# Pilot Evaluation of Model Based Design Tooling for Guidance, Navigation, and Control Flight Software Development

Brian R. Jamison, Mike R. Hannan,
James T. Kaidy, Juan I. Orphee, Nick S. Olson

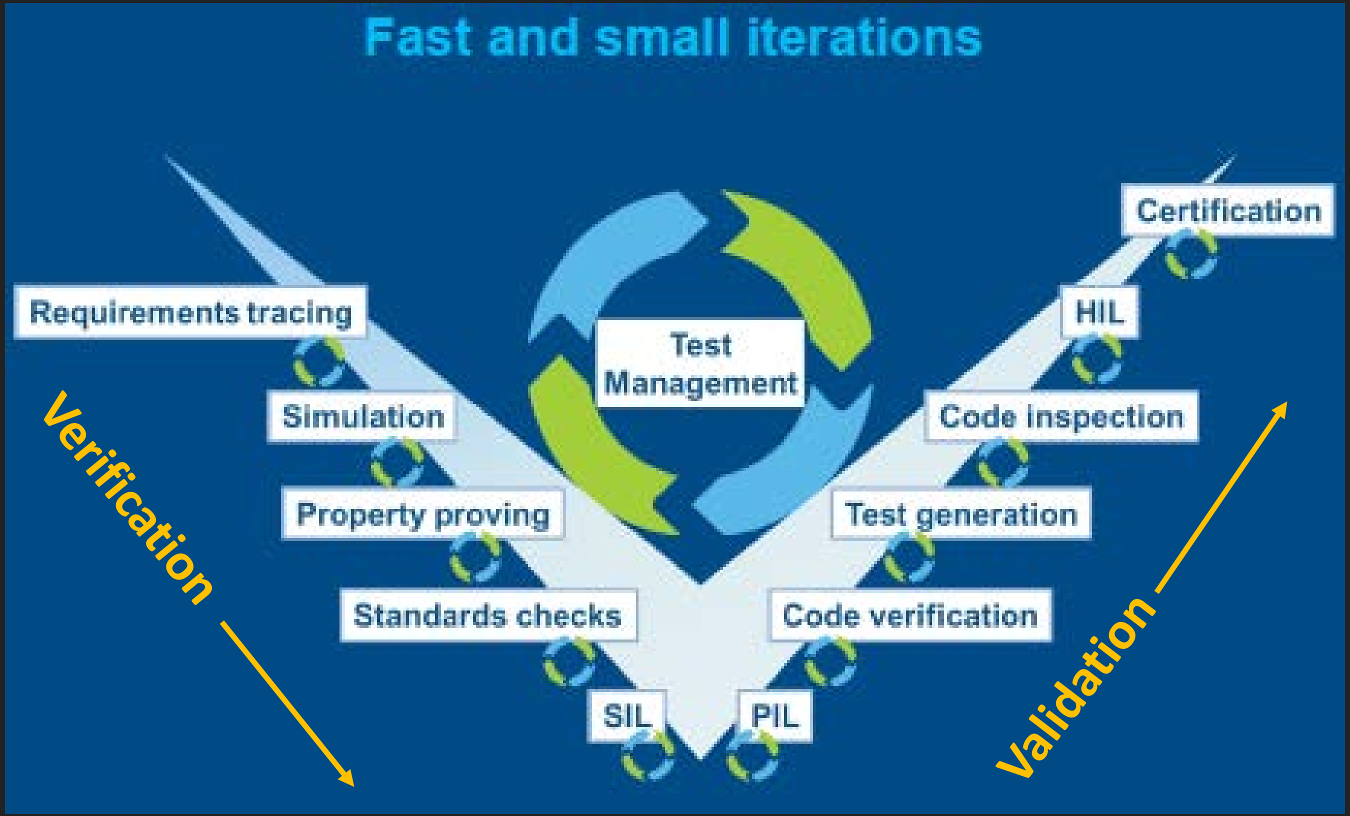
Guidance, Navigation, & Control

NASA Marshall Space Flight Center

9-12 December 2019

# What & Why Model Based Design (MBD)

- MBD systematically uses models throughout the development process for requirements, design, analysis, simulation, verification and validation, and documentation
- An MBD approach seeks to incorporate models into an automated, concurrent design process intended to minimize potential for human error
- Improvements offered by an MBD approach include efficiency improvements by automating aspects of requirements testing and documentation
- An advantage of MathWorks MBD tooling is the model visualization Simulink naturally incorporates into the design process



# NASA LADEE MBD Experience

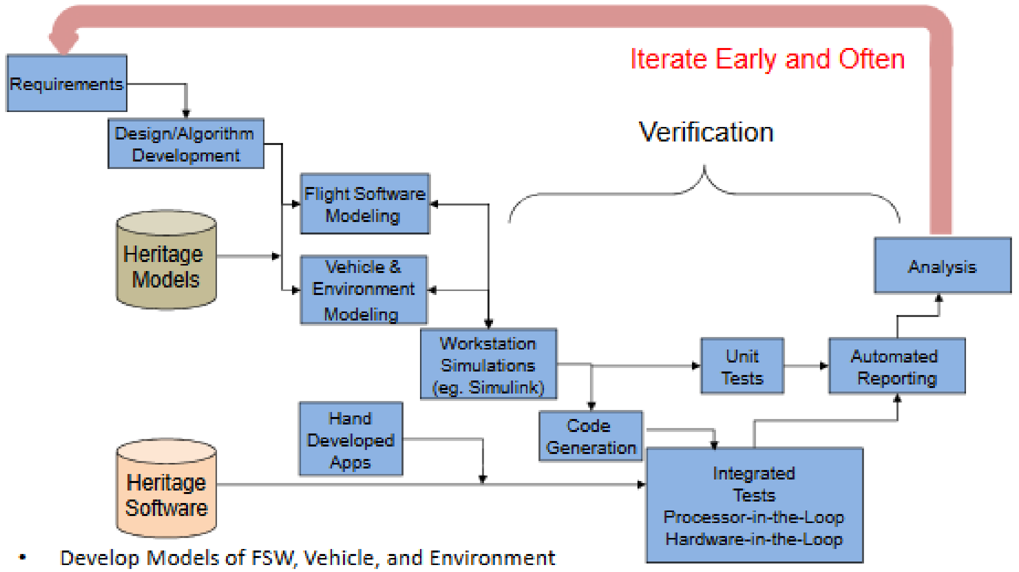
- "Compared with using Model-Based Design, hand-coding the flight software would have taken longer and made collaboration more difficult. Managers and hardware system engineers understand Simulink models, making it easy to achieve consensus because everyone knows what's going on in the software."

  \- *Dr. Karen Gundy-Burlet, NASA Ames Research Center*

## Model Based Development

- Issues:
  - Low Cost Mission and fixed schedule demanded rapid, low cost flight software development process
  - Simulations needed for FSW Verification and Mission Operations development, training, and command verification.
- Solution:
  - Use model based development approach
    - Automatic conversion of Models to FSW allows development and testing of algorithms which then becomes Software. Avoids "throwing it over the fence to be coded".
  - Developed multiple simulators of varying degrees of fidelity (WSIM, PIL, HIL)
  - Developed Simulink Interface Layer
    - Allows immediate translation from models to Code allowing rapid turnaround
  - Developed an automated test harness for rapid turnaround of verification results
- Result:
  - Model Based Development coupled with "push button" code generation and testing was highly effective for rapid software development.
  - Models and Simulations used extensively in Mission Operations.
    - WSIM provided faster than real time capability for rapid command verification.
    - Processor in the Loop and Hardware in the Loop simulations provided high fidelity simulations for critical maneuver verification, Ops training, and debugging anomalies
    - Fully Coupled Simulations (Power, Thermal, Propulsion, Attitude Control) provided better insight for coupled problems.

## Iterative Development




- Develop Models of FSW, Vehicle, and Environment
- Automatically generate High-Level Control Software
- Integrate with hand-written and heritage software.
- Iterate while increasing fidelity of tests – Workstation Sim (WSIM), Processor-In-The-Loop (PIL), Hardware-in-the-Loop (HIL)
- Automated self-documenting tests providing traceability to requirements

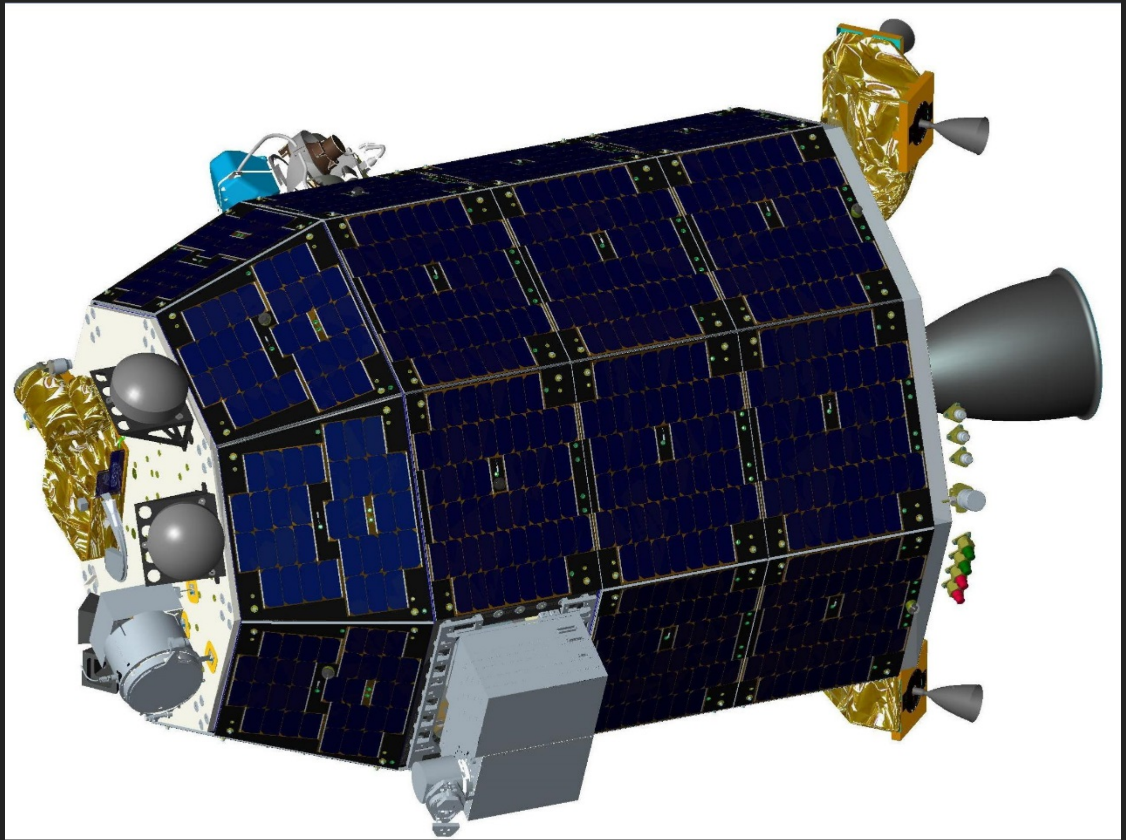
# NASA LADEE MBD Experience

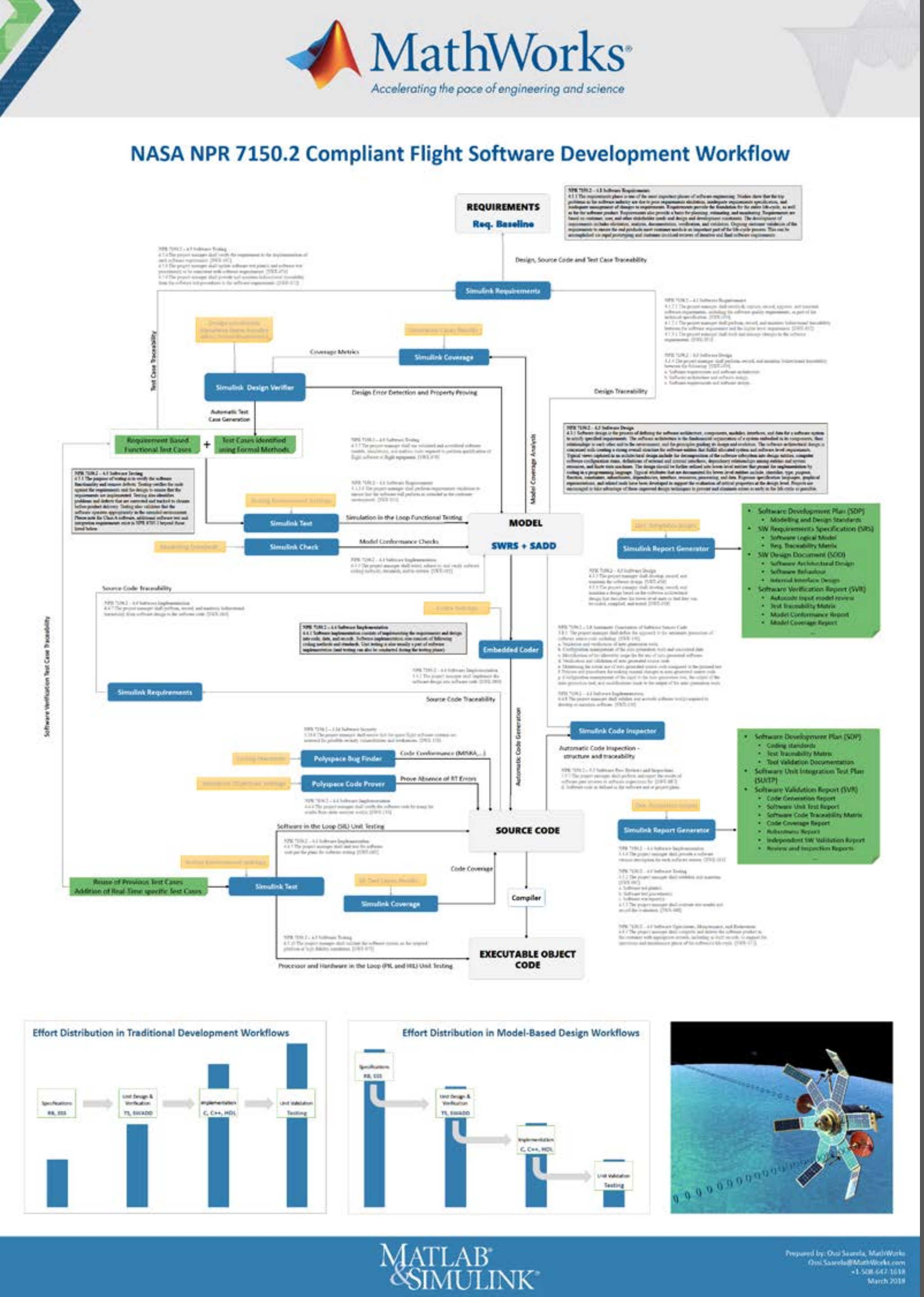
- "Compared with using Model-Based Design, hand-coding the flight software would have taken longer and made collaboration more difficult.   Managers and hardware system engineers understand Simulink models, making it easy to achieve consensus because everyone knows what's going on in the software."

  \- *Dr. Karen Gundy-Burlet, NASA Ames Research Center*

## Model Based Development

- Issues:
  - Low Cost Mission and fixed schedule demanded rapid, low cost flight software development process
  - Simulations needed for FSW Verification and Mission Operations development, training, and command verification.
- Solution:
  - Use model based development approach
    - Automatic conversion of Models to FSW allows development and testing of algorithms which then becomes Software. Avoids "throwing it over the fence to be coded".
  - Developed multiple simulators of varying degrees of fidelity (WSIM, PIL, HIL)
  - Developed Simulink Interface Layer
    - Allows immediate translation from models to Code allowing rapid turnaround
  - Developed an automated test harness for rapid turnaround of verification results
- Result:
  - Model Based Development coupled with "push button" code generation and testing was highly effective for rapid software development.
  - Models and Simulations used extensively in Mission Operations.
    - WSIM provided faster than real time capability for rapid command verification.
    - Processor in the Loop and Hardware in the Loop simulations provided high fidelity simulations for critical maneuver verification, Ops training, and debugging anomalies
    - Fully Coupled Simulations (Power, Thermal, Propulsion, Attitude Control) provided better insight for coupled problems.






# MathWorks MBD Tooling

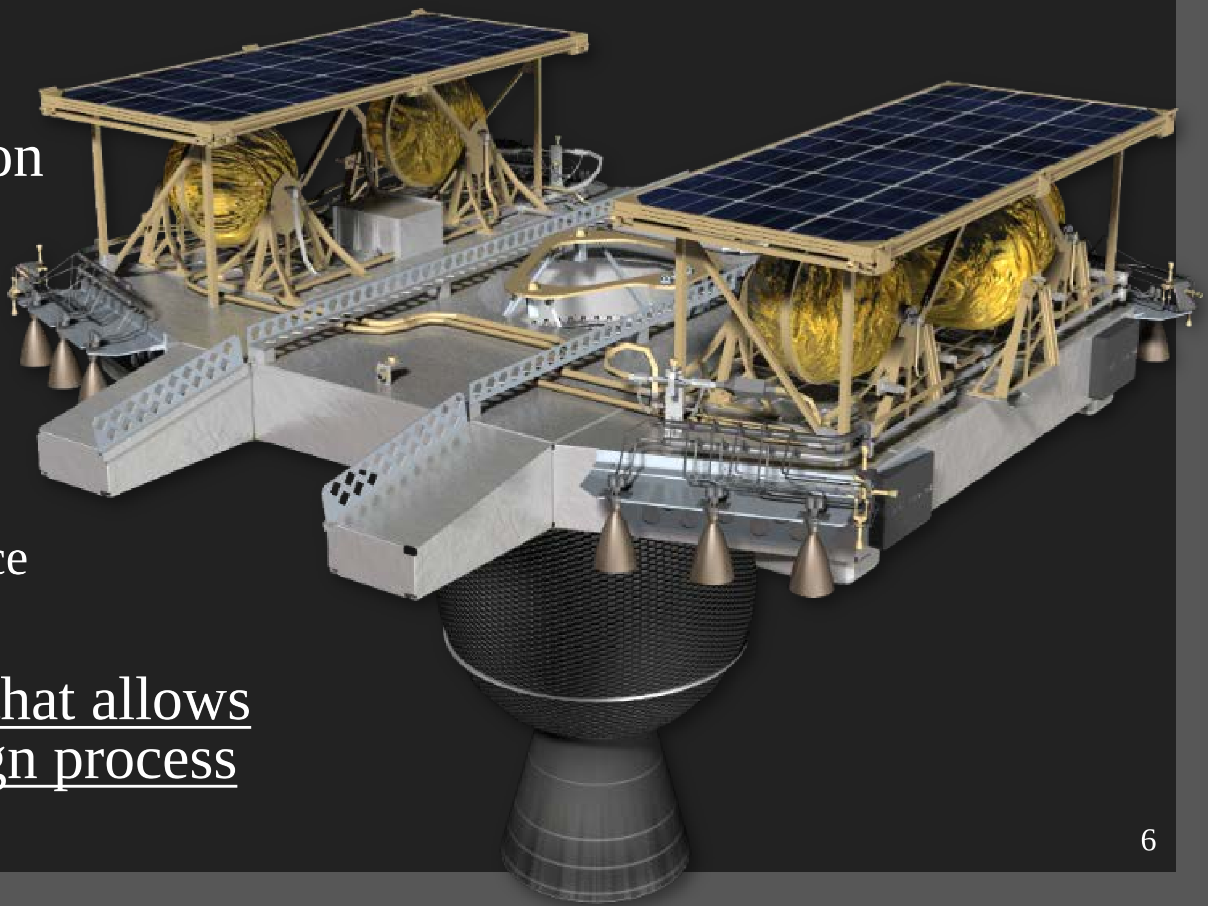
- MathWorks worked with the NASA Engineering & Safety Center (NESC) to develop tooling that addresses about 80% of NPR 7150.2 requirements, including:
  - Software requirements
  - Software design
  - Software implementation
  - Software testing
- NPR 7150.2 Requirements outside of the MathWorks workflow include:
  - Software architecture requirements
  - Project management requirements
- Consultation ongoing with the Marshall Space Flight Center (MSFC) software division to ensure they concur with our MBD approach

# VIPER Lunar Lander Pilot Program

- MSFC Guidance, Navigation, & Control (GNC) group used the VIPER Lander as a pilot program to use of MathWorks MBD tools for GNC FSW development
- Goal: Apply an MBD approach and tooling to condense schedule, reduce needed resources, and improve quality

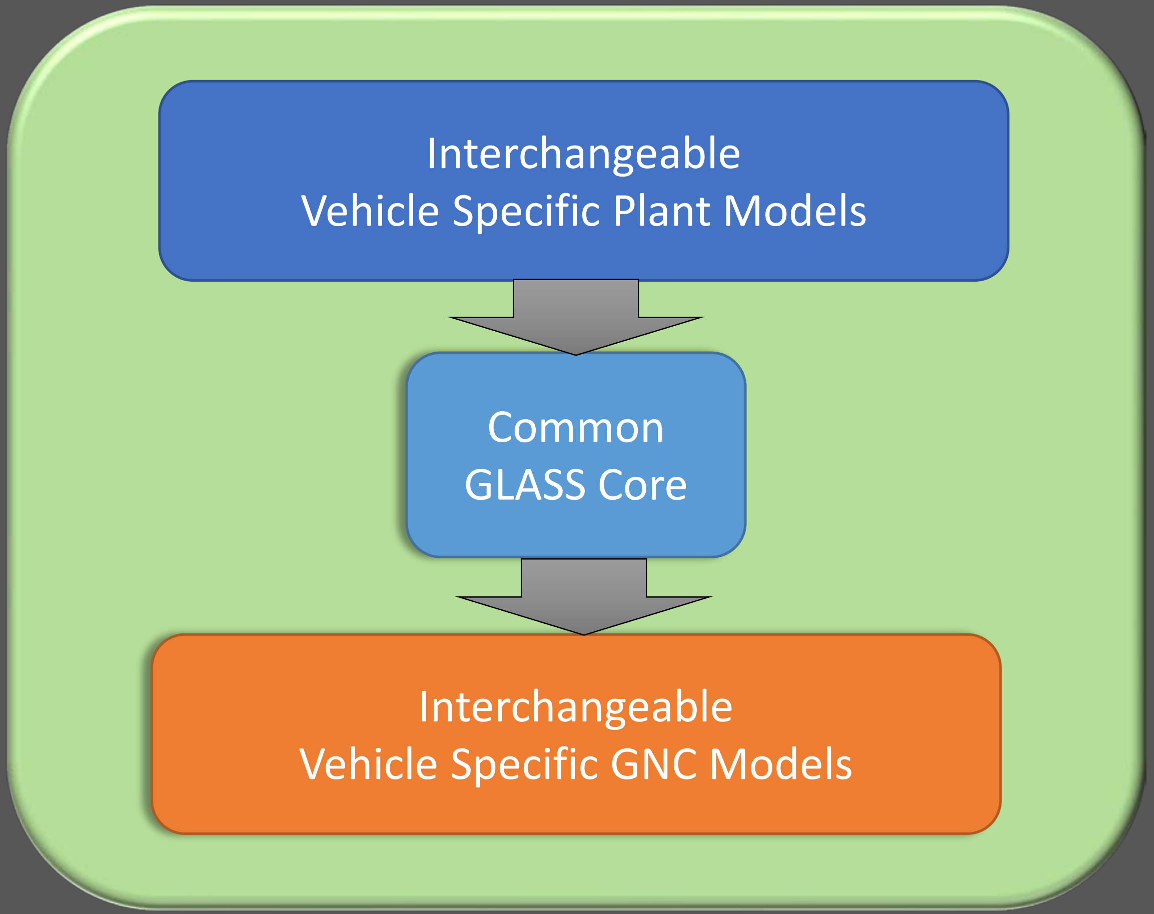
Expected MBD approach benefits:

- Facilitates requirements implementation verification
- Automates:
  - Requirements verification testing
  - Continuous model and flight code testing
  - Modeling standards (DO-178C) enforcement
  - Code-generation from the Simulink model
  - Static code analysis to ensure coding standard compliance
  - Report generation
- Establish a highly automated, disciplined process that allows repeated testing of the system throughout the design process



# Generalized Lander Simulation in Simulink (GLASS)

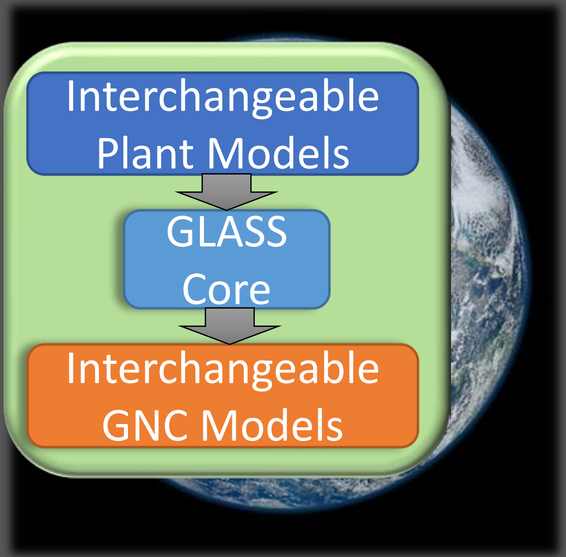
- 6-DoF aerospace vehicle simulation environment designed to be:
  - Modern
  - Flexible
  - User-friendly
- Features
  - Simulink Framework
    - Interfaces with MathWorks MBD products
    - Supports auto-coding to C/C++
  - Dynamics constructed using Simscape Multibody
    - Provides flexible & modular physics engine for simulation
    - GLASS Core is common to all simulations
  - Modular GNC algorithms
    - Mirror FSW functions in generated code



# MBD for GNC FSW Development

**Check Requirements: Simulink Requirements & Simulink Test**

- ✓ Ensure code satisfies requirements




```
// NASA MSFC GNC
Quality
Autocode
GNC
FSW
```

**Model & GNC Code Development: Simulink & Model Advisor**

- ✓ Highly modular software development
- ✓ Enforce DO178 and custom standards

**Static Code Check: Polyspace**

- ✓ Catch run time errors,
- ✓ Enforce MISRA, JSF, etc.

**Auto-code GNC Software: Simulink Embedded Coder**

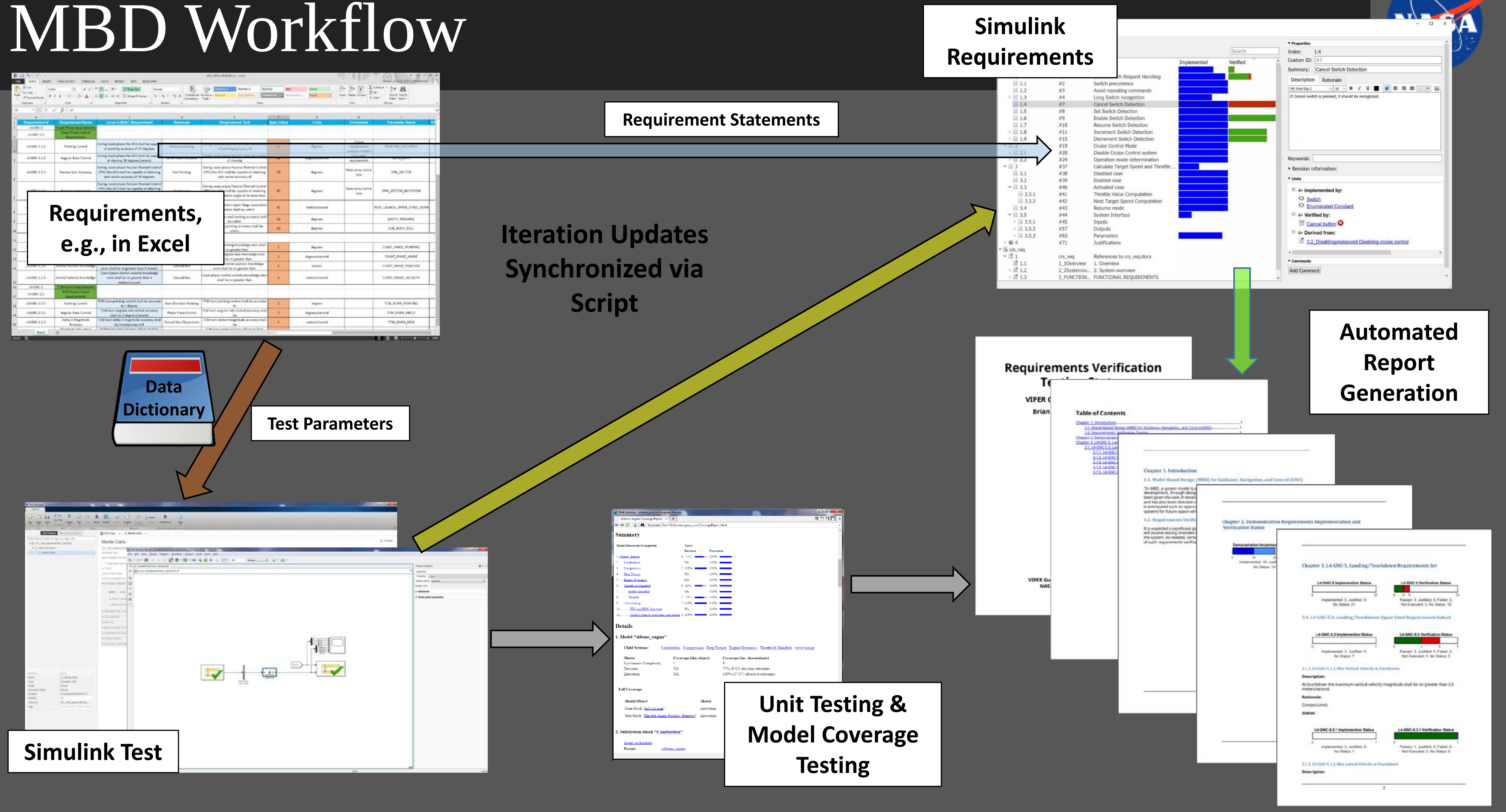
- ✓ Customize auto-code to meet FSW standards
- ✓ Enforce selected standards
- ✓ cFS compatible (optional)

**Generate Automated Reports: Report Generator**

**Software Unit Testing: Simulink Model & Code Coverage**

- ✓ Ensure software modules perform as expected
- ✓ Ensure all code is exercised

# MBD Workflow



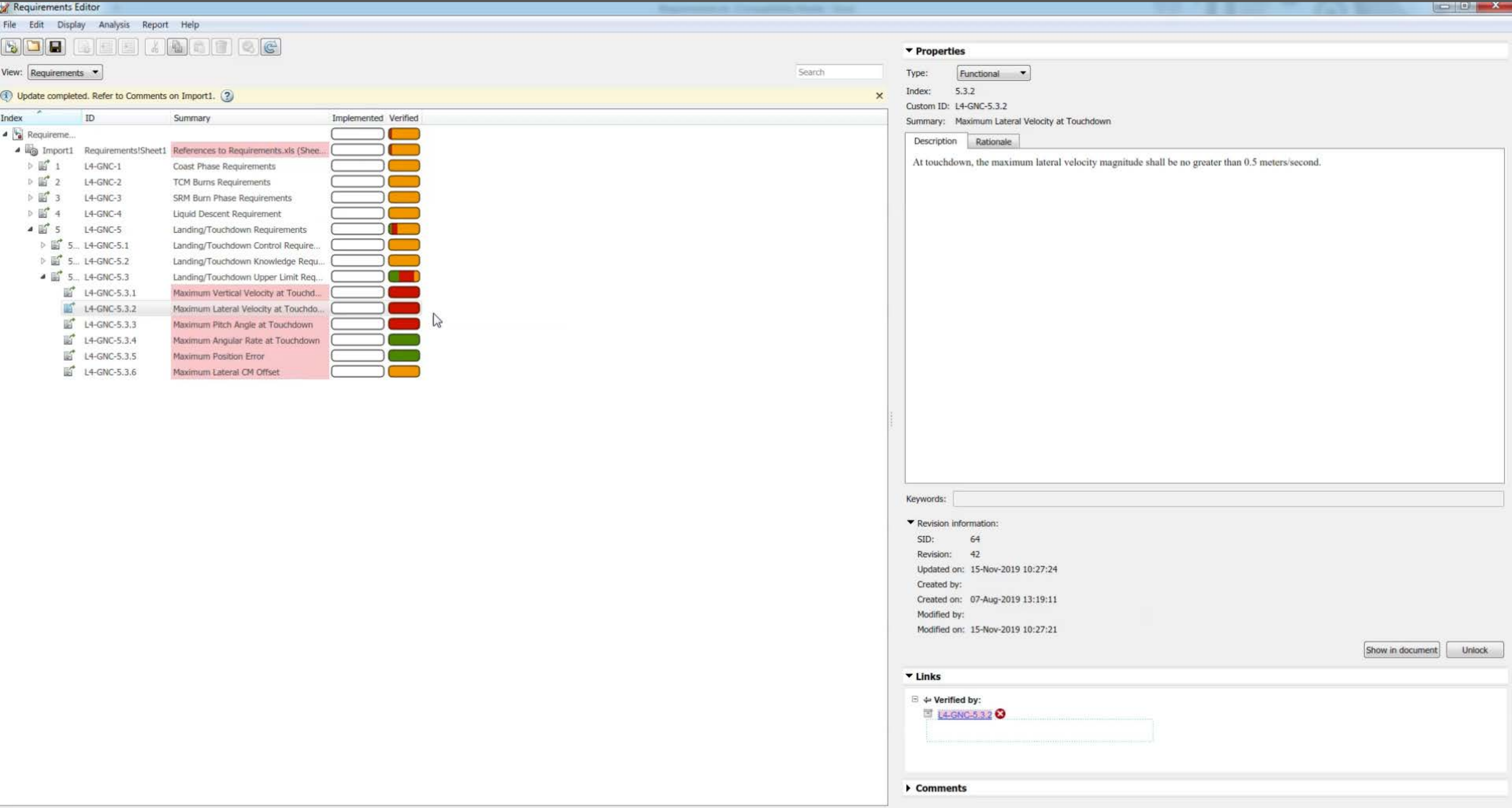
Requirements.xls [Compatibility Mode] - Excel

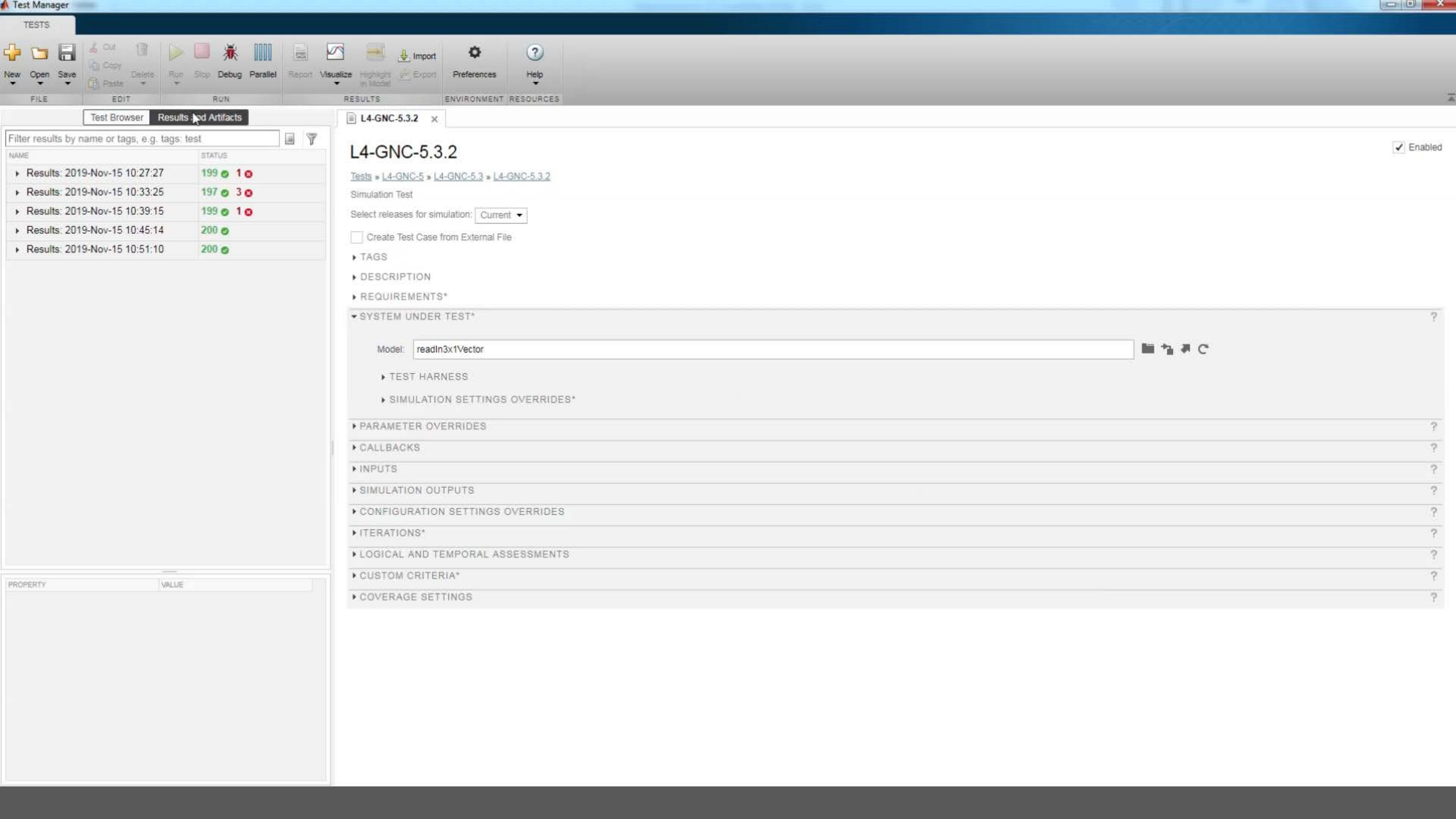
I64 | 0.5

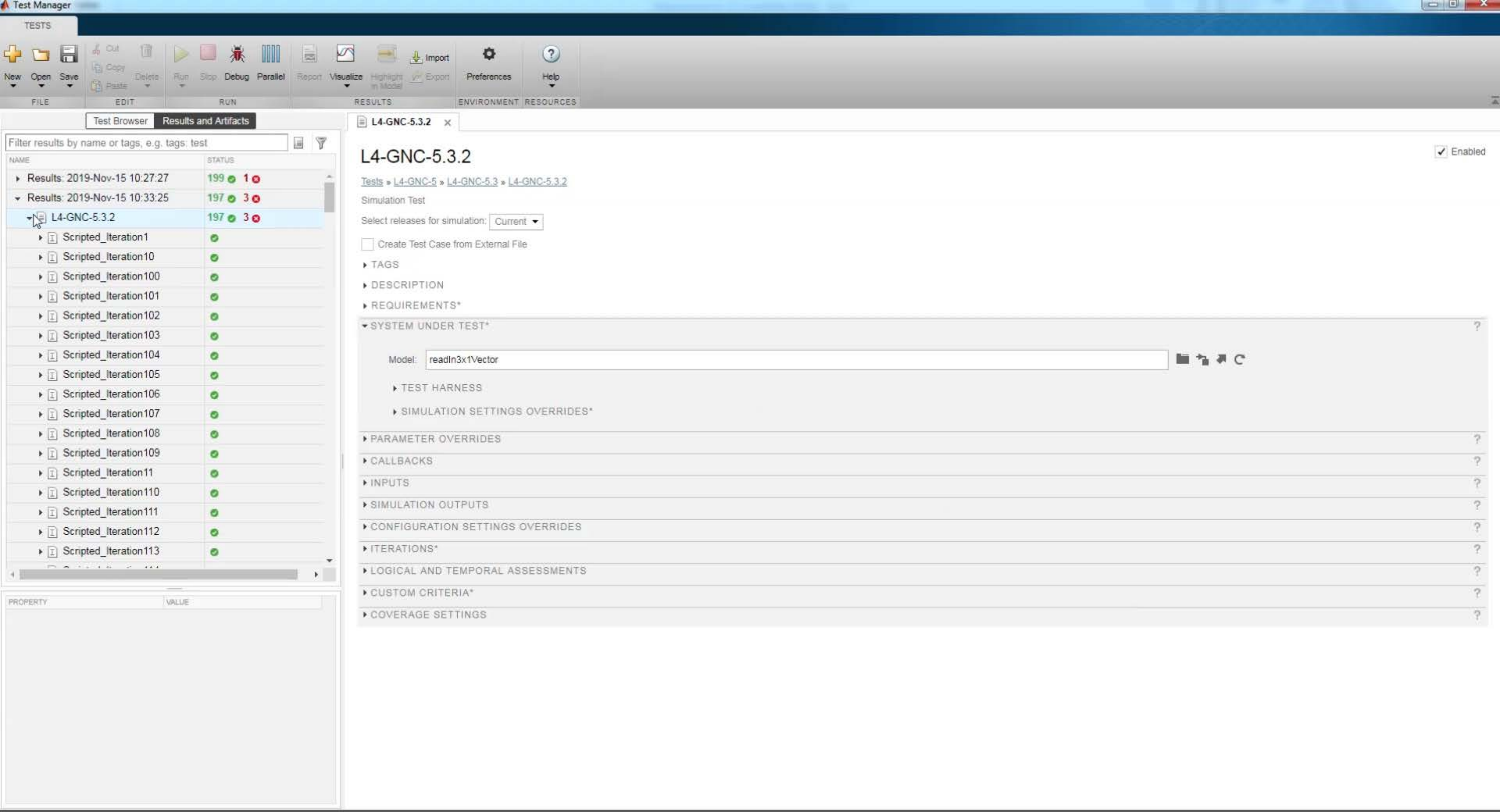
| Row | Requirement ID | Requirement Name | Level 4 GN&C Requirement | Rationale | Level 4 GN&C Requirement Text | Spec Value | Units |
|---|---|---|---|---|---|---|---|
| 62 | L4-GNC-5.3 | Landing/Touchdown Upper Limit Requirements | | | | | |
| 63 | L4-GNC-5.3.1 | Maximum Vertical Velocity at Touchdown | At touchdown, the maximum vertical velocity magnitude shall be no greater than 2.5 meters/second. | Contact Limits | At touchdown, the maximum vertical velocity magnitude shall be no greater than | 2.5 | meters/second |
| 64 | L4-GNC-5.3.2 | Maximum Lateral Velocity at Touchdown | At touchdown, the maximum lateral velocity magnitude shall be no greater than 0.5 meters/second. | Contact Limits | At touchdown, the maximum lateral velocity magnitude shall be no greater than | 0.5 | meters/second |
| 65 | L4-GNC-5.3.3 | Maximum Pitch Angle at Touchdown | At touchdown, the maximum pitch angle with respect to local vertical shall be no greater than 3 degrees. | Tip Over Limits | At touchdown, the maximum pitch angle with respect to local vertical shall be no greater than | 3 | degrees |
| 66 | L4-GNC-5.3.4 | Maximum Angular Rate at Touchdown | At touchdown, the maximum magnitude of the inertial angular rate shall be no greater than 0.1 degrees/second. | Tip Over Limits | At touchdown, the maximum magnitude of the inertial angular rate shall be no greater than | 0.1 | degrees/second |
| 67 | L4-GNC-5.3.5 | Maximum Position Error | At touchdown, the maximum position error from the nominal landing site shall be no greater than 100 meters. | Landed Payload Limits | At touchdown, the maximum position error from the nominal landing site shall be no greater than | 100 | meters |
| 68 | L4-GNC-5.3.6 | Maximum Lateral CM Offset | At touchdown the lateral CM offset shall be no greater than 77 millimeters. | Control Stability Margin | At touchdown the lateral CM offset shall be no greater than | 77 | millimeters |

Sheet1 | Functional | FlightSoftwareQualityChecks | ImportantMBDInstructions

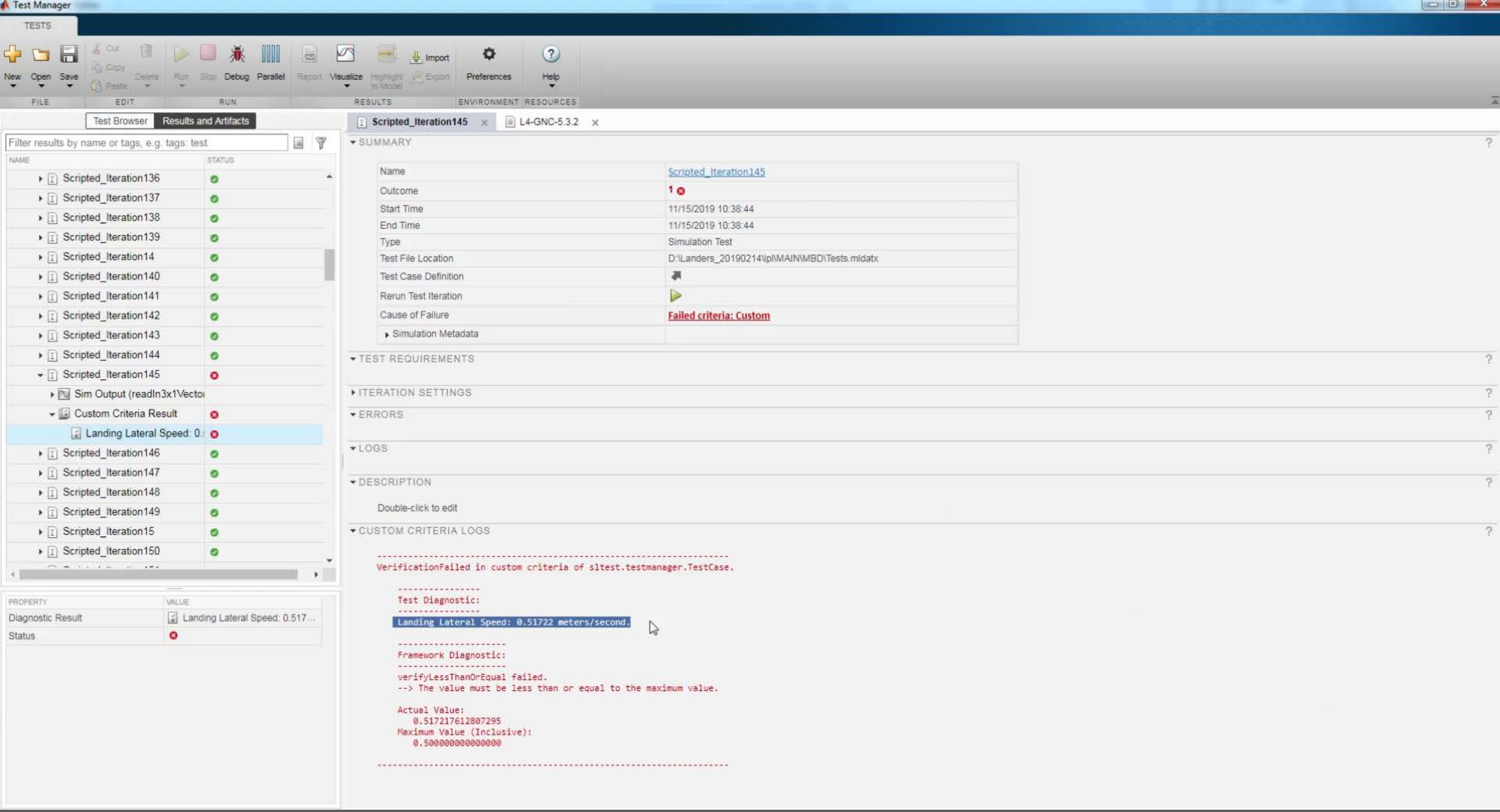
READY


Requirements Editor
File Edit Display Analysis Report Help
View: Requirements
Search
Update completed. Refer to Comments on Import1.
Index ID Summary Implemented Verified
Requireme...
Import1 Requirements!Sheet1 References to Requirements.xls (Shee...
1 L4-GNC-1 Coast Phase Requirements
2 L4-GNC-2 TCM Burns Requirements
3 L4-GNC-3 SRM Burn Phase Requirements
4 L4-GNC-4 Liquid Descent Requirement
5 L4-GNC-5 Landing/Touchdown Requirements
5... L4-GNC-5.1 Landing/Touchdown Control Require...
5... L4-GNC-5.2 Landing/Touchdown Knowledge Requ...
5... L4-GNC-5.3 Landing/Touchdown Upper Limit Req...
L4-GNC-5.3.1 Maximum Vertical Velocity at Touchd...
L4-GNC-5.3.2 Maximum Lateral Velocity at Touchdo...
L4-GNC-5.3.3 Maximum Pitch Angle at Touchdown
L4-GNC-5.3.4 Maximum Angular Rate at Touchdown
L4-GNC-5.3.5 Maximum Position Error
L4-GNC-5.3.6 Maximum Lateral CM Offset
Properties
Type: Functional
Index: 5.3.2
Custom ID: L4-GNC-5.3.2
Summary: Maximum Lateral Velocity at Touchdown
Description Rationale
At touchdown, the maximum lateral velocity magnitude shall be no greater than 0.5 meters/second.
Keywords:
Revision information:
SID: 64
Revision: 42
Updated on: 15-Nov-2019 10:27:24
Created by:
Created on: 07-Aug-2019 13:19:11
Modified by:
Modified on: 15-Nov-2019 10:27:21
Show in document Unlock
Links
Verified by:
L4-GNC-5.3.2
Comments


Test Manager
TESTS
New
Open
Save
Cut
Copy
Paste
Delete
Run
Stop
Debug
Parallel
Report
Visualize
Highlight in Model
Import
Export
Preferences
Help
FILE
EDIT
RUN
RESULTS
ENVIRONMENT
RESOURCES
Test Browser
Results and Artifacts
Filter results by name or tags, e.g. tags: test
NAME
STATUS
Results: 2019-Nov-15 10:27:27 199 1
Results: 2019-Nov-15 10:33:25 197 3
Results: 2019-Nov-15 10:39:15 199 1
Results: 2019-Nov-15 10:45:14 200
Results: 2019-Nov-15 10:51:10 200
PROPERTY
VALUE
L4-GNC-5.3.2
Enabled
L4-GNC-5.3.2
Tests » L4-GNC-5 » L4-GNC-5.3 » L4-GNC-5.3.2
Simulation Test
Select releases for simulation: Current
Create Test Case from External File
TAGS
DESCRIPTION
REQUIREMENTS*
SYSTEM UNDER TEST*
Model: readIn3x1Vector
TEST HARNESS
SIMULATION SETTINGS OVERRIDES*
PARAMETER OVERRIDES
CALLBACKS
INPUTS
SIMULATION OUTPUTS
CONFIGURATION SETTINGS OVERRIDES
ITERATIONS*
LOGICAL AND TEMPORAL ASSESSMENTS
CUSTOM CRITERIA*
COVERAGE SETTINGS



Test Manager
TESTS
New
Open
Save
Cut
Copy
Paste
Delete
Run
Stop
Debug
Parallel
Report
Visualize
Highlight in Model
Import
Export
Preferences
Help
FILE
EDIT
RUN
RESULTS
ENVIRONMENT
RESOURCES
Test Browser
Results and Artifacts
Filter results by name or tags, e.g. tags: test
NAME
STATUS
Results: 2019-Nov-15 10:27:27 199 1
Results: 2019-Nov-15 10:33:25 197 3
L4-GNC-5.3.2 197 3
Scripted_Iteration1
Scripted_Iteration10
Scripted_Iteration100
Scripted_Iteration101
Scripted_Iteration102
Scripted_Iteration103
Scripted_Iteration104
Scripted_Iteration105
Scripted_Iteration106
Scripted_Iteration107
Scripted_Iteration108
Scripted_Iteration109
Scripted_Iteration11
Scripted_Iteration110
Scripted_Iteration111
Scripted_Iteration112
Scripted_Iteration113
PROPERTY
VALUE
L4-GNC-5.3.2
L4-GNC-5.3.2
Enabled
Tests » L4-GNC-5 » L4-GNC-5.3 » L4-GNC-5.3.2
Simulation Test
Select releases for simulation: Current
Create Test Case from External File
TAGS
DESCRIPTION
REQUIREMENTS*
SYSTEM UNDER TEST*
Model: readIn3x1Vector
TEST HARNESS
SIMULATION SETTINGS OVERRIDES*
PARAMETER OVERRIDES
CALLBACKS
INPUTS
SIMULATION OUTPUTS
CONFIGURATION SETTINGS OVERRIDES
ITERATIONS*
LOGICAL AND TEMPORAL ASSESSMENTS
CUSTOM CRITERIA*
COVERAGE SETTINGS


Test Manager
TESTS
New
Open
Save
Cut
Copy
Paste
Delete
Run
Stop
Debug
Parallel
Report
Visualize
Highlight in Model
Import
Export
Preferences
Help
FILE
EDIT
RUN
RESULTS
ENVIRONMENT
RESOURCES
Test Browser
Results and Artifacts
Filter results by name or tags, e.g. tags: test
NAME
STATUS
Scripted_Iteration136
Scripted_Iteration137
Scripted_Iteration138
Scripted_Iteration139
Scripted_Iteration14
Scripted_Iteration140
Scripted_Iteration141
Scripted_Iteration142
Scripted_Iteration143
Scripted_Iteration144
Scripted_Iteration145
Sim Output (readIn3x1Vecto
Custom Criteria Result
Landing Lateral Speed: 0.
Scripted_Iteration146
Scripted_Iteration147
Scripted_Iteration148
Scripted_Iteration149
Scripted_Iteration15
Scripted_Iteration150
PROPERTY
VALUE
Diagnostic Result
Landing Lateral Speed: 0.517...
Status
Scripted_Iteration145
L4-GNC-5.3.2
SUMMARY
Name
Scripted_Iteration145
Outcome
1
Start Time
11/15/2019 10:38:44
End Time
11/15/2019 10:38:44
Type
Simulation Test
Test File Location
D:\Landers_20190214\lp\MAIN\MBD\Tests.mldatx
Test Case Definition
Rerun Test Iteration
Cause of Failure
Failed criteria: Custom
Simulation Metadata
TEST REQUIREMENTS
ITERATION SETTINGS
ERRORS
LOGS
DESCRIPTION
Double-click to edit
CUSTOM CRITERIA LOGS
VerificationFailed in custom criteria of sltest.testmanager.TestCase.
Test Diagnostic:
Landing Lateral Speed: 0.51722 meters/second.
Framework Diagnostic:
verifyLessThanOrEqual failed.
--> The value must be less than or equal to the maximum value.
Actual Value:
0.517217612807295
Maximum Value (Inclusive):
0.500000000000000

Requirements.xls [Compatibility Mode] - Excel

I65 | 3.25

| Requirement ID | Requirement Name | Level 4 GN&C Requirement | Rationale | Level 4 GN&C Requirement Text | Spec Value | Units |
|---|---|---|---|---|---|---|
| L4-GNC-5.3 | Landing/Touchdown Upper Limit Requirements | | | | | |
| L4-GNC-5.3.1 | Maximum Vertical Velocity at Touchdown | At touchdown, the maximum vertical velocity magnitude shall be no greater than 2.52 meters/second. | Contact Limits | At touchdown, the maximum vertical velocity magnitude shall be no greater than | 2.52 | meters/second |
| L4-GNC-5.3.2 | Maximum Lateral Velocity at Touchdown | At touchdown, the maximum lateral velocity magnitude shall be no greater than 0.52 meters/second. | Contact Limits | At touchdown, the maximum lateral velocity magnitude shall be no greater than | 0.52 | meters/second |
| L4-GNC-5.3.3 | Maximum Pitch Angle at Touchdown | At touchdown, the maximum pitch angle with respect to local vertical shall be no greater than 3.25 degrees. | Tip Over Limits | At touchdown, the maximum pitch angle with respect to local vertical shall be no greater than | 3.25 | |
| L4-GNC-5.3.4 | Maximum Angular Rate at Touchdown | At touchdown, the maximum magnitude of the inertial angular rate shall be no greater than 0.1 degrees/second. | Tip Over Limits | At touchdown, the maximum magnitude of the inertial angular rate shall be no greater than | 0.1 | |
| L4-GNC-5.3.5 | Maximum Position Error | At touchdown, the maximum position error from the nominal landing site shall be no greater than 100 meters. | Landed Payload Limits | At touchdown, the maximum position error from the nominal landing site shall be no greater than | 100 | meters |
| L4-GNC-5.3.6 | Maximum Lateral CM Offset | At touchdown the lateral CM offset shall be no greater than 77 millimeters. | Control Stability Margin | At touchdown the lateral CM offset shall be no greater than | 77 | millimeters |

Highlighting indicates
and is expected to be
executing
updateRequirementsT

For more information,
ImportandMBDInstruc
2.

Sheet1 | Functional | FlightSoftwareQualityChecks | ImportantMBDInstructions

Cell I65 commented by Jamison, Brian R. (MSFC-EV40)[ESSSA]

Editor - D:\Landers_20190214\Ipl\MAIN\MBD\updateRequirementsTesting.m

EDITOR | PUBLISH | VIEW

New | Open | Save | Find Files | Compare | Print | Go To | Find | Insert | Comment | Indent | Breakpoints | Run | Run and Advance | Run Section | Advance | Run and Time

FILE | NAVIGATE | EDIT | BREAKPOINTS | RUN

```
%% Beginning from an Excel spreadsheet, as determined necessary, this script creates or
% updates a Simulink Data Dictionary and the corresponding Simulink Requirements file, and
% loads and reruns the associated Simulink Test files.

clearvars
clc

%% display a pop-up indicating the need to now close open Excel requirements window to avoid the possiblity of errors when clearing highlighting in the .xls requirement file
uifig = figure;  % Create and then hide the UI as it is being constructed.
h = uicontrol('Position',[180 20 200 40],'String','Continue','Callback','uiresume(gcbf)','FontSize',16);
uidirections1 = 'To prevent errors during execution of updateRequirementsTesting.m, ensure ';
uidirections2 = 'the Excel Requirements.xls window is closed, and then select Continue.';
htext  = uicontrol('Style','text','String',[uidirections1 uidirections2],'Position',[40,100,480,300],'FontSize',23);
uiwait(gcf);
close(uifig);

%% Open the MAIN project
open('..\MAIN.prj')   % among other things, sets up needed path(s), including to readDict and editDict
cd('MBD')  % change directory back to MBD directory since previous command went back to MAIN directory

%% Setup filenames and ensure they are valid
xlsFilename = 'Requirements.xls';
slreqxFilename = 'Requirements.slreqx';
slddFilename = 'Dictionary.sldd';
testsFilename = 'Tests.mldatx';
testsResultsFilename = 'TestsResults_20190807_1434.mldatx';
if ~strcmp(xlsFilename(1:end-3),slreqxFilename(1:end-6))
    error('The filenames of the .xls and .slreqx files should match.')  % e.g., relative to importing materials from the .xls for creation of the .slreqx file if necessary
end

%% Create the data dictionary if necessary
if ~exist(slddFilename,'file')  % if the data dictionary does not exist, ...
    Simulink.data.dictionary.create(slddFilename);  % create it
end

%% specify the number of Monte Carlo iterations, and, as necessary, define or update a corresponding data dictionary parameter
NUM_MONTE_CARLO_ITERATIONS_ThisTime = 200;%0;
try  % if the data dictionary has a NUM_MONTE_CARLO_ITERATIONS parameter such that the following line doesn't result in an error, ...
    NUM_MONTE_CARLO_ITERATIONS = readDict(slddFilename,'NUM_MONTE_CARLO_ITERATIONS');  % get it's value (which is expected to be an integer) to see if it needs to be updated
    if (NUM_MONTE_CARLO_ITERATIONS ~= NUM_MONTE_CARLO_ITERATIONS_ThisTime)  % if the NUM_MONTE_CARLO_ITERATIONS parameter does not match the one already in the data dictionary, ...
        editDict(slddFilename,'NUM_MONTE_CARLO_ITERATIONS',NUM_MONTE_CARLO_ITERATIONS_ThisTime);  % update the NUM_MONTE_CARLO_ITERATIONS parameter in the data dictionary
    end
catch  % otherwise, ...
    editDict(slddFilename,'NUM_MONTE_CARLO_ITERATIONS',NUM_MONTE_CARLO_ITERATIONS_ThisTime);  % add the NUM_MONTE_CARLO_ITERATIONS parameter to the data dictionary
end

%% Check the xls file to see if it has been modified, and if so, set the flag to update the data dictionary and the requirements file
xls_file_obj = dir(xlsFilename);  % expected to contain 'name' (filename), 'folder' (folder location and name), 'date' (string), 'isdir' (boolean), and 'datenum' (float) fields
```

- Initialization.mlx
- oldLandedTests.m
- setupCallback.m
- RefineFinalDescentTimeEstimate.m
- compareGLASSResults.m
- MBDReqsImplStatusColorbarImagePath.m
- MBDReqsVerifStatusColorbarImagePath.m
- addMBDReqsStatusImages.m
- datenum.m
- updateRequirementsTesting.m
- loadMCIterationsLandingVelocities.m
- loadMCIterationsLandingAngularVelocities.m
- loadMCIterationsLandingQuaternions.m
- loadMCIterationsLandingPositions.m
- testTouchdownVerticalVelocityResult.m
- testTouchdownLateralSpeedResult.m
- testTouchdownPitchResult.m
- testTouchdownAngularRateResult.m
- testTouchdownDistanceResult.m
- GNC_MBD_RptGen.m

script | Ln 63 | Col 50

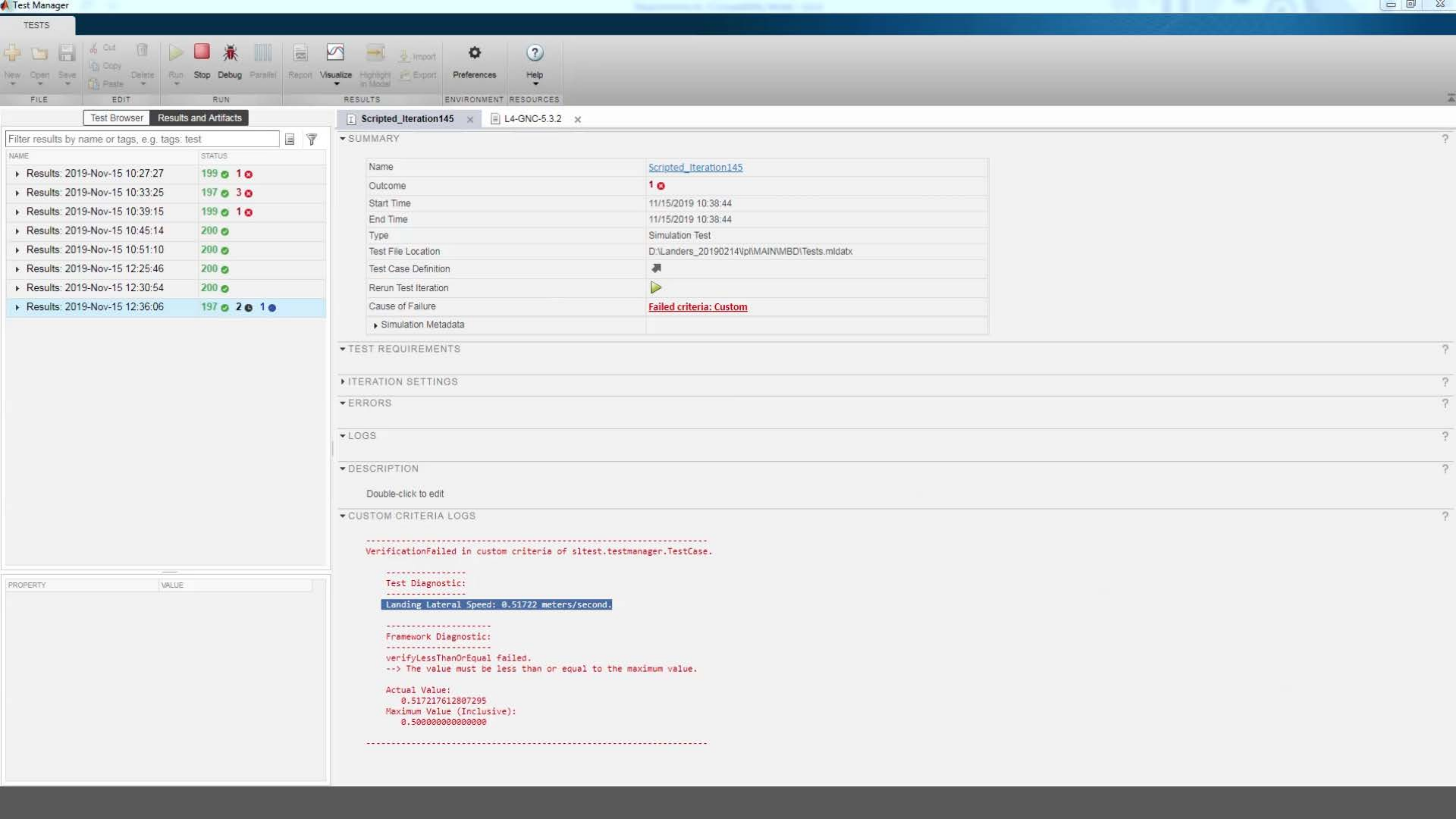
Requirements.xls [Compatibility Mode] - Excel

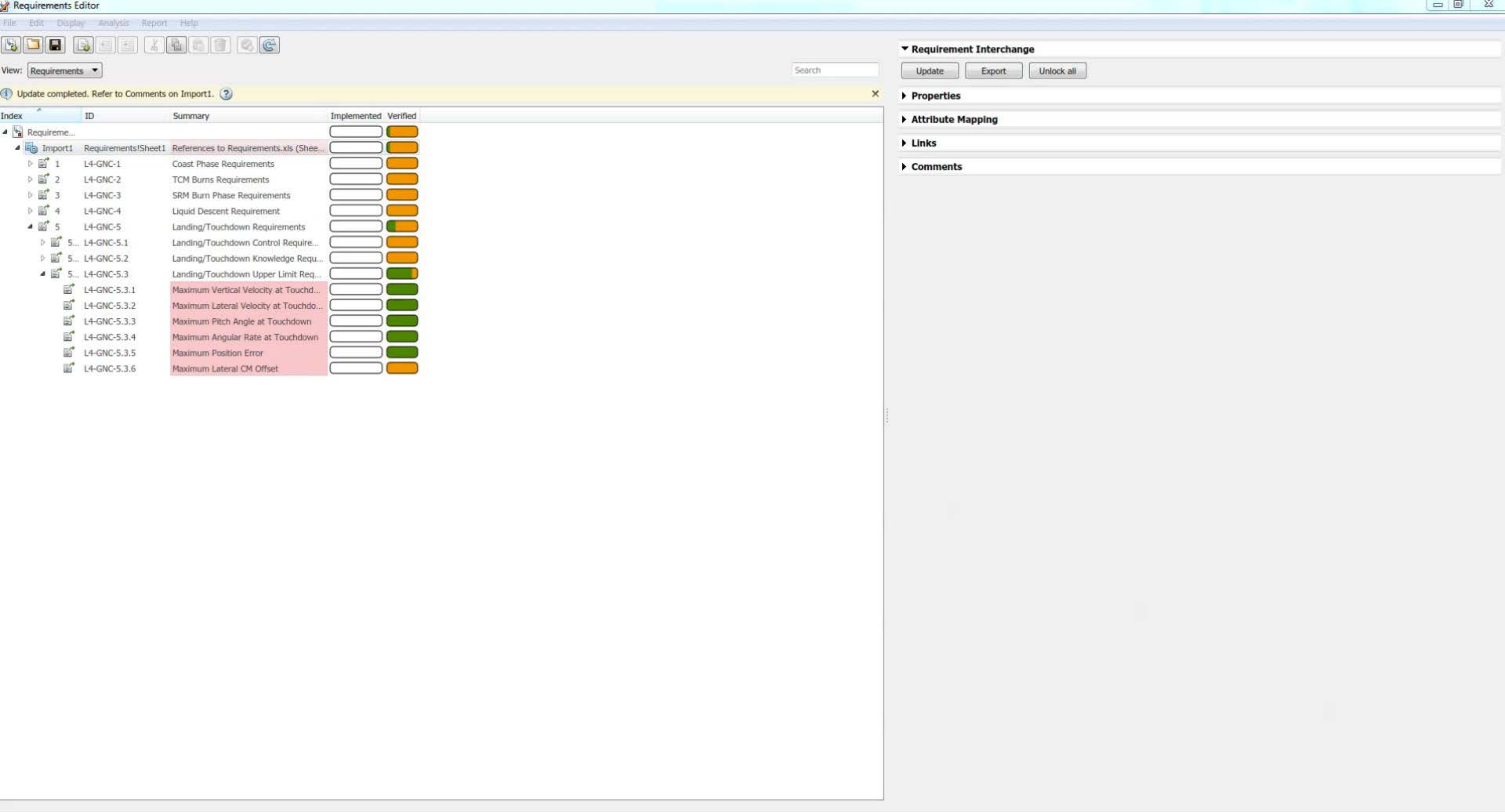
I66 | 0.1

| Requirement ID | Requirement Name | Level 4 GN&C Requirement | Rationale | Level 4 GN&C Requirement Text | Spec Value | Units |
|---|---|---|---|---|---|---|
| L4-GNC-5.3 | Landing/Touchdown Upper Limit Requirements | | | | | |
| L4-GNC-5.3.1 | Maximum Vertical Velocity at Touchdown | At touchdown, the maximum vertical velocity magnitude shall be no greater than 2.52 meters/second. | Contact Limits | At touchdown, the maximum vertical velocity magnitude shall be no greater than | 2.52 | meters/second |
| L4-GNC-5.3.2 | Maximum Lateral Velocity at Touchdown | At touchdown, the maximum lateral velocity magnitude shall be no greater than 0.52 meters/second. | Contact Limits | At touchdown, the maximum lateral velocity magnitude shall be no greater than | 0.52 | meters/second |
| L4-GNC-5.3.3 | Maximum Pitch Angle at Touchdown | At touchdown, the maximum pitch angle with respect to local vertical shall be no greater than 3.25 degrees. | Tip Over Limits | At touchdown, the maximum pitch angle with respect to local vertical shall be no greater than | 3.25 | degrees |
| L4-GNC-5.3.4 | Maximum Angular Rate at Touchdown | At touchdown, the maximum magnitude of the inertial angular rate shall be no greater than 0.1 degrees/second. | Tip Over Limits | At touchdown, the maximum magnitude of the inertial angular rate shall be no greater than | 0.1 | degrees/second |
| L4-GNC-5.3.5 | Maximum Position Error | At touchdown, the maximum position error from the nominal landing site shall be no greater than 100 meters. | Landed Payload Limits | At touchdown, the maximum position error from the nominal landing site shall be no greater than | 100 | meters |
| L4-GNC-5.3.6 | Maximum Lateral CM Offset | At touchdown the lateral CM offset shall be no greater than 77 millimeters. | Control Stability Margin | At touchdown the lateral CM offset shall be no greater than | 77 | millimeters |

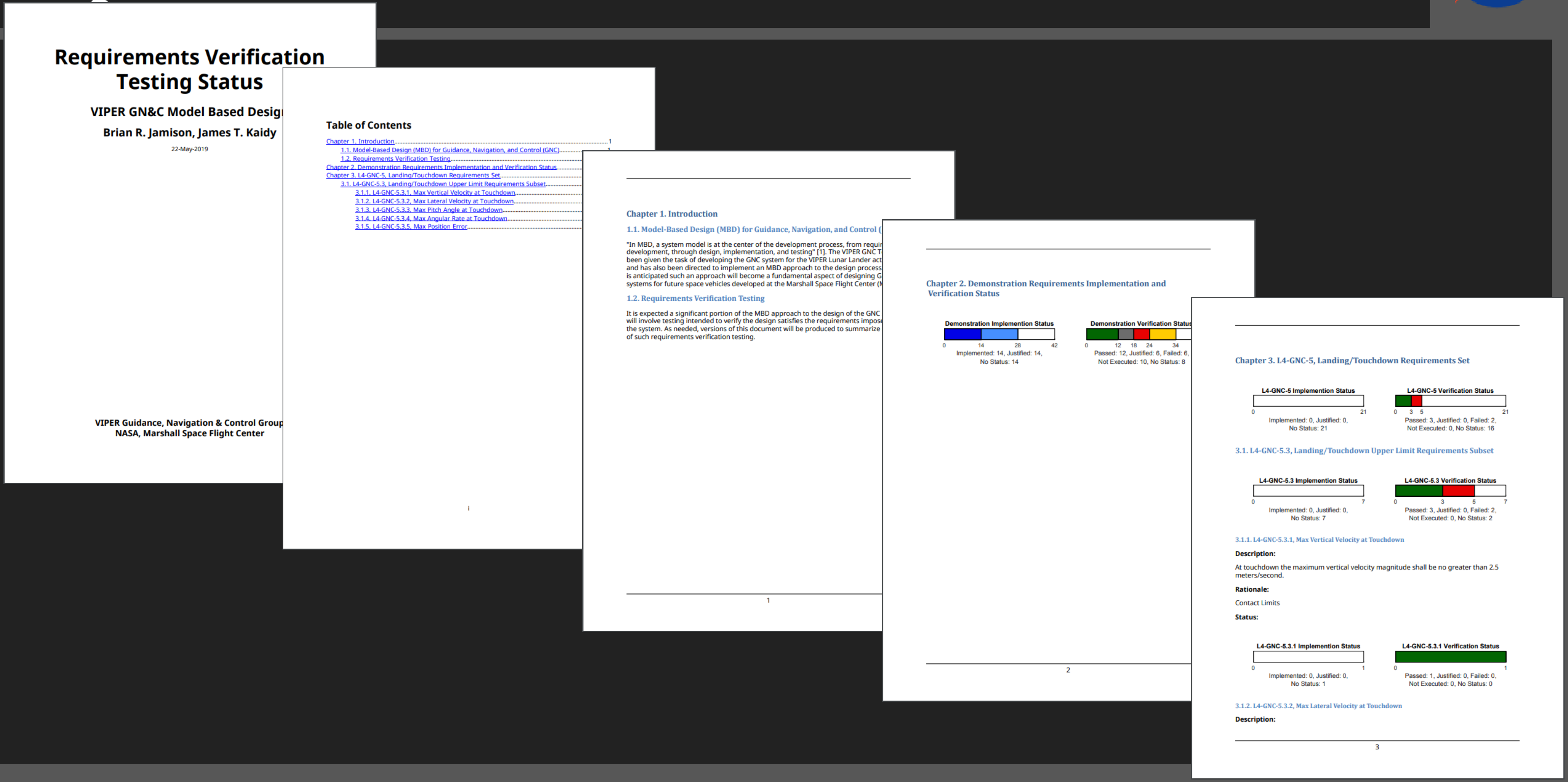
Sheet1 | Functional | FlightSoftwareQualityChecks | ImportantMBDInstructions

READY


Test Manager
TESTS
New
Open
Save
Cut
Copy
Paste
Delete
Run
Stop
Debug
Parallel
Report
Visualize
Highlight in Model
Import
Export
Preferences
Help
FILE
EDIT
RUN
RESULTS
ENVIRONMENT
RESOURCES
Test Browser
Results and Artifacts
Filter results by name or tags, e.g. tags: test
NAME
STATUS
Results: 2019-Nov-15 10:27:27 199 1
Results: 2019-Nov-15 10:33:25 197 3
Results: 2019-Nov-15 10:39:15 199 1
Results: 2019-Nov-15 10:45:14 200
Results: 2019-Nov-15 10:51:10 200
Results: 2019-Nov-15 12:25:46 200
Results: 2019-Nov-15 12:30:54 200
Results: 2019-Nov-15 12:36:06 197 2 1
PROPERTY
VALUE
Scripted_Iteration145
L4-GNC-5.3.2
SUMMARY
Name
Scripted_Iteration145
Outcome
1
Start Time
11/15/2019 10:38:44
End Time
11/15/2019 10:38:44
Type
Simulation Test
Test File Location
D:\Landers_20190214\lpl\MAIN\MBD\Tests.mldatx
Test Case Definition
Rerun Test Iteration
Cause of Failure
Failed criteria: Custom
Simulation Metadata
TEST REQUIREMENTS
ITERATION SETTINGS
ERRORS
LOGS
DESCRIPTION
Double-click to edit
CUSTOM CRITERIA LOGS
VerificationFailed in custom criteria of sltest.testmanager.TestCase.
Test Diagnostic:
Landing Lateral Speed: 0.51722 meters/second.
Framework Diagnostic:
verifyLessThanOrEqual failed.
--> The value must be less than or equal to the maximum value.
Actual Value:
0.517217612807295
Maximum Value (Inclusive):
0.500000000000000

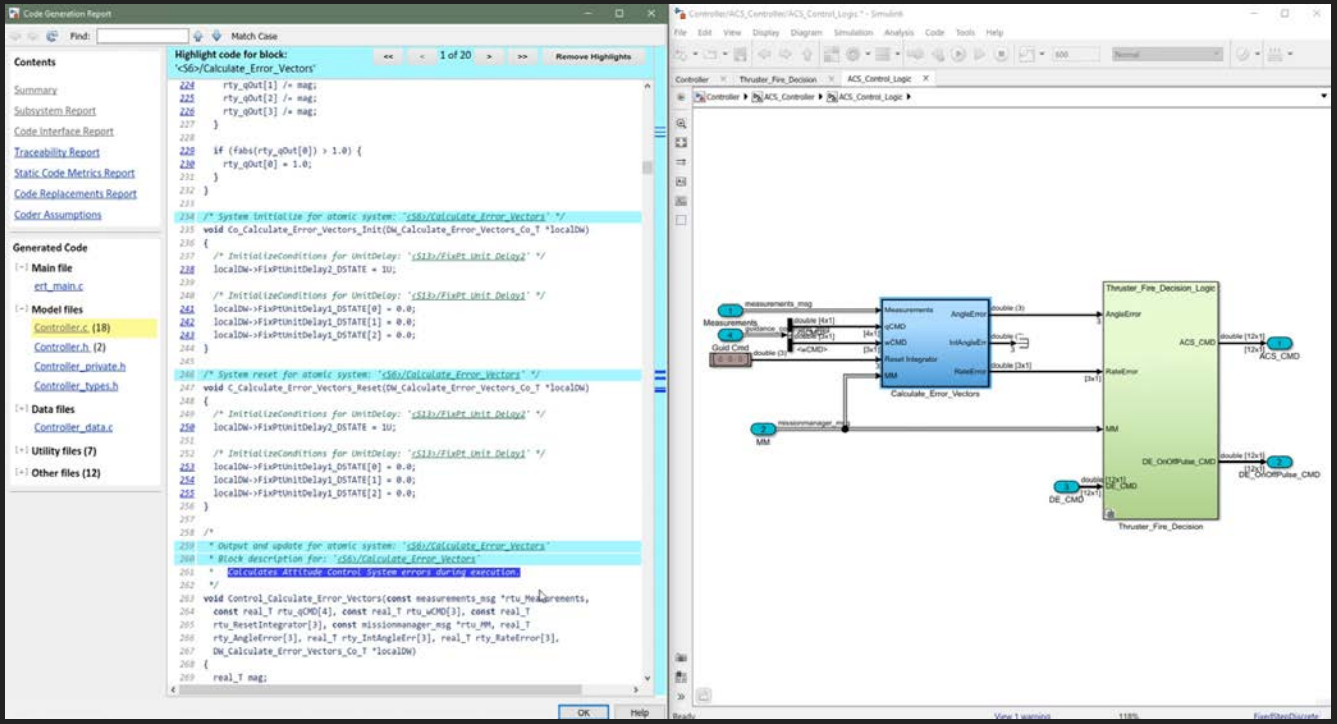

Requirements Editor
File Edit Display Analysis Report Help
View: Requirements
Search
Update completed. Refer to Comments on Import1.
Index ID Summary Implemented Verified
Requireme...
Import1 Requirements!Sheet1 References to Requirements.xls (Shee...
1 L4-GNC-1 Coast Phase Requirements
2 L4-GNC-2 TCM Burns Requirements
3 L4-GNC-3 SRM Burn Phase Requirements
4 L4-GNC-4 Liquid Descent Requirement
5 L4-GNC-5 Landing/Touchdown Requirements
5... L4-GNC-5.1 Landing/Touchdown Control Require...
5... L4-GNC-5.2 Landing/Touchdown Knowledge Requ...
5... L4-GNC-5.3 Landing/Touchdown Upper Limit Req...
L4-GNC-5.3.1 Maximum Vertical Velocity at Touchd...
L4-GNC-5.3.2 Maximum Lateral Velocity at Touchdo...
L4-GNC-5.3.3 Maximum Pitch Angle at Touchdown
L4-GNC-5.3.4 Maximum Angular Rate at Touchdown
L4-GNC-5.3.5 Maximum Position Error
L4-GNC-5.3.6 Maximum Lateral CM Offset
Requirement Interchange
Update Export Unlock all
Properties
Attribute Mapping
Links
Comments

# Report Generation



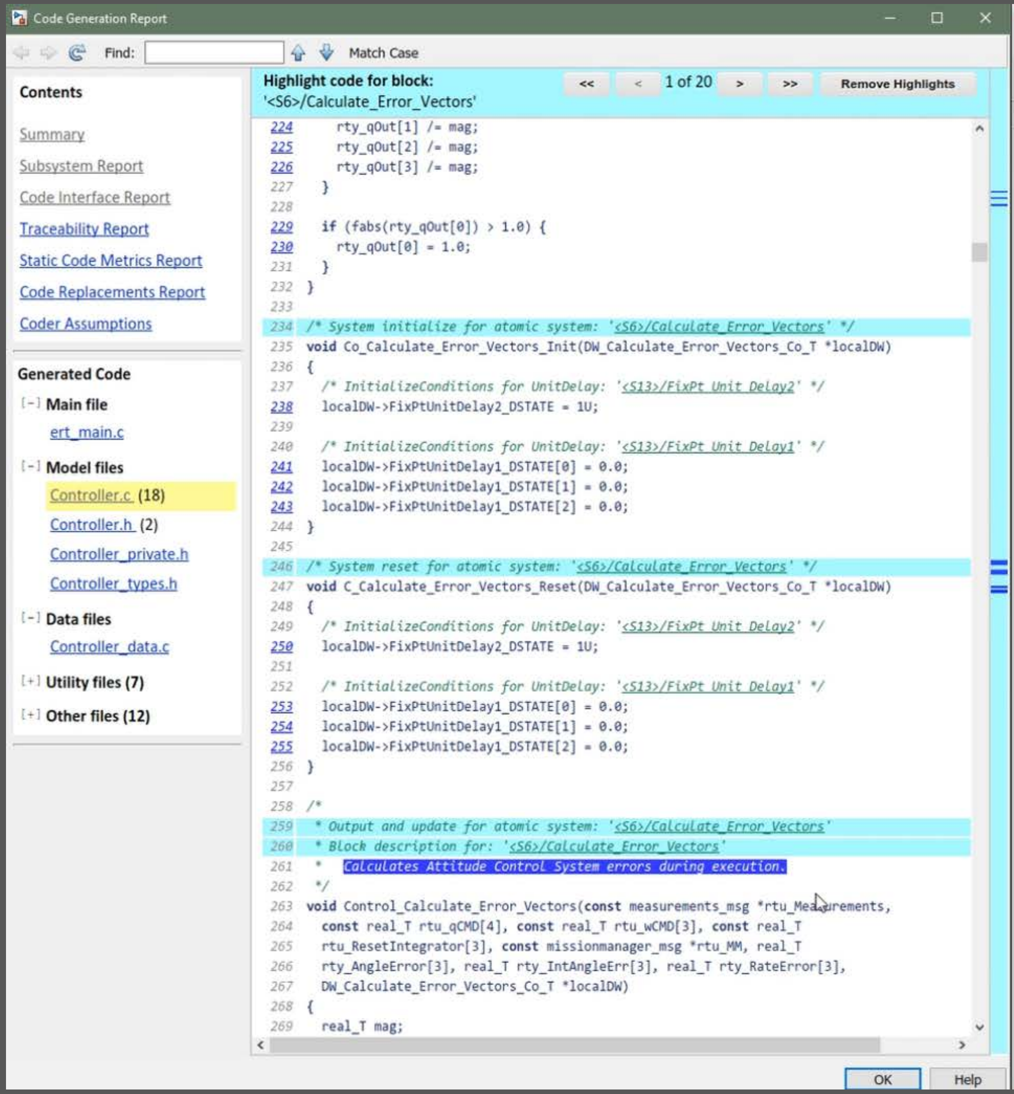
# Simulink Auto-Coding for MBD

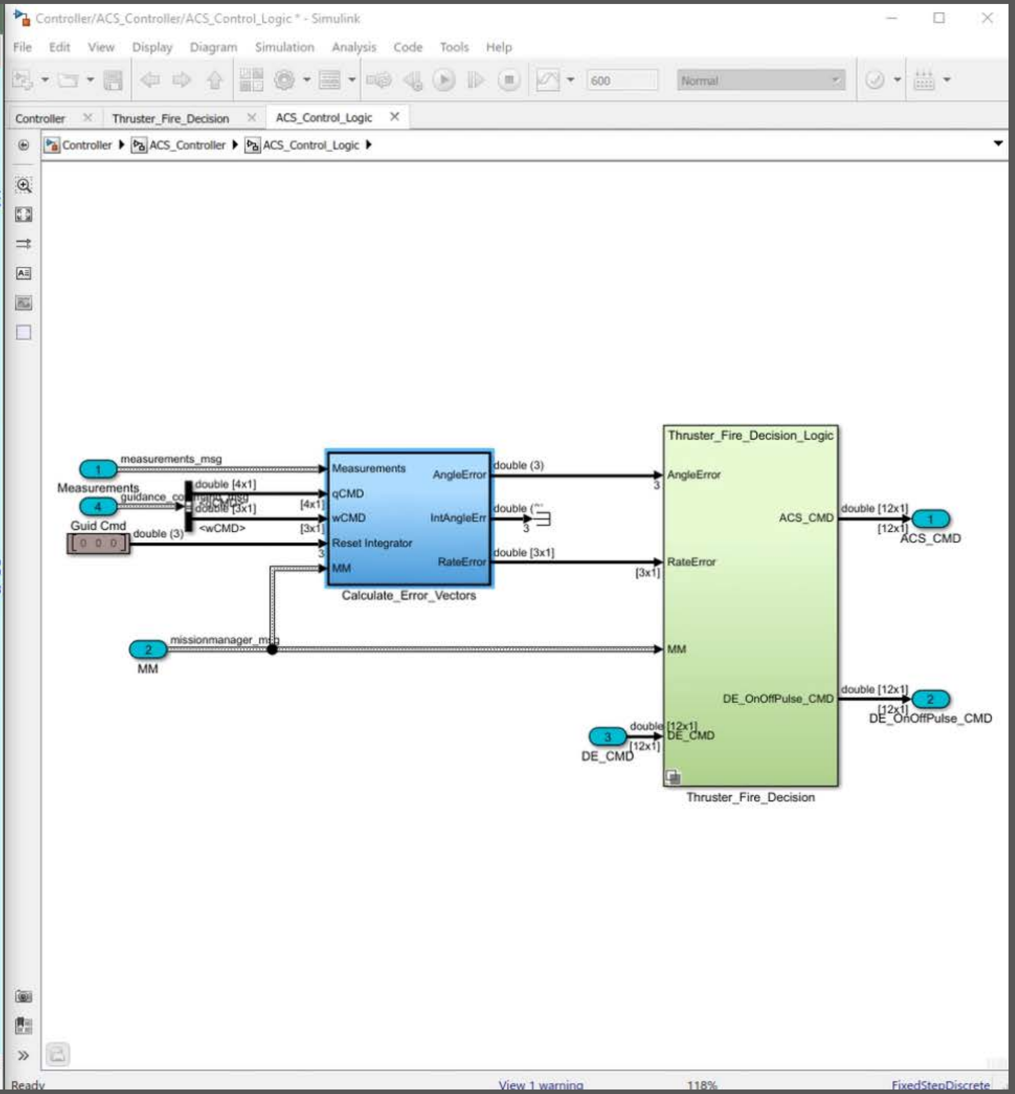
- Auto-coding enforces coding standards automatically and consistently
- Utilizes Simulink Coder & Embedded Coder to generate C/C++ code directly from Simulink models
  - One interface for plant modeling, algorithm development, & code deployment
  - Access to MathWorks control toolboxes and other analysis tools
- Used by:
  - NASA
    - GLASS
    - NEA-Scout
    - Orion GN&C
    - Goddard programs – PACE, JEDI
  - APL
  - Lockheed Martin
  - Automotive Industry



# Simulink Auto-Coding for MBD – Example

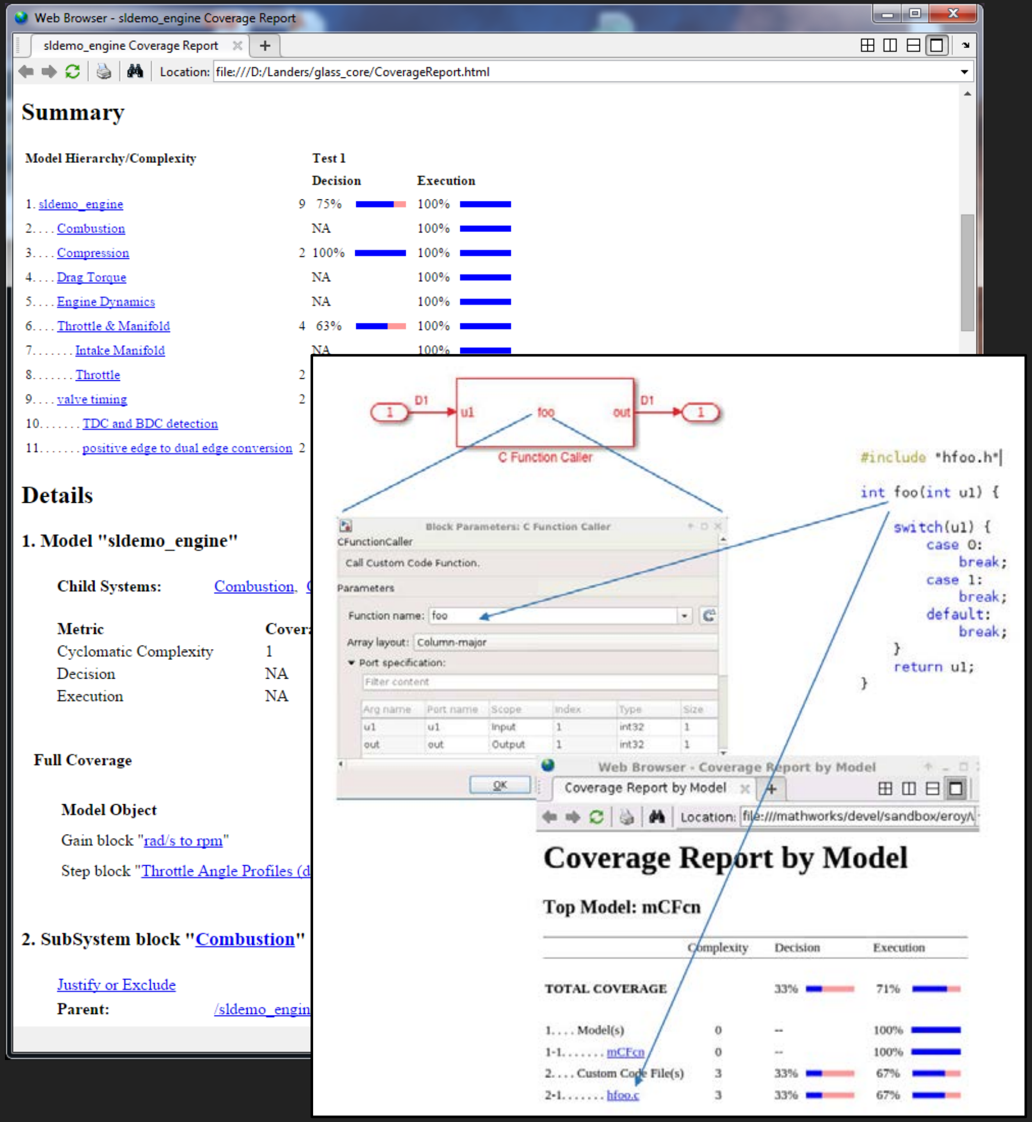
- Code & model linking
- Code comments
- Improved readability



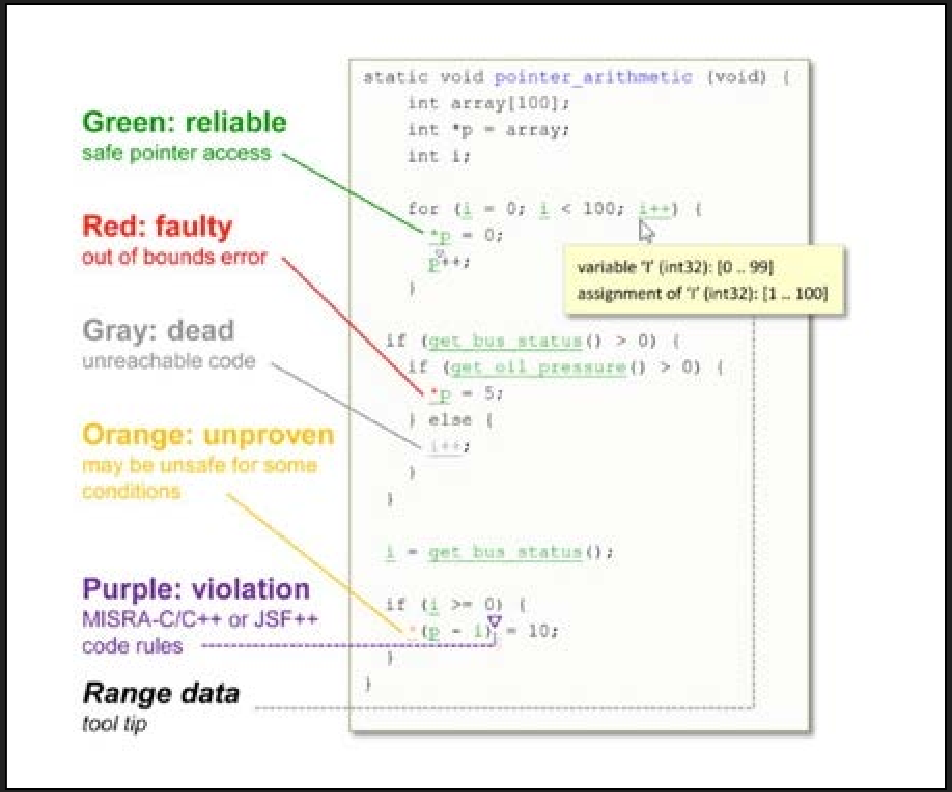
# Coverage Testing

- Ensures model/code are fully exercised
  - Used during unit testing
- Checked in the Simulink model and generated code
- Report links un-executed portions of the model and code
  - Simplifies repair/justification
- Report provides metric about work remaining



# MBD Static Code Checking

- Static Code Check – Polyspace
  - Integrated with Simulink for traceability from the source code back to the original model
  - Looks for concurrency issues, security vulnerabilities, & runtime errors, including arithmetic overflow, buffer overrun, division by zero, out-of-bounds array access, and others
    - Ariane 5 failed (4 June 1996) due to overflow
  - Enforces coding guidelines
    - MISRA C, MISRA C++, JSF++, CERT® C, CERT® C++, etc.
  - Static code checks are typically required for FSW



# MBD Static Code Checking

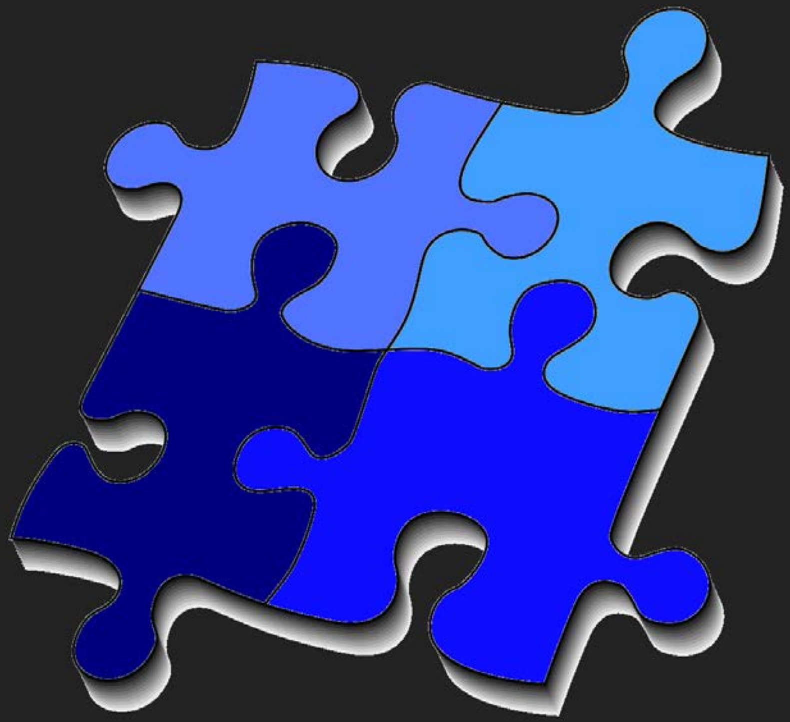
- Static Code Check – Polyspace
  - Integrated with Simulink for traceability from the source code back to the original model
  - Looks for concurrency issues, security vulnerabilities, & runtime errors, including arithmetic overflow, buffer overrun, division by zero, out-of-bounds array access, and others
    - Ariane 5 failed (4 June 1996) due to overflow
  - Enforces coding guidelines
    - MISRA C, MISRA C++, JSF++, CERT® C, CERT® C++, etc.
  - Static code checks are typically required for FSW

# Example: Traditional vs. MBD

- Traditional (manual, labor intensive, error prone):
  - Near Earth Asteroid Scout (NEA-Scout) Project had ~33 GNC L4 requirements, verified through 4 major analysis packages
  - When changes were introduced, analyses and documentation were (manually) repeated to assess impacts to requirements
  - NEA-Scout relied on the FSW integrator (JPL) to run static code checks, involving manual trace-backs from the code to the model
- MBD (largely automated):
  - L4 requirements are checked within the model, and impacts to the design changes can be assessed with every execution of the model
  - When change is introduced, automated testing confirms requirements are satisfied
  - Static code checks are done by code developer prior to delivery to FSW integrator using Polyspace, which seamlessly links code violations with the source code and the model

# A Synergistic Environment

- Automated processes
  - Testing, Auto-coding, report generation
- Testing
  - Tools such as Simulink Test are capable of exercising both the Simulink model and the generated code for requirements validation
- Auto-coding
- Reporting is part of the process
- All pieces work together to produce a highly automated, disciplined process



# Summary

- A preliminary process has been established to generate quality GNC code using MBD tools from MathWorks (and Microsoft Excel)
- Initial discussions with the NASA MSFC Flight Software group have taken place to ensure processes work together
- Goals:
  - Increase development speed
  - Reduce manual tasks (e.g., testing, hand coding, report writing)
  - Traceability from requirements to model/code verification
  - Consistent quality

# Image Sources

- NASA logo: https://commons.wikimedia.org/wiki/File:NASA_logo.svg
- LADEE image: https://www.nasa.gov/mission_pages/ladee/multimedia
- Dictionary image: https://commons.wikimedia.org/wiki/File:Gnome-dictionary.svg
- Simulink Requirements Editor: https://www.mathworks.com/products/simulink-requirements/features.html#author-and-organize-requirements-in-simulink
- Puzzle pieces image, modified from: https://commons.wikimedia.org/wiki/Jigsaw_puzzle#/media/File:Jigsaw.svg