Guidelines for deployment of MathWorks® R2010a toolset within a DO-178B-compliant process

UK MathWorks Aerospace & Defence Industry Working Group

Guidelines for deployment of MathWorks® R2010a toolset within a DO-178B-compliant process:
by UK MathWorks Aerospace & Defence Industry Working Group

Table of Contents

List of Tables

* *UK Aerospace and Defence industry guidelines for the application of MathWorks® tools in the development of high integrity systems using model-based design.*

Publication Notice

First published January 2012

By MathWorks, Inc.

Matrix House

Cambridge Business Park

Cambridge,

CB4 0HH

UK MathWorks Aerospace and Defence Industry Working Group:

MathWorks UK invited selected aerospace and defence companies to join a working group entitled "Graphical modelling and code generation for high-integrity systems". The companies invited all have or are developing software development processes that take advantage of Simulink® and its code generation capability. The objective of the working group will be to explore how this technology can best be used in the context of developing high-integrity systems.

This guidance is published by MathWorks, Inc. on behalf of the UK MathWorks Aerospace and Defence Industry Working Group. Neither the UK MathWorks Aerospace and Defence Industry Working Group nor MathWorks, Inc., accept any liability for the use or application of these guidelines.

Document source: AeroDef_Working_Group_guidance_matrix.xls Revision: 62

Document generation script: AeroDefWorkingGroupReport.rpt Revision: 65

Acknowledgements

The UK MathWorks Aerospace and Defence Industry Working Group on “Graphical modelling and code generation in the development of high-integrity systems” would like to thank the following organisations for their support and contribution in developing these guidelines

Airbus Operations Ltd

BAE Systems plc

MBDA UK Ltd

Selex Galileo Ltd

...and contributors from five other UK companies.

Document Scope

This document provides guidance on the application of Model-Based Design to the development of high integrity systems and software. This guidance produced by the Working Group is based on the capability of MathWorks tools at release R2010a. Users of this guidance should note:

1. Model-Based Design and design tools can help projects meet the requirements of DO-178B. As with all guidance, the recommendations made are unlikely to be sufficient on their own to address any requirement of DO-178B and in some cases may not be necessary to meet those requirements. Following this guidance can help a project satisfy its certification requirements but is not a replacement for formal review of system and software development processes.

2. The document provides a rationale for an activity's contribution towards an objective. The extent of objective coverage is described in Table 1 (DO-178B References with guidance from this document).

3. Model-Based Design techniques and design tool capabilities are continuously improving and, therefore, additional information on current capabilities should be obtained to support any assessment or evaluation of tools for a specific project.

4. MathWorks can provide additional information based on the capabilities of any specific version of the tools on request.

Document Revision. This document will be revised periodically. This will depend on the timing of updates to DO-178, MathWorks tools and feedback from the application of this guidance. Comments and suggestions for improvement on this guidance can be made to the Working Group through the MathWorks by email to ukaerodefstandards@mathworks.com.

Activities. Activity descriptions are based on industrial experiences from high-integrity UK projects working to DO-178B, DEFSTAN 00-56 and other safety-related standards. The guidance is intended to supplement the Model-Based Design Workflow for DO-178B information provided in MathWorks documentation (http://www.mathworks.com/help/toolbox/qualkitdo/do_workflow/bsevkdt-1.html) by describing why an activity can contribute towards a DO-178B objective.

The activities are grouped by DO-178B reference with the primary focus being on those relating to the modelling of High Level Requirements (HLR) and Low Level Requirements (LLR). The relevance of the activities to the software development process are based on the assumption that these requirements models are described primarily using MathWorks tools (Simulink®, Stateflow® and Embedded MATLAB®). Where possible, the activities are not dependent on a specific release of MATLAB®. However, not all of the products referred to in the activities will be available in some older releases of the product.

All activities described in this document can help satisfy DO-178B objectives more efficiently. The guidance assumes a project will assess whether objectives are satisfied using formal review. Many of the verification activities described in this document are supported by the MathWorks DO Qualification Kit. When using a qualified verification tool, the pass/fail indication can be used directly in the review without further inspection of supporting detail. This further increases review efficiency. Further information on the qualification kits and extent of DO-178B coverage is available and should be consulted at: http://www.mathworks.com/products/do-178/.

Each activity is divided into two sections:

- Rationale: A textural description of the activity.

- Review Artefacts: A checklist of sources for the review process.

Activity Numbering: Each activity is identified with a unique activity number. This number reflects the activity's position in the database and should not be used to indicate a suggested sequence of activities in the software development process. The activity numbering is used in the document for cross referencing purposes.

MathWorks Tools

Detailed documentation for the latest release of the MathWorks products referred to in this text can be found at http://www.mathworks.com/access/helpdesk/help/helpdesk.html. The key products that are used in the verification and validation of Simulink models are as follows:

Simulink® Verification and Validation™. - The umbrella product which includes:

- The Requirements Management Interface (RMI) that links Simulink® and Stateflow® objects to locations in requirements documents, providing fast navigation between the two. Reports documenting which objects link to which documents are created automatically.
- Model verification blocks that monitor model signals and characteristics and check that they remain within specified bounds during simulation.
- Model coverage that helps you validate your model tests by measuring how thoroughly the model objects are tested. Model coverage is a measure of how thoroughly a test case tests a model and the percentage of pathways that a test case exercise.
- The Model Advisor which allows the programmatic checking of Simulink models and sub-systems for conformance to modelling standards and guidelines. A number of standard checks are included in the product and it is extensible to include user/company specific checks and configurations.

Simulink Design Verifier. uses formal analysis methods to:

- Automatically generate test cases to achieve model coverage (e.g. MCDC) and user defined objectives.
- Verify user defined properties and provide counter examples for violations where these exist.
- Document test cases and objectives coverage through automatic report generation

Real-Time Workshop® Embedded Coder™. can be used to automatically generate C code from Simulink and Stateflow models.

Polyspace®. can be used for code-based verification to prove the absence of overflow, divide by zero, out-of-bounds array access, and other run-time errors in source code. It does this without requiring program execution, code instrumentation, or test cases, using abstract interpretation techniques to verify code. Polyspace can be used to verify handwritten code, generated code, or a combination of the two.

Simulink®. provides a number of features to assist in the management of models created and updated using different versions of MATLAB. Users are encouraged to visit the Simulink product documentation on setting Simulink Preferences e.g. for model File change notifications, model loading and saving options, and save_system for help and guidance on these features.

DO-178B References and Workflow Activities

The following table summarises the DO-178B references, from Tables A-3 to A-6, for which guidance is offered by this document. It should be noted that there are some references for which no guidance is offered by this document. This may be for one of a number of reasons, including the Working Group has yet to consider them, or they have been reviewed and are not currently addressed by MathWorks tools as of release R2010a.

Table 1. DO-178B References (Tables A-3 to A-6) with guidance from this document

DO-178B Table Ref	Table A-3	Table A-4	Table A-5	Table A-6
1	6.3.1a	6.3.2a	x	6.4.3
2	6.3.1b	6.3.2b	x	6.4.3
3	6.3.1c	6.3.2c	x	6.4.3
4	6.3.1d	6.3.2d	6.3.4d	6.4.3
5	6.3.1e	6.3.2e	6.3.4e	6.4.3
6	6.3.1f	6.3.2f	x	n/a
7	6.3.1g	6.3.2g	x	n/a
8	n/a	x	n/a	n/a
9	n/a	6.3.3b	n/a	n/a
10	n/a	x	n/a	n/a
11	n/a	x	n/a	n/a
12	n/a	6.3.3e	n/a	n/a
13	n/a	x	n/a	n/a

'n/a' indicates the Table Ref does not exist in that particular table. 'x' indicates the Table Ref exists but no guidance is offered by this document.

6.3.1a - Software high-level requirements comply with system requirements

DO-178B Objective: "The objective is to ensure that the system functions to be performed by the software are defined, that the functional, performance, and safety-related requirements of the system are satisfied by the software high level requirements, and the derived requirements and the reason for their existence are correctly defined. " [1]

Use simulation to demonstrate a requirement is satisfied - Activity 5

Rationale:

The software high level requirements may be developed as a Simulink model. The visual representation of the model can be the HLR document itself or a separate textual document that can be developed at the same time. An HLR model expressed in Simulink is executable and deterministic. Simulation of an HLR model is considered to be a form of analysis. This means test cases, expressed as time series of input values, can be applied to the model and the model's output can be checked against expected behaviour. Simulation can be open or closed loop. An open loop simulation comprises of input test cases, the HLR model and record of the expected output. A closed loop simulation consists of input test cases, an HLR model, a "plant" or environment model and a record of the expected output. The input test cases and expected output will be traceable to the system requirement they are designed to demonstrate.

Open loop simulation is well suited to the verification of functional requirements. Closed loop simulation is well suited to the verification of performance requirements. A review is likely to consider the input test cases, expected and recorded output and the traceability to system requirements.

Review Artefacts:

- Requirements document (s)
- HLR model (& model review output)
- Simulation input cases
- Simulation output
- Traceability from system requirements to HLR model, test cases and results

Use Simulink Design Verifier Property Proving to verify requirements properties (Formal methods analysis of model) - Activity 4

Rationale:

Formal methods can be applied to the HLR model to verify that the HLR model satisfies a system requirement. The system requirement is expressed independently as a model property, as contained in a "verification subsystem" in Simulink. This expected behaviour will be developed independently of the HLR model itself, for example by different engineers. The verification subsystem should also trace to the system requirement that it verifies. Simulink Design Verifier can then be used to analyse the model and verification subsystem together and verify that no model inputs exist that violate the property expressed in the verification subsystem. Simulink Design Verifier uses automated mathematical reasoning to verify properties and is built around a third party proving engine from Prover® Technology AB. Should Simulink Design Verifier identify a case that violates the expressed property, a counterexample test case that causes the violation will be generated.

Analysis using formal methods provides a means of demonstrating HLR model compliance with system requirements that is independent of time series simulation (see Activity 5). A review will consider the proof report generated by

Simulink Design Verifier, the construction of the verification subsystem that expresses the proof and any additional constraints or assumptions that have been applied to the system. Searching for counterexamples using formal methods is an example of negative testing ("does any case exist that violates this property") and complements the positive testing approach ("does this input give the expected output") that engineers use when designing time series tests (see Activity 5). DO-178B and DEFSTAN 00-56 both recognise this complementary contribution. Property proving is well suited to the verification of functional and safety requirements.

The possible outputs from Simulink Design Verifier are:

- Falsified - A test case has been found to violate the proof objective. This means the highlighted part of the model traces to a contradictory requirement or the implementation in the HLR model contains an error.
- Proven Valid - No test case exists. The proof objectives are valid and no counter examples exist.
- Undecided - Simulink Design Verifier was not able to complete its analysis within the time limit.

Certain modelling constructs can restrict analysis by Simulink Design Verifier. In order to conduct its analysis, Simulink Design Verifier may need to make some approximations or block substitutions such as to replace floating point numbers with rational numbers or non-linear arithmetic (e.g. look-up tables) with linear approximations. The user should fully understand the impact of these approximations and verify the results are unaffected when test cases are reapplied to the original model. Please see Simulink Design Verifier product documentation for further detail on approximations and substitutions:
http://www.mathworks.com/access/helpdesk/help/toolbox/sldv/sldv_product_page.html.

Review Artefacts:

- Requirements document(s)
- Simulink Design Verifier Proof report
- HLR model (& model review output)
- HLR verification subsystem

6.3.1b - High-level requirements are accurate and consistent

DO-178B Objective: "The objective is to ensure that each high level requirement is accurate, unambiguous and sufficiently detailed and that the requirements do not conflict with each other. " [1]

Use simulation to demonstrate a high-level requirement is satisfied - Activity 7

Rationale:

As with Activity 5, a Simulink model may be developed to represent the high level software requirements (the HLR model). Test harnesses, traceable to system requirements, may also be developed. The simulation (analysis) results also contribute towards the accuracy and consistency objective and can be assessed in review. The contribution of open and closed loop simulation output has already been described in Activity 5.

For a Simulink model to simulate, the Simulink language semantics require that it is specified in a fully deterministic way. This contributes towards the accuracy and consistency objective since the language will force the modeller to resolve conflicts before simulation is permitted.

Additionally structural and signal range coverage of the HLR model can be recorded during test case execution. This aids the assessment of requirements consistency at review. For example, the root cause of incomplete structural coverage can be contradictory requirements.

Review Artefacts:

- Requirements document (s)
- HLR model (& model review output)
- Simulation input cases
- Simulation output
- Traceability from system requirements to HLR model, test cases and results
- Model Coverage report

Run DO-178B Model Advisor checks - Activity 8

Rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Model Advisor report
- HLR model

6.3.1c - High-level requirements are compatible with target computer

DO-178B Objective: "The objective is to ensure that no conflict exists between the high level requirements and the hardware/software features of the target computer, especially, system response times and input/output hardware. " [1]

Run DO-178B Model Advisor checks - Activity 9

Rationale:

Same rationale as Activity 8.

If the project is also generating code using Real-Time Workshop Embedded Coder, the Model Advisor checks can be used to verify the code generator settings that relate to target CPU architecture. For example, the Simulink language replicates target computational precision (e.g. data type, size and format) for simulation but also supports features that may not be compatible with the target computer. The Model Advisor Checks identify whether such features are enabled in a model.

Activity 8 rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Model Advisor report
- HLR model

6.3.1d - High level requirements are verifiable

DO-178B Objective: "The objective is to ensure that each high-level requirement can be verified. " [1]

Measure Coverage during HLR model simulation - Activity 14

Rationale:

The high-level requirements are verifiable if all conditions and decisions are reached during simulation. This is measured directly by the Model Coverage tool. In projects working directly to DO-178B it is expected that a complete set of tests will be developed.

Where incomplete coverage is found, some of the possible root causes can be contradictory requirements, errors in the model, incomplete models, redundant parts of the model, errors in the test cases or incomplete tests (other sources may

exist). The traceability reviews assist in identifying incomplete models, redundant parts of the models and incomplete test cases.

Test generation using Simulink Design Verifier will return a test case for all reachable coverage objectives and is best applied to extend an existing suite of requirements based tests (See Activity 4 for further detail on the use of Simulink Design Verifier).

Review Artefacts:

- Model Coverage report
- HLR model

6.3.1e - High level requirements conform to standards

DO-178B Objective: "The objective is to ensure that the Software Requirements Standards were followed during the software requirements process and that deviations from the standards are justified. " [1]

Run DO-178B Model Advisor checks - Activity 11

Rationale:

Same rationale as Activity 8.

The Model Advisor supports modelling standards for DO-178B, IEC 61508, and the MathWorks Automotive Advisory Board (MAAB). It may also be extended to support other projects specific standards. The Model Advisor checks assess conformance of HLR models to these standards.

Activity 8 rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Model Advisor report
- Project specific modelling standard
- HLR model

6.3.1f - High level requirements are traceable to system requirements

DO-178B Objective: "The objective is to ensure that the functional, performance, and safety-related requirements of the system that are allocated to software were developed into the software high-level requirements. " [1]

Run Model Advisor Requirements consistency check - Activity 13

Rationale:

Same rationale as Activity 8, specifically:

In the event that the Requirements Management Interface is being used to link requirements documents to models, the DO-178B checks verify that the traceability links between HLR models and requirements documents are valid. The report identifies:

- Requirement links with missing documents
- Requirement links that specify invalid locations within documents
- Selection-based links having description that do not match their requirements document text

Activity 8 rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Requirements document(s)
- Model Advisor Requirements Consistency report
- HLR model

Use Requirements Management Interface to highlight model components without linkage to requirements - Activity 2

Rationale:

In the event that the Requirements Management Interface is being used to link requirements documents to models, it can be used to identify model components without links to requirements which may indicate one of the following:

- Incomplete requirements
- HLR Model is not consistent with requirements
- Requirements links incomplete

Use of this tool prior to manual review should improve review efficiency by early detection of missing traceability.

Review Artefacts:

- Requirements document(s)
- Model Advisor Requirements report
- HLR model

Use Requirements Management Interface to generate requirements report - Activity 3

Rationale:

In the event that the Requirements Management Interface is being used to link requirements documents to models, the use of the Model Advisor Requirements report as an input to the manual review process may improve review efficiency. The requirements report includes hyperlinks to both the model and the corresponding requirements document. This can be used to provide fast navigation between the requirements documents and the HLR model.

Review Artefacts:

- Requirements document(s)
- Model Advisor Requirements report
- HLR model

6.3.1g - Algorithms are accurate

DO-178B Objective: "The objective is to ensure the accuracy and behaviour of the proposed algorithms, especially in the area of discontinuities. " [1]

Run DO-178B Model Advisor checks - Activity 15

Rationale:

Same rationale as Activity 8, specifically:

The DO-178B checks statically verify the data types used within the model. This Activity will typically be supplemented by simulation based tests, as described in Activity 7.

Activity 8 rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend

time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Model Advisor report
- HLR model

6.3.2a - Low-level requirements comply with high-level requirements

DO-178B Objective: "The objective is to ensure that the software low-level requirements satisfy the software high-level requirements and that derived requirements and the design basis for their existence are correctly defined. " [1]

Model simulation with HLR test case re-use - Activity 19

Rationale:

When both the high level and low level requirements are expressed as models, the test cases that demonstrated HLR model compliance with system requirements can be reused against the LLR model. The HLR model will be designed to express the system requirements. The LLR model will be a representation of the HLR model augmented with all the necessary software design and architecture details (e.g. integer data types not float, functional partitioning, etc). Using these same cases demonstrates that the LLR model satisfies the system requirements and therefore exhibit the same behaviour as the HLR models.

Engineers tend to write positive tests, i.e. demonstrating that something will happen, not the negative cases which are more useful but harder to design. This type of testing (or verifying with Simulink Design Verifier Property Proving) should also be considered for these LLR models.

Review Artefacts:

- Activity 5 test harnesses
- LLR model simulation output
- LLR Coverage report

Use Simulink Design Verifier Property Proving to verify requirements properties - Activity 20

Rationale:

Mathematical analysis of the LLR model using Simulink Design Verifier can be used to verify that no test case exists that violate requirement proof properties. If counterexamples exist for legitimate reasons (e.g. Input values fall outside of achievable range) the test case will be an input to the review process.

If system requirements have already been verified against HLR models (see Activity 4) then the properties can be reused as proofs on the LLR models. The HLR and the LLR models must satisfy the same system requirements, so the proof result should be the same in both cases.

Review Artefacts:

- Requirements document(s)
- Simulink Design Verifier Proof report
- LLR model (& model review output)
- LLR verification subsystem

Use of XML compare tool to compare HLR and LLR models - Activity 21

Rationale:

Where HLR and LLR models are both in Simulink, the XML comparison report is a very useful contribution to the review process.

Review Artefacts:

- HLR model
- LLR model
- XML comparison report

6.3.2b - Low-level requirements are accurate and consistent

DO-178B Objective: "The objective is to ensure that each low-level requirement is accurate and unambiguous and that the low-level requirements do not conflict with each other. " [1]

Use simulation to demonstrate a requirement is satisfied - Activity 54

Rationale:

Same rationale as Activity 7, but for LLR.

Activity 7 rationale:

As with Activity 5, a Simulink model may be developed to represent the high level software requirements (the HLR model). Test harnesses, traceable to system requirements, may also be developed. The simulation (analysis) results also contribute towards the accuracy and consistency objective and can be assessed in review. The contribution of open and closed loop simulation output has already been described in Activity 5.

For a Simulink model to simulate, the Simulink language semantics require that it is specified in a fully deterministic way. This contributes towards the accuracy and consistency objective since the language will force the modeller to resolve conflicts before simulation is permitted.

Additionally structural and signal range coverage of the HLR model can be recorded during test case execution. This aids the assessment of requirements consistency at review. For example, the root cause of incomplete structural coverage can be contradictory requirements.

Review Artefacts:

- Requirements document (s)
- LLR model (& model review output)
- Simulation input cases
- Simulation output
- Traceability from system requirements to LLR model, test cases and results
- Model Coverage report

Run DO-178B Model Advisor checks - Activity 23

Rationale:

Same rationale as Activity 8, specifically:

The Model Advisor DO-178B checks statically analyse a Simulink model for possible sources of ambiguity. These static checks complement the dynamic verification through simulation described in Activity 19 and Activity 54.

Activity 8 rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Model Advisor report
- LLR model

6.3.2c - Low-level requirements are compatible with target computer

DO-178B Objective: "The objective is to ensure that no conflict exists between the software requirements and the hardware/software features of the target computer, especially, the use of resources (such as bus-loading), system response times, and input/output hardware. " [1]

Assessment of block execution/sort order - Activity 22

Rationale:

Simulink determines the execution order for blocks in the model. This is termed the "Sorted order". Wherever possible, it is best to accept the order implied by the model. Note that this order will also be replicated in code generated from the model. When adding software detail to a model, i.e. developing the LLR model, it is sometimes appropriate to override the default order. An example is to "load balance" a computationally expensive calculation across multiple model ticks. Typically this would be done to meet a software non-functional requirement such as "the calculation must complete within X ms".

Simulink allows the "sorted order" of its blocks to be displayed (Format -> Block Displays -> Sorted Order). This shows the fixed, deterministic execution order Simulink has calculated for all model elements. Similarly, Stateflow has a transition order annotation feature which displays transition order on the chart (View -> Show Transition Execution Order). With these diagnostics turned on, if Simulink Report Generator or Simulink's "Export to Web" feature is used to generate a report of the model, the execution order will also be displayed. The generated report can be assessed in review. This assessment would usually be supplemented by execution timing data from the actual target.

Review Artefacts:

- Requirements document(s)
- LLR Model including sorted order data

Use of simulation to support target computer selection criteria - Activity 25

Rationale:

The LLR models will reflect implementation specific detail, such as function partitioning, data typing and execution rates. Restrictions imposed by the target computer or operating system are one input to these design decisions. Simulation of LLR models allows the effects of these choices on algorithmic behaviour to be assessed. Simulation results cannot be the sole means of compliance with this objective and final compatibility must be verified on the target hardware.

Review Artefacts:

- LLR model review
- LLR model simulation output

Run DO-178B Model Advisor checks - Activity 26

Rationale:

Same rationale as Activity 8, specifically:

If the project is also generating code using Real-Time Workshop Embedded Coder, the Model Advisor checks can be used to verify the code generator settings that relate to target CPU architecture. The Simulink language replicates target computational precision (e.g. data type, size and format) for simulation.

Activity 8 rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Model Advisor report
- LLR model

6.3.2d - Low-level requirements are verifiable

DO-178B Objective: "The objective is to ensure that each low-level requirement can be verified. " [1]

Measure Coverage during LLR model simulation - Activity 29

Rationale:

Same rationale as Activity 14, but for LLR.

Activity 14 rationale:

The high-level requirements are verifiable if all conditions and decisions are reached during simulation. This is measured directly by the Model Coverage tool. In projects working directly to DO-178B it is expected that a complete set of tests will be developed.

Where incomplete coverage is found, some of the possible root causes can be contradictory requirements, errors in the model, incomplete models, redundant parts of the model, errors in the test cases or incomplete tests (other sources may exist). The traceability reviews assist in identifying incomplete models, redundant parts of the models and incomplete test cases.

Test generation using Simulink Design Verifier will return a test case for all reachable coverage objectives and is best applied to extend an existing suite of requirements based tests (See Activity 4 for further detail on the use of Simulink Design Verifier).

Review Artefacts:

- Model Coverage report
- LLR model

6.3.2e - Low-level requirements conform to standards

DO-178B Objective: "The objective is to ensure that the Software Design Standards were followed during the software design process and that deviations from the standards are justified. " [1]

Run DO-178B Model Advisor checks - Activity 34

Rationale:

Same rationale as Activity 8.

The Model Advisor supports modelling standards for DO-178B, IEC 61508, and the MathWorks Automotive Advisory Board (MAAB). It may also be extended to support other projects specific standards. The Model Advisor checks assess conformance of LLR models to these standards.

Activity 8 rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Model Advisor report
- Project specific modelling standard
- LLR model

6.3.2f - Low-level requirements are traceable to high-level requirements

DO-178B Objective: "The objective is to ensure that the high-level requirements and derived requirements were developed into the low-level requirements. " [1]

Use of XML compare tool to compare HLR and LLR models - Activity 36

Rationale:

Where HLR and LLR models are both in Simulink, the XML comparison report is a very useful contribution to the review process. It enables the user to trace how elements of the LLR model have derived from the HLR model. When High Level Requirements are not expressed in a model, the guidance for HLR traceability to system requirements (Activities 2 and 3) can be re-applied to LLR models.

Review Artefacts:

- XML comparison report
- HLR model
- LLR model

Run DO-178B Model Advisor checks - Activity 37

Rationale:

Same rationale as Activity 8, specifically:

In the event that the Requirements Management Interface is being used to link requirements documents to models, the DO-178B checks verify that the traceability links between LLR models and requirements documents are valid. The validity of associating "system requirement A" with "model element B" is not machine checkable and must be examined by review.

Activity 8 rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Model Advisor report
- LLR model

6.3.2g - Algorithms are accurate

DO-178B Objective: "The objective is to ensure the accuracy and behaviour of the proposed algorithms, especially in the area of discontinuities. " [1]

Use of XML compare tool to compare HLR and LLR models - Activity 39

Rationale:

Where HLR and LLR models are both in Simulink, the XML comparison report is a very useful contribution to the review process. It enables the user to identify how the LLR model has derived from the HLR model and make a judgement (by manual review) that the refinement steps have not compromised the intent of the HLRs.

Review Artefacts:

- XML comparison report
- HLR model
- LLR model

Run DO-178B Model Advisor checks - Activity 42

Rationale:

Same rationale as Activity 8, specifically:

The DO-178B checks statically verify the data types used within the model. This Activity is typically supplemented by simulation based tests, as described in Activity 54.

Activity 8 rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Model Advisor report
- LLR model

6.3.3b - Software architecture is consistent

DO-178B Objective: "The objective is to ensure that a correct relationship exists between the components of the software architecture. This relationship exists via a data flow and control flow. " [1]

Run DO-178B Model Advisor checks - Activity 50

Rationale:

Same rationale as Activity 8, specifically:

Simulink has built-in diagnostics that will identify whether or not the model architecture is internally compatible (e.g. data types are explicitly defined at function interfaces and are the same either side of the boundary). The diagnostics help the engineer identify the presence of architectural inconsistencies. The diagnostics are user selectable and the Model Advisor DO-178B checks are used to confirm that these diagnostics are enabled. This ensures, when inconsistencies are detected, an error preventing simulation or code generation is produced.

Simulink's Model Reference feature may be used to construct a single top level model that references all other LLR models. When used, this construction will allow the architectural diagnostics to examine the complete model hierarchy.

Activity 8 rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Model Advisor report
- LLR model

6.3.3e - Software architecture conforms to standards

DO-178B Objective: "The objective is to ensure that the Software Design Standards were followed during the software design process and that deviations from the standards are justified, especially complexity restrictions and design constructs that would not comply with the system safety objectives. " [1]

Run DO-178B Model Advisor checks - Activity 51

Rationale:

Same rationale as Activity 8, specifically:

The DO-178B checks include tests for explicit ordering of Stateflow states and transitions, conformance of state machines implemented in Stateflow to the specified type (i.e. Mealy or Moore semantics), diagnostic settings for sample time, and diagnostic settings for solvers. User defined custom checks may also be implemented to test conformance of the LLR model architecture to the machine checkable requirements of the Software Design Standard.

Activity 8 rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Model Advisor report
- LLR model

6.3.4d - Source code complies to standards

DO-178B Objective: "The objective is to ensure that the Software Code Standards were followed during the development of the code, especially complexity restrictions and code constraints that would be consistent with the

system safety objectives. Complexity includes the degree of coupling between software components, the nesting levels for control structures, and the complexity of logical or numeric expressions. This analysis also ensures deviations to the standards are justified. " [1]

Use Polyspace to check for conformance to coding standards - Activity 47

Rationale:

Polyspace helps partially to identify deviation from coding standards. When used in conjunction with the DO Qualification kit, Polyspace provides a qualified compliance report. Polyspace partially checks against the following standards:

- MISRA ®
- JSF++

Review Artefacts:

- Source Code
- Polyspace Compliance Report

6.3.4e - Source code traceable to low level requirements

DO-178B Objective: "The objective is to ensure that the software low-level requirements developed into Source Code. " [1]

Use HTML Code generation report to support source code review - Activity 48

Rationale:

Real-Time Workshop Embedded Coder Code Generation Report, which includes a traceability report, can be used as an input to the manual review process and may improve source code review efficiency. The code generation report provides fast navigation between LLR model and HTML source code representation as it provides direct two-way linking via hyperlinks in the report.

Review Artefacts:

- Real-Time Workshop Embedded Coder Code Generation Report
- LLR Model
- Source Code

Run DO-178B Model Advisor checks - Activity 52

Rationale:

Same rationale as Activity 8.

The Model Advisor DO-178B checks verify that the Real-Time Workshop Embedded Coder settings are applied to maximise traceability between model and generated code.

Activity 8 rationale:

The DO-178B Model Advisor checks may be used to statically check a Simulink model for possible sources of ambiguity and to assess the model against a defined standard. Projects can define additional (e.g. in-house) standards for modelling and the Model Advisor is commonly extended to include these custom checks.

Custom checks in the Model Advisor can also improve review efficiency by enforcing model style guides. A well enforced style guide means that all reviewers can readily understand the models presented and do not need to spend time correcting format or layout problems. For example, a style guide may limit the blocks in a subsystem to ensure it is readable on-screen and in print. User defined custom checks could also be related specifically, for example, to the constraints of the target environment.

The Model Advisor DO-178B checks can be qualified when used in conjunction with the DO-Qualification Kit. The steps required to achieve qualification are detailed in the relevant Test Cases, Procedures, and Results document. When used as a qualified tool, a Model Advisor report indicating that all checks have passed is reviewed.

If the Model Advisor is extended to include custom checks, then the qualification kit must also be extended by the user. Where aspects of the modelling standard are not machine checkable, then this must be assessed directly by manual review.

These static checks complement the dynamic verification through other simulation and test activities.

Review Artefacts:

- Model Advisor Report
- LLR Model

6.4.3 - Requirements-Based Testing Methods

DO-178B Objective: "Requirements-based testing methods consist of methods for requirements-based hardware/software integration testing, requirements-based software integration testing, and requirements-based low-level testing.With the exception of hardware/software integration testing, these methods do not prescribe a specific test environment or strategy" [1]

Use target-based code verification with LL test re-use in external tool - Activity 46

Rationale:

The simulation data can be used to validate that the object code complies with low level requirements. Simulation test cases are exported and re-run against the code in an external environment. The results achieved in the external environment should match the simulation results achieved in Simulink. Acceptable tolerances should be pre-defined. Differences in the results should be explained with justification.

There are a number of potential sources of differences between source and object code, e.g. due to task ordering. The assessment may be used to differentiate between deterministic differences where a close match would be expected and non-deterministic differences where effects such as jitter may be observed.

Review Artefacts:

- Exported LLR Test cases
- Exported LLR model simulation output
- Results in external tool

7.1a - Provide a defined and controlled configuration of the software throughout the software lifecycle

DO-178B Objective: "The objective is to provide a defined and controlled configuration of the software throughout the software life cycle. " [1]

Use of Simulink Manifest feature to support model traceability - Activity 44

Rationale:

The Simulink Manifest report shows model dependencies which may be used as a check list for the review activity, supporting the configuration management objectives.

The Manifest analysis includes dependencies for:

- Model reference
- Library links
- MATLAB functions (including block and model call-backs, Embedded MATLAB dependencies)
- Data files
- Files for code generation (e.g. templates, TLC, etc)
- Requirements documents (when using requirements linking)
- User toolboxes
- MATLAB script files

Review Artefacts:

- Simulink Manifest report

Glossary

Glossary of terms, document usage and abbreviations.

Activity Numbering:

Each activity is identified with a unique activity number. This number reflects the activity's position in the database and should not be used to indicate a suggested sequence of activities in the software development process.

Condition and Decision Coverage (See Simulink product documentation for Using Model Coverage):

Decision Coverage - Decision coverage analyzes elements that represent decision points in a model, such as a Switch block or Stateflow states. For each item, decision coverage determines the percentage of the total number of simulation paths through the item that the simulation actually traversed.

Condition Coverage - Condition coverage analyzes blocks that output the logical combination of their inputs (for example, the Logical Operator block) and Stateflow transitions. A test case achieves full coverage when it causes each input to each instance of a logic block in the model and each condition on a transition to be true at least once during the simulation, and false at least once during the simulation. Condition coverage analysis reports whether the test case fully covered the block for each block in the model.

Modified Condition/Decision Coverage - Modified condition/decision coverage analysis by the Simulink Verification and Validation software extends the decision and condition coverage capabilities. It analyzes blocks that output the logical combination of their inputs and Stateflow transitions to determine the extent to which the test case tests the independence of logical block inputs and transition conditions.

DO Qualification Kit - Contains the Tools Qualification Plan and supporting process documents and models to qualify many of the verification activities described in this document.

HIL output - Output signals logged during hardware-in-the loop testing.

HLR model - A model which describes the properties of the High Level Requirements.

LLR model - A model which describes the properties of the Low Level Requirements.

Model Advisor qualification report - The report that is generated, when using the DO Qualification Kit, giving the qualification status of the Model Advisor.

Model Advisor report - The report that is generated by the Model Advisor giving the pass/fail/warning status of each model advisor test as applied to the selected model.

Model Advisor Requirements Consistency report - The report that is generated by the Model Advisor giving the status of the consistency check for each requirements link within the selected model.

Model Advisor Requirements report - The report that is generated by the Model Advisor showing which elements of the selected model are linked to a requirements document.

Model Coverage report - The report generated by the Simulink coverage tool showing the extent of

coverage achieved by the model during simulation.

Modelling standard - A set of modelling rules that could, for example, aid readability, improve re-use, ensure corporate workflow compatibility, etc. These could be externally published (e.g. MathWorks Automotive Advisory Board MAAB) or company/project specific standards.

Polyspace Compliance Report - The report generated by Polyspace showing the status of the source code analysis.

Real-Time Workshop Embedded Coder Code Generation Report - The report generated by Real-Time Workshop Embedded Coder at code generation time. It is an HTML representation of the generated code including trace hyperlinks to the source models, requirements documents, etc.

Requirements document - A textual document (e.g. in Microsoft ® Word, DOORS®, etc) which describes the system requirements.

Simulation input cases - Model input signals/data used to drive simulation cases

Simulation output - Model output signals logged during simulation testing.

Simulink Design Verifier Proof report - The report generated by Simulink Design Verifier giving the status (and any counter examples) of a model proof analysis.

Simulink Manifest report - The report generated by the Simulink Manifest tool which shows model and library interdependencies.

Sorted order data - A view of the model showing the execution order of the elements of the model.

Source Code - e.g. C/C++, Fortran, Ada, etc

Test cases - Sets of simulation input data and the resulting output data.

Verification subsystem - A special case model sub-system which is excluded from any code generation analysis by Real-Time Workshop. Requirements based verification tests can be implemented in a verification sub-system allowing code to be generated, excluding the verification tests, without the need to change the model.

XML comparison report - The report generated by the comparison of XML files using Simulink Report Generator.

Abbreviations:

CAST	Certification Authorities Software Team
CPU	Central Processor Unit
HIL	Hardware in the Loop - testing the production software in real-time against production hardware and hardware simulations.
HLR	High Level Requirements
LLR	Low level requirements
MISRA	Motor Industry Software Reliability Association
PIL	Processor in the Loop - testing the production object code on the target processor, communicating with the Simulink environment for test inputs and outputs

SIL	Software in the Loop - testing the production source code in Simulink environment
XML	Extensible Markup Language

Bibliography

[1] Software Considerations in Airborne Systems and Equipment Certification, Document No. RTCA/DO-178B, December 1, 1992, prepared by SC-167.

[2] Certification Authorities Software Team CAST Position Papaers (http://www.faa.gov/aircraft/air_cert/design_approvals/air_software/cast/)

[3] Defence Standard 00-56, Safety Management Requirements For Defence Systems, Ministry Of Defence, Issue 4, 01 June 2007

[4] DO Qualification Kit Product Documentation (http://www.mathworks.com/access/helpdesk/help/toolbox/qualkitdo)

[5] Guidelines for the Use of the C Language in Critical Systems, MISRA, ISBN 0 9524156 2 3 (paperback), ISBN 0 9524156 4 X (PDF), October 2004.

[6] Model-Based Design For Safety Critical Or Mission Critical DO-178B Applications Using MathWorks Software (http://www.mathworks.com/support/solutions/en/data/1-1ZLDDE/?solution=1-1ZLDDE)

[7] Real-Time Workshop Embedded Product Documentation (http://www.mathworks.com/access/helpdesk/help/toolbox/ecoder)

[8] Simulink Product Documentation (http://www.mathworks.com/access/helpdesk/help/toolbox/simulink)

[9] Simulink Design Verifier Product Documentation (http://www.mathworks.com/access/helpdesk/help/toolbox/sldv)

[10] Simulink Verification and Validation Product Documentation (http://www.mathworks.com/access/helpdesk/help/toolbox/slvnv)