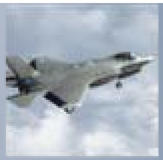
The MathWorks MATLAB®&SIMULINK®

Verification and Validation

Introducing Simulink Design Verifier

Goran Begic, Technical Marketing
Goran.Begic@mathworks.com

June 5, 2007



MathWorks
Aerospace and Defense Conference '07

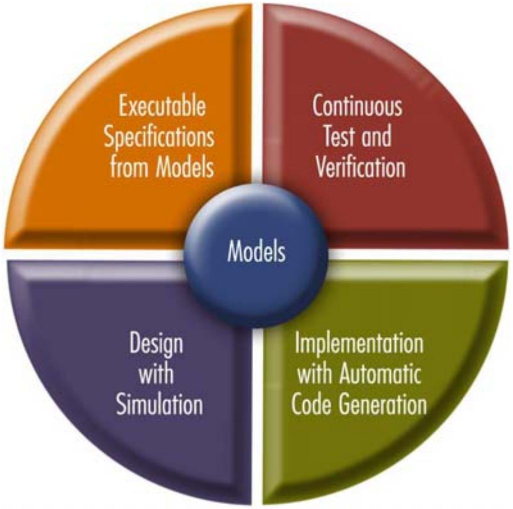
Agenda

- Verification and Validation in Model-Based Design
  - Overview of verification and validation activities and products that support them
- Introducing Simulink Design Verifier
  - What is Simulink Design Verifier?
  - Why is it important?
  - How does it work?
  - Demonstration
- Summary
- Questions

Verification and Validation in Model-Based Design

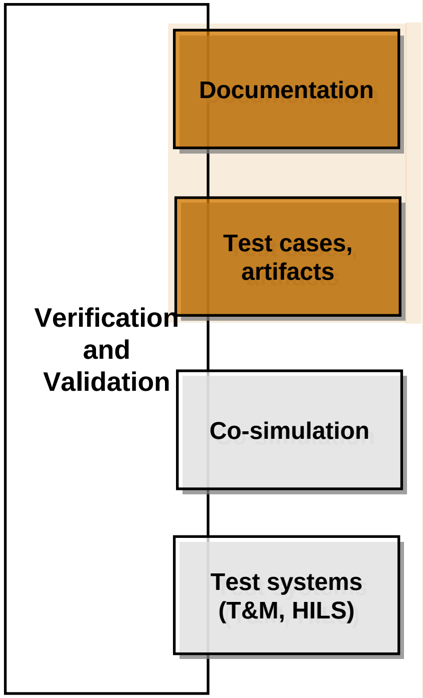
- Verification and Validation is one of the inherent benefits of Model-Based Design

- **Continuous Test and Verification**
  - Important design concepts
    - “It should work”
  - Implementation of requirements
    - “It works“
  - Objective evidence


Executable Specifications from Models
Continuous Test and Verification
Models
Design with Simulation
Implementation with Automatic Code Generation

Verification and Validation

Overview



Verification and Validation

Documentation

Test cases, artifacts

Co-simulation

Test systems (T&M, HILS)

- **Reporting**
  - Automatic collection and archiving of artifacts

Simulink Report Generator

- **Model Checking**
  - Static checking of Simulink and Stateflow models
- **Requirements Traceability**
  - Linking and tracing requirements from and to models
- **Model Coverage**
  - Coverage of model structure and logic

Simulink Verification and Validation

- **Functional Testing**
  - Parameter sweeps
  - Monte Carlo Analysis

SystemTest

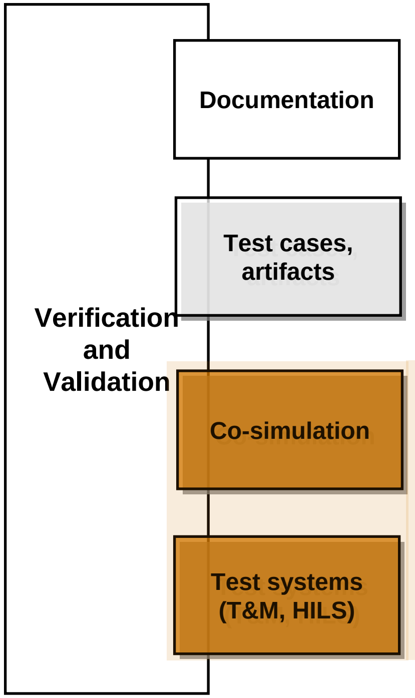
- **Formal Methods**
  - Test generation
  - Property Proving

NEW IN R2007a

Simulink Design Verifier

Verification and Validation

Overview



- **Processor In The Loop**
  - Verification of generated object code on the target processor
  - Co-simulation

Embedded Software
Link for Code Composer Studio™
Link for TASKING®
Target for Freescale™ MPC5xx
Target for Infineon C166®
Digital Hardware
Link for Cadence® Incisive®
Link for ModelSim®

- **Hardware In The Loop**
  - Real-Time verification of software inside embedded controller against plant simulation

xPC Target
xPC Target Embedded Option

- **Test And Measurement**
  - Data acquisition
  - Instrument control

Data Acquisition Toolbox
Instrument Control Toolbox

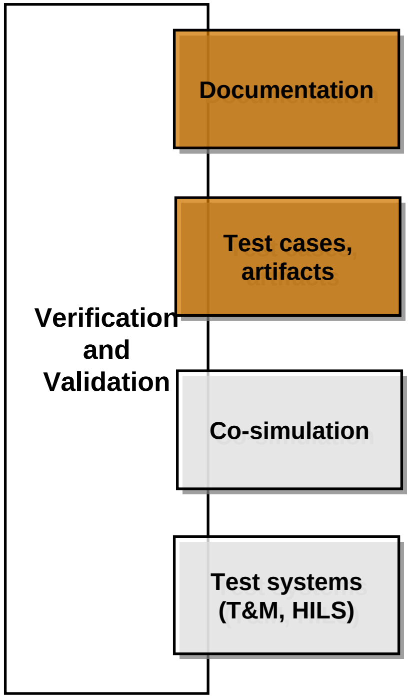
Verification and Validation

Overview

- **Formal Methods**
  - Test generation
  - Property Proving

NEW IN R2007a

Simulink Design Verifier



- Introducing Simulink Design Verifier
  - What is Simulink Design Verifier?
  - Why is it important?
  - How does it work?
  - Demonstration

Simulink Design Verifier

What is it?

- Formal analysis
- Not simulation

- **New model verification and validation product**
  - New verification technology for Simulink and Stateflow

- **Based on formal analysis engine from Prover Technology**

prover plugged in

- **Key Features**
  - Generates tests for Simulink® and Stateflow® models
  - Detects unreachable design elements in models
  - Proves model properties and generates example of violations
  - Includes blocks for definition of properties
  - Produces detailed test-generation and property-proving analysis reports

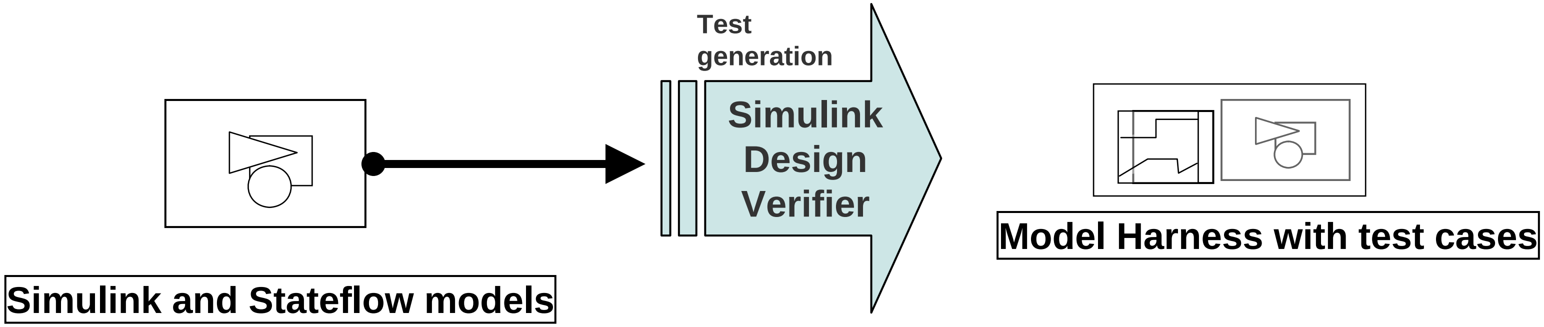
Simulink Design Verifier

Why is it important?

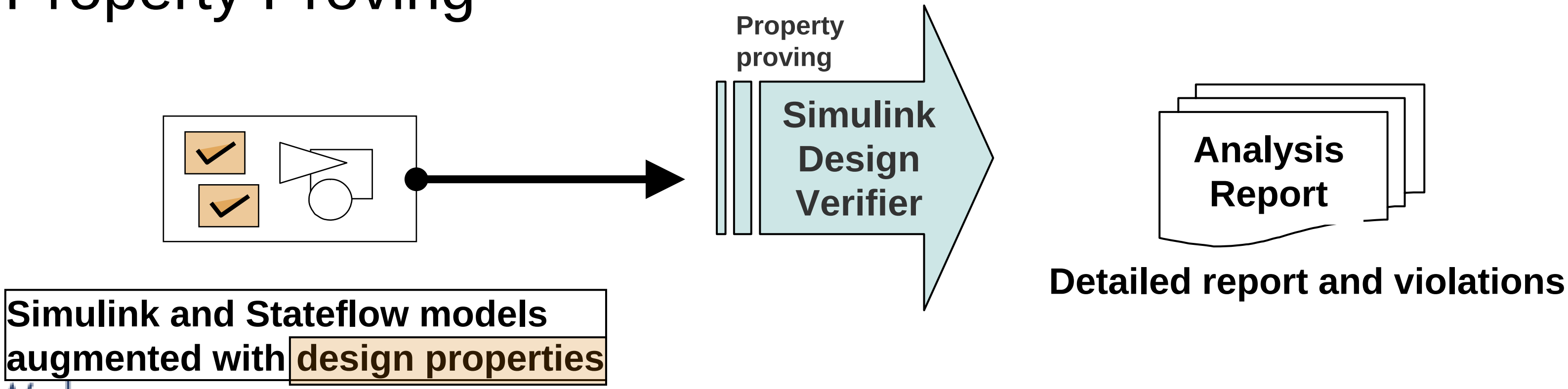
- **Building exhaustive tests is hard and time consuming**
  - Example: Achieving 100% MC/DC coverage
- **Some functional requirements are difficult to prove via simulation**
  - Example requirement: Reverse thrust operation shall not engage when aircraft in flight
- **Particularly relevant for:**
  - Safety critical applications
  - Complex Stateflow models
  - Component based development

Working with Simulink Design Verifier

- Test Generation



- Property Proving

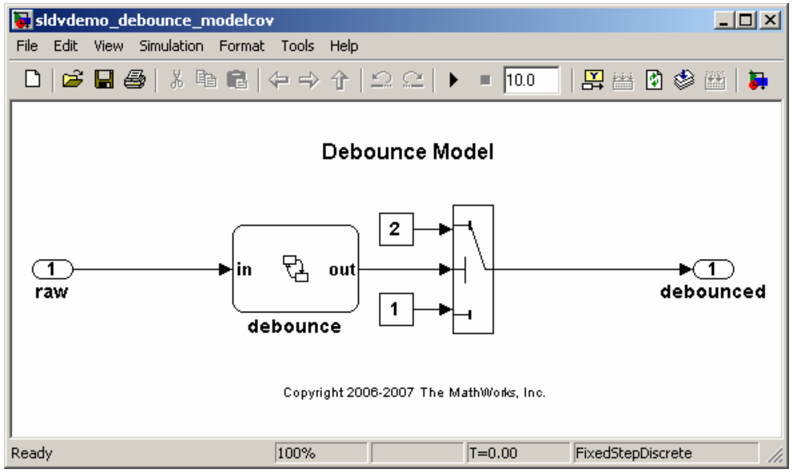

Property proving
Simulink Design Verifier
Analysis Report
Detailed report and violations
Simulink and Stateflow models augmented with design properties

Simulink Design Verifier

Demonstration

- Example model available in product help

- Test generation for model coverage
- Property definition and proving


sldvdemo_debounce_modelcov
File Edit View Simulation Format Tools Help
10.0
Debounce Model
1
raw
in
out
debounce
2
1
1
debounced
Copyright 2006-2007 The MathWorks, Inc.
Ready
100%
T=0.00
FixedStepDiscrete

Demonstration using debounce model

Summary

- Design Verifier
  - Uses static analysis to verify model behavior
  - Complete and exhaustive analysis that uncovers problems that are very difficult to detect using simulation only
- Practical implementation of formal methods in control design applications
- Minimizes the risk of unknown and unexpected execution scenarios

Resources

- Product web page

http://www.mathworks.com/products/sldesignverifier/

  - Demo recordings
  - Data sheet
  - User's Guide
- Webinar on June 12
- Exhibit Hall
  - Verification and Validation station

Questions

- Goran Begic
  - gbegic@mathworks.com
  - 508.647.4313