Automated Model Based Requirement Coverage Analysis Tool

Chethan C U

cchethan@moog.com
chethan.cu@gmail.com
Outline

- DO 178B guidelines for Software Testing
- Functional Requirements
- Integrator – Anti-wind up Analysis
- Functional Metric Characteristics
- Metric Definition – Integrator, PersistenceOnOff
- Structural Coverage Analysis using Simulink Design verifier, Reactis
- Moog Functional Coverage Tool
- Mutation Testing
Introduction


- The DO-178B Guidelines suggests to perform:
  - Requirement Based Testing – verify the correct functionality of the Software.
  - Structural Coverage Analysis – to determine which parts of the code was not exercised during the ‘Requirement Based Testing’.

- 100 % Structural Coverage DOES NOT mean 100% Requirement Coverage.
Process Followed

- Requirements
- Test Cases
- Code
- Execute
- Manual Functional Reviews
- Structural Coverage
- Results Expected/Actual
Ambiguous Requirements

- Integrator function **shall** be implemented as per the algorithm below:
  - During Initialization:
    
    ```
    out=IC
    prevo=IC
    previ=input
    ```
  - During Normal Operation:
    
    ```
    out=prevo + DT*input
    ```
    
    ```
    if out>UL
        out=UL
    elseif out<LL
        out=LL
    ```

  Where DT is sample Time.

  Equivalent Code – I

  ```
  if(initialized==0)
  {
    out=IC;
    prevo=IC;
    previ=input;
  }
  else
  {
    out=prevo + DT*input;
    if (out>UL)
        out=UL;
    elseif(out<LL)
        out=LL;
    previ= input;
    prevo=out;
  }
  ```

  Equivalent Code - II

  ```
  if(initialized==0)
  {
    out=IC;
    prevo=IC;
    previ=input;
  }
  else
  {
    out=prevo + DT*input;
    previ= input;
    prevo=out;
    if (out>UL)
        out=UL;
    elseif(out<LL)
        out=LL;
  }
  ```

**OR**

Improper Code !!!
Integrator – Design Issues

Limits the State

Limits the Output
Integrator coming out of Saturation
Proper Functional Requirements

- The Integrator shall be implemented as per the equation:
  \[ \text{Output} = \text{Previous Output} + DT \times \text{input} \]
  Where, Previous Output is the output obtained at the previous execution frame and DT is the sample time.

- The output During the first frame of execution shall be equal to IC.

- If the Output is greater than UL then Output shall be made equal to UL.

- If the Output is less than LL then the Output shall be made equal to LL.

- When the Integrator Output has reached a limit, the output will be limited such that any Input sign change will be immediately reflected in the Output.
Coverage Metrics in Existence

- **Structural Code Coverage**
  - Statement Coverage, Decision Coverage, Condition Coverage, Multiple Condition Coverage, Condition/Decision Coverage, Modified Condition/Decision Coverage

- **Simulink block coverage**
  - Decision, Condition, MC/DC, Look-up Table, Signal Range

- **Drawback**
  - These metrics do not talk about the functionality of the control system element
Characteristics of a Functional Metric

- The metric should be functionality based.
- The metric should be based on the input - output relation of the block under test.
- The metric should be independent of the platform being used.
- The metric should have an capability of test Case optimization.
Metric Definition

- We define a pair of cells for each functional requirements.
- The first cell of the pair is discrete (TRUE/FALSE) which tells if a particular functional requirement is exercised or not.
- The second cell defines a “distance to coverage”, a continuous metric which can be minimized to ensure coverage.

<table>
<thead>
<tr>
<th>T/F</th>
<th>Distance to coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td></td>
</tr>
<tr>
<td>False</td>
<td></td>
</tr>
</tbody>
</table>
Integrator – Metric Definition

Distance from Min value of output and the LL

Distance from Max. value of output and the UL

Time in secs
Integrator -- Metric Definition
## Integrator Metrics

<table>
<thead>
<tr>
<th>#</th>
<th>Discrete Metric</th>
<th>Continuous Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Output of the Integrator has reached the UL</td>
<td>$\text{abs}(\min(\text{Output}-\text{UL}))$</td>
</tr>
<tr>
<td>2</td>
<td>Output of the Integrator has reached the LL</td>
<td>$\text{abs}(\min(\text{Output}-\text{LL}))$</td>
</tr>
<tr>
<td>3</td>
<td>Output is non-zero and lesser than UL and greater than LL</td>
<td>$\text{abs}(\min(\text{Output(NonZero)}-(\text{UL+LL})/2))$</td>
</tr>
<tr>
<td>4</td>
<td>Integrator comes out of saturation from UL</td>
<td>When Output $\approx$ UL, Drive input towards values $&lt;0$</td>
</tr>
<tr>
<td>5</td>
<td>Integrator comes out of saturation from LL</td>
<td>When Output $\approx$ LL, Drive input towards values $&gt;0$</td>
</tr>
</tbody>
</table>
Integrator -- Simulink Design Verifier

Test Case 1

Test Case 2

Test Case 3

100% Structural Coverage
## SDV – Integrator Report

<table>
<thead>
<tr>
<th>#:</th>
<th>Type</th>
<th>Model Item</th>
<th>Description</th>
<th>Test Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Decision</td>
<td><strong>Discrete-Time Integrator</strong></td>
<td>integration result &lt;= lower limit F</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Decision</td>
<td><strong>Discrete-Time Integrator</strong></td>
<td>integration result &lt;= lower limit T</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Decision</td>
<td><strong>Discrete-Time Integrator</strong></td>
<td>integration result &gt;= upper limit F</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Decision</td>
<td><strong>Discrete-Time Integrator</strong></td>
<td>integration result &gt;= upper limit T</td>
<td>2</td>
</tr>
</tbody>
</table>
**Integrator -- Reactis**

Test Case 1

Test Case 2
# Reactis – Integrator Report

System "Integrator"

<table>
<thead>
<tr>
<th>Subsystems</th>
<th>Covered</th>
<th>Local</th>
<th>Uncovered</th>
<th>Unreachable</th>
<th>Uncovered</th>
<th>Cumulative</th>
<th>Covered</th>
<th>Local</th>
<th>Uncovered</th>
<th>Unreachable</th>
<th>Uncovered</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Branches</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>100%</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>States</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>CSEPT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>Condition Actions</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>Transition Actions</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>Decisions</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>Conditions</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>MC/DC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>Boundaries</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>100%</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Lookup Targets</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>User-Defined Targets</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>Assertion Violations</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>Total</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>100%</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>100%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target</th>
<th>Type</th>
<th>Status</th>
<th>Test</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete-Time Integrator%b1</td>
<td>Branches</td>
<td>covered</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Discrete-Time Integrator%b2</td>
<td>Branches</td>
<td>covered</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Discrete-Time Integrator%b3</td>
<td>Branches</td>
<td>covered</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>in1=0.0</td>
<td>Boundaries</td>
<td>covered</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
Output Comparison for Structural Coverage
Tests of Integrator

![Graph showing output comparison for structural coverage tests of integrator.](image)
## Integrator – Moog Functional Coverage Tool

<table>
<thead>
<tr>
<th>Metric</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output &gt;= UL</td>
<td>65 frames</td>
</tr>
<tr>
<td>Output &lt;= LL</td>
<td>41 frames</td>
</tr>
<tr>
<td>Output is non-zero, Output&lt;LL and &gt;UL</td>
<td>395 frames</td>
</tr>
<tr>
<td>Coming out of saturation of UL</td>
<td>1 transition</td>
</tr>
<tr>
<td>Coming out of saturation of UL</td>
<td>1 transition</td>
</tr>
</tbody>
</table>

The graph shows the input and output signals with the following lines:
- **Input**: Blue line
- **Output**: Green line
- **UL**: Red line
- **LL**: Cyan line
Persistence On/Off -- Algorithm

Inputs: INP, Init, Pers_On, Pers_Off
Output: OUT
During Initialization: OUT = Init
During normal operation:
if (INP is TRUE and has remained TRUE for Pers_On frames)
  OUT = TRUE
elseif (INP is FALSE and has remained FALSE for Pers_Off frames)
  OUT = FALSE
Else
  OUT = Previous frame value of OUT
Persistence On/Off – Design Variations

Signal 1
Signal Builder
Scope

PersistenceOnOff
Pers_On = 0.5 sec
Pers_Off = 1 sec

PersistenceOn
Pers_On = 0.5 sec

PersistenceOff
Pers_Off = 1 sec
Persistence On/Off – Design Comparison
Persistence On + Persistence Off $\sim$ Persistence OnOff
Metric Definition for Persistence OnOff
## Persistence OnOff Metrics

<table>
<thead>
<tr>
<th>#</th>
<th>Discrete Metric</th>
<th>Continuous Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IC is tested for TRUE value</td>
<td>NA</td>
</tr>
<tr>
<td>2</td>
<td>IC is tested for FALSE value</td>
<td>NA</td>
</tr>
<tr>
<td>3</td>
<td>Input has a TRUE pulse whose width is less than $\text{PersOn}$</td>
<td>$\text{abs}(\text{PersOn}/2 - \text{min. TRUE pulse Width})$</td>
</tr>
<tr>
<td>4</td>
<td>Input has a TRUE pulse whose width is greater than $\text{PersOn}$</td>
<td>$\text{abs}(\text{PersOn} - \text{max. TRUE pulse Width})$</td>
</tr>
<tr>
<td>5</td>
<td>Input has a FALSE pulse whose width is less than $\text{PersOff}$</td>
<td>$\text{abs}(\text{PersOff}/2 - \text{min. FALSE pulse Width})$</td>
</tr>
<tr>
<td>6</td>
<td>Input has a FALSE pulse whose width is greater than $\text{PersOff}$</td>
<td>$\text{abs}(\text{PersOn}/2 - \text{max. FALSE pulse Width})$</td>
</tr>
</tbody>
</table>
Persistence On/Off – Simulink Design Verifier

Test Case 1 (IC = 1)

Test Case 2 (IC = 0)
## SDV – Persistence On/Off Report Summary

<table>
<thead>
<tr>
<th>Metric</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision (D1)</td>
<td>100% (12/12) decision outcomes</td>
</tr>
<tr>
<td>Condition (C1)</td>
<td>100% (2/2) condition outcomes</td>
</tr>
</tbody>
</table>

100 % Structural Coverage
Persistence On/Off -- Reactis

Test Case 1

Test Case 2
### Reactis – Persistence On/Off Report

System "**PersistenceOnOff**"

<table>
<thead>
<tr>
<th></th>
<th>Covered</th>
<th>Unreachable</th>
<th>Uncovered</th>
<th>Cumulative</th>
<th>Unreachable</th>
<th>Uncovered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subsystems</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Branches</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>100%</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>States</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CSEPT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Condition Actions</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Transition Actions</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Decisions</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>100%</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Conditions</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>100%</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>MC/DC</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>100%</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Boundaries</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>100%</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Lookup Targets</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>User-Defined Targets</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>100%</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Assertion Violations</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>26</td>
<td>0</td>
<td>0</td>
<td>100%</td>
<td>28</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Child</th>
<th>Subsystems</th>
<th>Branches</th>
<th>States</th>
<th>CSEPT</th>
<th>Condition Actions</th>
<th>Transition Actions</th>
<th>Decisions</th>
<th>Conditions</th>
<th>MC/DC</th>
<th>Boundaries</th>
<th>Lookup Targets</th>
<th>User-Defined Targets</th>
<th>Assertion Violations</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Delay</td>
<td>--</td>
<td>(2) 100%</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>External IC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(2) 100%</td>
</tr>
</tbody>
</table>
## Persistence On/Off – Moog Functional Coverage Tool

<table>
<thead>
<tr>
<th>Metric</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC tested for TRUE</td>
<td>0</td>
</tr>
<tr>
<td>IC tested for FALSE</td>
<td>1</td>
</tr>
<tr>
<td>Input has a TRUE pulse whose width is less than PersOn</td>
<td>1</td>
</tr>
<tr>
<td>Input has a TRUE pulse whose width is greater than PersOn</td>
<td>1</td>
</tr>
<tr>
<td>Input has a FALSE pulse whose width is lesser than PersOff</td>
<td>1</td>
</tr>
<tr>
<td>Input has a FALSE pulse whose width is greater than PersOff</td>
<td>1</td>
</tr>
</tbody>
</table>

### Input vs. Output Graph

- **Inp**
- **Out**

- **PersOn = 0.5 sec**
- **PersOff = 1 sec**
Moog Functional Coverage Review Tool

Preprocessing:
- Log All Signals of the model
- List of Functional Blocks
- Identify Review function for each functional block

Parser:
- Execute the Model

Post Processing:
- Run the Review function with I/O of each functional block
- Report Generation

Manually Developed Review Function
Model Under Test

AnalogIn1

1

AnalogIn2

2

15
Constant1

<=
Relational Operator

>=
Relational Operator1

-15
Constant2

K T_s (z+1)
2(z-1)
Discrete-Time Integrator
UL=10,
LL=-10

Relational Operator1

AND

Switch
u2 =~ 0

1
Constant

1
AnalogOut1
### Functional Coverage Report

<table>
<thead>
<tr>
<th><strong>Discrete-Time Integrator</strong></th>
<th>Dmetric</th>
<th>Cmetric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of samples Output is equal to UL and Input is greater than 0</td>
<td>158</td>
<td>0</td>
</tr>
<tr>
<td>Number of samples Output is equal to LL and Input is less than 0</td>
<td>62</td>
<td>0</td>
</tr>
<tr>
<td>Number of samples the Output is less than UL and greater than LL and is non-zero</td>
<td>280</td>
<td>0.003</td>
</tr>
<tr>
<td>Number of times the output comes out of saturation from UL</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Number of times the output comes out of saturation from LL</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>AND</strong></th>
<th>Dmetric</th>
<th>Cmetric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Showing the effect of Input1 of AND on the Output</td>
<td>1</td>
<td>N/A</td>
</tr>
<tr>
<td>Showing the effect of Input2 of AND on the Output</td>
<td>1</td>
<td>N/A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Switch</strong></th>
<th>Dmetric</th>
<th>Cmetric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of frames the Input1 and Input3 are unequal and Input2 is TRUE</td>
<td>269</td>
<td>0.64463</td>
</tr>
<tr>
<td>Number of frames the Input1 and Input3 are unequal and Input2 is FALSE</td>
<td>232</td>
<td>0.97403</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Relational Operator</strong></th>
<th>Dmetric</th>
<th>Cmetric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of frames the output is TRUE</td>
<td>381</td>
<td>0.0106</td>
</tr>
<tr>
<td>Number of frames the output is FALSE</td>
<td>120</td>
<td>0.0827</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Relational Operator1</strong></th>
<th>Dmetric</th>
<th>Cmetric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of frames the output is TRUE</td>
<td>389</td>
<td>0.1711</td>
</tr>
<tr>
<td>Number of frames the output is FALSE</td>
<td>112</td>
<td>0.0197</td>
</tr>
</tbody>
</table>
Test case were designed to provide complete functional coverage as defined using the Moog Functional metrics.
Different Mutants

- Logical Mutants (&, |, ~<, ^)
- Arithmetic mutants (+, -, *, /)
- Data Mutants (add +/- error of 0.001)
- Variable Mutants (replace Variable with another Variable in the code)
- Manual Mutants

-- Intern students from Colleges are making these for us.
Nand, Xor Mutant Analysis

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A &amp;B</th>
<th>~(A&amp;B)</th>
<th>A XOR B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

These are the Test Cases that prove the MCDC of an AND gate.
MCDC Analysis for Complex Combinational Logic

- 100 Complex Combinational Logics with 16 inputs.
- Automated test cases to achieve MCDC.
- 165 mutants for each combinational logic. <All possible mutants>.

Average Probability of killing a Mutant is 98 %
What are we working on now?

- Complete Qualified tool as per DO-178B guidelines.
- Automated Test Case generation using the Functional Metrics as assertions to be achieved.
- Defining System level functional metrics
- Integrate the functional metrics to Simulink Design verifier!!??
Genetic Algorithm for test Case generation

- We are researching various Optimization tool for Test case optimization.
- GA has provided the initial results for an Integrator with limits.
- A sine signal whose Amplitude, Frequency and Bias are varied by the GA algorithm to achieve Zero Dmetrics
Test Case generation for Integrator using GA

```matlab
1 - global Amp
2 - global Freq
3 - global Bias
4 - global accumulate;
5 - accumulate=[];
6 - % A1=10;B1=15;C1=20;
7 - % A2=0.1;B2=0.002;C2=2;
8 - % A3=-10;B3=-1;C3=13;
9 - A1=2;B1=4;C1=6;
10 - A2=0.1;B2=0.02;C2=1;
11 - A3=-2;B3=0;C3=2;
12 - pause(10)
13 -
14 - IntPop=[A1 A2 A3
15 - A1 B2 B3
16 - A1 C2 C3
17 - B1 A2 B3
18 - B1 B2 C3
19 - B1 C2 A3
20 - C1 A2 C3
21 - C1 B2 A3
22 - C1 C2 B3 ];
23 - options = gaoptimset('PlotFcns',...}
24 - {@gaplotbestf,@gaplotbestindiv,@gaplotdistance,@gaplotscores},...}
25 - 'PopulationSize',20,...
```
Take Away

- 100% Structural code Coverage doesn’t mean 100 % functional coverage.
- Test, Test, Test !! Add Random tests..
- Verify the quality of the tests using mutations.
- Follow process – Like doing reviews.. few errors are easily found by reviews than tests…
- There are no short cuts in testing.. Processes can be automated but cannot be removed.
Thank You

- [Email](mailto:cchethan@moog.com)
- LinkedIn: Chethan C U

Next Generation Flight Controls
Moog Provides a complete end to end solution in the development and V&V of the Lateral Control Electronics for Boeing 747-8 Aircraft