

# Formal methods for control engineering

Ursula Martin  
University of St Andrews  
[www-theory.dcs.st-and.ac.uk/~um](http://www-theory.dcs.st-and.ac.uk/~um)

- School of Computer Science
- University of St Andrews
  - RAE 5, “Commendable” teaching
  - 12.5 staff, 20 graduates per year
- Research
  - Software architectures
  - Symbolic computation
    - GAP discrete maths software
    - Symmetry and constraints
    - Light formal methods - NAG
    - Formal methods and analysis - NASA
    - Computational logic for computer algebra
      - Funding: EPSRC, EC, NAG Ltd, Qinetiq, Microsoft, SRI
      - SAFA consortium: Edinburgh, Glasgow, St Andrews

- **A very short history of formal methods**

- 1949 Turing

Explain why program right using values at intermediate stages -- assertions

- 1967 Hoare

Gave formal rules for tracking assertions through programmes -- Hoare logic

$\{A\}$  prog  $\{B\}$  denotes “if A is true and we run prog then B is true”

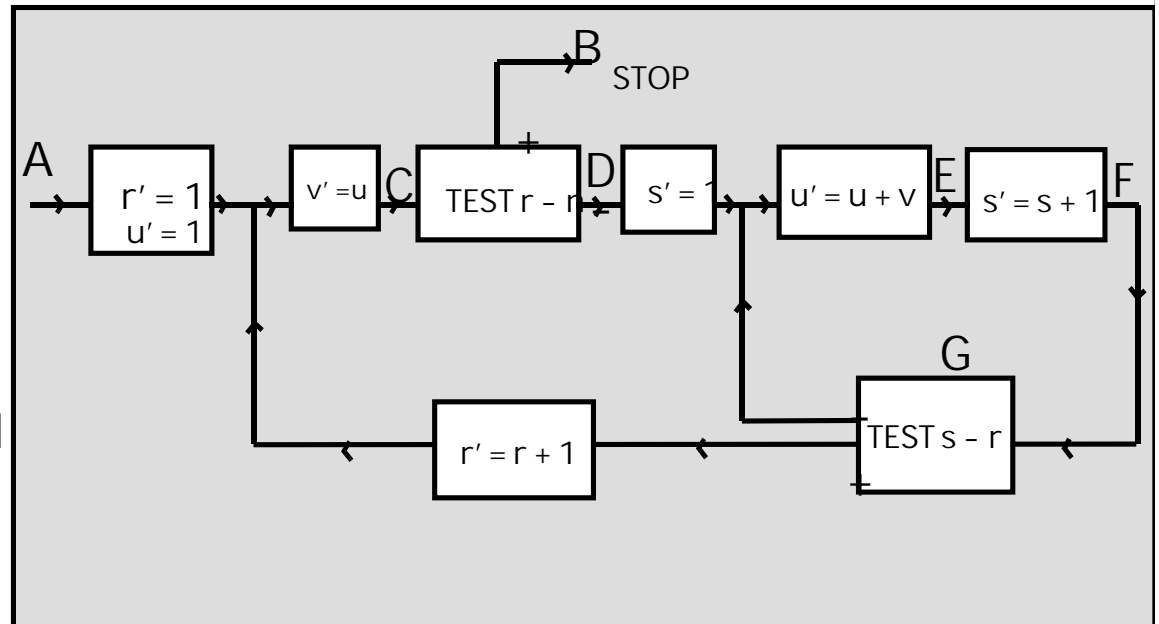
- To exploit this needed

- More theory

- Software for logical manipulation (theorem provers)

- Effective scalability

- Compelling commercial benefits



Theorem proving: use of a computer to produce or check formal proofs within a computer representation of a system of formal logic

$$*110\cdot64. \quad \vdash . 0 +_c 0 = 0 \quad [*110\cdot62]$$

$$*110\cdot641. \quad \vdash . 1 +_c 0 = 0 +_c 1 = 1 \quad [*110\cdot51\cdot61 . *101\cdot2]$$

$$*110\cdot642. \quad \vdash . 2 +_c 0 = 0 +_c 2 = 2 \quad [*110\cdot51\cdot61 . *101\cdot31]$$

$$*110\cdot643. \quad \vdash . 1 +_c 1 = 2$$

*Dem.*

$$\vdash . *110\cdot632 . *101\cdot21\cdot28 . \supset$$

$$\vdash . 1 +_c 1 = \hat{\xi} \{ (\exists y) . y \in \xi . \xi - \iota' y \in 1 \}$$

$$[*54\cdot3] = 2 . \supset \vdash . \text{Prop}$$

The above proposition is occasionally useful. It is used at least three times, in \*113·66 and \*120·123·472.

- Numerical computation      MATLAB, NAG library

Solve  $x^2 - 2x - 4 = 0$       Soln:  $x = 3.236, -1.236$

Integrate  $\cos(x)$  between  $0, \pi/2$       Soln: 1.0

- Symbolic computation      Maple, Mathematica

Solve  $x^2 - 2x - 4a = 0$       Soln:  $x = 1 + \sqrt{1 + 4a}, 1 - \sqrt{1 + 4a}$

Differentiate  $\sin(\cos(x))$       Soln:  $-\sin(x) \cdot \cos(\cos(x))$

- Computational logic      HOL, PVS

Prove that  $x^2 - 2x - 4a = 0$  has a real solution for  $a > -1/4$

Prove that  $x = 3.236$  is a “solution” of  $x^2 - 2x - 4 = 0$  with error ...

Prove that this implementation of Newton-Raphson is....

Prove that the Nichols plot from this design has...

## ■ Theorem proving and formal methods

PVS theorem prover: SRI International Menlo Park USA

HOL theorem prover : Cambridge UK

formal proof + fast decision procedures + computation+ high level of automation + architecture for other techniques

Proof-power : ICL/ Lemma-One

*Qinetiq, Eurofighter certification*

Intel: verification of floating point division for IA-64

*HOL: analysis, numerical analysis, floating point*

NASA Langley: verification of free flight air traffic control

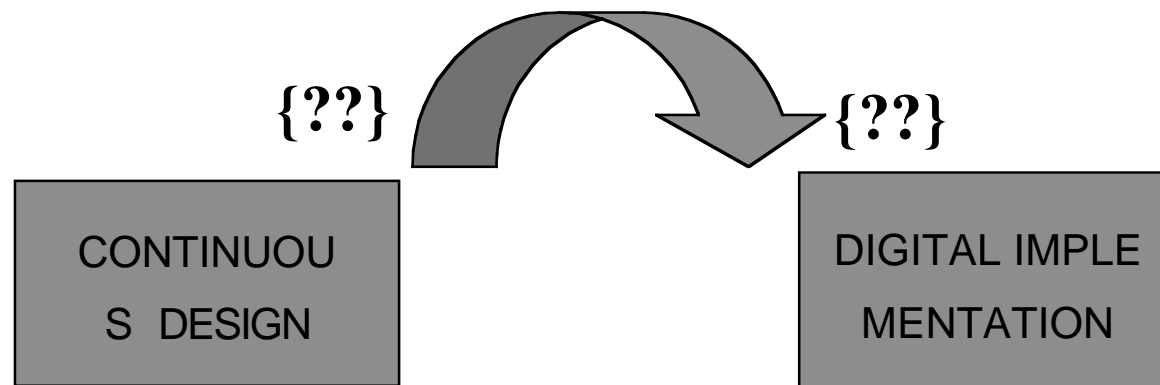
*PVS: analysis, trigonometry*

Logica: verification of Mondex smart card : protocol verification

SUN: Java, proof carrying code : program verification

Ford: automotive components : model checking+FSA

- **Formal methods for control engineering**
- **Goal** Investigate formal methods and theorem proving in support of control engineering
- **Long term aim**
  - Understand control theory and control engineering better
  - Improved or novel assurance techniques
  - Streamlined code generation and certification
- **Starting point**
  - Investigate the connection between continuous and discrete models



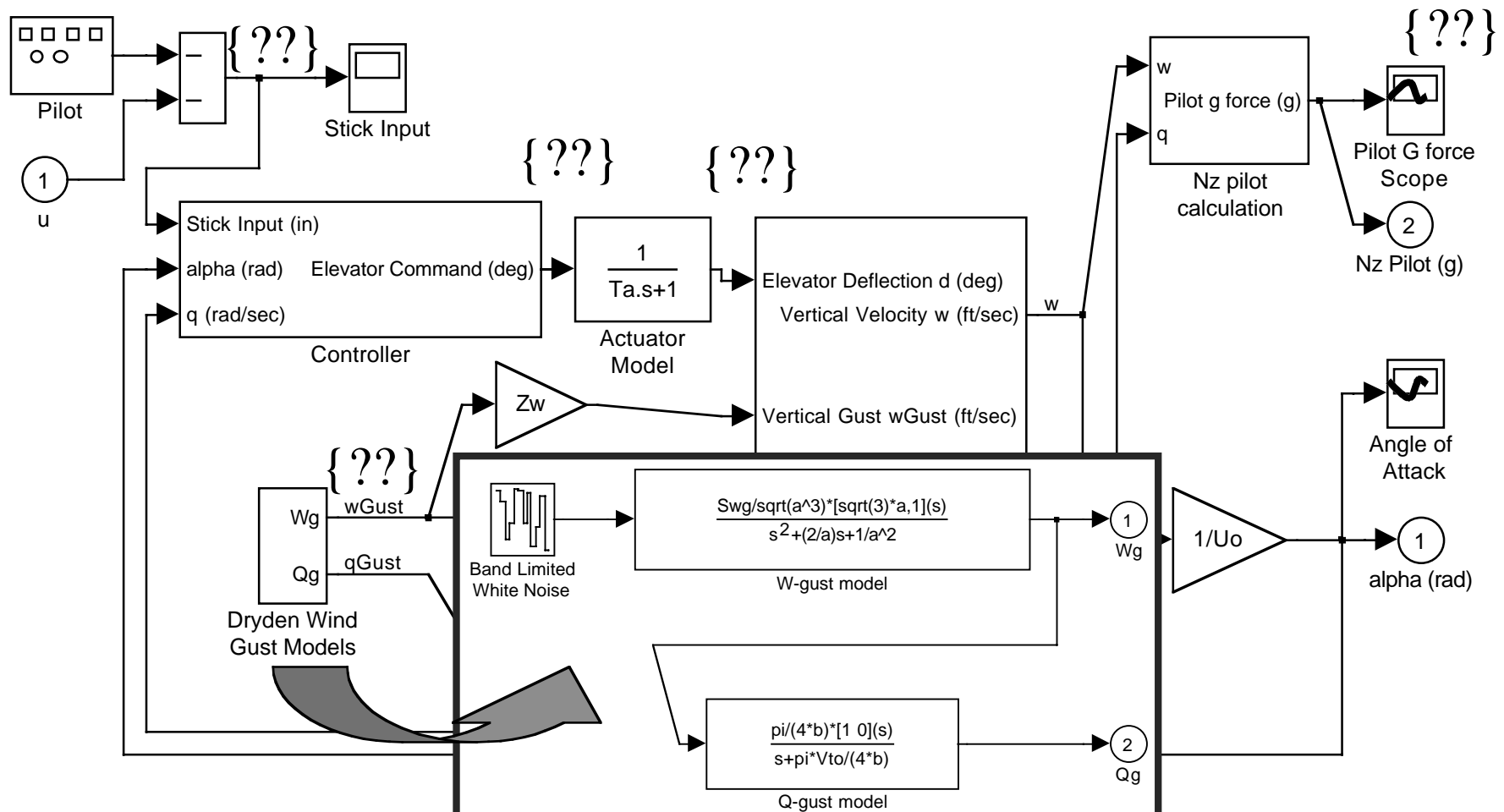
## Progress to date

How would you formulate and reason about assertions?

What would a Hoare logic for Block diagrams look like?

What sort of assertions are currently used?

How can we reason about them?



- **Progress to date**

- How would you formulate and reason about assertions?

- What would a Hoare logic for Block diagrams look like?

- Key observation: composition of phase and gain round block diagram

2002 Boulton, Hardy and Martin, Hoare logic for block diagrams in terms of phase and gain, paper forthcoming, pilot implementation in HOL

Next steps.... extend to other parameters, discrete ...

incorporate in methodologies eg Clawz

- **Progress to date**

- How would you formulate and reason about assertions?

- What would a Hoare logic for Block diagrams look like?

- Key observation: composition of phase and gain round block diagram

2002 Boulton, Hardy and Martin, Hoare logic for block diagrams in terms of phase and gain, paper forthcoming, pilot implementation in HOL

Next steps.... extend to other parameters, discrete ...

incorporate in methodologies eg Clawz

- What sort of assertions are currently used?

- How can we reason about them?

- Key observation: requirements analysis in terms of standard plots

2002 Boulton, Hardy, Kelsey and Martin, Symbolic tests for Nichols plots, paper forthcoming, pilot implementation in Maple and PVS

Next steps.... currently shadowing existing technology

extend to MIMO, parameters, other “plots”

- **What next?**

- **Research**

- Develop Hoare logic work
- Composability
- Develop alternative assurance tests

- Code generation and certification
- Generation and management of assertions
- Link with web service architectures?

- **Tools**

- Develop required theories in computational logic system
- Integration with Simulink: via Clawz? Via MATLAB/Prosper/Maple?

■ Thank you!