Crossover to Model-Based Design

Ben Ciavola

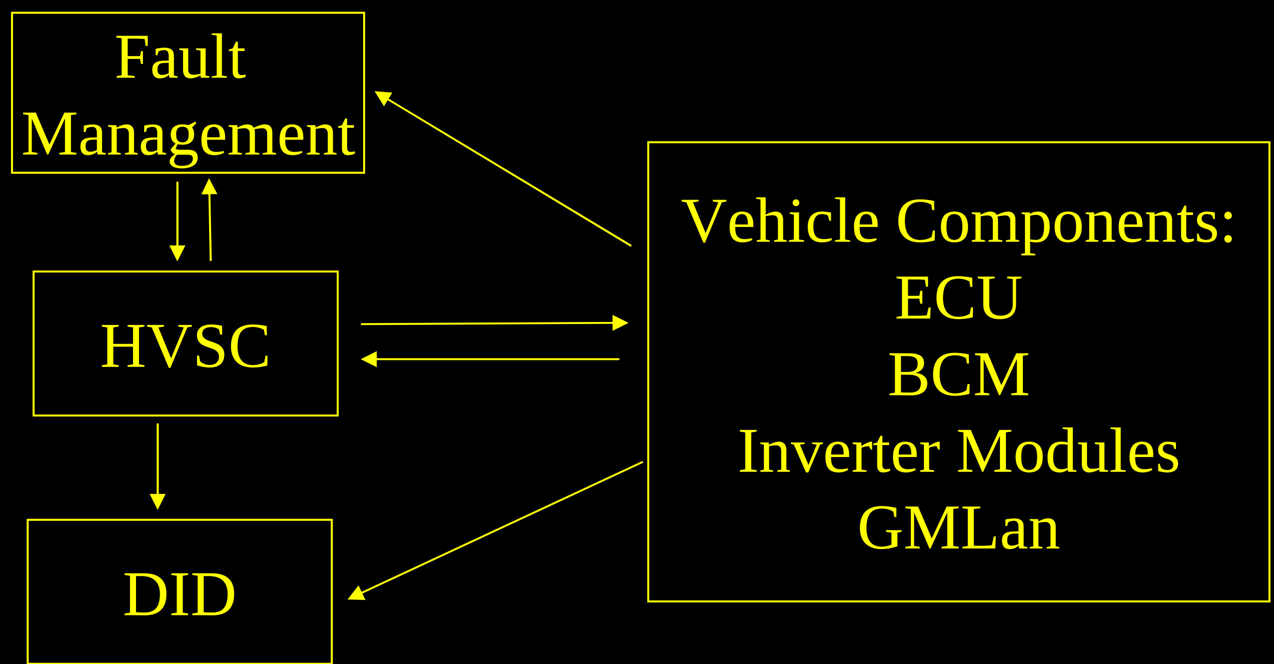
Model-Based Design Applied to Vehicle Development Process (VDP)

- MathWorks Tools:
  - Used Throughout the Development Process
  - Drive the Use of All Other Software Tools
  - Power-Split Design Impossible Without The Tools
  - SimDriveline, Stateflow, SystemTest, Distributed Computing Toolbox

Tools Used Throughout VDP

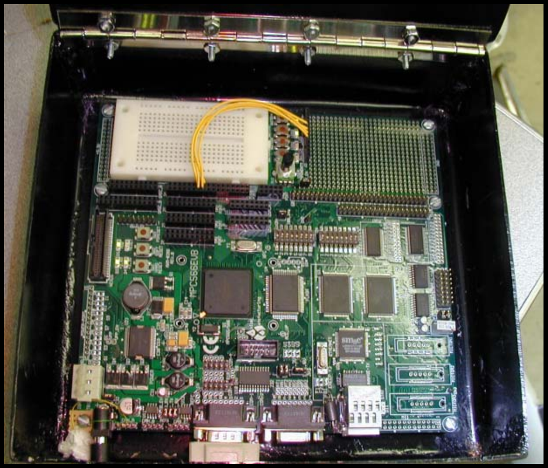
- Basic Physical Understanding
- Component Modeling
- Plant Modeling
- Controller Design
- SIL, HIL
- Implementation on Target (MotoHawk)
- Controller Tuning
- Data Collection (ISDL) and Analysis (Matlab)
- Verification and Validation

Overall System Block Diagram



Fault Management System (FMS)

- MPC566EVB
- Observe Vehicle for Faults
- Notify HVSC if Problem is Detected
- Detect Special Fault Conditions
  - Broken Shafts
  - Engine Alive
  - Component Alive
  - Over Temperature



Application of Model-Based Design

- Basic Physical Understanding, Component Modeling, Plant Modeling
  - SimDriveline allowed for rapid development and adaptation of physical model
  - Enabled control system adaptation to take place in a matter of weeks after complete drivetrain architecture change

Application of Model-Based Design

- Controller Design
  - Ability to simultaneously develop controller and vehicle model allowed for easy modification of control architecture
  - Immediately see effects of control system changes

Application of Model-Based Design

- HIL
  - Compiling of auto-generated code to multiple targets enabled easy use of multiple real-time systems
    - Custom Labview DLL wrapper for communication
    - MPC566 real-time diagnostics
    - MotoTron MotoHawk for years 3/4 in-car HVSC

Application of Model-Based Design

- In-car tuning of controller
  - Calibration, override, and probe capabilities of MotoHawk in Simulink enabled tuning of numerous speed-based gain schedules
  - Fast, robust system for making in-vehicle controller changes

Application of Model-Based Design

- Data Collection and Analysis
  - ISDL Programmed with Simulink/MotoHawk
  - Data Converted with Matlab
  - Data Analyzed with Matlab
  - Software-in-the-Loop simulation allowed rapid drivecycle development for model to vehicle performance comparisons

Application of Model-Based Design

- Verification and Validation tools
  - SystemTest
    - Enabled one-click testing of control system for large number of initial conditions and configurations
  - MonteCarlo analysis
    - Seamless interoperability of Matlab and Simulink allowed for custom systemtest-like application development

Plant Modeling

- Basic and Advanced Features Applied to Plant Model
  - Extensive Use SimDriveline
  - Matlab Init File used to specify Parameters
  - Matlab Post-Processing File to Analyze and Summarize results

Plant Modeling

- Features to Allow Fast Execution
  - Simulink Accelerator
  - Look-up tables for modeling component behavior
  - Distributed testing environment using Distributed Computing Toolbox (still) in development
    - RHIT cluster of 10 dual-processor nodes
    - Available for modeling-intensive EcoCar Year 1

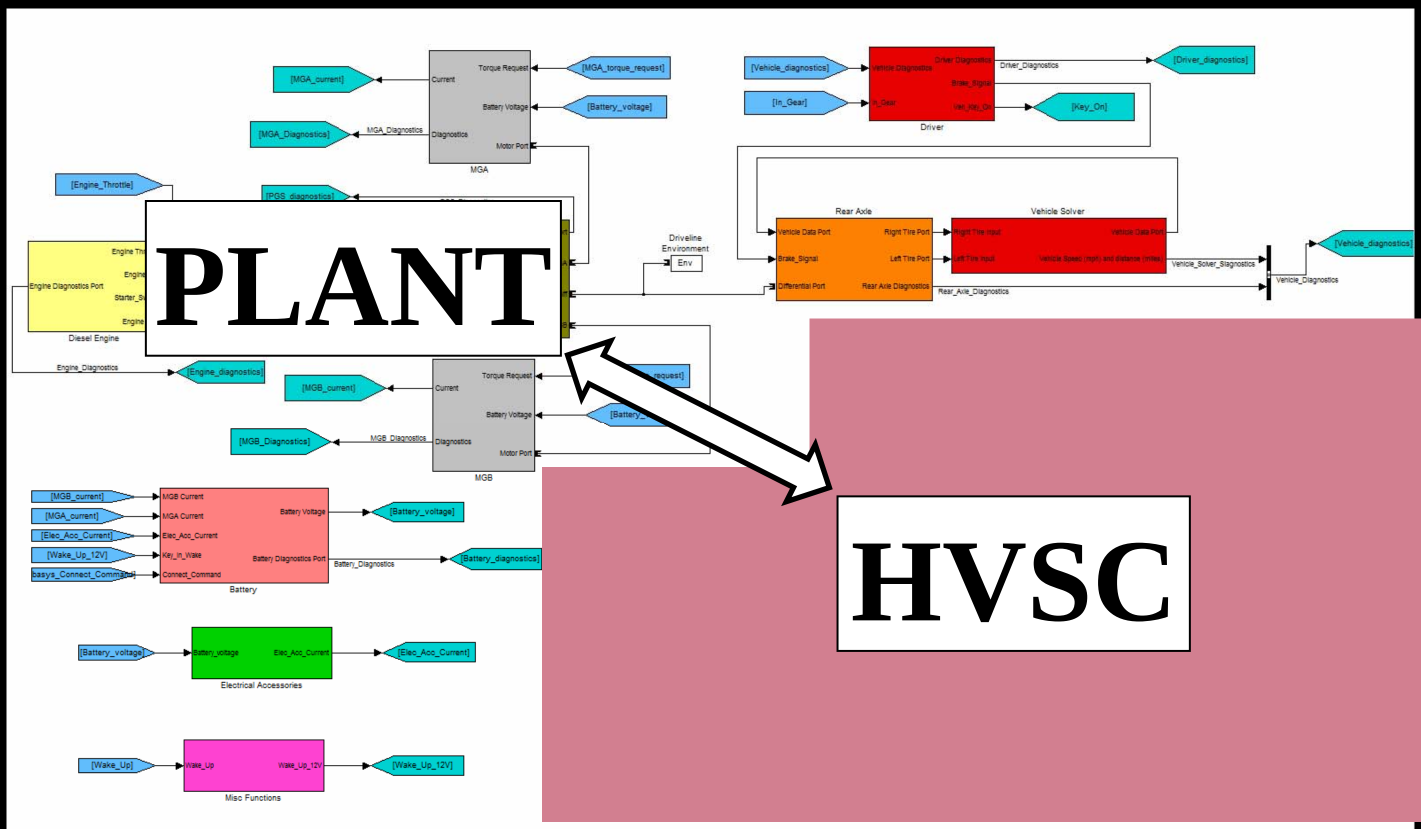
Controller Modeling

- Developed entirely in Simulink
- Real-Time Testing
  - PXI Real-Time Simulations
  - Network Level Simulation (MPC566/CAN)
- In Vehicle Testing.
- Data Collected and Analyzed
- Test Plan to Verify Stateflow Logic

Controller Modeling

- Ease of Use and Understanding:
  - Components Split into Subsystem Blocks
  - Controller Functions grouped by Function and placed in Subsystems
  - Logic implemented with Stateflow
  - Continuous Functions implemented with Simulink Blocks
  - Model Designed Modularly to promote Ease of Use and Understanding
  - Others

Application of Model-Based Design Philosophy

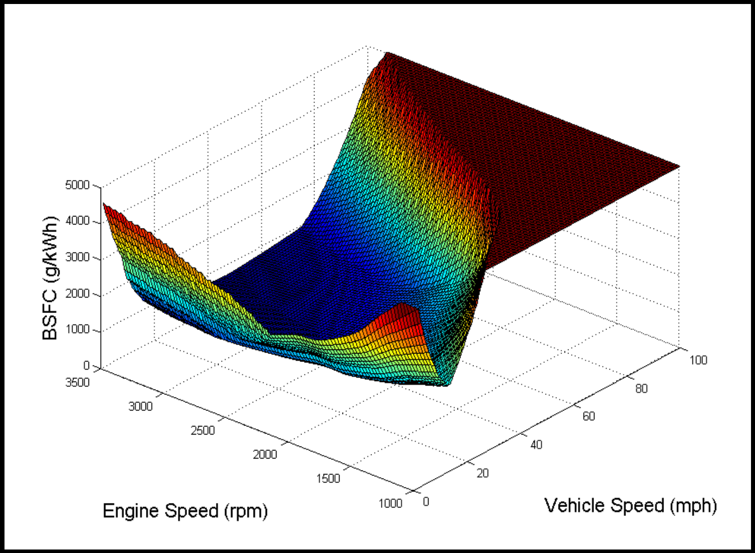


Modeling Improved Understanding

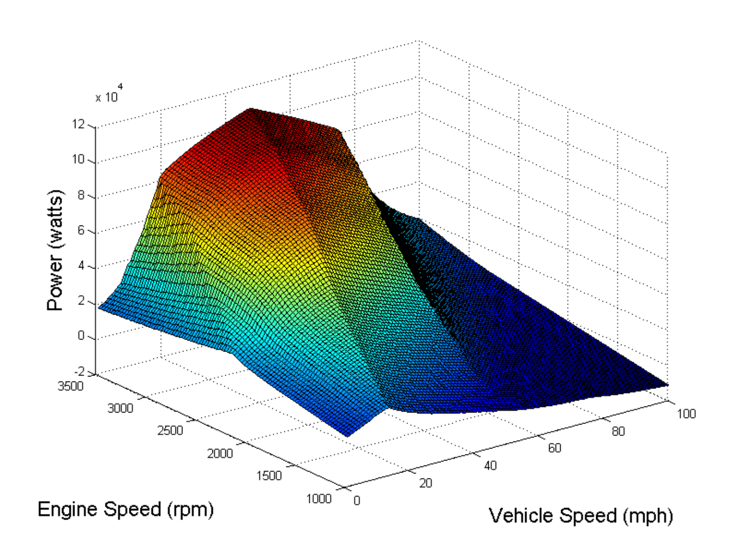
Development of the model allowed team members to come to an intuitive system-level understanding of vehicle operation.

This resulted in increased sophistication of optimization efforts.

Brake-specific fuel consumption

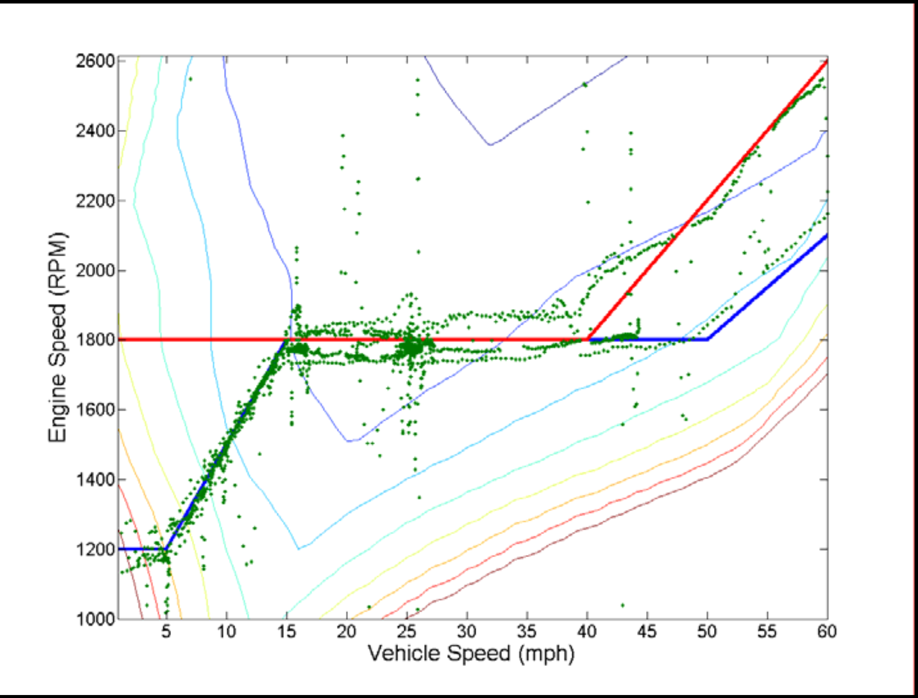


Maximum power output



Data Analysis

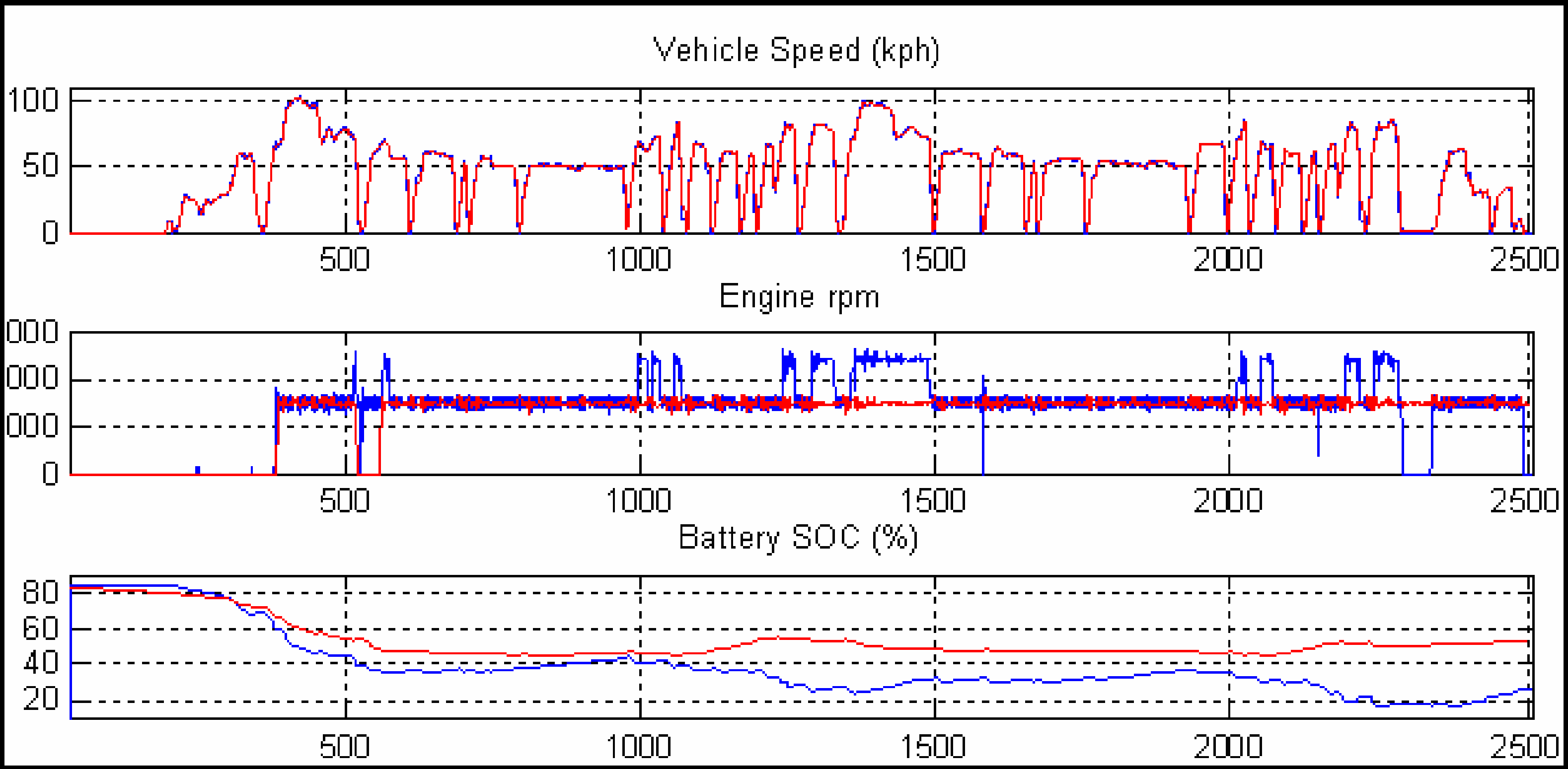
- Matlab Used to Enable Data Aquisition
  - ISDL Programmed With Simulink/MotoHawk
- Data Logged By Motohawk ISDL for later retrieval
  - Collected ISDL data post-processed with Matlab
  - Matlab Used to Visualize Results



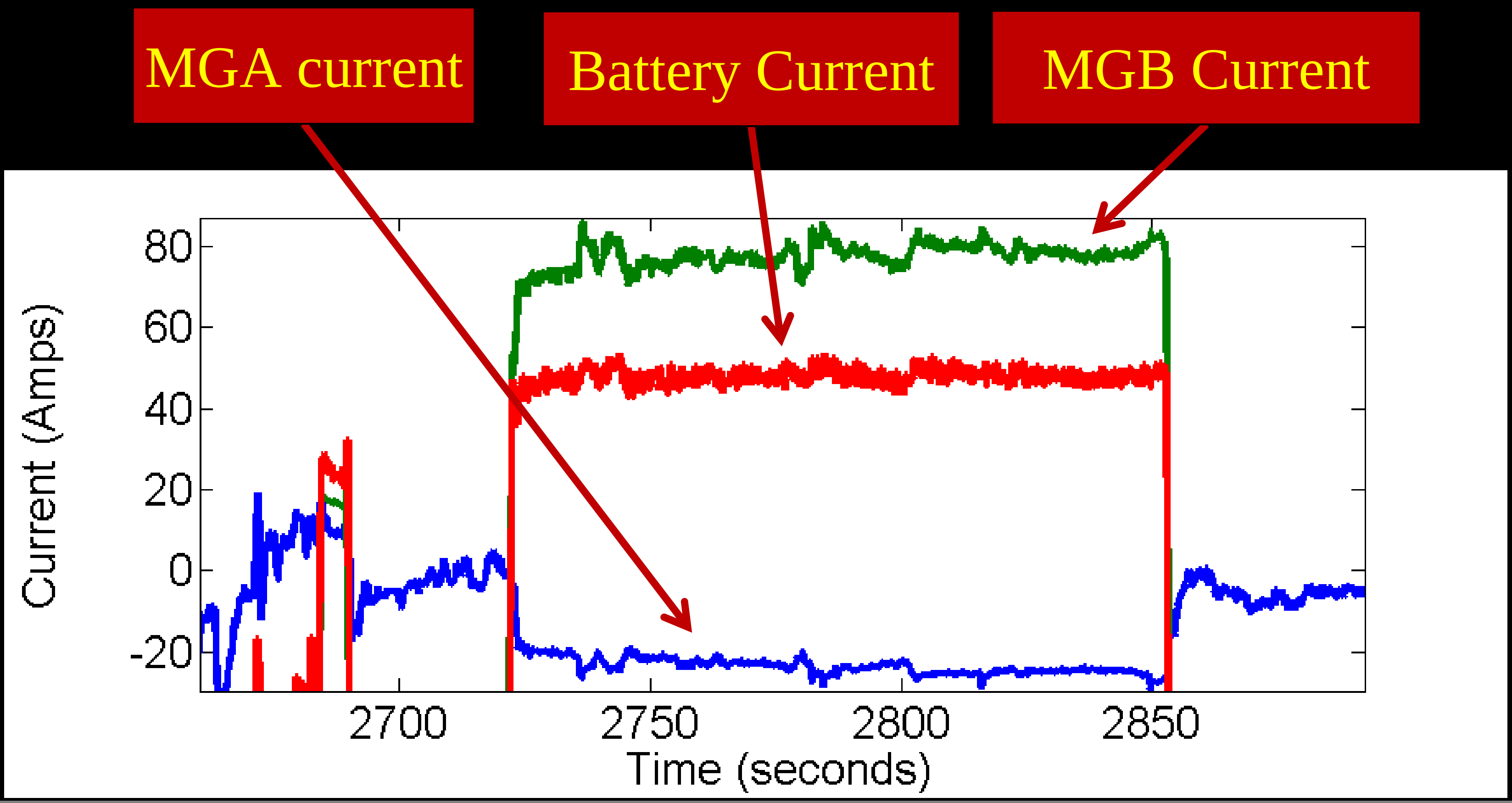
Plant Verification

- Run drivecycles from collected data, make comparisons to vehicle performance

Red: Simulated Blue: Actual Car Data



Controller Verification

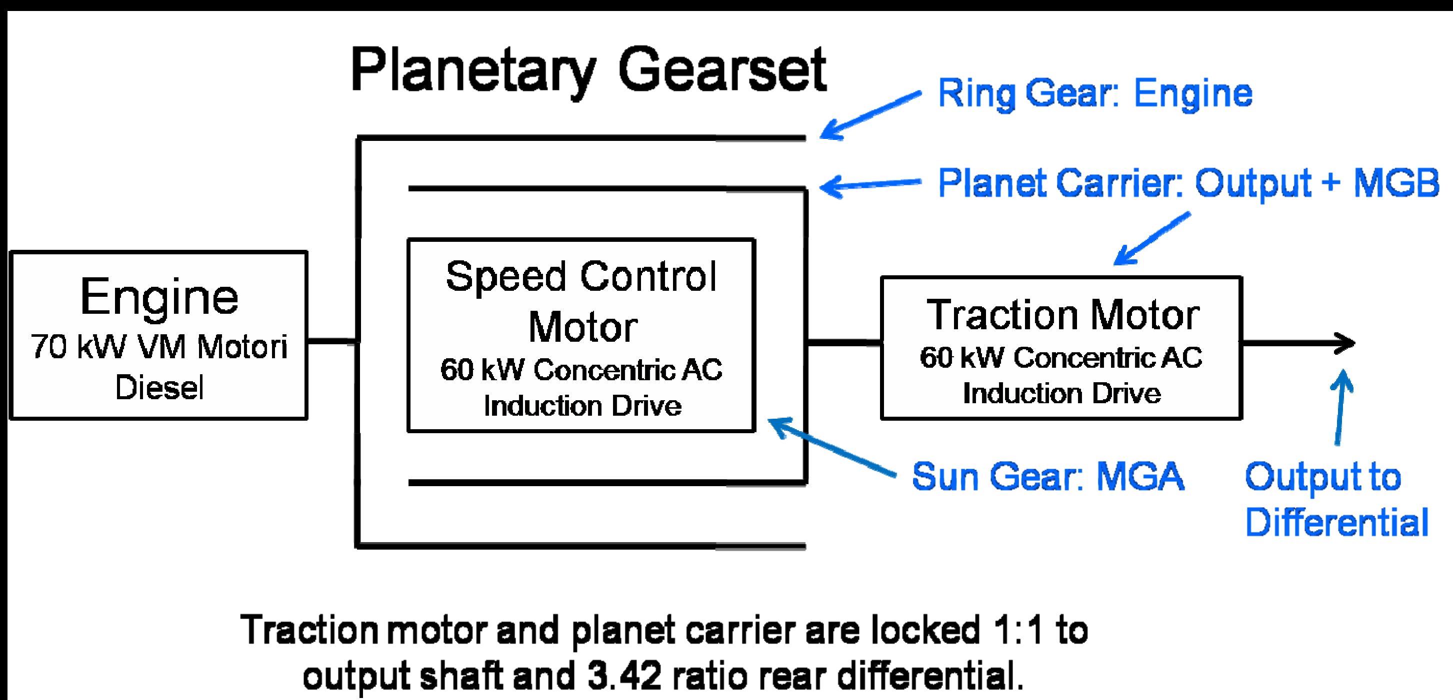


Technology Transfer

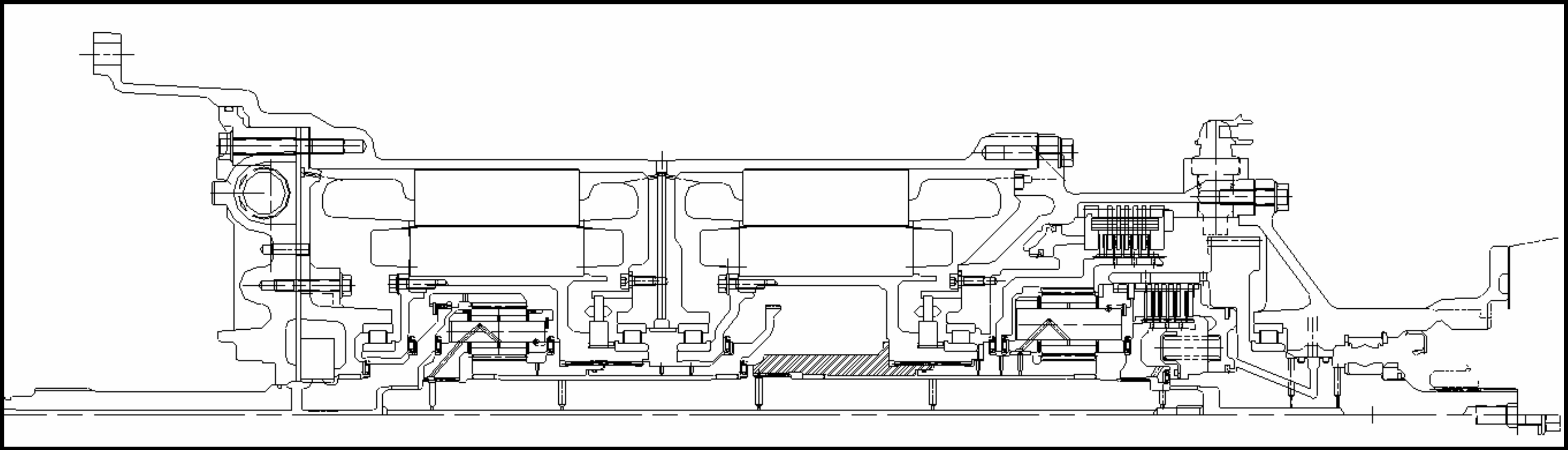
From the Competition to The Classroom

History

For the ChallengeX Competition, Rose-Hulman has successfully deployed a powersplit architecture



History



The complexity of this system required extensive component and plant modeling to develop a robust controller. This is the **most important step**!

Technology Transfer

- Rose-Hulman has used the Challenge X competition as a vehicle to:
  - Update existing classes
  - Create new classes
  - Create new laboratories
- These improvements move beyond the scope of Challenge X and are available to all students

Technology Transfer

- Areas impacted include
  - Project Management Methods
  - System Modeling
  - Real-Time Control
  - System Design from Concept to Implementation and Testing

Technology Transfer

- New Courses Created
  - Short course
    - 2-day seminar
  - Introduction to Model-Based System Design
    - Junior Level
    - Pre-Requisite: Differential Equations
  - Advanced Model-Based System Design
    - Senior Level
    - Pre-Requisite: Intro Course

MBSD Teaching History

- Rose has provided three MBSD workshops to interested CX participants
- During the third year of competition, a two day MBSD short course was required of all team members
  - Only half showed up
  - REV and UM ChX attended

Introduction to MBSD Course

- Model-Based Design for a small system
- Model Design and Simulations
- Real-time simulations
- Introduction to Digital Systems
- RAppID Toolbox
- Deploy controller on MPC5553/4 target
- PIL Real-Time Simulations
- Test controller on physical system
- Model Verification
- DoE - Improve Motor and Generator Models
- Model Refinement and Re-Verification

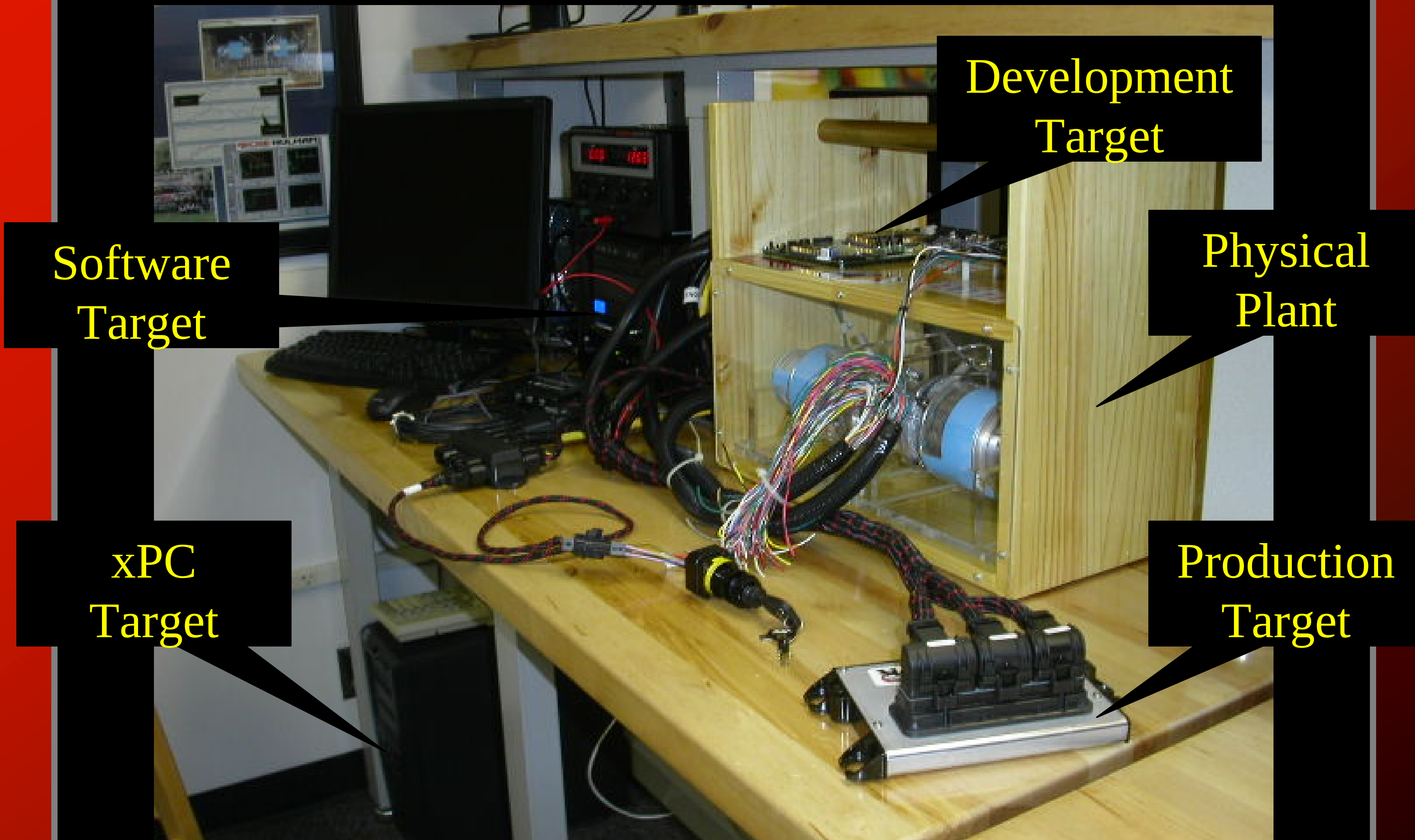
Advanced MBSD Course

- Modeling a Series Hybrid-Electric Vehicle
- Introduction to CAN
- Introduction to MotoHawk
- Controller Deployed on MPC555
- Plant Deployed on MPC555 (Band-Aid)
- Sensors and Interfacing (Driver Controls)
- Data Collection

Technology Transfer

- Existing Courses Affected by MBSD
  - ME Senior Design
    - Control of PTO driven CTV
    - Hybrid vehicle optimization
  - Machine Design
    - Dynamic response of clutches, brakes, and shafts
  - Conservation and Accounting Principles
    - Dynamic response of first principle systems

Challenge X Hybrid Vehicle Project



Conclusions

- MathWorks tools essential to our development cycle
  - Ability to bypass low-level first principles modeling with SimDriveline speeds up development
  - Automatic Code Generation indispensible
  - High-Level tools allow focus on development vs. implementation
  - Easy model comparison with real-world data
- Give us Discretionary Points!