MATLAB EXPO 2017
Pratiquez la Vérification et Validation en toute sérenité

Mathieu Cuenant, MathWorks
Simulation models are primary meant to support V&V activities
Maneuverability

Instability

Grumman X-29

Constructor: Grumman
Role: Experimental aircraft
Status: Programme terminated
First flight: December 14, 1984
Retirement date: 1991
Number built: 2
Integrated Design and Control of a Flying Wing Using Nonsmooth Optimization Techniques

Yann Denieul, Joël Bordeneuve, Daniel Alazard, Gilles Taquin - 2015

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

Dynamic Testing

Static Analysis
Setting the scene

Pacemaker software model

Heart physical model
Algorithm design with Simulink

mode and failure management

signal processing
Checking while editing

OFF entry: y = 0;

ON

Nominal entry: y = 1;

Degraded entry: y = 2;

[Switch]

[Failure]

[-Failure]

[after(2, sec)]

Tempo

Nominal entry: y = 1;

Delegated entry: y = 2;

Symbol Wizard

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch</td>
<td>Data</td>
<td>Input</td>
</tr>
<tr>
<td>Failure</td>
<td>Data</td>
<td>Input</td>
</tr>
<tr>
<td>y</td>
<td>Data</td>
<td>Output</td>
</tr>
</tbody>
</table>

MATLAB EXPO 2017
Interactive testing
Reactive testing
Requirement Proving

Step

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

Step

PropertyProving
sldv.prove(~ (Failure & Mode == 1))
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

MATLAB EXPO 2017
Code Testing

Equivalence Testing

Simulation mode: Software-in-the-loop (SIL)
Code interface: Accelerator

Simulation 1

System under test
Model: ExpoVideo
Harness: ReactiveTesting
Simulation mode: Normal

Simulation 2

Copy settings from Simulation 1
System under test
Model: ExpoVideo
Harness: ReactiveTesting
Simulation mode: Software-in-the-loop (SIL)
Code Proving

Prove absence of run-time errors

✅ Polyspace Code Prover

Detect interface mismatch and concurrency issues

✅ Polyspace Bug Finder
Is this enough?

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

By Zhihao Jiang and Rahul Mangharam, University of Pennsylvania
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