GM's Model-Based Design Methods Facilitate Global Development

Karla Berger
Sr. Manager Controls Systems
General Motors Powertrain Europe

GM Powertrain Globally

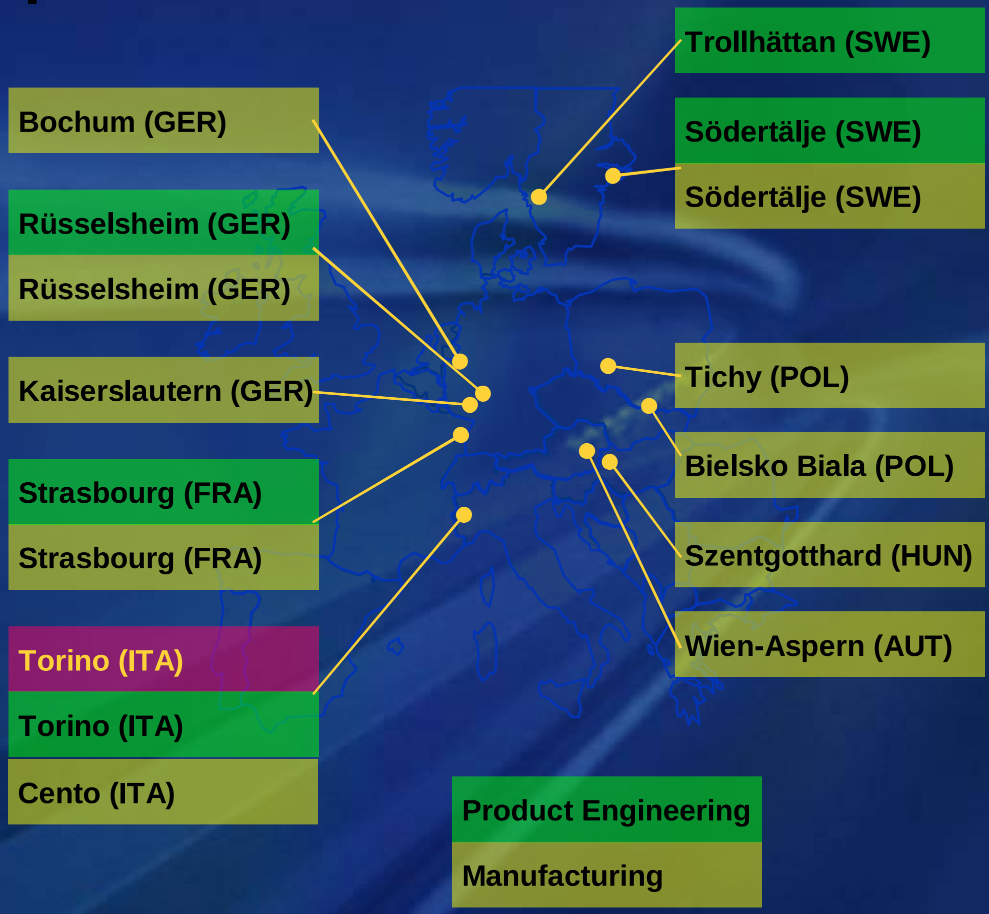
- 89 facilities
  (Engineering and Manufacturing)
- 14 countries

- Over 59,000 people
- 36,000 engines/day
- 32,000 transmissions/day

GM Powertrain Europe

Global Lead Responsibilities:

- Small diesel engines
- Diesel controls
- Automated Manual controls
- Small gasoline engines
- Manual transmissions



GM Powertrain Engineering Torino

- Diesel Engine Engineering
  - Computer Aided Engineering (CAE) & Simulation
  - Diesel Engine design & testing
  - Exhaust Aftertreatment
  - Turbocharging
  - Engine Management System Components
- Controls Engineering
  - Engine/Transmission Control Algorithms
  - Diagnostics
  - *GM is the only auto maker with complete software and systems expertise in control of gasoline & Diesel engines and transmissions*
- Hybrid Powertrain Engineering
  - System configuration modelling & optimization
  - HEV powertrain design and testing

New Torino Facility

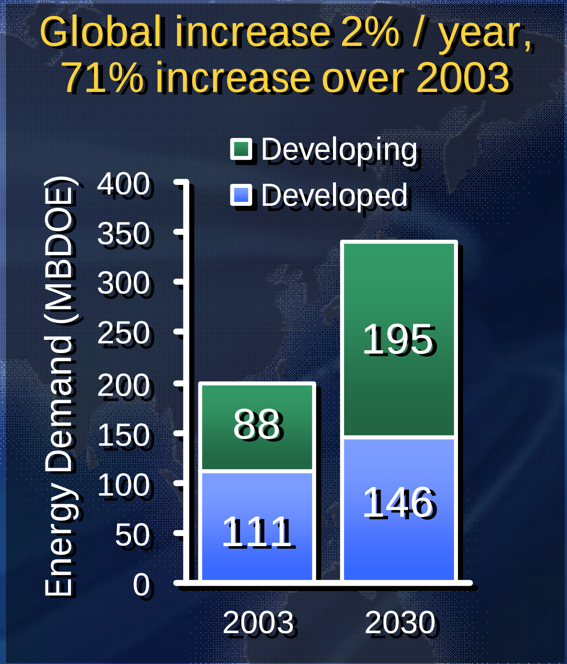
- Innovative cooperation with Politecnico di Torino

Automotive Control System Constraints

- Fuel Economy
- Emissions
- Performance
- Quality
- Reliability
- Verification
- Durability
- Time to Market
- Ease of Tune-ability
- Scale-ability
- Workforce Training
- Validation

The Challenge

- Global energy demand exceeds our current glide path for supply
- There are several risks that can disrupt the existing supply
  - Above ground infrastructure
  - Natural disasters
  - Wars
  - Hostile regimes
- Growing concern about Global Warming due to CO_2
- Potential for regulations that exceed both technical capability and business feasibility

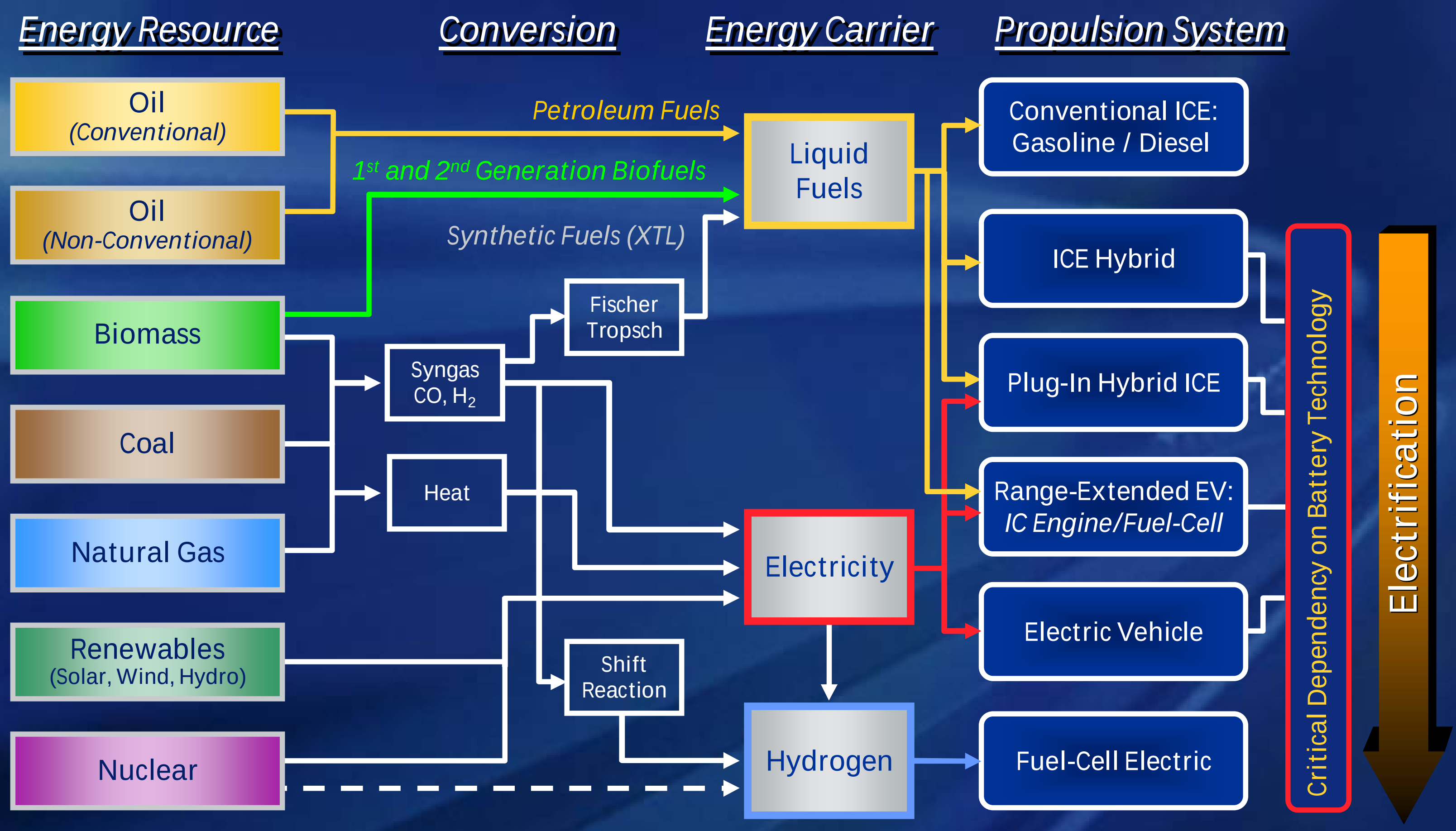


Source: DOE Energy Information Agency

MBDOE: Millions of barrels per day oil equivalent

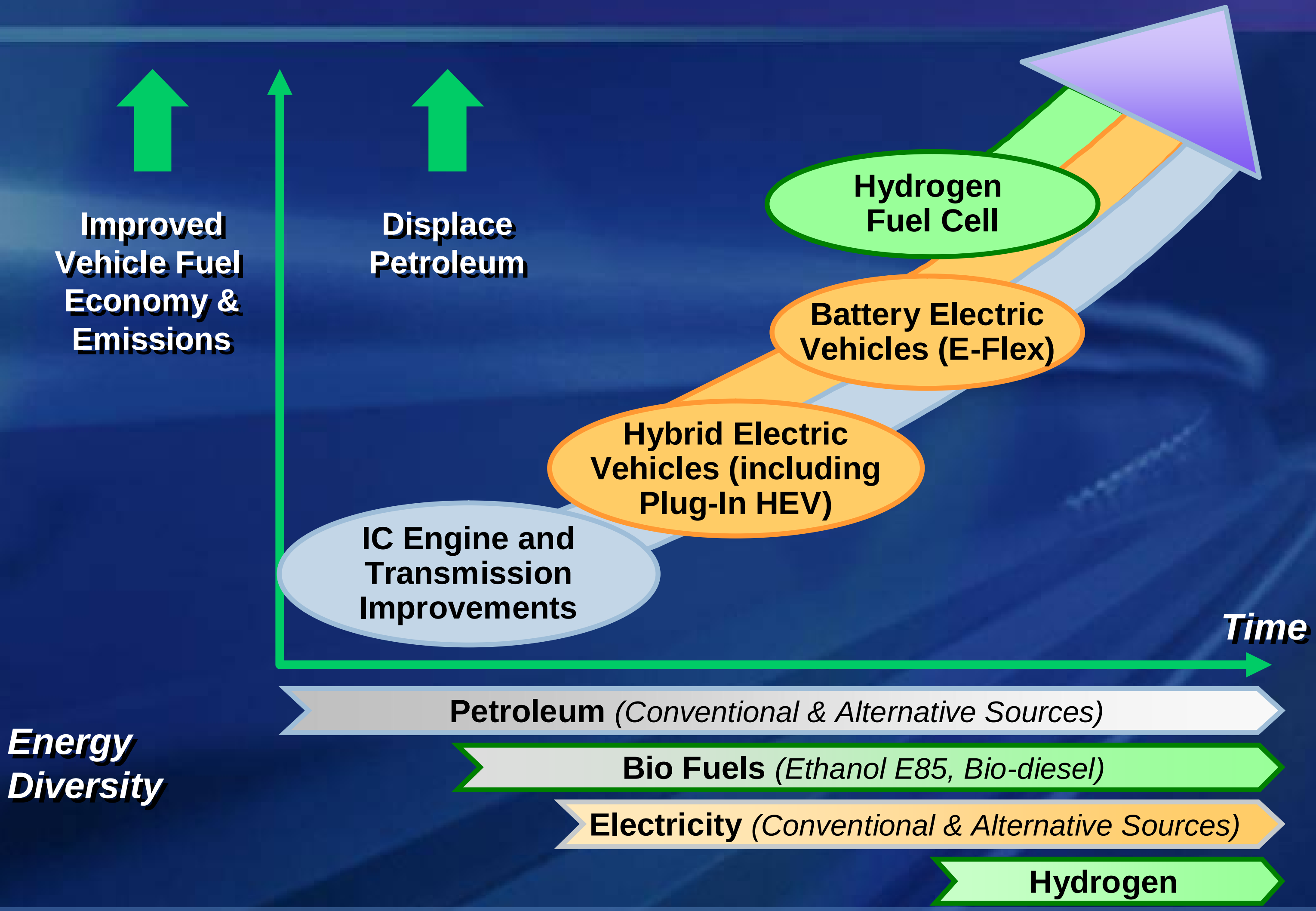
Energy Diversity

Liquid Fuels / Electricity / Hydrogen as the In-Vehicle Energy Carriers



Application Map

Advanced Propulsion Technology Strategy



Model Based Design

Methods that facilitate global controls design to meet challenging automotive constraints and translate our strategies into products

Controls Development before MBD

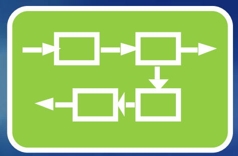
- Lengthy architecture trade off studies prior to start of control system design
- High reliance on hardware based on “cut and try” empirical test methods for development of algorithms
- Requirements captured in text based documentation
- Difficult to manage complexity and parallelism
- Algorithms and software implementation separated from requirements
- Verification and validation labor intensive

Controls Development after MBD

- Hardware and Controls design becomes integral to architecture studies
- Plant models used throughout design cycle at varying levels of fidelity
- Models capture the requirements for algorithm design and software execution
- Verification and Validation testing automation improved
- Key behavioral tests developed prior to hardware availability

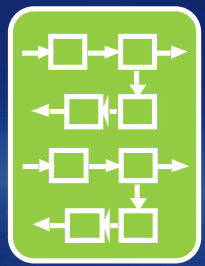
Modeling Integral to the Control Engineering Process

Concept ······ *Model elaboration* ······▶ Production System

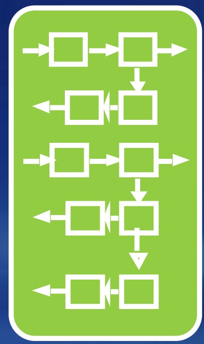


Models and Simulation

Executable Specification



Executable Specification



Automated:
- *code*
- *testing*
- *documentation*
- *calibration*

EI&S

Advanced Controls Engineering
- *Using GMPT Library*
- *Following GMPT model structure and guidelines*
- *Single/Boolean data typing*
- *Model descriptions added*
- *Model in 65% form*
- *Functional test cases*
- *Model checker tool used to*

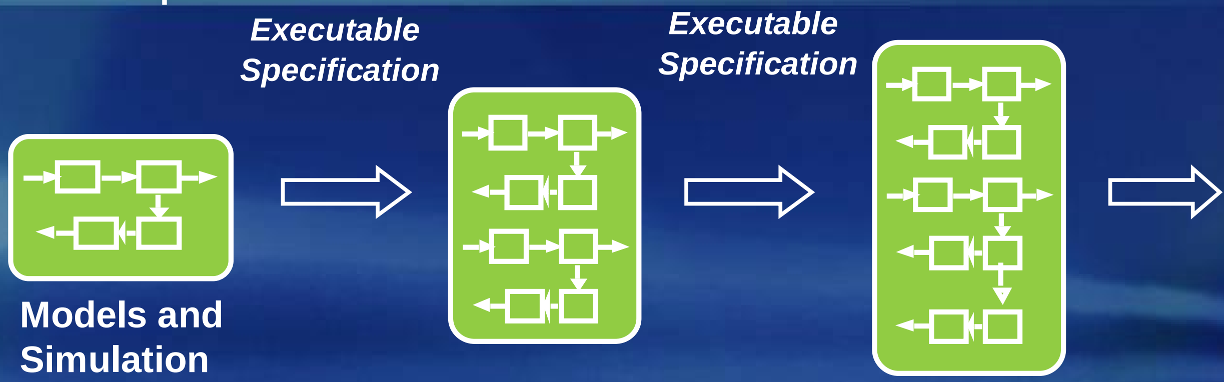
Algorithm Engineering
- *Finalize model structure (Adhere to naming standards, select interface parameters (Aspen))*
- *Add initialization logic*
- *Finalize all comments*
- *Run model checker tool*
- *Run Functional test cases and add additional test cases*
- *Generate ADD (Populate DD)*
- *Generate Reactis test cases*

Software Engineering
- *Add MPT to the model*
- *Configure settings for file structure and data scope*
- *Generate code*
- *Compile code*
- *Perform Code Inspection*
- *Run code checker tool*
- *Check-in to CM Synergy*

Software Integration
- *Test the System*
- *Execute Reactis test cases*
- *Verify Code-To_Model testing (CTM) using RIBETT*

Modeling Integral to the Control Engineering Process

Concept ······ *Model elaboration* ······> Production System



***Automated*:**
- *code*
- *testing*
- *documentation*
- *calibration*

EI&S

Advanced Controls Engineering
- *Using GMPT Library*
- *Following GMPT model structure and guidelines*
- *Single/Boolean data typing*
- *Model descriptions added*
- *Model in 65% form*
- *Functional test cases*
- *Model checker tool used to*

Algorithm Engineering
- *Finalize model structure (Adhere to naming standards, select interface parameters (Aspen))*
- *Add initialization logic*
- *Finalize all comments*
- *Run model checker tool*
- *Run Functional test cases and add additional test cases*
- *Generate ADD (Populate DD)*
- *Generate Reactis test cases*

Software Engineering
- *Add MPT to the model*
- *Configure settings for file structure and data scope*
- *Generate code*
- *Compile code*
- *Perform Code Inspection*
- *Run code checker tool*
- *Check-in to CM Synergy*

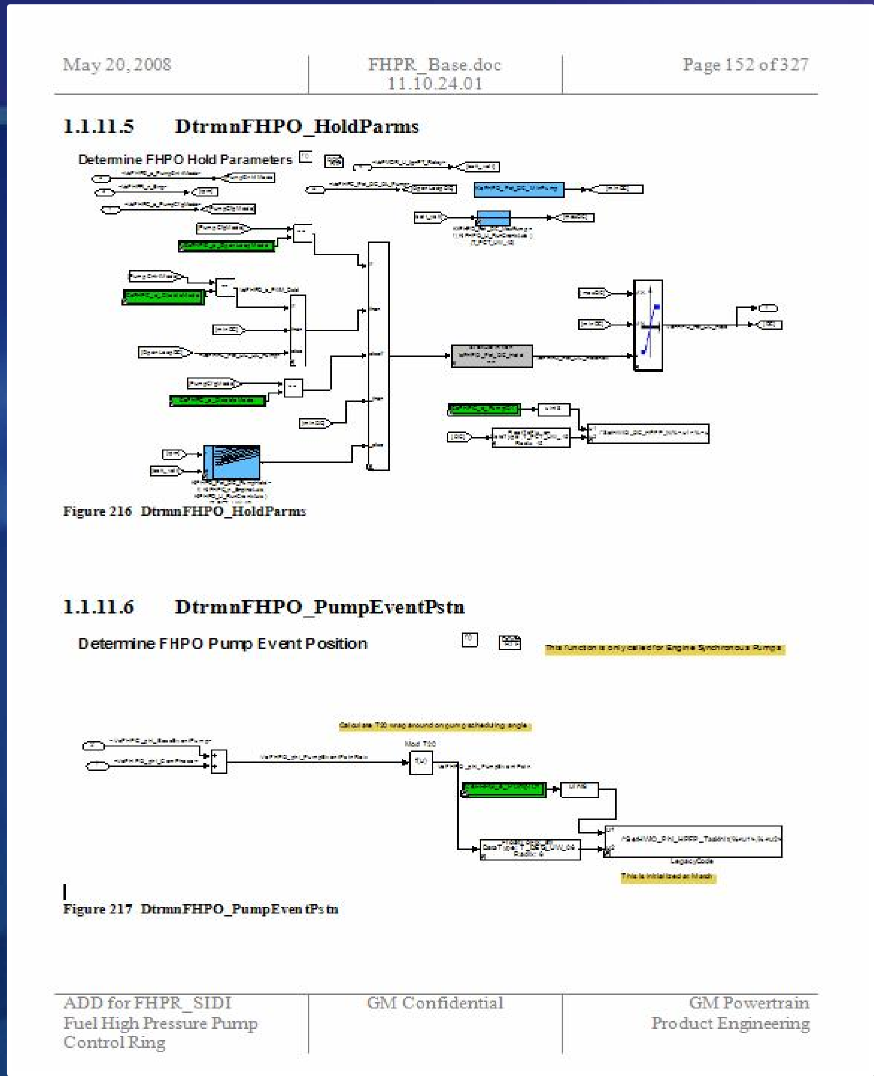
Software Integration
- *Test the System*
- *Execute Reactis test cases*
- *Verify Code-To_Model testing (CTM) using RIBETT*

Critical to have a consistent/stable tool chain throughout the process and life cycle of the program!

Algorithm Description Documents

Algorithm Engineering

- Algorithm Description Documents (ADDs) describe the control strategy to the level that production code can be produced
- ADDs for the core SIDI functions contain extractions of the Simulink/Stateflow algorithm models from advanced development


May 20, 2008 | FHPR_Base.doc 11.10.24.01 | Page 152 of 327

1.1.11.5 DtrmnFHPO_HoldParms

Figure 216 DtrmnFHPO_HoldParms

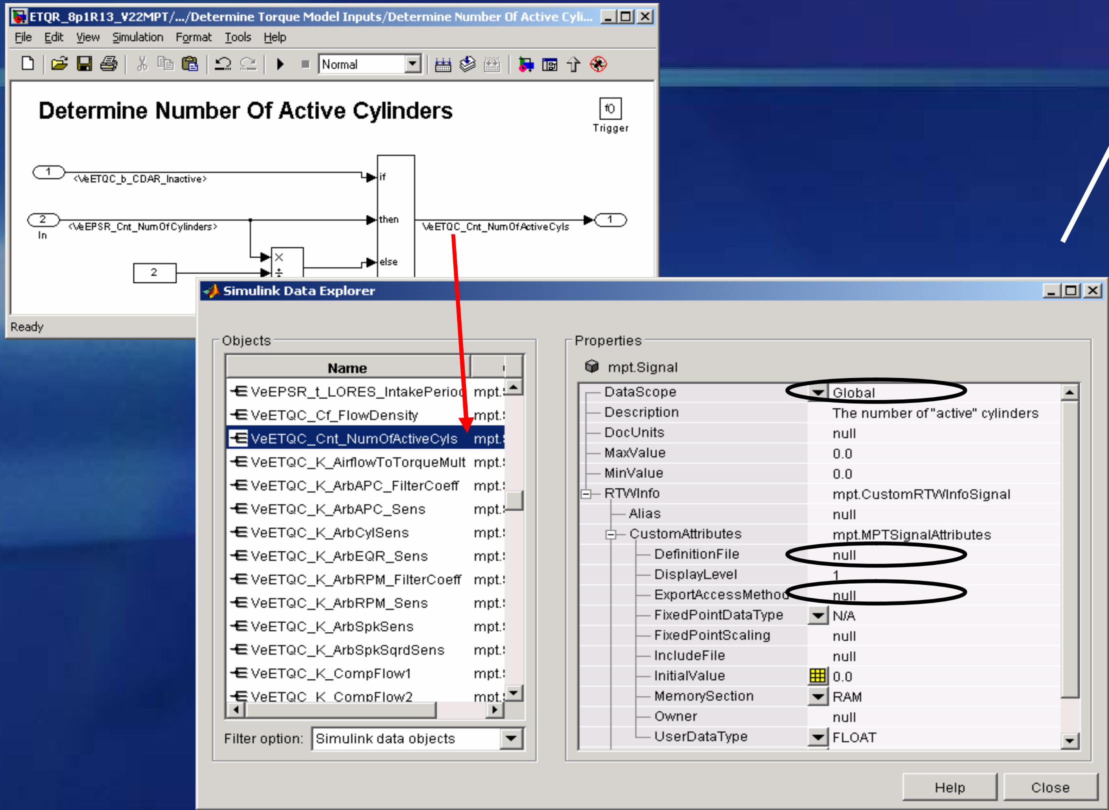
1.1.11.6 DtrmnFHPO_PumpEventPstn

Figure 217 DtrmnFHPO_PumpEventPstn

ADD for FHPR_SIDI Fuel High Pressure Pump Control Ring | GM Confidential | GM Powertrain Product Engineering

Production Code Generation

Software Engineering



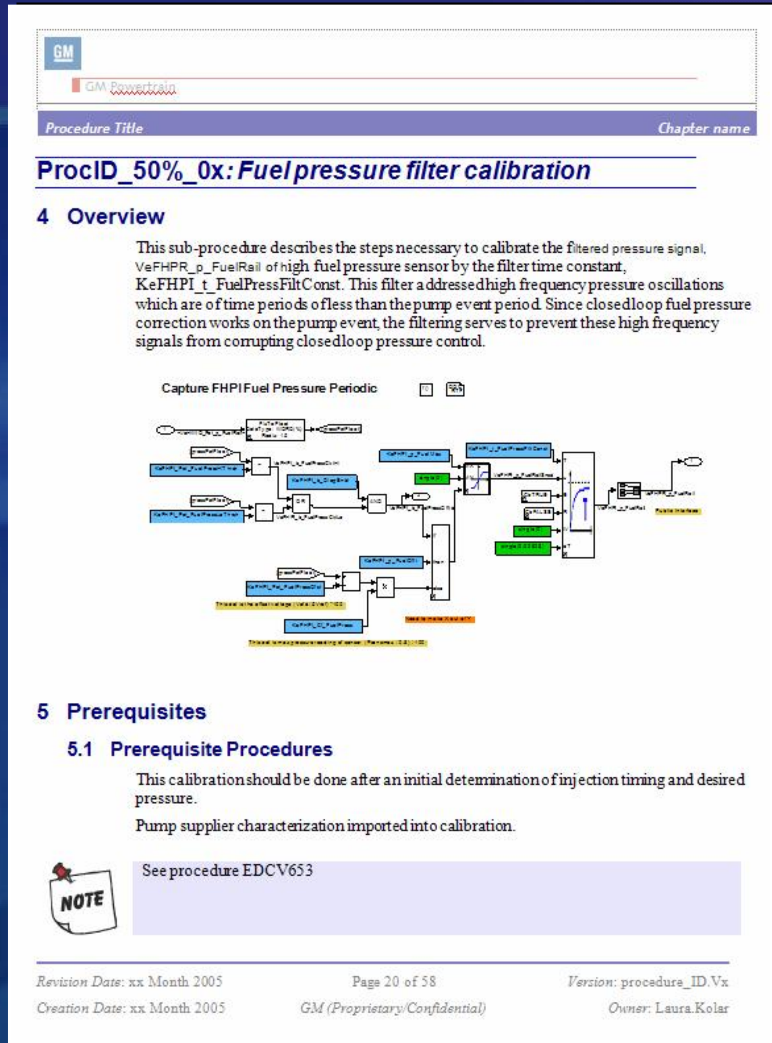
Production code for the core SIDI functions produced using RTW Embedded Coder

```
etqcmeqr.c
/*****************************************************
**************************
 * Inclusion of Other Header Files
*****************************************************
*************************/
#include "etqrsxlt.h"
#include "fulrpitf.h"
/*****************************************************
*************************/
/* Output and update for function-call system:
'<S9>/Determine Equivalence Ratio' */
void DtrmnETQC_EffectiveEQR(void)
{
  /* local block i/o variables */
  FLOAT rtb_Switch2_e;
  if ((GetFULR_FuelCntrlState() ==
CeFULR_ClosedLoop)) {
    VeETQC_eqr_EquivRatio = GetETQR_eqr_EquivRatio();
  } else {
    if ((GetETQR_OpenLoopEquivRatio() == 0.0F)) {
      rtb_Switch2_e = 1.0F;
    } else {
      rtb_Switch2_e = (GetETQR_eqr_EquivRatio() /
GetETQR_OpenLoopEquivRatio());
    }
    VeETQC_eqr_EquivRatio = rtb_Switch2_e;
```

Calibration Procedures

Integration / Documentation

- Calibration procedures are documents detailing the steps to follow in order to calibrate control strategy
- Calibration procedures fo the the core SIDI functions contain extractions of the Simulink/Stateflow algorithm models



GM Powertrain

Procedure Title | Chapter name

ProcID_50%_0x: *Fuel pressure filter calibration*

4 Overview

This sub-procedure describes the steps necessary to calibrate the filtered pressure signal, VeFHPR_p_FuelRail of high fuel pressure sensor by the filter time constant, KeFHPI_t_FuelPressFiltConst. This filter addressed high frequency pressure oscillations which are of time periods of less than the pump event period. Since closedloop fuel pressure correction works on the pump event, the filtering serves to prevent these high frequency signals from corrupting closedloop pressure control.

5 Prerequisites

5.1 Prerequisite Procedures

This calibration should be done after an initial determination of injection timing and desired pressure.

Pump supplier characterization imported into calibration.

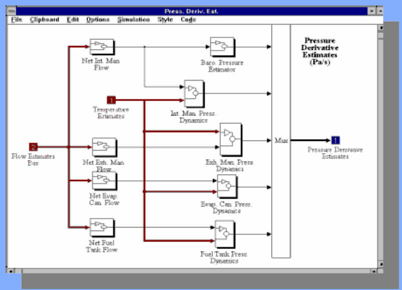
NOTE: See procedure EDCV653

Revision Date: xx Month 2005 | Page 20 of 58 | *Version*: procedure_ID.Vx

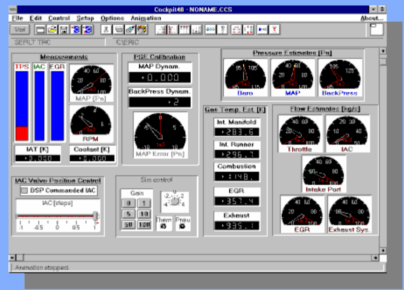
Creation Date: xx Month 2005 | *GM (Proprietary/Confidential)* | *Owner*: Laura.Kolar

Auto-Code Targeted for Rapid Prototyping Controller

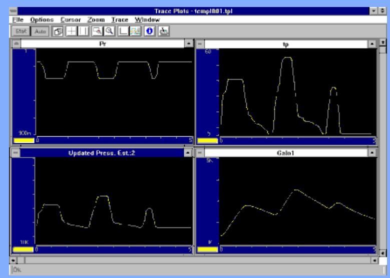
Algorithm Development | **Control/Monitor Interface** | **Data Logging Interface**



Matlab/Simulink/Stateflow



Control Desk



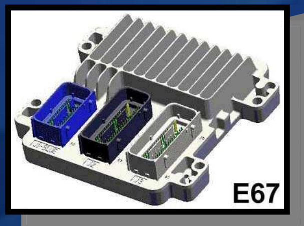
Control Desk Logging

Automatically Generated Code

PC

Rapid Prototyping Controller (dSPACE Micro-Autobox)

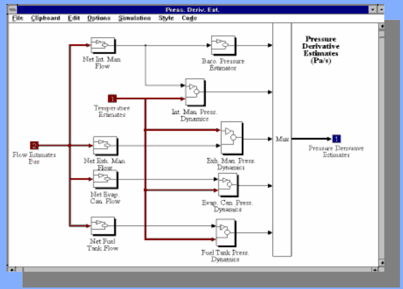
Dual-Port RAM Controller Interface



Production ECM / TCM

Auto-Code Targeted for Production Controller Hardware

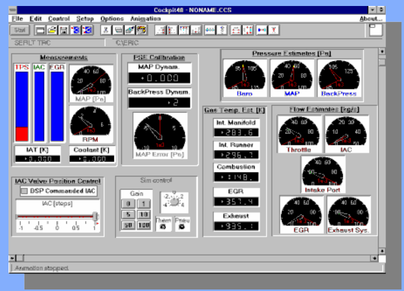
Algorithm Development



Matlab/Simulink/Stateflow

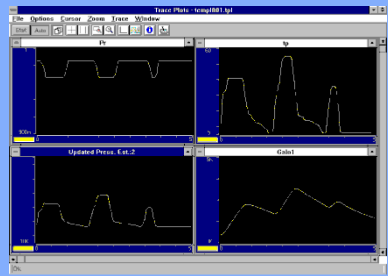
Automatically Generated Code

Control/Monitor Interface



Control Desk

Data Logging Interface



Control Desk Logging

PC

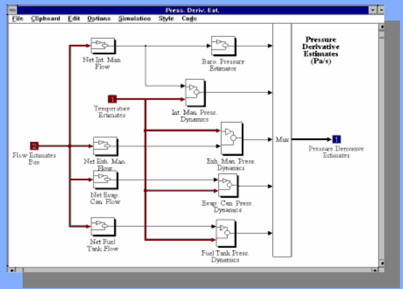
ECM / TCM

Auto-Code Targeted for Production Controller Hardware

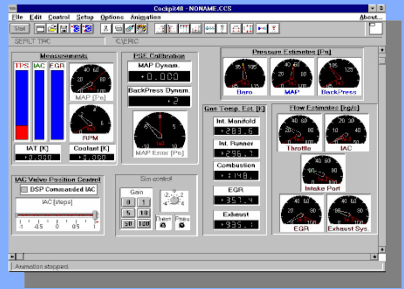
Algorithm Development

Control/Monitor Interface

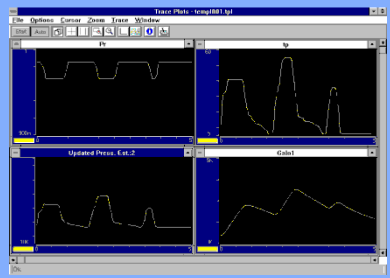
Data Logging Interface



Matlab/Simulink/Stateflow



Control Desk

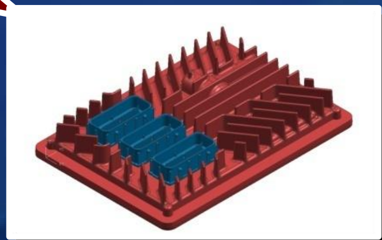


Control Desk Logging

Automatically Generated Code

PC

….SIDI algorithm models and auto-code will continue to be used across many applications, model years (MY2012, 2013, 2014,…) and controllers!



Production ECM / TCM

Controls Development using MBD

SIDI Controls System Development Example

- Rapid prototyping controls enabled development of engine hardware analysis
  - Compression Ratio Analysis
  - Injector Spray Patterns and Orientation
  - Dual Injection for Knock and Emissions Control
  - CMCV Analysis
  - NVH Development
- Algorithm models developed and tested for new SIDI functionality including fuel delivery, pump control, spark, torque, and cam phasers
- Plant models utilized in SIL, MIL and HIL
- Advanced development architecture used for initial fuel economy, emissions and drivability development, environmental testing including altitude and cold weather

SIDI Control Development Architecture

Advanced Development

GM Controller (Corp C Software)

High Pressure Fuel Sensor

58X Crank Sensor

4X Cam Sensor

Dspace Micro-Autobox

- High Pressure Fuel Pump Control
- Fuel Delivery

Injector Driver Box
Various Injector Suppliers

Injectors

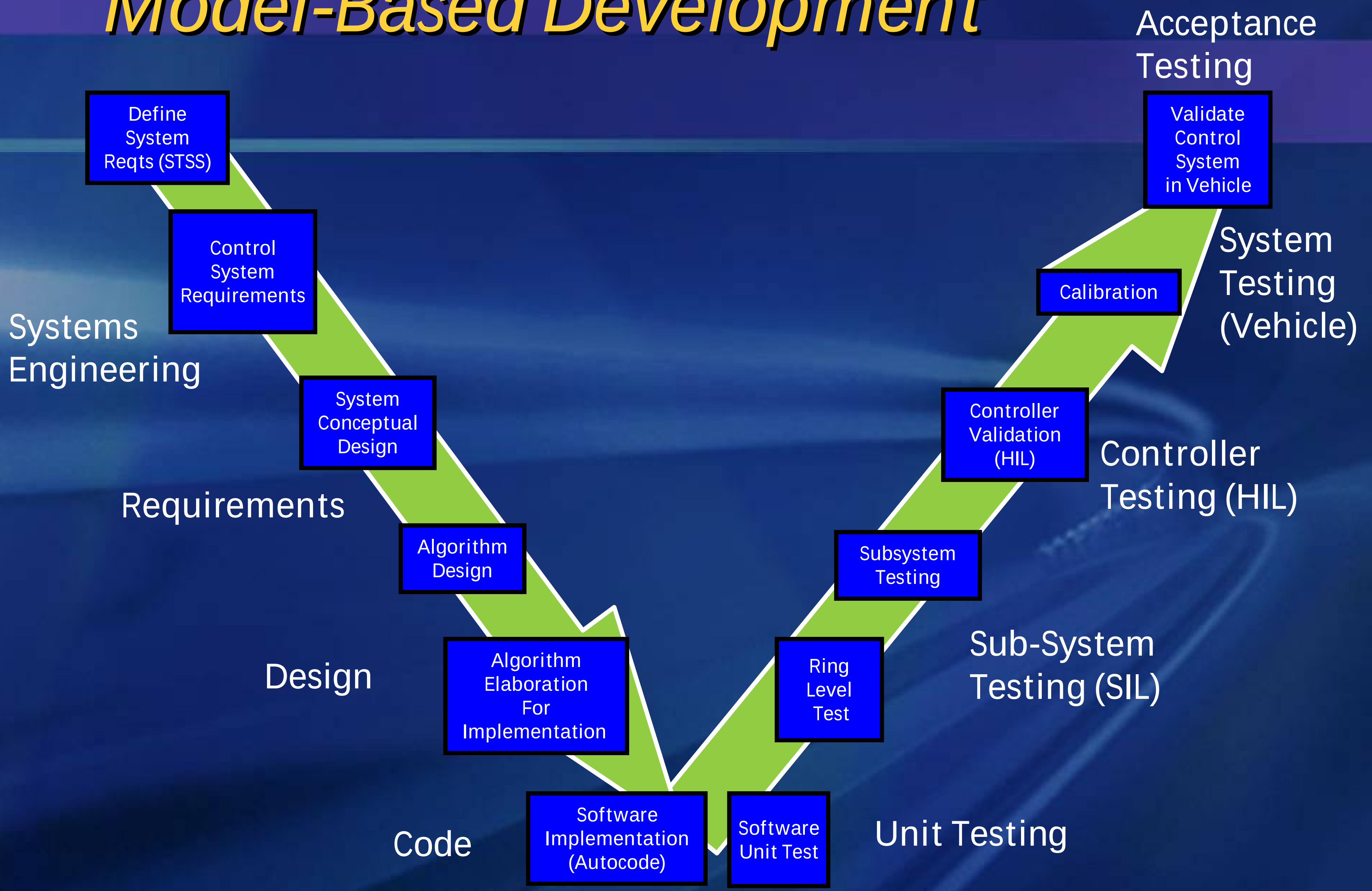
Pump Driver Box
Various Engine Architectures

Flow Control Valve

SIDI Functions

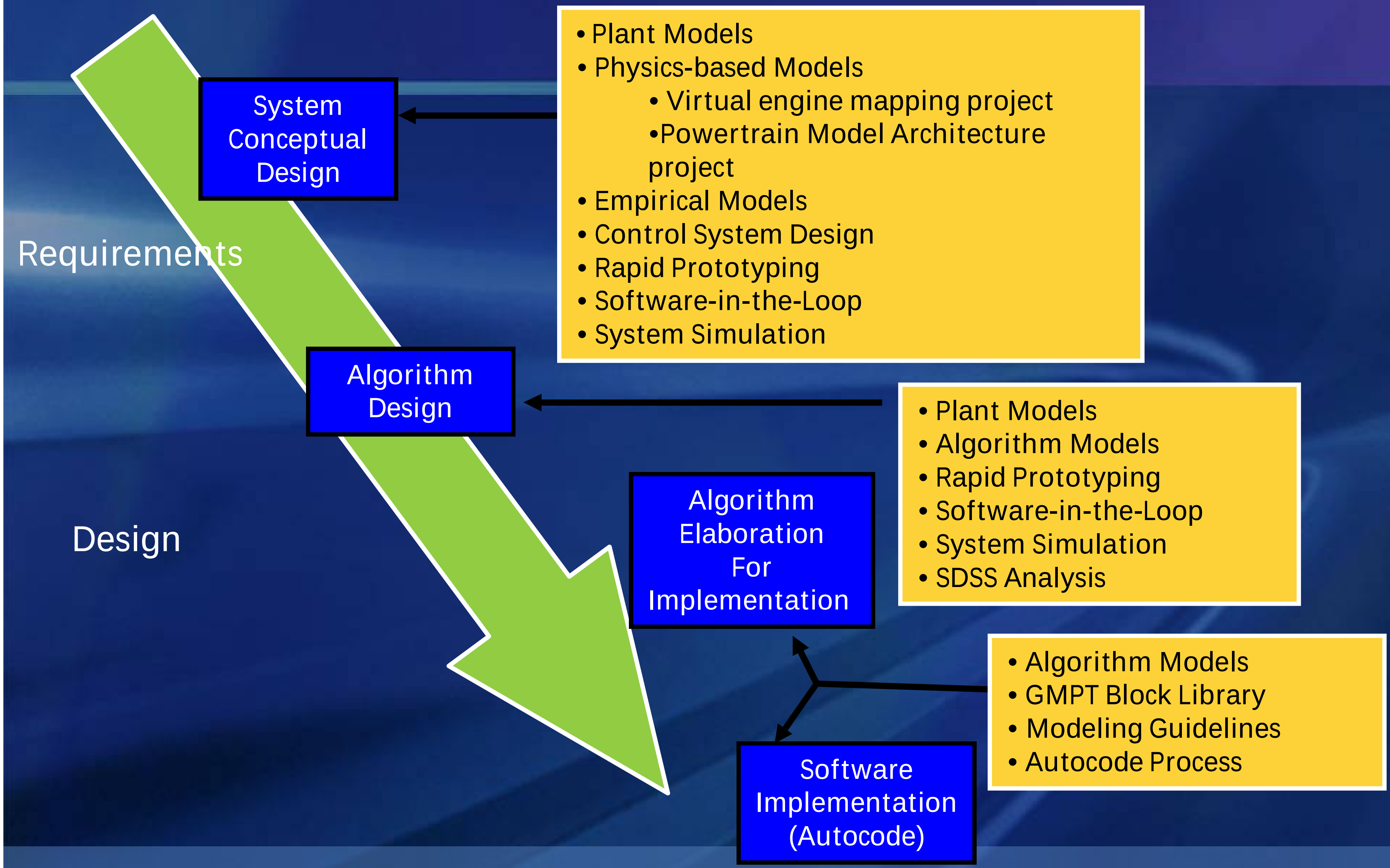
Core SIDI functions developed in Simulink/Stateflow models and executed on rapid prototyping hardware

Model-Based Development

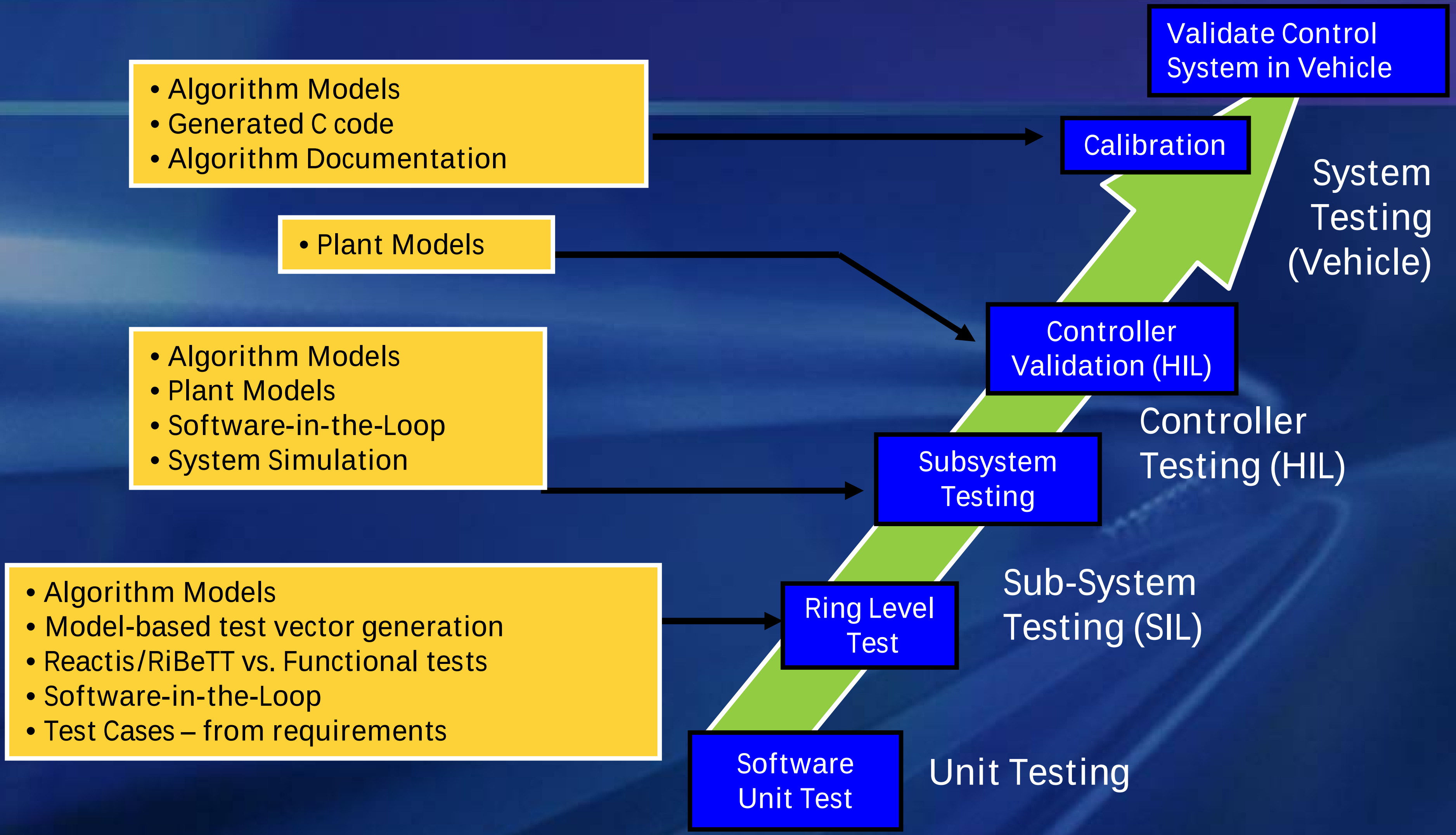


24

Model-Based Development – Design & Implementation



Model-Based Development – Testing & Validation



Benefits of Model Based Design

- Seamless transition

  Requirements ->Design->Implementation->Verification and Validation

- Enables global development by eliminating translation and re-interpretation between steps
- Permits what-if studies via virtual simulation
- Allows variation analysis without hardware
- Enables controls development prior to hardware availability
- Enables increase of global domain knowledge


GM
Thank You for your Attention.