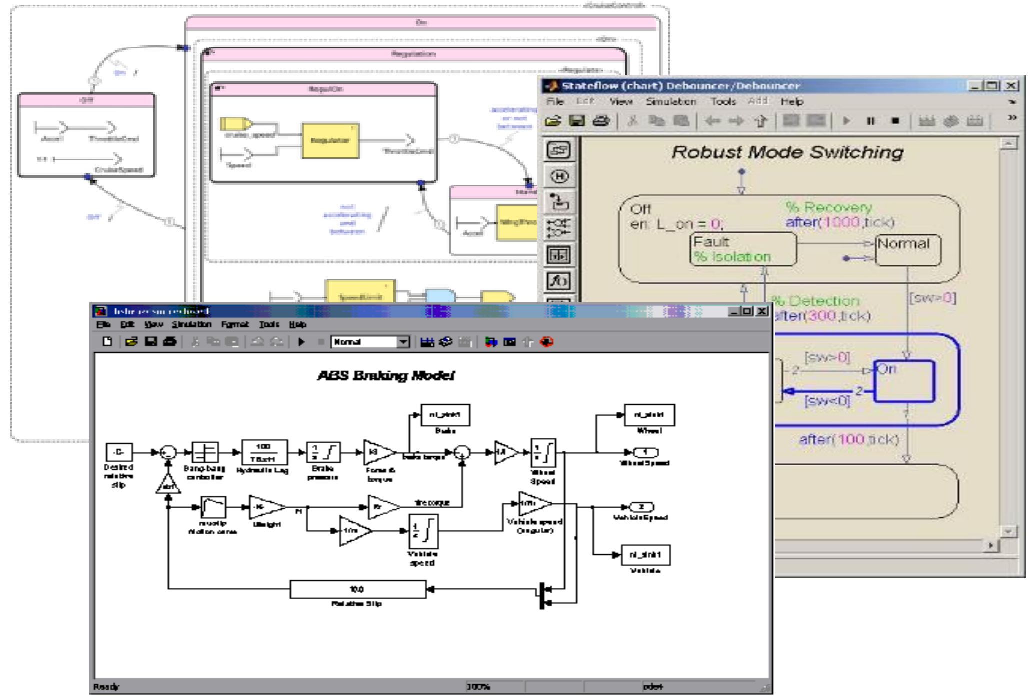
FM at Rockwell Collins France

- Since March 2009, 1 research engineer in Toulouse
- 2011 to 2013: PhD student – Combination of different techniques (model checking, abstract interpretation, …)
- Objectives:
  - Extension of the Automated Analysis section in the US
  - Participate in French and European Research Projects
  - Collaboration with industrial partners and customers and share experiences with them
  - Contact with European Research Institutions
  - Evaluation of tools (especially open source)

Activities in Model Checking

- Application in Model-Based Development
  - MATLAB Simulink®, Esterel Technologies SCADE Suite™
  - Enable early simulation and debugging
- Development of an in-house tool
  - Translator framework as front-end to different proof systems

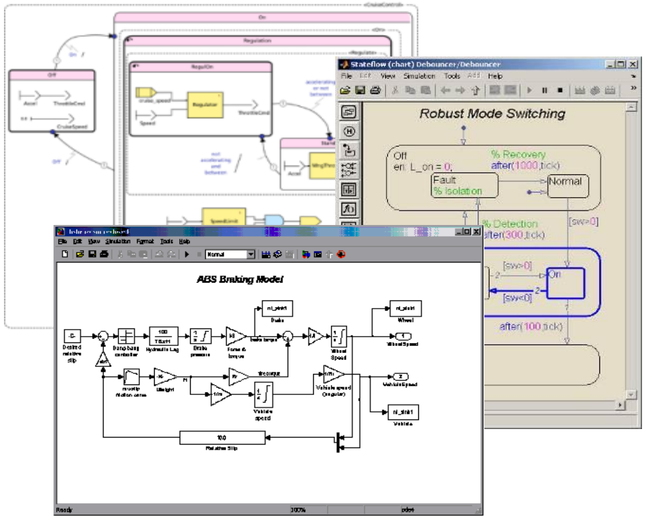
Reduce Costs and Improve Quality By Using Analysis to Find Errors During **Early** Design



In-House Tool

TRANSLATOR FRAMEWORK

Our In-House Tool:
The Rockwell Collins Translator Framework



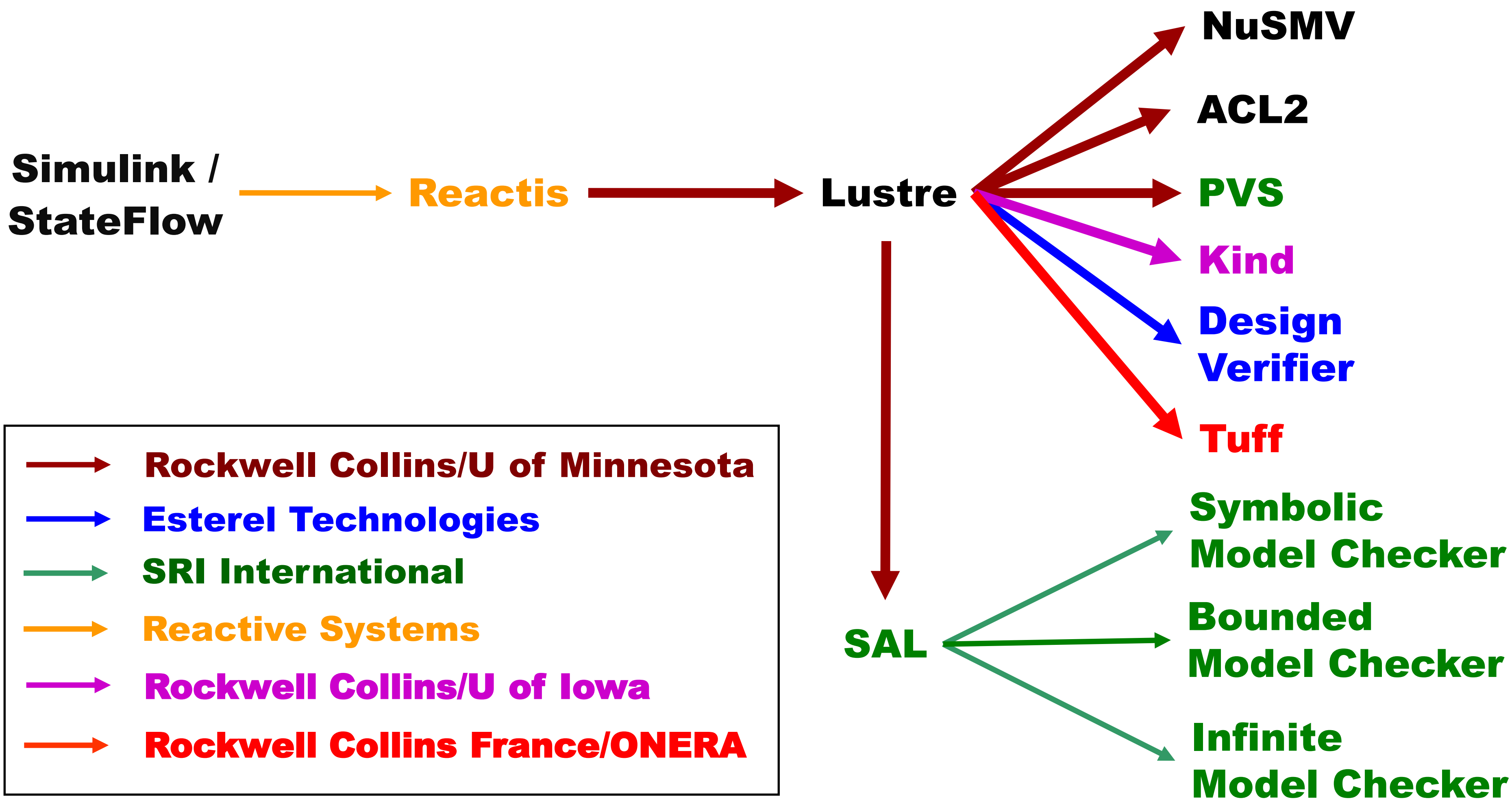
- Purpose : Formal Analysis of **SCADE™** and **MATLAB Simulink©** models

- **Long term effort** in the domain of formal methods

- Used on **several projects** (see articles by Steven Miller and Michael Whalen, e.g. *Software model checking takes off,* CACM 53(2), 2010)

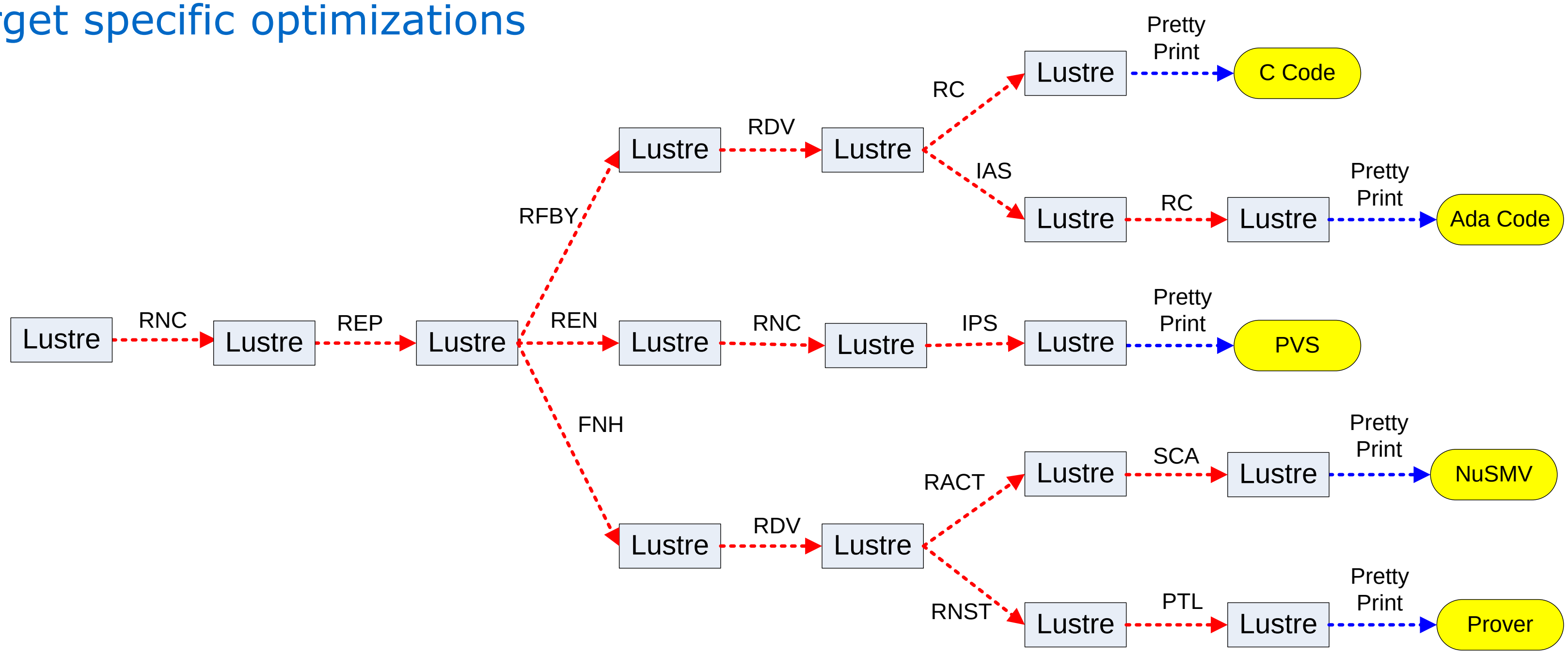
- Can output **optimized descriptions** in input languages of several **different analyzers**

The Rockwell Collins Translator Framework



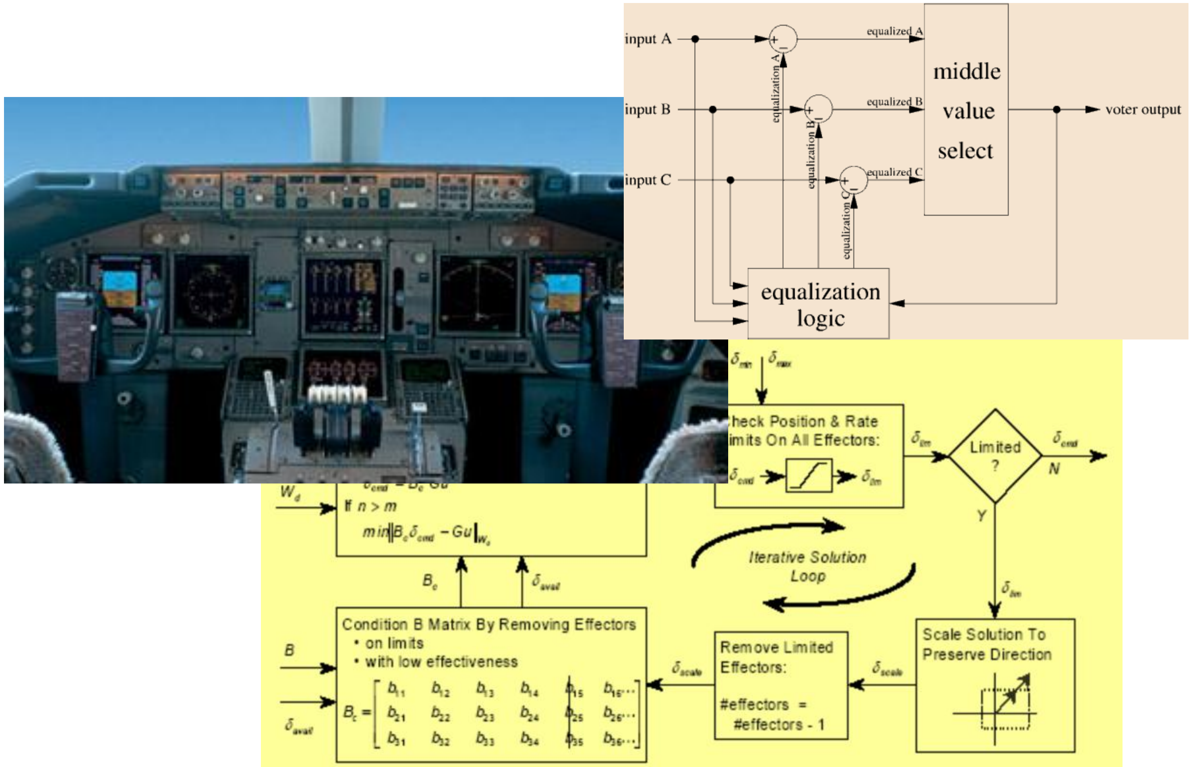
A Product Family of Translators

- Many small Lustre-to-Lustre translation passes
- Each pass refines closer to the target language
- Target specific optimizations



Translators Optimize for Specific Analysis Tools

Model	CPU Time (For NuSMV to Compute Reachable States)		Improvement
	Before	After	
Mode1	> 2 hours	11 sec	> 650x
Mode2	> 6 hours	169 sec	> 125x
Mode3	> 2 hours	14 sec	> 500x
Mode4	8 minutes	< 1 sec	480x
Arch	34 sec	< 1 sec	34x
WBS	29+ hours	1 sec	105,240x



Model Checking

CASE STUDIES

ADGS-2100 Adaptive Display & Guidance System

Modeled in Simulink

Translated to NuSMV

4,295 Subsystems

16,117 Simulink Blocks

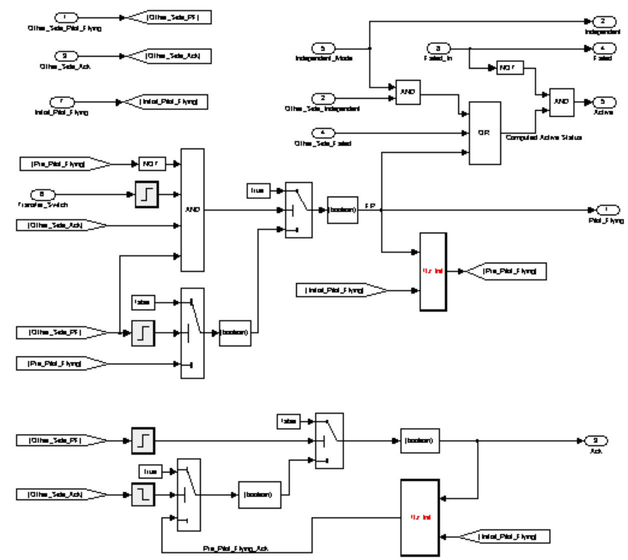
Over 10^{37} Reachable States

Example Requirement:

The Cursor Shall Never be Positioned on an Inactive Display

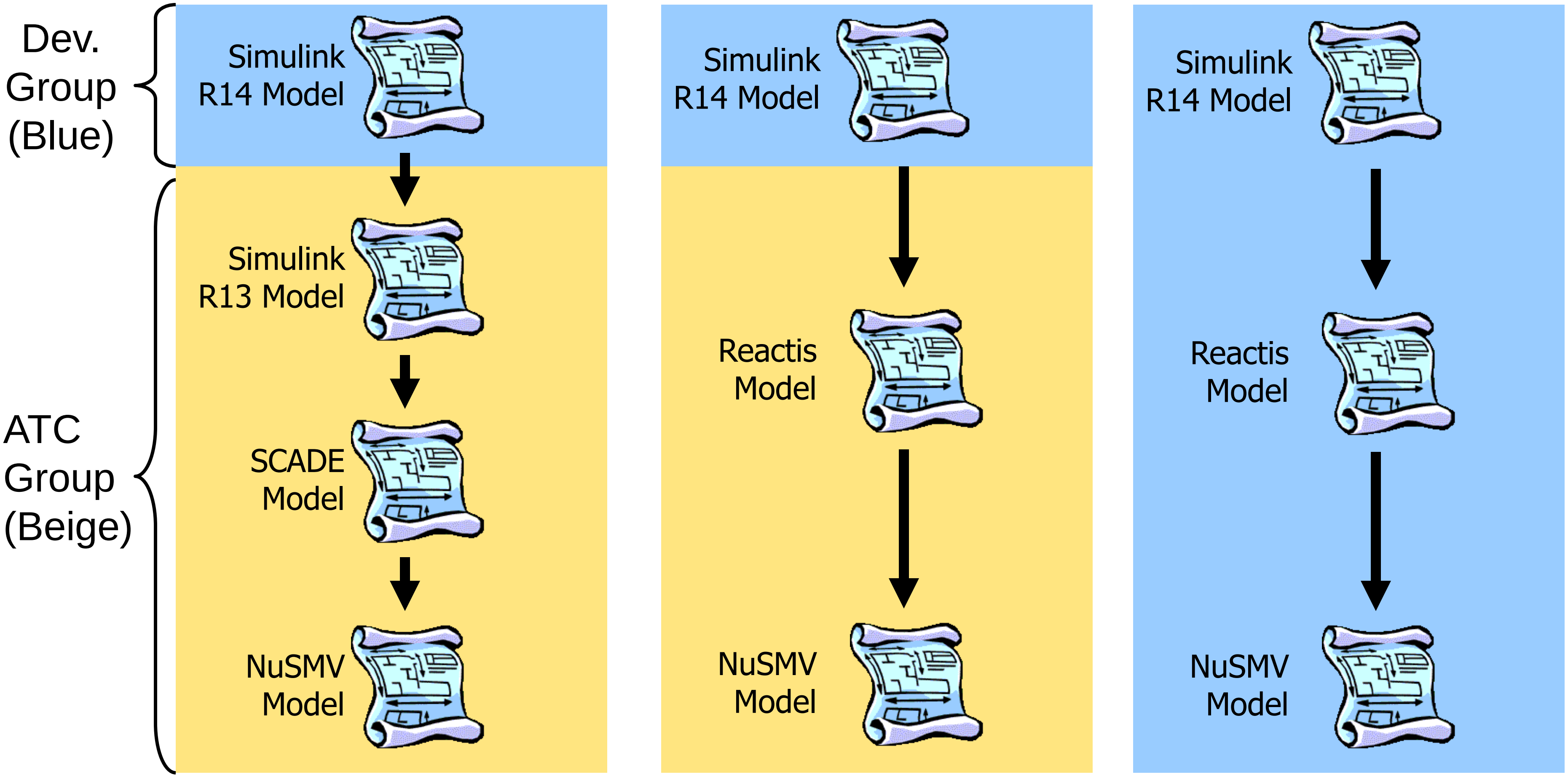
Counterexample Found in 5 Seconds

Checked 563 Properties - Found and Corrected 98 Errors in Early Design Models



ADGS-2100 Technology Transfer

Iteration 1 | Iteration 2 | Iteration 3



Iteration 1	Iteration 2	Iteration 3
Translation Time: 1-4 Hours	Translation Time: 10 Minutes	Translation Time: 10 Minutes
Turnaround: 1 Day to 1 Week	Turnaround: 3 Hours to 2 Days	Turnaround: 10 Minutes

Conclusion of this case study

Model Checking is successful in finding errors in early design models of our products

Case study for CerTA FCS Project (US)

- Sponsored by the Air Force Research Labs
- Can formal verification complement or replace some testing?
- Example Model – Lockheed Martin Adaptive UAV Flight Control System

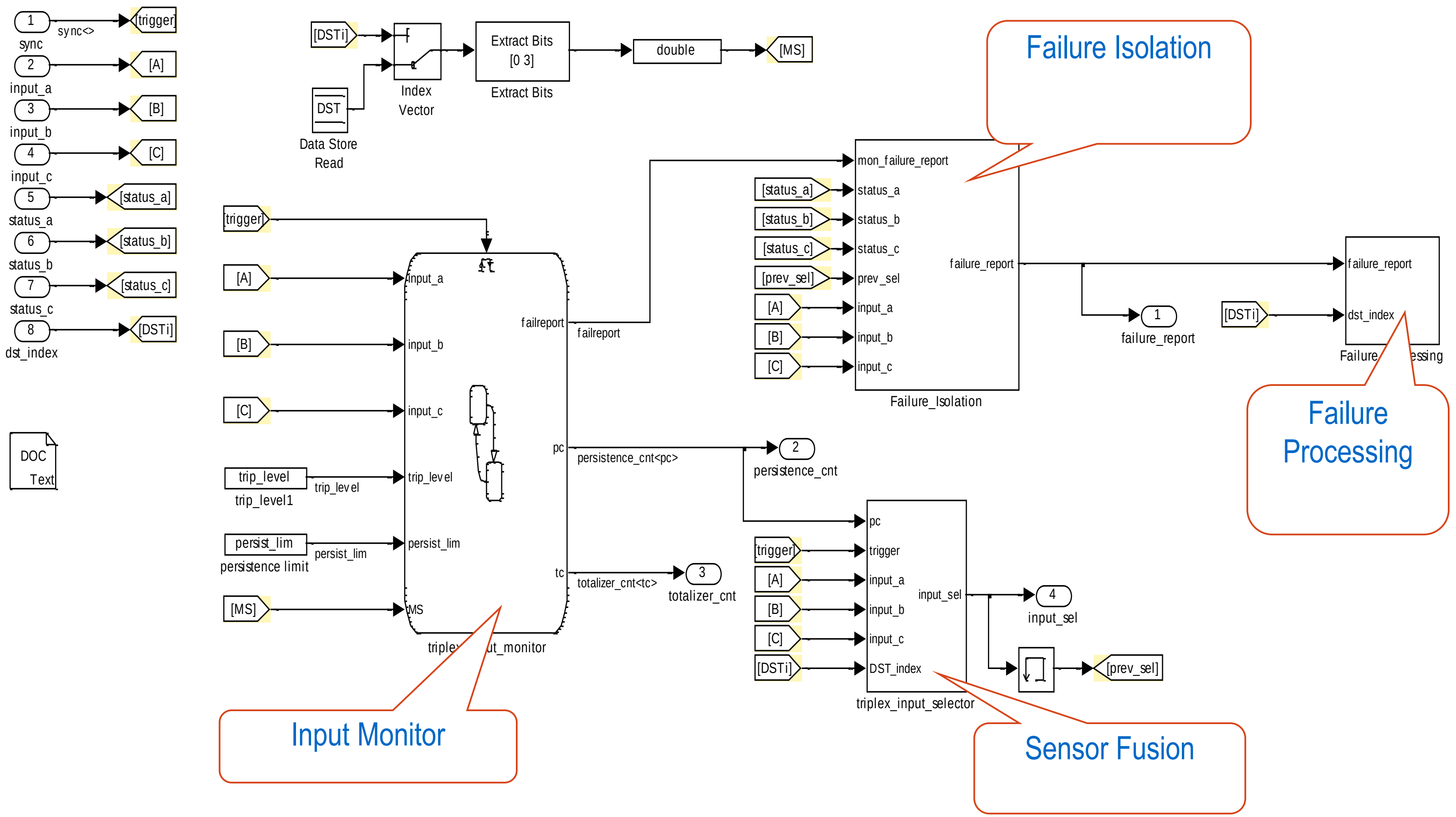
Lockheed Martin Aero

- Based on Testing
- Developed Tests from Requirements
- Executed Tests Cases on Test Rig

Rockwell Collins

- Based on Model-Checking
- Developed Properties from Requirements
- Proved Properties using Model-Checking

CerTA FCS Phase I - OFP Redundancy Management Logic



CerTA FCS Phase I – Errors Found

Errors Found in Redundancy Manager

	Model Checking	Testing
Triplex Voter	**5**	**0**
Failure Processing	**3**	**0**
Reset Manager	**4**	**0**
Total	**12**	**0**

- Model-Checking Found 12 Errors that Testing Missed
- Spent More Time on Testing than Model-Checking
  - 60% of total on testing vs. 40% on model-checking

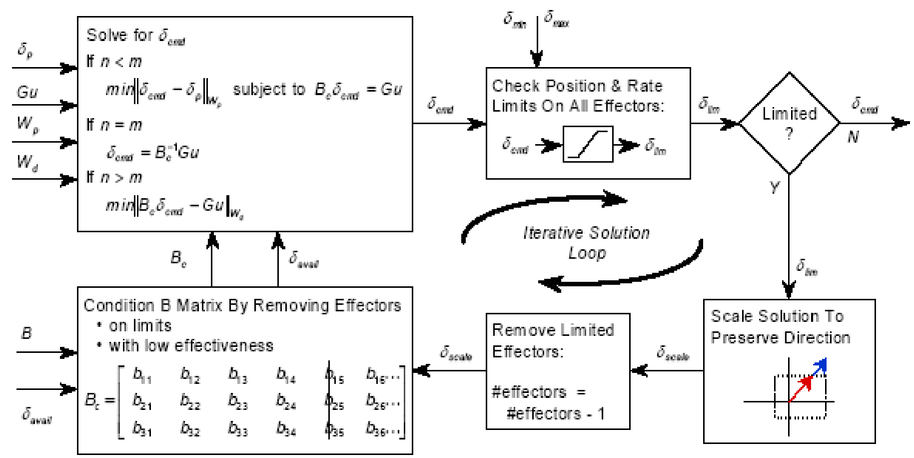
Rockwell Collins

Conclusion of this case study

Model-checking was more
cost effective
than testing at finding errors
in design models of our products

Second use case for CerTA FCS Project (US)

- Sponsored by the Air Force Research Labs
- Can Model Checking be Used on Numerically Complex Systems?
- Example Model
  - Lockheed Martin Adaptive UAV Flight Control System
  - Generates actuator commands for aircraft control surfaces
  - Matrix arithmetic of real numbers



CerTA FCS Phase II – Verification of Floating Point Numbers

- Translate Floating Point Numbers into Fixed Point
  - Extended translation framework to automate this translation
  - Convert floating point to fixed point (scaling provided by user)

- Advantages & Issues
  - Use bit-level integer decision procedures for model checking
  - Results unsound due to loss of precision
  - Very valuable tool for debugging

CerTA FCS Phase II - Results

- Errors Found
  - Five previously unknown errors that would drive actuators past their limits
  - Several implementation errors were being masked by defensive programming

Conclusion of this case study

Model-Checking is useful for debugging numerically complex systems

Analysis of a Triplex Sensor Voter (RCF)

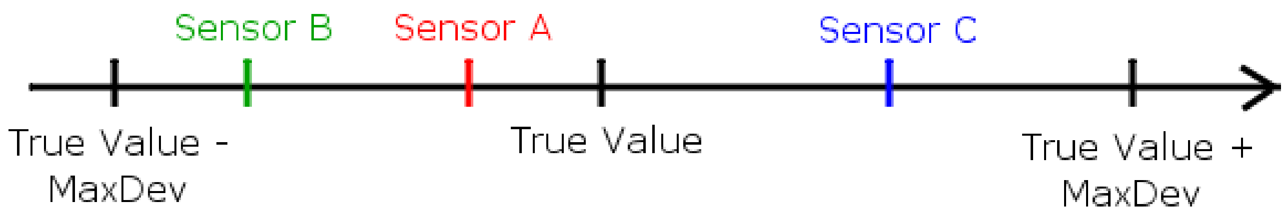
- Prove
  - Stability
  - Absence of runtime errors
  - Correct choice of parameters
- Analysis based on modern SMT solvers
- No abstraction of real numbers

Case Study : Triplex Sensor Voter

- Compute an output from input of **three redundant sensors**
- Modelled in **Simulink**
- Uses arithmetical operations on **real values**
- Includes low pass filtering, so has **internal state**

Sensor Characteristics

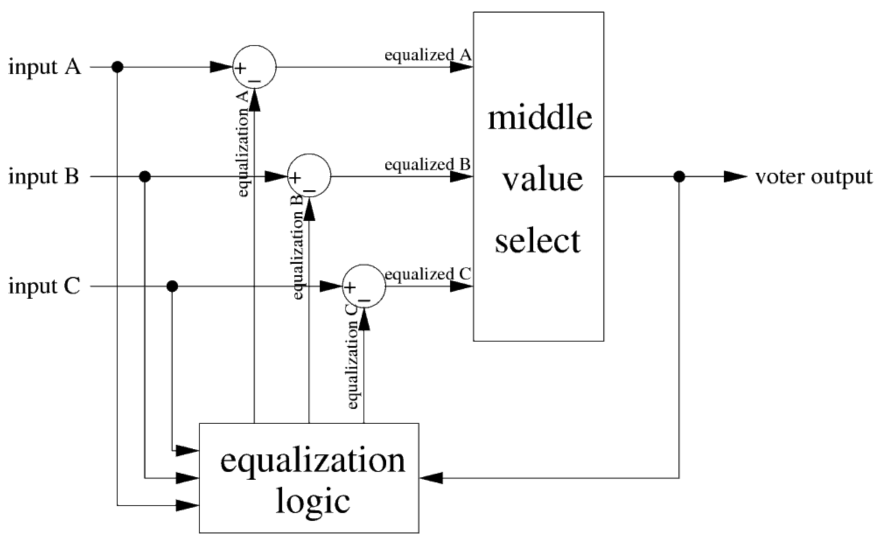
- Non-faulty sensors furnish a value within an interval around true value determined by a constant **MaxDev**

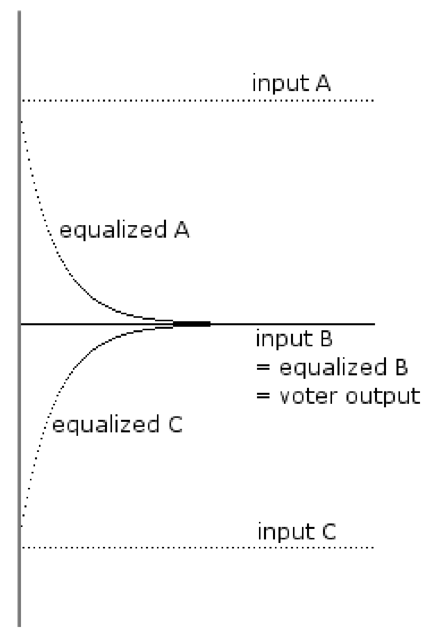


- In our analysis, we assume that sensors are **non-faulty**
- Result allows to paramerterize automatic **fault detection**

Structure and Operation of the Voter

- From each of the three inputs, subtract an equalization value
- Output is middle value of equalized values
- Equalization based on integration (has internal state)





Industrial Context of the Analysis

- **Legacy** model (~20 years old)
- Reverse engineering – **why** and **how** does it work ?
- Finding right **parameters** by testing is **very time consuming**
- Has been **qualified**, high confidence
- **Modifications** are made now
  - Better usage of Simulink
  - 4th input ?
- **New application** areas
- **No experience** in how to analyse it

Objectives of the Analysis

- Prove that a **transient peaks** cannot occur
  - Bounded-input bounded-output stability
- Choose good **parameters** for fault detection
  - a non-faulty sensor is never eliminated
- Experiment our **translator framework** on this kind of system
  - Feedback to implementors of proof engines

Equations of the Normal Operation Mode

$$EqualizationA_0 = 0.0$$
$$EqualizationB_0 = 0.0$$
$$EqualizationC_0 = 0.0$$

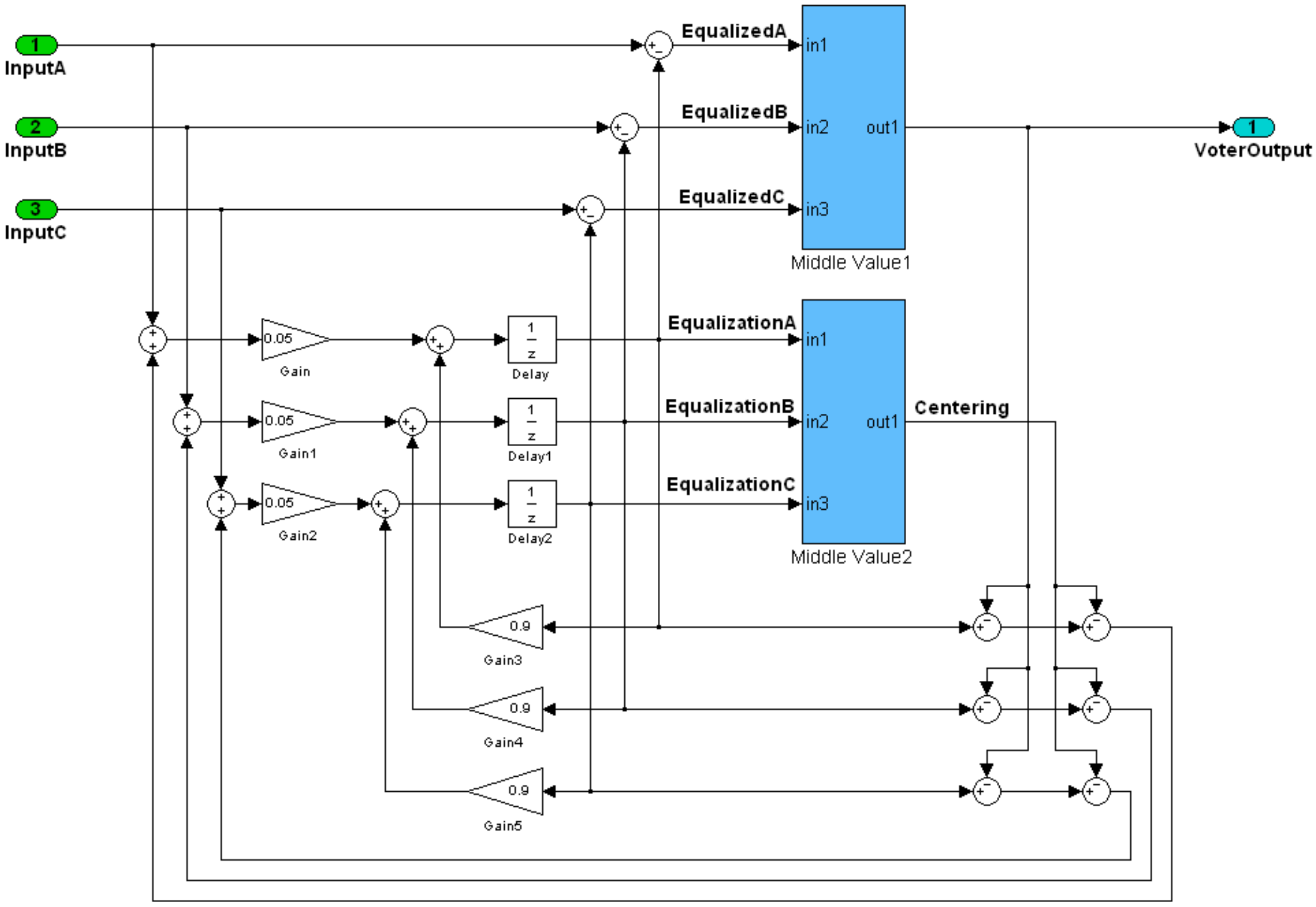
$$Centering_t = middleValue(EqualizationA_t, EqualizationB_t, EqualizationC_t)$$

$$EqualizedA_t = InputA_t - EqualizationA_t$$
$$EqualizedB_t = InputB_t - EqualizationB_t$$
$$EqualizedC_t = InputC_t - EqualizationC_t$$

$$VoterOutput_t = middleValue(EqualizedA_t, EqualizedB_t, EqualizedC_t)$$

$$EqualizationA_{t+1} = EqualizationA_t + 0.05 * (sat_{0.5}(EqualizedA_t - VoterOutput_t) - sat_{0.25}(Centering_t))$$
$$EqualizationB_{t+1} = EqualizationB_t + 0.05 * (sat_{0.5}(EqualizedB_t - VoterOutput_t) - sat_{0.25}(Centering_t))$$
$$EqualizationC_{t+1} = EqualizationC_t + 0.05 * (sat_{0.5}(EqualizedC_t - VoterOutput_t) - sat_{0.25}(Centering_t))$$

MATLAB Simulink Model of the Voter



Questions for the Analysis

- Is this system **stable** if sensors are non-faulty, i.e. is the output always within some bound from the true value? *Bounded-Input-Bounded-Output stability*

- Is an **implementation** using floating point arithmetic stable? Can there be an **accumulation** of rounding errors, causing loss of stability / overflow?

- Observation: system is stable if Equalization values are bounded -> prove that **Equalization values** are bounded

Model Level Analysis Result

- Set MaxDev = 0.2 (typical value)
- Model level analysis can **prove stability**
- The following property can be found and proven **automatically:**

$$|EqualizationA| \leq 0.4 \text{ and}$$
$$|EqualizationB| \leq 0.4 \text{ and}$$
$$|EqualizationC| \leq 0.4$$

- Automated analysis based on the research results of our PhD student Adrien Champion

Key to Analysis Objectives : Inductive Invariant

For MaxDev = 0.2

|EqualizationA| ≤ 0.4
|EqualizationB| ≤ 0.4
|EqualizationC| ≤ 0.4

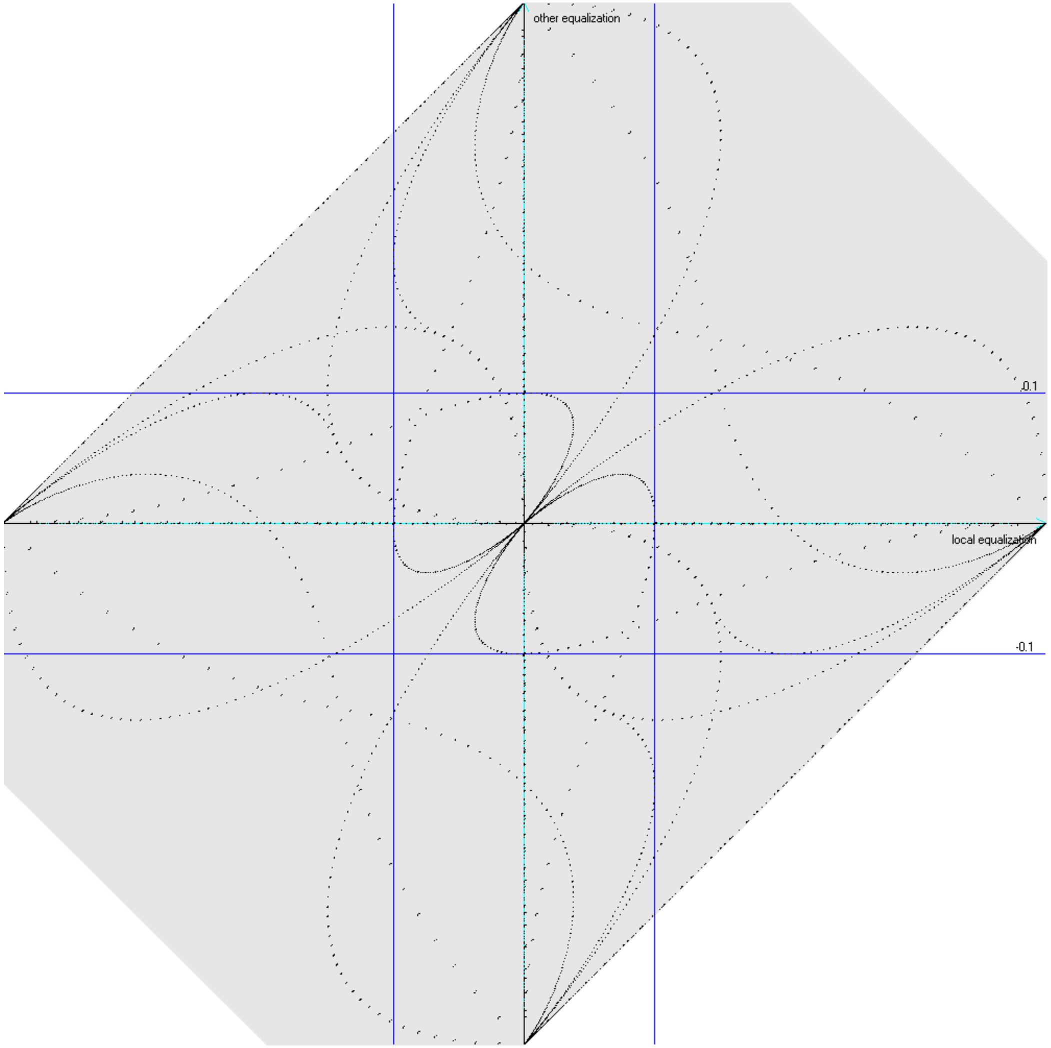
Proof objective

|EqualizationA - EqualizationB| ≤ 0.4
|EqualizationA – EqualizationC| ≤ 0.4
|EqualizationB – EqualizationC| ≤ 0.4

|EqualizationA + EqualizationB + EqualizationC| ≤ 0.66

Automatically generated lemmas

Inductive Octagonal Invariant



Code level analysis (floating point)

- Proof on model level assumes that no rounding errors occur
- In an implementation using floating point, rounding errors may accumulate
- **The invariant was partially confirmed on a C implementation using Astrée (abstract interpretation) based on the result from model checking**
  - **Combination of MC and AI**
- At the current state, a complete proof with Astrée is not possible
- Rounding errors can be over-approximated at model level, but this lacks scalability

Conclusion of this case study

Model-Checking is useful for proving properties of numerically complex systems and their floating point implementation

Systematic Industrial Application

- Despite the conclusive case studies, there is still no systematic application of model checking at RC

- **Why ?**

Obstacles to Systematic Application

- Still too much user skills required
  - Difficult for domain engineers
  - But there is progress in automated invariant generation
- Difficulty to express formal properties
  - But formal requirements engineering might help
- Scalability
  - Considerable progress in SMT solving
- Limited Scope
  - Lack of support for non-linear functions
- Cost is difficult to predict

Certification

- Objective: use analysis results as evidence for certification
- Not yet done today
- Enabled by latest standard DO-178C
- A research project is ongoing at RC with University of Iowa (Cesare Tinelli) based on the kind2 tool

Future Work: Cyber Security

- Cyber security of embedded systems is an issue
- Use model checking on cyber security requirements
- Prove the absence of security flaws in our systems
- We intend to initiate a collaborative project on the application of formal methods to cyber security

Further interests in formal methods at RC

- Combining analysis methods
  - PhD student, French research project CAFEIN
- Architectural analysis (AADL, SysML)
  - Participation in French « Project P », projects in the US
- Requirements engineering (generation of properties)
  - French research project co-submitted
- Automated Test Generation
  - Participation in ARTEMIS project MBAT

It's time for

Questions

Thank you for your attention