Model-Based Design for Automotive Control Systems
Agenda

- Overview of The MathWorks
- Introduction of Model-Based Design
- Key MathWorks Products
- Early Verification and Validation Tools
- RP and HIL Solution
The MathWorks at a Glance

Headquarters:
Natick, Massachusetts USA

USA:
California, Michigan, Washington DC, Texas

Europe:
UK, France, Germany, Switzerland, Italy, Spain, the Netherlands, Sweden

Asia-Pacific:
China, Korea, Australia, Japan, India

Worldwide training and consulting

Distributors in 25 countries

Earth’s topography on an equidistant cylindrical projection, created with MATLAB and Mapping Toolbox
MathWorks Today

- Revenues ~$500M in 2008
- Privately held
- More than 2,200 employees worldwide
- Worldwide revenue balance: 45% North America, 55% international
- More than 1,000,000 users in 175+ countries
Key Industries

- Aerospace and Defense
- Automotive
- Biotech and Pharmaceutical
- Chemical/Petrochemical
- Communications
- Education
- Electronics
- Financial Services
- Industrial Automation and Machinery
- Power and Energy
- Semiconductor
MathWorks Product Family Overview

Simulink Product Family

Application-Specific Products

MATLAB Product Family
Agenda

- Overview of The MathWorks
- Introduction of Model-Based Design
- Key MathWorks Products
- Early Verification and Validation Tools
- RP and HIL Solution
Current Challenges in Automotive

- Environmental issues
  - Oil shortage and resulting price development

- Legal regulations
  - CO2 Emission Limits; e.g. 120g/km by 2015 in EU

- Customer Requirements
  - Cost efficient and save vehicles
Current Challenges in Automotive Applications under development

- Lights
- Climate Controls
- Voice Recognition
- Power
- Engine Control
- Navigation
- Windows
- Obstacle Detection
- Instrumentation
- Adaptive Front Lighting Systems
- Traction Control
- Driver Drowsiness
- Infotainment
- Collision Avoidance
- Airbags
- Passenger Doors
- ABS
- Adaptive Cruise Control
- Tire Pressure Monitoring
- 2010: Electronics expected to comprise 40% of material cost (SAE AEI March 2005)
Current Challenges in Automotive Applications under development

- Hybrid Electrical Vehicles
- Brake-Energy Recuperation
- Battery Management Systems
- Electrical Motors
- Electrical Power Steering
- Electrical Accessories
- Pure Electrical Vehicles
- Start-Stop Systems
- Fuel Cells
- Accessories
Explosive Growth of Code

Estimates for Lines of Code (LOC)
- Today’s powertrain: 500,000 LOC
- Today’s vehicles: 1,000,000 LOC
- 2015 vehicles: 100,000,000 LOC

* 2006 figures from SAE AEI

“Growth of top end automotive embedded software has been exponential.”

Robert Gee
Director of Strategy for Motorola Automotive

“…no longer possible to validate and verify functionality by brute force testing alone.”

Jim Kolhoff
Director of Software Engineering at GM Powertrain

Automotive Engineering, “Managing for Software Success” – Aug 2006
sae.org/automag/electronics/08-2006/1-114-8-34.pdf
Model Based Design

- **Requirements**
- **System Design**
- **Software Design**
- **Software Integration**
- **Coding**
- **Vehicle Integration & Calibration**

**Simulation**
- Verification and Validation
- Matlab/Simulink/Stateflow
- Design Verifier
- Physical Modeling

**Rapid Prototyping**
- Matlab/Simulink/Stateflow
- Design Verifier
- Physical Modeling

**Hardware/Software Integration**
- Processor-in-the-Loop Testing
- PIL
- Target Support

**Vehicle Integration & Calibration**
- Hardware-in-the-Loop Testing
- HIL
- xPC

**Production Code Generation**
- Fixed-Point RTW/E-Coder
- Target Support

**PolySpace SIL**

**Design with Simulation**
- Executable Specifications from Models
- Continuous Test and Verification

**Implementation with Automatic Code Generation**

**On-Target Rapid Prototyping**

**Software-in-the-Loop Testing**

**Hardware-in-the-Loop Testing**

**Processor-in-the-Loop Testing**

**12**
Problems with Traditional Development

Requirements and Specs

Test documents prevents rapid iteration

Design

Physical prototypes incomplete, expensive

Implementation

Manual implementation separate tools & human error

Test and Verification

Traditional testing errors found late in process
Applying Model-Based Design

Requirements and Specs

Design

Implementation

Test and Verification

Physical prototypes incomplete, expensive

Manual implementation separate tools & human error

Traditional testing errors found late in process

Executable Specification

- Unambiguous spec, supplemented by text
- One set of models for all teams
- Model whole system including environment
- Block diagram description
- Early validation and test development
Applying Model-Based Design

Requirements and Specs

Design

Implementation

Test and Verification

Design with Simulation

- Systematic design exploration and optimization
- Find flaws before implementation
- Apply to both controller and physical plant
- Incremental design from system level to implementation

Manual implementation separate tools & human error

Traditional testing errors found late in process
Applying Model-Based Design

Requirements and Specs

Design

Implementation

Test and Verification

Automatic Code Generation
- No manual coding errors
- Hardware target portability
- Improved testability due to repeatability
- Bridge between domain, software and hardware knowledge
- Hardware-in-Loop for physical model

Traditional testing errors found late in process
Applying Model-Based Design

Requirements and Specs → Design → Implementation → Test and Verification

Model elaboration → Continuous verification

Continuous Test and Verification
- Detect errors early in development
- Reduce dependency on physical prototypes
- Implementations that work the first time
- Reuse test suites across development stages
Model-Based Design Workflow

Requirements and Specs

Design

Implementation

Test and Verification
The Value of Model-Based Design

Model-Based Design
- Executable specification
- Design with simulation
- Implementation through code generation
- Continuous test and verification

Innovation
- Rapid design iterations
- “What-if” studies
- Unique features and differentiators

Quality
- Reduce design errors
- Minimize hand coding errors
- Unambiguous communication internally and externally

Cost
- Reduce expensive physical prototypes
- Reduce re-work
- Automate testing

Time-to-market
- Get it right the first time
Agenda

- Overview of The MathWorks
- Introduction of Model-Based Design
- Key MathWorks Products
- Early Verification and Validation Tools
- RP and HIL Solution
MathWorks provide MBD tools and Physical modeling tools to integrate control software design and control target modeling to in a common platform.
Core MathWorks Products

MATLAB

The leading environment for technical computing

- The *de facto* industry-standard, high-level programming language for algorithm development
- Numeric computation
- Data analysis and visualization
- Toolboxes for signal and image processing, statistics, optimization, symbolic math, and other areas
- Foundation of MathWorks products
Simulink

- System-level modeling
  - Multidomain
  - Graphical
  - Interactive
  - Hierarchical
- Algorithm design
- Simulation
  - Model is an “executable specification”
Stateflow

- Extend Simulink with a design environment for developing state machines and flow charts
- Design systems containing control, supervisory, and mode logic
- Describe logic in a natural and understandable form with deterministic execution semantics
Implement Design with Code Generation

Real-Time Workshop Embedded Coder

- Generate ANSI/ISO C code from Simulink models
  - Readable
  - Traceable
  - Compact
  - Efficient
  - Consistent
  - For deployment to microprocessors and DSPs

- Generating HDL Code
  - For deployment to FPGAs and ASICs

```c
rtY.heat = (int16_T)(rtU.Sensor >> 5);
```
Agenda

- Overview of The MathWorks
- Introduction of Model-Based Design
- Key MathWorks Products
- Early Verification and Validation Tools
- RP and HIL Solution
Tracing Requirements ↔ Model
Simulink Verification and Validation

Creating links between textual documents and model objects
Tracing Requirements ↔ Source Code
Simulink Verification and Validation
Real-Time Workshop Embedded Coder

Including requirements in the generated source code
Tracing Model ↔ Source Code
Real-Time Workshop Embedded Coder

Bidirectional navigation between the model and the generated code
Traceability – Summary

Simulink Verification and Validation
Real-Time Workshop Embedded Coder

Requirements Traceability – Report
Simulink Verification and Validation

Traceability Report
Real-Time Workshop Embedded Coder
ISO 26262 工具资质认证

首个具有ISO/DIS 26262资质的产品代码生成器
- Real-Time Workshop Embedded Coder和PolySpace已经通过TÜV SÜD的资质验证
- MathWorks创建的工具认证包和开发流程也通过了TÜV SÜD的评估
- 用户通过对工具认证包的客户化及参考相应的证书/认证报告来进行工具资质认证

包含如下模板：
- 软件工具资质计划
- 软件工具文档
- 软件工具分类分析
- 软件工具资质报告
ISO 26262 模型自动校验
MISRA-C:2004 自动校验
模型测试

*Simulink Design Verifier*

- Building exhaustive tests is hard and time consuming
  - Example: Getting 100% model coverage

- Exhaustive structural testing is particularly important for
  - Safety critical applications
  - Complex logic
  - Component based development

- Generated test cases are reusable
  - Test design in the model form
  - Test code after implementation
  - Test code on target processor
模型覆盖率测试
Simulink Verification and Validation
- Model Coverage tool reports coverage metrics

Subsystem "Logical Operator"

Parent: coverage_example_harness/Test Unit (copied from coverage_example)

Uncovered Links:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclomatic Complexity</td>
<td>0</td>
</tr>
<tr>
<td>Condition (C1)</td>
<td>100% (4/4) condition outcomes</td>
</tr>
<tr>
<td>MCDC (C1)</td>
<td>50% (1/2) conditions reversed the outcome</td>
</tr>
</tbody>
</table>

Conditions analyzed:

<table>
<thead>
<tr>
<th>Description</th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>input port 1</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>input port 2</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

MC/DC analysis (combinations in parentheses did not occur)

<table>
<thead>
<tr>
<th>Decision/Condition</th>
<th>True Out</th>
<th>False Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>input port 1</td>
<td>T</td>
<td>T(T)</td>
</tr>
<tr>
<td>input port 2</td>
<td>T</td>
<td>T(F)</td>
</tr>
</tbody>
</table>
Agenda

- Overview of The MathWorks
- Introduction of Model-Based Design
- Key MathWorks Products
- Early Verification and Validation Tools
- RP and HIL Solution
From Simulation to Real-Time Testing

- Models designed in Simulink and Stateflow
- You want to run, test, and prove your Simulink design with your hardware under test at its normal operating frequency, speed, or timing.
- But how can you do this testing in real-time?
xPC Target

- Complete, fully assembled, real-time testing solution
- Combines xPC Target (software) with a real-time target machine and IO modules (hardware)
- RP, HIL
Today’s Car Depends on Electronic Controls

- Engine Management System
- Hybrid Delivery Truck
- Construction Equipment
- CNG/LPG EMS
- HVAC
- Common Rail Diesel
- Battery Management System
- Active Suspension Control
- Hybrid Powertrain
- Vehicle Controller
- Central Body Controller
- Telematics
MathWorks

Change the world by

Accelerating the pace
of discovery, innovation, development, and learning

in engineering and science