Decentralized Engine Control System Simulator
DECSS
The hard real-time platform with analog and digital I/O for hardware testing

John McArthur, Bobbie Hegwood, O.A. (Bud) Watts
11 December 2013
Agenda

- Why DECSS and HIL
- DECSS Description
  - Hardware
  - Capabilities
- DECSS Models and Suggested Architecture
- HIL Testing – DECSS Efficiencies
  - Legacy
  - Distributed
- DECSS Operation
  - Flexible and Simplified Modeling from all simulation to HIL
- Summary
HIL testing is important

DECSS provides the environment

- High fidelity testing of hardware in the loop provides validation of all the interfaces and logic before risking expensive hardware and life on the test stand.
  - Facilitates rapid and inexpensive debugging of hardware components
  - Enables low-risk and low-cost environment iterative development with real control system hardware
  - Allows identification and verification of realistic minimum bandwidth, latencies, packet drop probabilities, and redundancies
  - Facilitates analysis of sensitivities to off-nominal component performance
    - Off-design engine, controller, actuator, sensor, and network can be tested for (system and/or component) robustness with low-cost and low-risk
DECSS Description

- DECSS is a hard Real-Time capable Simulation and Test Environment developed to support Distributed and Decentralized Engine Control development and integration with vehicles
  - Hardware is Linux based real-time operating system with interfaces compatible with WinOS and multiple control system communication bus structures plus 16 channels of A/D and 4 channels of D/A
  - Software is built around industry standard MatLab Simulink with support packages to interface with multiple external platforms and to simplify Hardware In the Loop testing and evaluation of developmental systems
Models and Architecture

Limitations:

- Single .mdl file
  - Replacing software with hardware is more difficult
  - Parallel development of model is more difficult

- Use of to/from blocks
  - Can introduce trouble in code generation
  - Cannot see system dependencies
Models and Architecture targeted for DECSS

Improvements:

- Blocks are .mdl files
  - Creates individual executables that can be replaced by hardware
  - Developers can work without worry of conflicts with other developers

- No to/from blocks
  - Eliminates concern of improper code generation
  - System dependencies are more clear
DECSS Models

Engine with Lumped Network Model
High-Level Network Characteristics
- Use Prob. for drops
- Use PDF for delay/jitter
- When network is at 85% full capacity, 5% of packets will be dropped, and delay will be 0.5 +/- 0.1 ms.
**DECSS Models**

High-Level Network Characteristics

- Use Prob. for drops
- Use PDF for delay/jitter
- When network is at 85% full capacity, 5% of packets will be dropped, and delay will be 0.5 +/- 0.1 ms.

For given conditions:

Packet drop prob. increases as traffic (bits/s) increases

Packet delay increases as traffic (bits/s) increases

Engine with Lumped Network Model
Decommissioned Models

High-Level Network Characteristics
- Use Prob. for drops
- Use PDF for delay/jitter
- When network is at 85% full capacity, 5% of packets will be dropped, and delay will be 0.5 +/- 0.1 ms.
**DECSS Models**

**Engine with Lumped Network Model**

### High-Level Network Characteristics
- Use Prob. for drops
- Use PDF for delay/jitter
- When network is at 85% full capacity, 5% of packets will be dropped, and delay will be 0.5 +/- 0.1 ms.
DECSS Models

Simulation with Detailed Network Model
Complex Event Based Comms
- Use clock drift, timing errors, and protocol to simulate traffic, collisions, delays, and over-flows
- Requires SimEvents, or other event/time triggered simulation block set
- Achieving real-time operation can be difficult
Complex Event Based Comms

- Use clock drift, timing errors, and protocol to simulate traffic, collisions, delays, and over-flows
- Requires SimEvents, or other event/time triggered simulation block set
- Achieving real-time operation can be difficult
**DECSS Models**

**Complex Event Based Comms**
- Use clock drift, timing errors, and protocol to simulate traffic, collisions, delays, and over-flows
- Requires SimEvents, or other event/time triggered simulation block set
- Achieving real-time operation can be difficult

**Example:** Modeling a time-triggered digital network where one node has a slow clock and causes a collision with other traffic.
Component Testing - Legacy

Legacy systems have multiple complex analog interfaces specific to each component/vendor/design/implementation.
Distributed systems have a single simplified open digital interface with analog elements contained in the components.
DECSS Operations

![Diagram showing DECSS Operations with Net Book, Work Station, TCP/IP, VGA, and DECSS Station connections.](image)
DECSS Operations

Step 1

TCP/IP

Net Book

Work Station

DECSS Station

VGA
DECSS Operations

Step 1

Net Book

Work Station

TCP/IP

Step 2

.obj

.exe

VGA

DECSS Station
DECSS Operations

Step 1
- Net Book
- Work Station
- TCP/IP

Step 2
- .obj
- .exe

Controller
- CAN EADIN TTE
- Sensor
- Actuator

DECSS Station
- VGA
DECSS Operations

Step 1: TCP/IP - Work Station

Step 2: .obj .exe

Controller

Sensor

Actuator

CAN EADIN TTE

DECSS Station

VGA

Net Book

Approved for Public Release

Slide 23 of 32
DECSS Operations

Step 1

Work Station

Net Book

TCP/IP

Step 2

.obj
.exe

Step 3

Controller

CAN EADIN TTE

Sensor

Actuator

DECSS Station

VGA
DECSS Operations

Step 1
- Work Station
- TCP/IP

Step 2
- .obj
- .exe

Step 3
- DECSS
- Station
- VGA

Step 4
- Controller
- CAN
- EADIN
- TTE
- Sensor
- Actuator
DECSS Operations

User Interface

TCP/IP

Net Book

Work Station

TCP/IP

.obj

.exe

CAN EADIN TTE

Controller

Sensor

Actuator

VGA

DECSS Station
DECSS Operations

User Interface

TCP/IP

.obj

.exe

Net Book

Work Station

TCP/IP

CAN EADIN TTE

Controller

Sensor

Actuator

VGA

DECSS Station
DECSS Operations

User Interface

TCP/IP

.obj .exe

UDP

Proprietary Model

CAN EADIN TTE

Controller

Sensor

Actuator

Net Book

TCP/IP

Work Station

TCP/IP

VGA

DECSS Station
DECSS Operations

User Interface -> TCP/IP -> Environmental, Health & Deep Interrogation

Net Book -> TCP/IP

Work Station -> TCP/IP

User Interface

TCP/IP

UDP

C-MAPSS40K Plant

Controller

CAN EADIN TTE

Actuator

DECSS Station

Proprietary Model

Plant input = Actuator output

Plant output = Sensor input

Pilot/Airframe input

 Rolls-Royce

Approved for Public Release
DECSS Operations

User Interface

Environment, Health & Deep Interrogation

Engine Plant Model (EPM)

Plant input = Actuator output

Plant output = Sensor input

Actuators
Control Laws
Sensors

NETWORK

Control System Platform (CSP)

Pilot/Airframe input
Summary

- DECSS provides friendly and capable Simulation and HIL test environment
  - Includes multiple base models to support system development
  - Monte Carlo testing on simulated system reduces risk
  - Real-Time HIL
    - Identifies critical limits (of network characteristics)
    - Demonstrates compatibility of sub-systems
  - Distributed system architecture simplifies interfaces
  - Flexible network modeling capability with adjustable corruption parameters
  - Real-time HIL testing of smart components

**DECSS is well suited to support Distributed Engine Control System Development**
Acknowledgements

- NASA Glenn Research Center (GRC)
  - Controls and Dynamics Branch
- Air Force Research Laboratories (AFRL)
- Distributed Engine Controls Working Group (DECWG)
Thank you