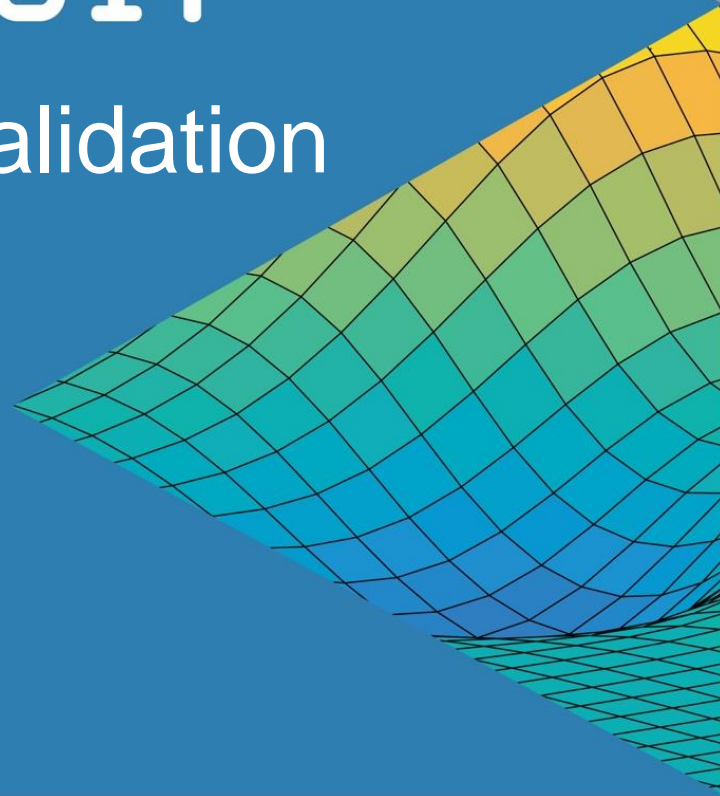


MATLAB EXPO 2017

Pratiquez la Vérification et Validation
en toute sérénité

Mathieu Cuenant, MathWorks



**Simulation models are primary
meant to support V&V activities**

Grumman X-29



Constructeur	 Grumman
Rôle	Avion expérimental
Statut	programme terminé
Premier vol	14 décembre 1984
Date de retrait	1991
Nombre construits	2

Maneuverability



Instability



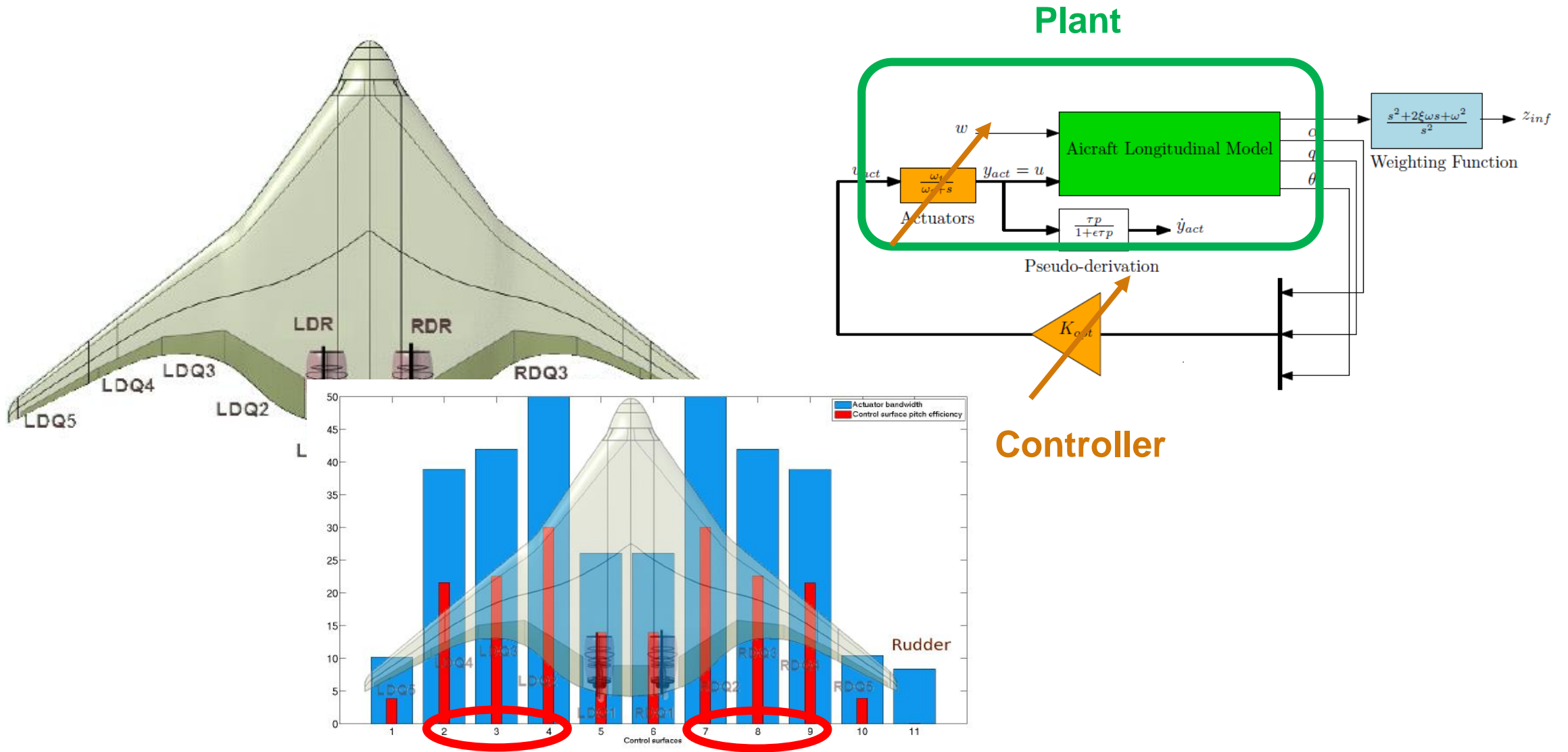


Fig. 9 Actuators tuned bandwidths (blue) and elevons pitch efficiencies (red).

V&V techniques

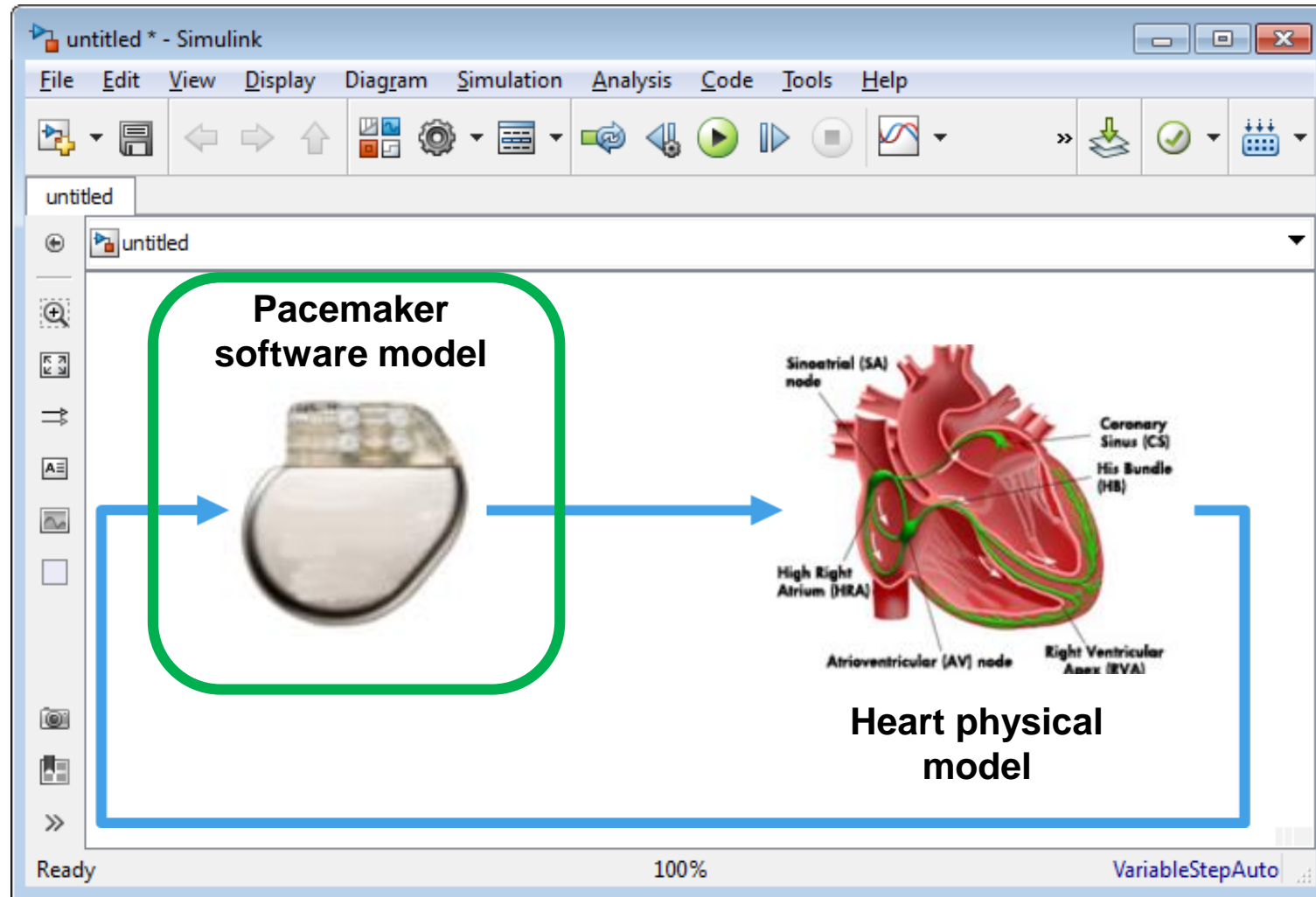


Dynamic Testing

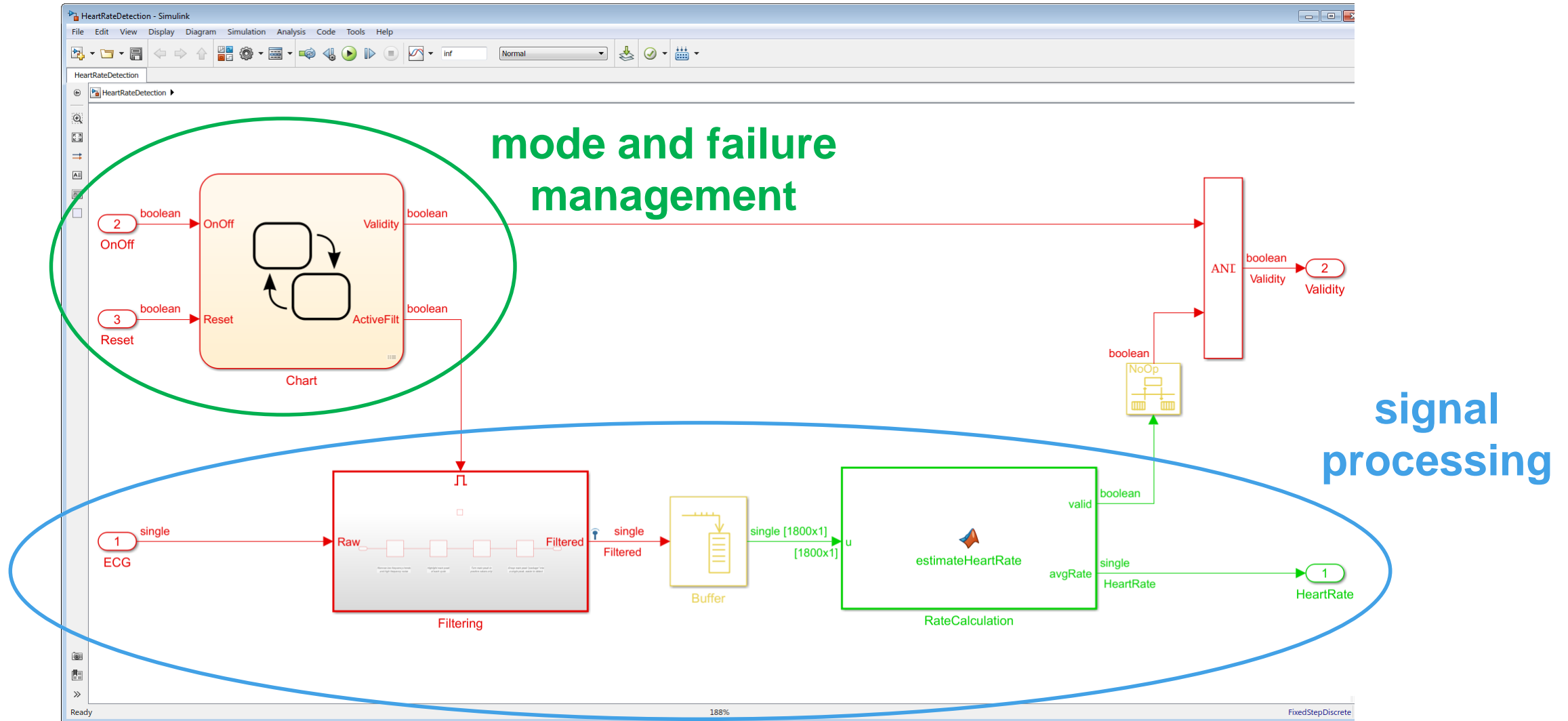


Static Analysis

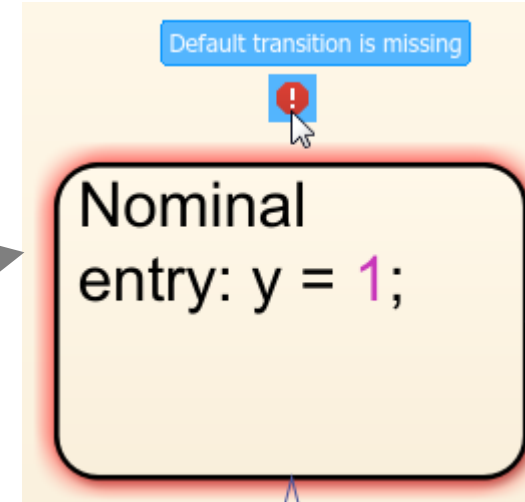
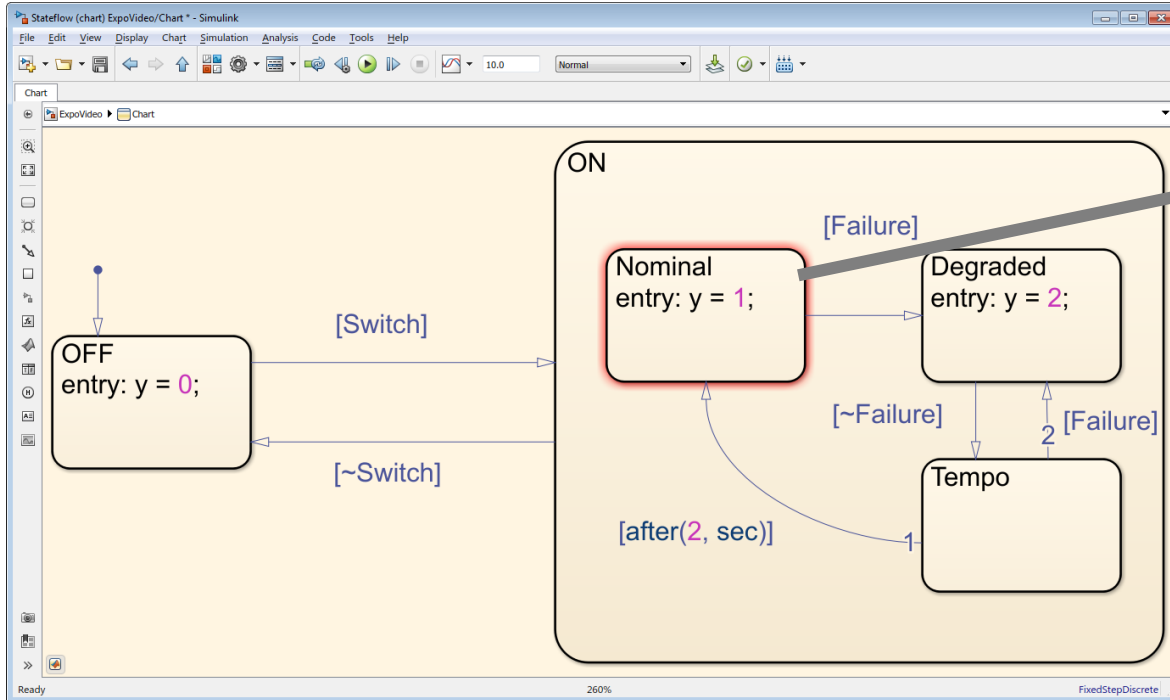
Setting the scene



Algorithm design with Simulink



Checking while editing



Symbol Wizard

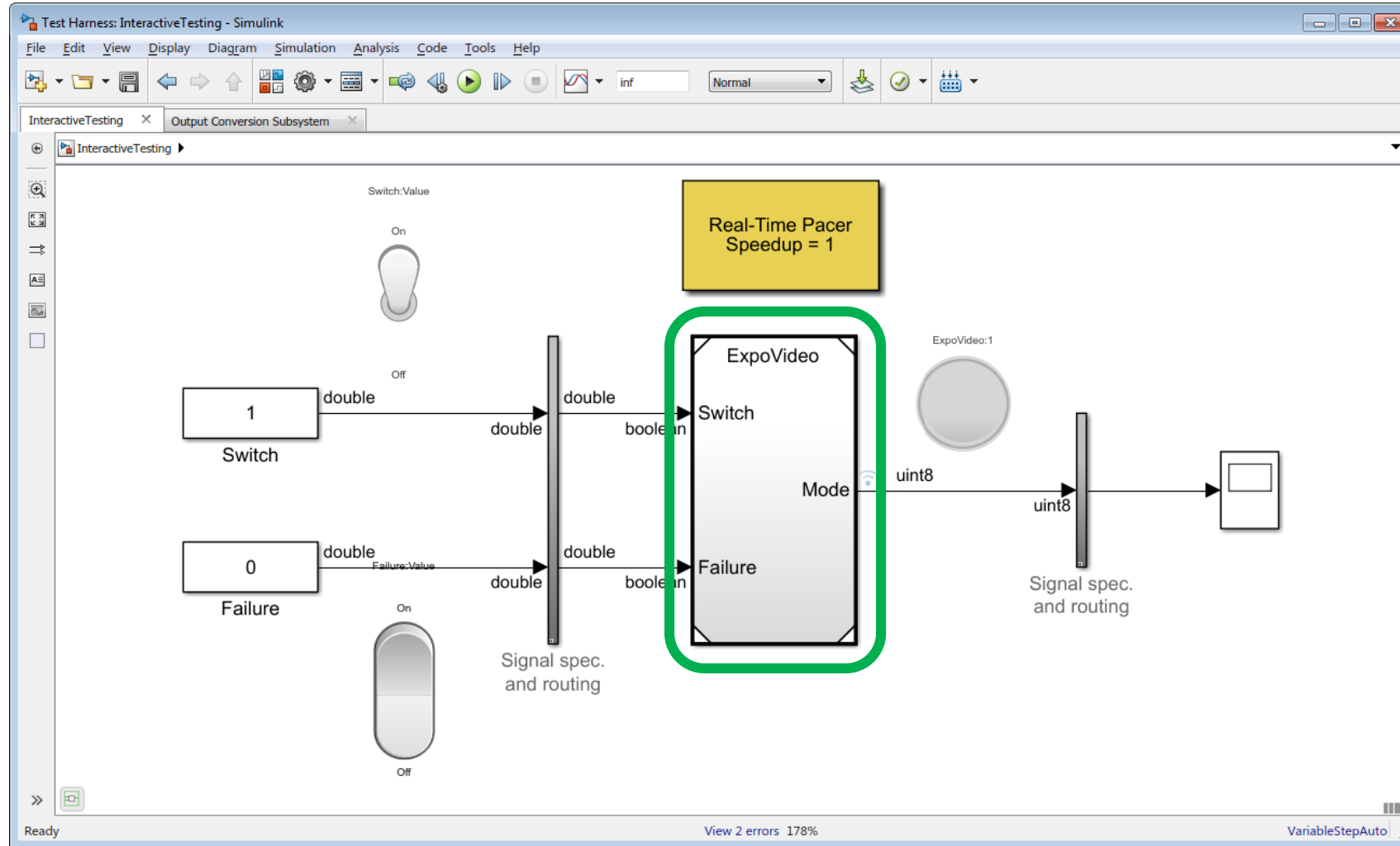
Unresolved symbols found in **Chart**. All selected data/events/messages will be created in the chart

Name	Class	Scope
<input checked="" type="checkbox"/> Switch	Data	Input
<input checked="" type="checkbox"/> Failure	Data	Input
<input checked="" type="checkbox"/> y	Data	Output

View created data/events/messages in Model Explorer

OK Cancel Help

Interactive testing



Reactive testing

The screenshot displays the Simulink Reactive Testing environment. The main window shows a test harness diagram with a 'Test Sequence' block (containing steps 1, 2, and 3) and an 'ExpoVideo' block. A 'Test Assessment Block' is also present, receiving inputs 'Switch_A', 'Failure_A', and 'Mode_A'. The diagram includes signal routing and specification blocks. Two inset windows provide detailed views of the test sequence and a safety property.

Test Sequence Table:

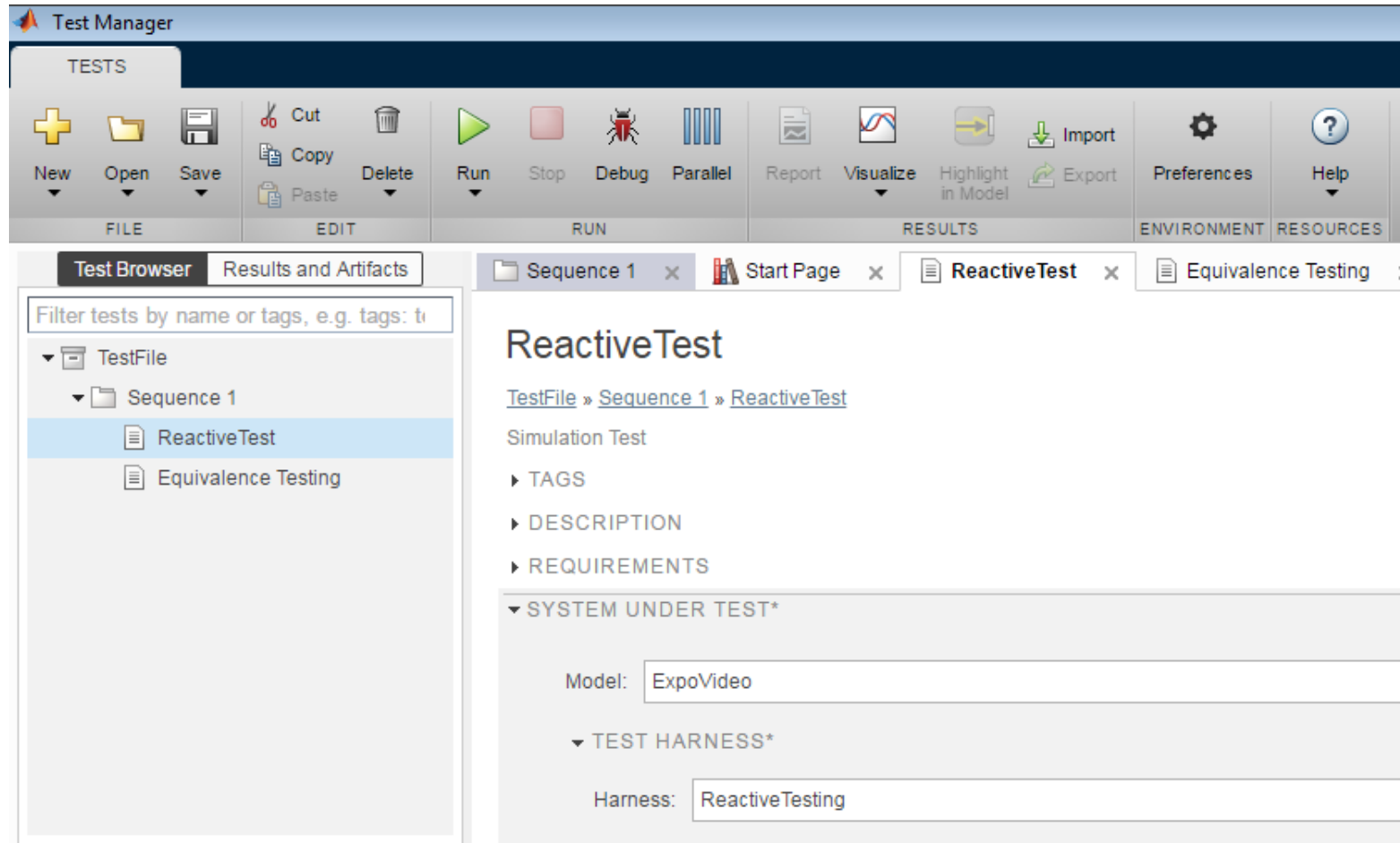
Step	Transition	Next Step
Init %% Initialize data outputs. Switch = false; Failure = false;	1. after(1,sec)	Startup
Startup Switch = true;	1. after(1,sec)	Fail
Fail Failure = true;	1. Mode == 2	BackT...
BackToNormal Failure = false;	1. Mode == 1	Switch...

Safety Property:

```

Step
SafetyProperty
verify( ~ (Failure & Mode == 1))
    
```

Test Manager



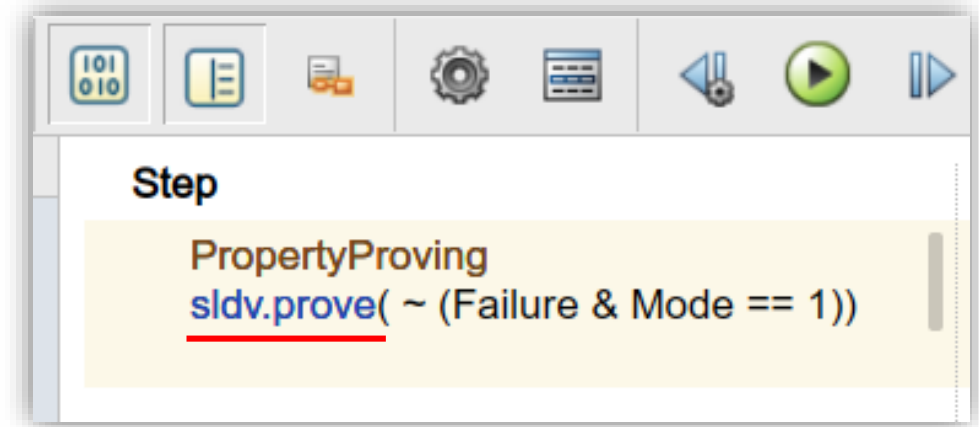
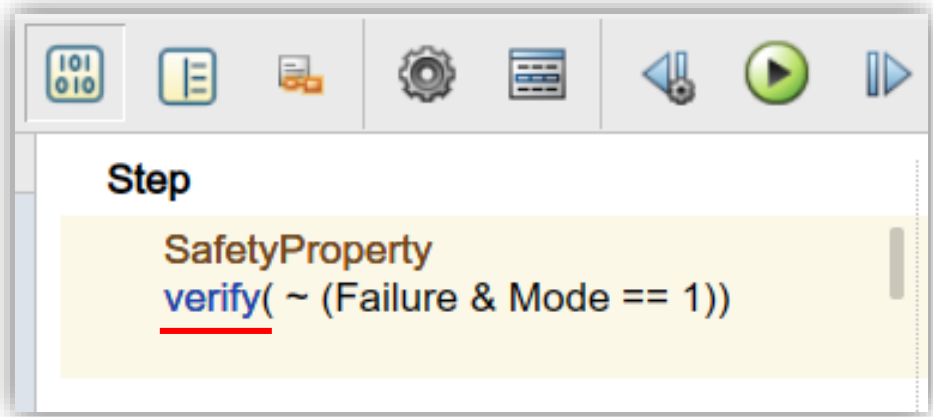
The screenshot displays the MATLAB Test Manager application window. At the top, there is a menu bar with the following sections: FILE (New, Open, Save), EDIT (Cut, Copy, Paste, Delete), RUN (Run, Stop, Debug, Parallel), RESULTS (Report, Visualize, Highlight in Model, Import, Export), ENVIRONMENT (Preferences), and RESOURCES (Help). Below the menu bar, there are two tabs: 'Test Browser' and 'Results and Artifacts'. The 'Test Browser' tab is active, showing a tree view of test files. The tree structure is as follows:

- TestFile
 - Sequence 1
 - ReactiveTest (selected)
 - Equivalence Testing

The 'ReactiveTest' file is selected, and its details are shown in the right-hand pane. The details include:

- ReactiveTest**
- TestFile » Sequence 1 » ReactiveTest
- Simulation Test
- ▶ TAGS
- ▶ DESCRIPTION
- ▶ REQUIREMENTS
- ▼ SYSTEM UNDER TEST*
 - Model: ExpoVideo
- ▼ TEST HARNESS*
 - Harness: ReactiveTesting

Requirement Proving



V&V journey



Interactive testing

Reactive Testing

Coverage Analysis

Code Testing

Dynamic Testing

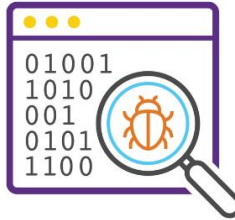
Edit-time checks

(Dead Logic Detection)
(Test Case Generation)

Requirement Proving

Code proving

Static Analysis



Code Testing

The screenshot displays the MATLAB Test Browser interface for configuring an Equivalence Test. The main window shows the test configuration for 'Equivalence Testing' under 'Sequence 1'. Two simulation configurations are visible:

- SIMULATION 1:**
 - Model: ExpoVideo
 - Harness: ReactiveTesting
 - Simulation Mode: Normal (highlighted with a green box)
- SIMULATION 2 (Copy settings from Simulation 1):**
 - Model: ExpoVideo
 - Harness: ReactiveTesting
 - Simulation Mode: Software-in-the-Loop (SIL) (highlighted with a red box)

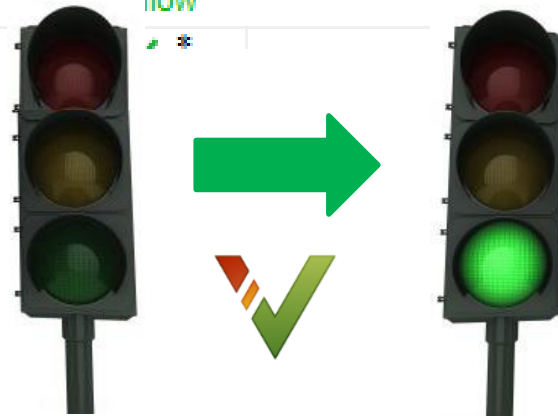
Overlaid on the right side of the interface are three callouts:

- Simulation mode dropdown:** Shows options: Software-in-the-loop (SIL), Normal, Accelerator, Software-in-the-loop (SIL), and Processor-in-the-loop (PIL). The 'Software-in-the-loop (SIL)' option is selected.
- ExpoVideo (SIL) block diagram:** Shows a block with inputs 'Switch' and 'Failure', and an output 'Mode'.
- COVERAGE SETTINGS*:**
 - COVERAGE TO COLLECT:
 - Record coverage for system under test
 - Record coverage for referenced models

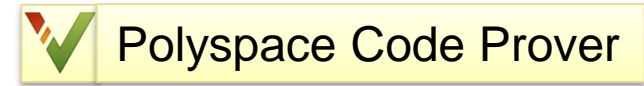
At the bottom left, the text 'MATLAB EXPO 2017' is displayed.

Code Proving

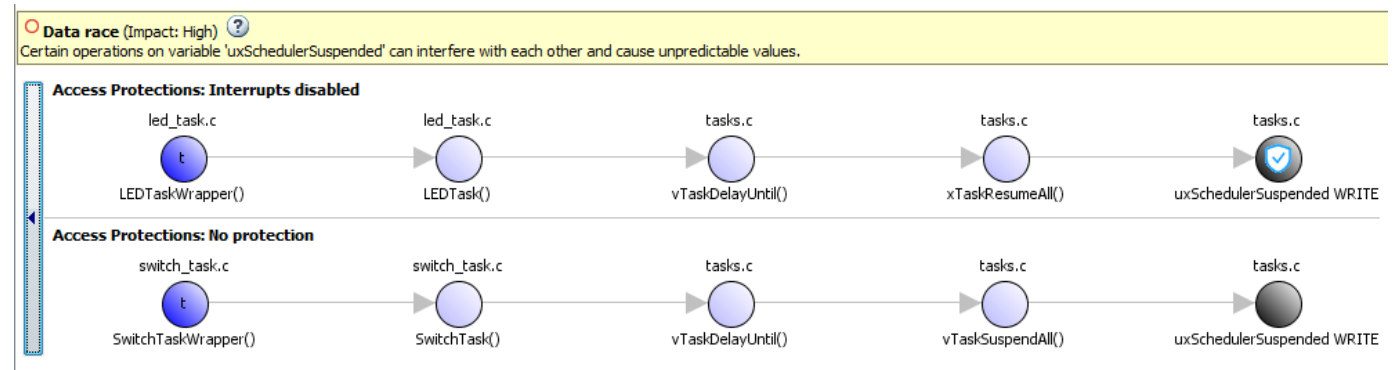
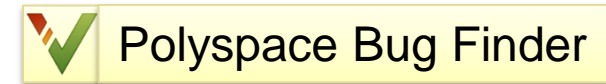
[-] Run-time Check	16
[-] Green Check	16
[-] Illegally dereferenced pointer	1
✓ *	ExpoVideo.c
[+] Invalid use of standard library	2
[-] Non-initialized pointer	1
✓ *	ExpoVideo.c
initialized variable	11
flow	1
✓ *	ExpoVideo.c



Prove absence of run-time errors



Detect interface mismatch and concurrency issues

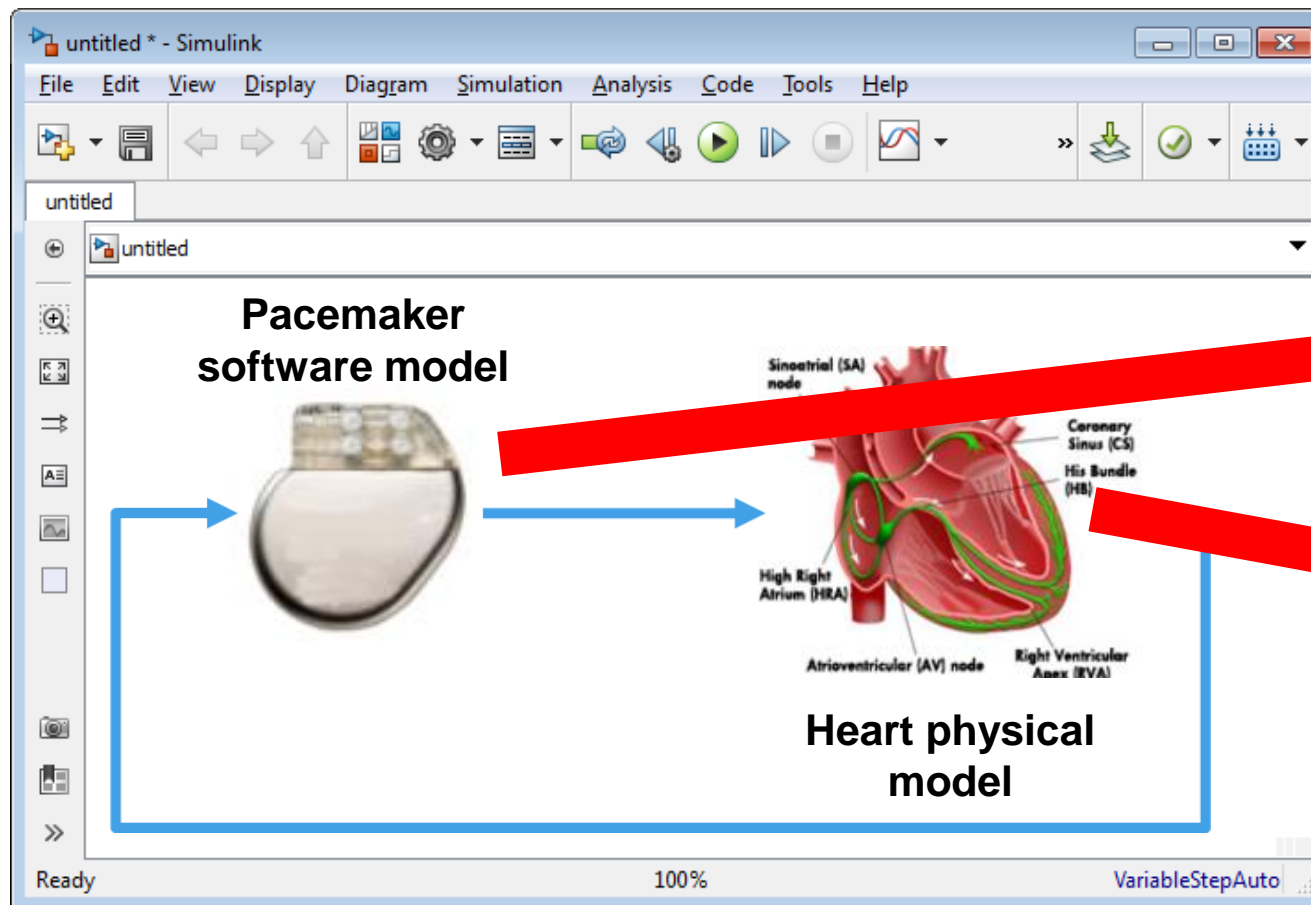


Is this enough ?

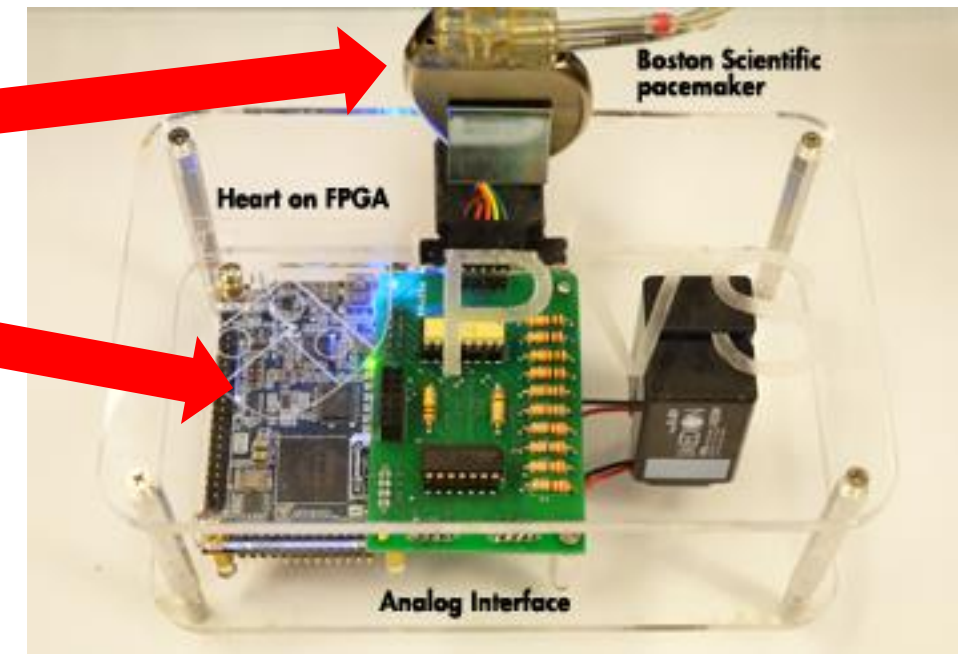
University of Pennsylvania Develops Electrophysiological Heart Model for Real-Time Closed-Loop Testing of Pacemakers

[Read article](#)

By Zhihao Jiang and Rahul Mangharam, University of Pennsylvania



[Learn more](#)



V&V journey



Interactive testing

Reactive Testing

Coverage Analysis

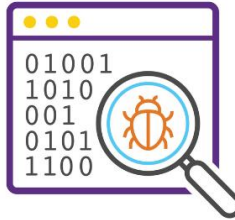
Code Testing

Edit-time checks

(Dead Logic Detection)
(Test Case Generation)

Requirement Proving

Code proving



**Simulation models are primary
meant to support V&V activities**