Hardware Software Co-Design and Testing Using Simulink® Real-Time™

Paul Berry and Brian Steenson
Overview

Process Development

• Introduction to THALES
• Overview of design process
• Development of autocode capability

Real Time Testing for the Lightweight Multirole Missile

• Guidance and control algorithm design
• Guidance and control algorithm implementation
• Guidance and control algorithm testing
THALES in the UK

Aerospace
- Avionics Systems
- Air Traffic Management
- In-Flight Entertainment
- Electrical Systems
- Training and Simulation

Defence
- Radar
- Watchkeeper
- Command and Control
- Cameras and Sensors
- Sonar Systems
- Threat Warning
- Short Range Defence

Ground Transportation
- Signalling Systems
- Integrated Comms & Supervision Systems
- Revenue Collection Systems

Security
- Secure Communication
- Network & Infrastructure Systems
- Protection Systems
- Critical Information Systems & Cybersecurity

Space
- Telecoms
- Observation
- Infrastructure
- Navigation
THALES in Belfast
Model Based Algorithm Development

System Requirements / Model

**DESKTOP**
- Algorithm Design
- Verify Design

**DEVELOPMENT**
- Verify Timing
- Test Cases
- Verify Design

**PRODUCTION**
- Real Time Simulation
- Real Time Validation

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2017 All rights reserved.
Model Based Algorithm Development

System Requirements / Model

DESKTOP
Algorithm Design
Verify Design

DEVELOPMENT
Test Cases
Verify Timing
Verify Design

SIMULINK

PRODUCTION
Real Time Simulation
Real Time Validation

HARDWARE IN THE LOOP
Goals and Objectives

- Remove human error
- Reduced code development time
- Early prototype on hardware
- Improved efficiency
- Fast turn around between iterations
- Common test environment
- Improved traceability
Evolution of Autocoding Capability

2005: Land Based Systems

- Algorithm development for target tracking algorithm
- Implemented in C from Simulink
- Floating point C code

2007: Missile Systems

- Guidance algorithm implemented in C from Simulink
- Target-specific libraries used to optimise speed
- Fixed point C code
Evolution of Autocoding Capability

2009: Hardware/Software Partitioning

- Motor control algorithms developed in Simulink
- Autogenerated C code used to quickly prove concept
- Final solution partitioned between C and HDL

2013: FreeFall-LMM rapid development

- Rapid prototyping of guidance and control algorithms
- Autogenerated C code
- 6 months development from concept to flight trials
Evolution of Autocoding Capability

2017: Next Generation Beam Steering

- Updated algorithms developed for new guidance unit
- μrad positional accuracy and stabilisation error
- Autocoded algorithms ported to System on Chip
- Improved linkages between model and implementation
Evolution of Autocoding Capability – Future?

Missile state machines
- Currently using legacy state machine layer
- Bring this logic within MBD process as complexity increases

Digital laser scanning
- Very high rate (ns), high precision control
- High fidelity simulation crucial to understanding
- MBD approach essential for rapid prototyping and implementation
Real Time Testing for the Lightweight Multirole Missile
LMM launch
Overview

Guidance and control algorithm design

• (Sub)system model development
• Algorithm development
• Performance verification in non-real time Simulink 6DOF simulation
• Generate algorithm autocode (C or HDL)

Guidance and control algorithm testing

• Real time simulator development
  • Real time 6DOF simulation
  • Hardware emulators
• Hardware in the Loop testing
• Verification and Validation
Model Development

Develop subsystem models:

- Aerodynamics
- Structural bending
- Inertial Measurement Unit (IMU)
- Laser Information Field (LIF)
- Semi-Active laser (SAL)
- Control Actuation System (CAS)
- Rocket motor(s)
- Missile dynamics
- Canister exit model
- Launch platform
- Guidance algorithms
Guidance algorithm design

- Controllers developed at key operating points
- Based on a continuous linearised/idealised model
- Algorithm design iteration may be required
- Discrete versions of algorithms created
- Performance quantified in 6DOF simulation
- Iteration may be needed

- Linearise aero functions at fixed Mach
- Create continuous controllers for fixed Mach number
- Test continuous controllers in 6DOF simulation (variable Mach)
- Test discrete controllers in 6DOF simulation
- Good performance?
  - Yes → Proceed to code generation
  - No → Algorithm design iteration may be required
- Good performance?
  - Yes → Proceed to code generation
  - No → Algorithm design iteration may be required
Testing in non-real time - Simulink 6DOF Simulation

- Discrete version
- Continuous version
- Guidance and control algorithms
- Actuator model(s)
- Sensor performance models
- Data Link Noise Model
- Disturbance noise
- Equations of motion
- Aero coefficient tables
- Aerodynamics and kinematics
- Force and moments experienced by missile
- Specific Data visible to missile (with noise)
- Specific Data visible to missile
- Control surface position

Ensure system performance integrity from continuous to discrete domain.
Autocode algorithms onto target - C

- Common autocode configuration settings across projects for code standard consistency
- Run Monte Carlo simulations replacing algorithms with autogenerated code
- Open loop tests using 6DOF generated test vectors performed on target hardware
- Verify executable code integrity and assess coverage
Autocode algorithms onto target - HDL

- Common autocode configuration settings across projects for code standard consistency
- Fixed point model required
- Open loop tests using 6DOF generated test vectors performed on target hardware
- Check executable code integrity and assess coverage
LMM Laser Beam Riding (LBR) HIL key components

**Missile Electronics**

- **Guidance Processing Unit**
  - Generate elevator angle demands
  - Simultaneously roll stabilise missile nose
- **Fin Control Actuation System**
  - Implement fin control algorithms

**System Emulators**

- **Laser Information Field emulator**
  - Provides missile with its position
- **Inertial Measurement Unit emulator**
  - Provides missile with rates and accelerations for inertial reference calculations
Hardware in the Loop (HIL) testing

- Autocoded algorithms implemented on hardware
- Simulink Real Time version of 6DOF simulation created
  - May require simplification - Larger step-size, remove high frequency dynamics, limit real time comms
- Model subsystems can be gradually removed from the 6DOF simulation and replaced with hardware or hardware emulators
- Run Monte-Carlo real time simulations on real time target
- Simulation version of algorithms can run on real time target in parallel to permit debugging
LMM Laser Beam Riding (LBR) HIL simulation

Close guidance loop with either:

- Fin deflections from actual missile hardware
- Fin deflections from real time 6DOF simulation

Missile Hardware

DSP1
FPGA1
G & C algorithms

DSP2
FPGA2

Fin Actuator 1

Fin Actuator 2

Fin Actuator 3

Missile Hardware Simulation

Aerodynamics
kinematics
IMU model
LIF model

LIF and IMU signals in missile interface electrical format

Real Time Computer

Laser Information Field emulator

IMU emulator

Emulation Hardware
Verification and validation

• HIL simulation is a key part of pre-flight tests
• Hardware stimulated closed-loop with realistic flight data
• Failure mode testing
• Iterative development – 6DOF (both non-real time and real time) validated against flight telemetry data
• Linkage from algorithm design model to hardware implementation tests accelerates rapid prototyping development and testing
Next steps

**Missile Hardware**

**Real Time Computer**

**Real Time Sensor Emulation**

**Real Time Scenario Generation**
Goals and Objectives

- Remove human error
- Reduced code development time
- Early prototype on hardware
- Improved efficiency
- Fast turn around between iterations
- Common test environment
- Improved traceability
LMM in action