Rapid Control Prototyping and Real-time Simulation of Distributed Embedded Systems with OpenSim::ModelPartition
Modern Automotive Systems

- Implemented over Distributed Architectures
- Multiple ECU’s
- Communication via CAN, network broadcast buses…

Diagram:
- Engine Control Unit
- Transmission Control Unit
- ABS
- Interface
- Instrument Panel
- SAE J1850
- CAN
Modern Automotive Systems

Reference: Car picture courtesy Adobe PDF presentation on the design of distributed control systems, found on the world wide web.
Why change current process?

Confluence of market pull and technology push forces change

Market pull

Current Process

Development costs

Technology push

- Model based design approaches
- Advances in Rapid prototyping
- Autocode generation for ECUs

“The pressure grows daily for Automotive Engineers to deliver new systems and products cheaper, better, and faster …”
Model-based control design

Control System Problem

- Physical Models
  - Process Model
  - Sensor Model
  - Actuators
  - A/D and D/A
  - Delta P, Position

- Code Generation
  - Auto Code
  - Embedded S/W
  - Real-time Link
  - ...

- Embedded H/W
  - Hardware-in-the-Loop
  - Rapid Prototype

- Control System Design
  - MPC
  - Adaptive
  - Nonlinear
  - Optimization
  - ...

- Control System Analysis
  - Performance
  - Stability
  - Execution
  - ...

- Design & Simulation
  - Non Real-Time

- Data Analysis
  - Animation
  - Plots
  - Numerical
  - ...

- Visualization
  - Real-Time

H/W-in-the-loop
To realize all the benefits of Model Based Control design in a real vehicle like distributed environment, a designer needs:

- Low Cost, Scalable, Open environment using COTS components
- Easily upgradeable systems to minimize product obsolescence risk
- A tool to go rapidly and effortlessly from off-line simulations to HIL simulations and/or rapid control prototyping of distributed systems
- No coding (especially for distributing the model to multiple real-time target or setting up the communications such as in CAN)
How to Lower Costs?

✧ Use **open architecture systems**: PC104, PCI, EBX etc or use ECUs (such as Motorola MPC5xx family), directly - Over **150** vendors manufacture compatible products

✧ Use ‘No Frills’ interfaces (**OpenSim:ModelPartition**) that are easy to use and cost less

✧ Do not overbuy on capacity, you may never use it

✧ Lower per unit (test stand) costs – Better utilization of resources

✧ Use easily upgradeable systems to minimize product obsolescence risk
Advantages of Open Architecture

- **Scalability** – Modular architecture allows you to add processors and I/O devices as your simulation needs increase
- **Openness** – Systems are built using commercial-off-the-shelf hardware and software, supporting Matlab/Simulink/RTW
- **Cost-Effectiveness** – Using standard single-board PCs, our solution is cost-effective without compromising performance
- **Flexibility** – The I/O devices supported include numerous analog/digital and data-bus interfaces
- **Reconfigurable** – Same system can be used for many different projects
OpenSim:ModelPartition enables seamless transition from design & simulation to rapid prototyping, hardware-in-the-loop testing, and ECU implementation in a vehicle like distributed environment with CAN communication.

OpenSim:ModelPartition enables you to develop a clean interface between your models and vendor provided models as shown in the above figure and also test the CAN network.
OpenSim:ModelPartition: Features

- Model partitioning system developed by Pathway Technologies Inc
  - Partition and distribute a Simulink model to multiple targets from one interface without writing a single line of code
  - Wide variety of virtual instruments with a Drag ‘n Drop interface
  - Intuitive tree view of signals and parameters from all targets in one GUI
  - Real-time visualization and tuning for faster “in-process” development
  - Collect data from all targets on the host PC for post-processing
  - Control model download and target execution
  - Can be run from a remote location
OpenSim:ModelPartition

- GUI for Real-time visualization
- Data logging and off-line analysis

Simulink™

Model Partition Utility

RTW™/Embedded Coder (Visual C++ 6.0, 7.0, .NET)

C Compiler

Target Management

Communication Layer

Target 1

TCP/IP
CAN
CCP
...

Target 2

Communication

I/O

- OpenSim:ModelPartition
Custom GUIs

Pathway Technologies Incorporated
Rapid prototyping & Hardware-in-the-loop

Pathway Technologies Incorporated

- Control System Problem
- Physical Models
  - Process Model
  - Sensor Model
  - Actuators
  - A/D and D/A
  - Delta P, Position
- Control System Design
  - MPC
  - Adaptive
  - Nonlinear
  - Optimization
- Control System Analysis
  - Performance
  - Stability
  - Execution
- Data Analysis
  - Animation
  - Plots
  - Numerical
- Code Generation
  - Auto Code
  - Embedded S/W
- Embedded H/W
  - Hardware-in-the-Loop
  - Rapid Prototype
- Non Real-Time
  - Design & Simulation
  - Non Real-Time
- Real-Time
- Visualization
- H/W-in-the-loop
Rapid Prototyping

Seamlessly move from Matlab/Simulink® models to hardware platform without having to write a single line of code

Target Support in

OpenSim:ModelPartition

✓ xPCTarget, Mathworks
✓ MPC555, Motorola
✓ MPC565, Motorola
✓ Custom ECU targets
✓ …
Drag and Drop for Rapid Control Prototyping/HIL Testing

Step 1: Open your Simulink model

If dlm or flash files are available, go to Step 6 and download code from GUI

Ref: Power Window Control model from Simulink Demos, The Mathworks.
Step 2: Insert CAN part blocks to partition your model into subsystems that will run on different targets.

Step 3: Start the model partition process to bring up the target selection GUI.
Step 4: Select the target to which you want to build and download a specific partition

Step 5: Specify parameters for communication between the targets
C code for each partition is generated by Real-time Workshop or Embedded Coder

This C code is downloaded to the respective targets

Host Computer
(Runs OpenSim:ModelPartition)
Communication layer between the target’s and the host PC is provided by OpenSim:ModelPartition

Step 6: Launch the monitoring GUI

If using dlm or flash files, download code to the targets else go to Step 7

Step 7: Launch the communication layer components

Step 8: Prepare a custom virtual display for real-time visualization of signals by selecting widgets from the widget library provided with OpenSim:ModelPartition

Start model execution, monitor your system, tune control parameters and/or log data for post processing, etc ...
Example Application: Dynamometer Testing

Pathway Technologies Incorporated

Host Computer

TCP/IP

xPC Target

TCP/IP

xPC Target

CAN

Reference data and Alarms

Dynamometer controller

Interface-box

Dynamometer

ENGINE

Test cell controller
Contact Information

Headquarters
510 Townshipline Road, Ste. 110
Blue Bell, PA 19422
Phone 215-542-4925
Fax 775-490-2068

Michigan Sales Office
17197 N. Laurel Drive, Ste. 120
Livonia, MI 48152
Phone 734 432 1006

Web: www.pathwaytechnologies.net