ABSTRACT

GM's Road-to-Lab-to-Math (RLM) initiative is a fundamental engineering strategy leading to higher quality design, reduced structural cost, and improved product development time. GM started the RLM initiative several years ago and the RLM initiative has already provided successful results.

The purpose of this paper is to detail the specific RLM efforts at GM related to powertrain controls development and calibration. This paper will focus on the current state of the art but will also examine the history and the future of these related activities.

This paper will present a controls development environment and methodology for providing powertrain controls developers with virtual (in the absence of ECU and vehicle hardware) calibration capabilities within their current desktop controls development environment. The end goal is to provide powertrain controls developers with virtual controls development and calibration capabilities which duplicate, and actually provide additional benefits over, vehicle/dynamometer (Road/Lab) capabilities.

ETAS was able to provide this needed capability to GM in the form of an INCA Simulink Integration Package (SIP). This paper will provide the technical details of this interface as well as the general benefits for GM Powertrain controls development and calibration. This capability has allowed GM Powertrain to meet the project goals of one of their next generation transmission programs. This paper will also describe how GM used this tool on this project as well as the benefits and limitations of this approach.

INTRODUCTION

Due to the multitude of external design constraints, such as increasing fuel economy standards, and decreasing emissions standards, developers of automotive powertrain controls have had to cope with increasing levels of system complexity while at the same time being forced by the marketplace to improve system quality, reduce development costs, and improve time to market.

While all vehicle manufacturers choose to meet these challenges in different ways, there are several common trends. The first trend is the reduction on the reliance on physical prototypes and prototype vehicles for development testing and validation. Physical prototypes and prototype vehicles are very expensive and the lead time to design, manufacture, and integrate these prototype vehicles can greatly extend the overall vehicle development process. In order to reduce or eliminate this process dependency, most vehicle manufactures have embraced physical/plant modeling and simulation tools. In addition to shorter development times, these plant modeling tools have resulted in higher powertrain quality at GM [1].

The next, and parallel, trend is the use of model-based controls development processes. Model-based controls development generally replaces the formerly utilized method of text-based controls requirements capture and controls design via a structured programming language such as C or C++ [2] [3]. The use of a model-based controls development process provides many opportunities for improved product quality and development productivity improvement [4]. This trend was predicated on the adoption of specialized modeling and simulation tools such as Simulink® from The Mathworks. Simulink® is now the predominant automotive
controls modeling and simulation tool and an integral part GM Powertrain's (GMPT) model-based controls development strategy.

The process of powertrain calibration traditionally involves the tuning of generic control algorithms in prototype vehicles to meet program specific specifications of performance, fuel economy, emissions, and drivability. With the trend to move the powertrain development process away from prototype vehicle usage with more reliance on modeling and simulation tools, the calibration process will inevitably have to adapt to this environment and process as well.

As ETAS’ INCA calibration and measurement tool has been the primary tool within GMPT used for this task since prior to 2000, it also stands to reason that INCA should play a role in this migration of the powertrain calibration from in-vehicle to a desktop modeling/simulation environment.

This paper will examine specifically how GM is progressing in their efforts to migrate their powertrain controls development process to one which provides as seamless a transition as possible between R/L/M environments and provides the organization with the most benefit with the least amount of learning curve.

THE DIFFICULTIES OF CALIBRATION IN A VIRTUAL WORLD

SILOS OF ENGINEERING
Engineering of powertrain components requires a diverse skill set. There are “performance” engineers who design the hardware to meet the performance requirements, there are algorithm & software engineers developing controls, and there are calibrators who calibrate the algorithms together with the hardware.

Performance engineers would like to use algorithms and calibrations to properly stimulate their plant models, algorithm engineers would like to use the plant models and control software to develop their algorithms, and calibrators need both. Unfortunately, these engineers often use different tools in their development and the development is often done sequentially: first the hardware, then the algorithms, then the software, and finally the calibrations. The result might look like silos of sequential engineering that takes too long and inhibits communication across disciplines. Refer to Figure 1 - Engineering Silos.

This engineering model can lead to hardware which can be built but not controlled and/or algorithms that cannot be calibrated. Error correction across disciplines in this model can take a very long time. GM needed the capability to reduce development time and enable co-development across disciplines. System Simulation provides that capability.

INTRODUCTION TO SYSTEM SIMULATION
Several years ago GM’s RLM leadership began working with GM’s Advanced Engineering and hardware engineers to develop a “System Simulation” tool that could meet the need described above. There have been a number of versions of System Simulation as a result, each with its own strengths and weaknesses.
During this same time, GM developed a proprietary process in which production source code can be brought into Simulink in a robust, repeatable, and easy to use manner. This Software-In-the-Loop Simulink block has become known as the SIL block.

In 2007 GM determined that it would embark on the development of an entirely new transmission. This transmission provided the perfect opportunity to bring together all that has been learned in various System Simulation activities and create a tool that would aid production in being quicker to market and first time capable. This new tool would eventually be called SysSim. The SysSim architecture is shown in Figure 2 - SysSim.

This architecture is reused from its predecessors and it integrates the newly developed Software-in-the-Loop (SIL) block, Figure 2 -- bottom left. The SIL block contains all of the software that is in the Transmission Control Unit (TCU). This includes the common software like function scheduling and CAN communication as well as Transmission Specific Algorithms. This code is either hand coded and/or generated from models. Once the code set is complete it is compiled into the SIL block. Software functions, variables and calibrations are made available to Simulink via the SIL block, enabling the other components of the SysSim to interact with the entire TCU software set.

On the top left of Figure 2 is the Simulink model libraries block. This block contains the new transmission specific algorithm models in Simulink. This is where the algorithm engineer will spend most of their time. The engineer will develop algorithms, and then test them against the rest of the software in the SIL block, the plant model and the vehicle / engine model. Feedback is immediate and the algorithm engineer can modify the algorithm and retest very quickly. This is the block that is most attractive to algorithm engineers.

The center block of Figure 2 is the Hardware Adaptation Layer. This is a group of Simulink algorithms which is designed to represent the TCU’s Hardware Input Output (HWIO) interface and the high level behavior of the TCU electronics. Future extensions of this application would be to integrate physical models of the TCU electronics. The Hardware Adaption Layer block is used to connect the application source code in the SIL block to the vehicle plant and transmission plant models. This is the block that is of most interest to electronics engineers.

The top right of Figure 2 is the Vehicle & Engine model. This model represents a generic vehicle and engine plant model. The plant model has enough fidelity to support some items like grade and torque reduction requests, but does not emulate any particular vehicle or engine. Future extensions of this application could be to build in a better vehicle model or even connect this transmission SysSim to an engine SysSim and execute them simultaneously. This is the block that is most attractive to fuel economy, vehicle and system engineers.
On the bottom right of figure 2 is the plant model. This plant model can be a simple Simulink model to allow for a near real time execution or a complex plant model that represents the physics of the transmission. A Performance Engineer can use this plant model to analyze hardware modifications and use the algorithms and calibrations to stimulate the plant. Conversely, a good physical plant model can support the development of better algorithms and calibrations to control the hardware. Performance Engineers have co-simulated their hardware in AMESim via CosiMate [5] within the SysSim in the development of the new transmission.

Now that each of the disciplines in the “silos of engineering” has access to each other's work products via SysSim, the next step is to enable virtual calibration engineering -- the topic of this paper. Refer to Figure 3 -Bridges between engineering silos via SysSim.

Calibration data can be provided to the model environment in one of two ways. One way is that the calibration data is written to the Matlab Workspace and is accessed by Simulink blocks which are present in the model environment. The other way is that the calibration data only exists in the computer's memory. This type of calibration data is brought into Simulink by use of the SIL process.

**HOW TO CALIBRATE?**

One of the problems with the SysSim or any type of system simulation containing native algorithm models and legacy source code is providing convenient calibration access to the parameters of interest. These calibration parameters need to be calibrated in order for the simulation to behave properly. This presents a problem. There are multiple ways to calibrate native Simulink algorithm models. However, calibrating legacy source code brought into Simulink by GM's proprietary SIL process is not easy. This can be accomplished basically in two ways. One is to modify the calibrations in the source code before it is brought into Simulink, and the other is to modify the calibrations in the compiled source code once brought into Simulink. Modifying the calibrations in the source code (which typically contain non-program specific default values), while possible, is very time consuming.

Of course, the most efficient way to for engineers to work is to provide them the capability to make calibration changes to the calibration parameters in the SIL block directly. ETAS was able to provide this capability through its INCA Simulink Integration Package (SIP) add-on. INCA is the tool that GM uses to solve similar issues with compiled source code in a target Electronic Control Unit (ECU), so why couldn't the same tool be used to do the same thing in the Simulink SysSim environment?

**SOLUTION DETAILS**

As mentioned earlier in this paper, ETAS' INCA tool has been GMPT's primary tool used for both in-vehicle and dynamometer-based powertrain calibration since prior to 2000. Because of this, it made complete sense for ETAS and GM to develop a seamless solution for the emerging Math/simulation desktop powertrain controls and calibration development environment. As also described earlier, another key component of this powertrain calibration solution involves the interface to GMPT's system simulation environment, SysSim, which is constructed around Simulink®.
CALIBRATION SOLUTION GOALS

As a primary goal of this solution, it was required that the operation of the tool/system should appear and act to the user the same way as it appears and acts to the user in the physical world (dynamometer/dyno or vehicle). Since these engineers were already starting from a common user interface (INCA), the goal was to minimize or eliminate any learning curves associated with the adoption of this new tool and process.

Another primary goal of any calibration solution was to be able to provide a solution which facilitated the easy reuse of calibration files/data across all phases of development (R - L - M). INCA already provides seamless calibration data exchange from in-vehicle to dyno development and back. What was lacking was this same level of seamless integration in the Math case - specifically to and from the SysSim environment. Of particular need was the ability to easily import production calibration data values into the SIL blocks.

Additionally, once the engineer had performed this initial import of production calibration values into SysSim, engineers also required the ability to perform calibration tasks on both the parameters located in the SIL block as well as the New Transmission Specific Algorithms. Of course this capability should be provided as seamlessly as it is available today for both in-vehicle and dyno-based calibration activities (i.e. without having to recompile the code with the new calibration values).

ETAS was able to accomplish all of these goals through the use of a standards-based interface to Matlab®/Simulink®, use of the published Matlab® Application Programmers Interface (API), cooperation and collaboration with The Mathworks, and the creation of custom automation scripts to streamline the operation of the tool as well as to provide the user transparency that was desired.

To describe the operation of the solution, let's walk through a typical calibration session using INCA-SIP.

CREATE A DESCRIPTION OF THE SYSTEM OF INTEREST

In order to connect any type of Measurement, Calibration, or Diagnostic (MCD) tool into either a physical or virtual system for engineering development purposes, system description information must be created and provided to the MCD tool. The system description information or file contains information such as signal or variable names, scaling, data type, conversion factors (to physical units such as miles per hour), and location in memory. The system description file allows the MCD tool to access, display, and act on the parameters of interest in the system [6].

Early on in the development of MCD tools, the many benefits of standardization related to ECU description files became apparent. In 1991, as an initiative of German vehicle manufacturers, ASAM, The Association of Standardization of Automation and Measuring Systems was formed. This organization took the form of several technical work groups each focused on the creation of standards for data models, interfaces, and syntax specifications for a variety of applications (e.g. testing, evaluation, simulation).

For the purposes of defining the required ECU description file standard, ASAM created the ASAM MCD-2 specification. As an automotive industry standard, ASAM MCD-2, or ASAP2 file, is widely accepted and, of course, supported by ETAS' INCA MCD tool [7].

In the virtual space, the requirement from the MCD tool is the same, but a Simulink model does not provide the direct memory address mapping that is available with a physical ECU. To solve this problem, the INCA-SIP product creates this required description file automatically.

The first step of the system description file creation is to make a determination of what parameters -measurement and calibration variables - are of interest to the user. While Simulink® provides for standard measuring blocksets (e.g. scope) as well as standard calibration blocksets (e.g. Look-up Table), it is very typical that users either do not make use of these standard blocksets or they supplement these standard blocksets with custom Simulink® S-functions - including GMPT in their SysSim environment. In order to allow for these types of customizations, ETAS has implemented Matlab .m scripts which specify which blocksets are of interest for the user to have access to in the MCD tool. These .m scripts can be easily tailored to cover user specific or even project specific needs.

The typical working session starts by the user opening the Simulink® model of interest. The user then selects the “connect to current model” option from the Simulink® tools menu. See Figure 4 - Connecting the MCD Tool to the Model, for more details.

Using the .m scripts described above to identify the blocksets and parameters of interest from the model, and adding to this information, their respective data types and scaling, and their location in the model, it is possible for the INCA-SIP interface to create the required MCD tool database objects automatically.

The files that are required by the MCD tool and created by the INCA-SIP interface are:

- The model variable description file - ASAP2 file
• The model HEX file which contains the values of Matlab® Workspace variables associated with the model which the MCD tool will have the ability to modify or calibrate

Additionally, the INCA-SIP interface creates the following MCD tool database objects from the ASAP2 and HEX files which are required to start a measurement and calibration session:

• The Project contains a description of all of the calibration data and measurement data
  ◦ The Project also contains the calibration data set values derived from the from the HEX file (or in this case from the Matlab Workspace data)
• The Workspace is the primary working structure in the MCD tool and contains a description of the tool interface (ASAM XCP on Ethernet in this case), the designated Project and calibration data set, and the designated Experiment (user interface details).

WORKING WITH INCA-SIP

By selecting the newly created Workspace (SimulINCADemo_Wsp from Figure 5), the MCD tool will open up a blank (initially) Experiment. It is of course possible to create and store multiple Experiments for use with INCA-SIP. The Experiment is where all of the user interaction with the Simulink® model will occur.

Different Types of Data are Required

It is important at this point to clarify the typical uses of an MCD tool. It is also once again important to draw analogies to powertrain development using an actual ECU and test vehicle. In a vehicle or dynamometer environment, it is typical for a powertrain controls developer or calibration engineer to need to perform calibration parameter changes in order to “tune” the powertrain control system to insure that this powertrain control system meets program goals such as fuel economy or emissions. To verify that these goals are met, measurement data must also be captured.

MCD tools are generally capable of capturing measurement data from multiple different sources and time aligning these multiple data sources. There are two primary different types of data sources. The first and primary data source is the internal ECU control variables which come from the real-time calculations occurring in the embedded ECU powertrain control software. This data is necessary to validate the correct operation of the powertrain control system as well as its desired operation relative to program goals or requirements.

The second type of measurement data which is required during powertrain controls development is data which is captured via externally mounted measurement sensors and equipment. This data allows powertrain controls developers or calibrators to verify that the ECU software observations/calculations and intended control outputs correlate to their respective physical measurements on the vehicle.

Figure 4. Connecting the MCD Tool to the Model

Figure 5. MCD Tool Database Objects

Please see Figure 5 - MCD Tool Database Objects, for more details.
While controls development requires the ability to measure and calibrate (i.e. read and write) from the ECU software or control model, only measurement (i.e. read) capability is necessary to capture the required data from the physical system being controlled, or analogously, the plant model in the virtual world.

Therefore, to provide the most complete and accurate virtual powertrain development environment, the SysSim environment includes the capability of including both the controls models and the physical system/plant models in the system simulation. Because of this need, INCA-SIP can provide appropriate access to both the Simulink controls and the Simulink plant models even during model execution - thereby completely replicating the powertrain calibration task in the physical environment (vehicle or dyno). See Figure 6 - INCA-SIP Model Access.

Providing Calibration Access to the SIL Block

As was described earlier and unique to the SysSim environment, are the production C code SIL blocks (refer to the lower left corner of the diagram in Figure 2 - SysSim. As these SIL blocks constitute a major portion of the SysSim environment, it was important that SysSim users could have both calibration and measurement access to the variables contained in these SIL blocks. Unfortunately, Simulink does not provide a very convenient method for gaining the required access to these variables, so a custom interface had to be created to allow calibration and measurement access. Due to the priority that was placed on having calibration access to the SIL blocks, calibration access to the SIL blocks has been implemented first. At the time of this paper being written, measurement access to the SIL blocks has been implemented and is undergoing initial testing. Please refer to Figure 7 - INCA-SIP DLL Access, for a complete overview of the proposed system.

Experiment Environment - Connecting to Simulink Data

Once the Experiment is open, it is now possible to select which measurement and calibration variables are of interest and into which type of editor or display they should be placed. This is accomplished through the use of a Variable Selection Dialog. See Figure 8 - Variable Selection Dialog.

Running and Using the Simulation Environment

Once the desired parameters have been added to the Experiment, it is then possible to:

- Monitor or record Simulink® simulation data values in the MCD tool by activating the simulation in Simulink® by starting the measurement or recording mode in the MCD tool.
  - This measurement data can then be reviewed in the MCD tool's data analysis tool
  - INCA-SIP supports two different measurement modes - real time emulation mode and fast emulation mode
    - Real time emulation mode is the default measurement mode. It is intended to emulate how calibrators work in the vehicle by forcing the Simulink® model to execute in near real time. This allows engineers the ability to see the immediate
results of their calibration changes within the MCD tool data display GUIs.

- Fast emulation mode allows the Simulink® model to execute as quickly as possible and writes a measurement data to a file for post processing/review

- Calibrate Matlab® Workspace variables or Simulink® blockset constants while the model simulation is running (only in real time emulation mode) – this is not easily possible in Simulink®
  - Allows users to see the immediate affect of a given calibration change totally analogous to the physical environment (vehicle or dyno)

- Store any calibration changes as a comprehensive calibration data set that can be easily reused by the same user or shared with other users
  - Matlab®/Simulink® is not an MCD tool, and is therefore not well equipped to manage the large complexities and calibration variants which are necessary for these applications
  - INCA-SIP provides seamless calibration data exchange across all engineering project phases (R-L-M)
  - INCA-SIP provides the capability to “download” a complete calibration data set to the selected model prior to a calibration and simulation session
  - These changes do not need to be maintained in the model or the Workspace data of Matlab® as they are maintained and tracked in the MCD tool

Concluding an INCA-SIP Session
After performing the required calibration and measurement tasks within the INCA-SIP and SysSim environment, controls developers have several options. Using the MCD tool, they are able to create and store as many different calibration variants as are required for their particular need. These different calibration variants or calibration data sets can then be easily shared across the entire development community.
Another possibility might be that the controls developer needs to make some additional changes to the Simulink/SysSim models. In both these cases, after they have completed the required tasks with the representative models, the user will need to disconnect the MCD tool from the selected Simulink model.

After disconnecting the MCD tool from the model, the user is then able to make any necessary changes to the model(s) in Simulink. Once those changes are made, the user can just as easily reconnect to the modified model. If only changes to the Matlab Workspace data is made (i.e. no changes to the model), then upon reconnecting to the model, the MCD tool gives users the choice of using (i.e. downloading) the calibration data set stored in the MCD tool or using (i.e. uploading) the new Matlab Workspace data and creating a new calibration data set.

If model changes are made, then the MCD tool must rebuild the required model description and HEX files before it is possible to reconnect to the model.

**Goals Accomplished**

Through the use of INCA-SIP, GMPT is able to achieve all of the required powertrain controls calibration functionality necessary to round out the controls development environment of SysSim.

- Equivalent calibration tool behavior to use in in-vehicle and dyno-based environments
- Seamless reuse of calibration data files across all development phases (R - L - M)
- Easily import production calibration data into the SysSim simulation environment
- Calibration of New Transmission Specific Algorithms
- Calibration of SIL Blocks without having to re-compile the code

**FUTURE OF SYSTEM SIMULATION AT GM**

As shown in **Figure 9 - Calibration Plan**, GM's current calibration effort occurs in the dynamometer and vehicle. However, with the advent of system simulation (i.e., virtual simulation) and tools like INCA-SIP, there is a possibility to move calibration further upstream where physical vehicle hardware is not required.

One of the concerns of creating valid calibrations using the virtual simulation is that it can depend greatly on the fidelity of the plant model used.

**SUMMARY/CONCLUSIONS**

ETAS's INCA calibration interface to Simulink has brought Calibration Engineering capabilities to the virtual space at GM. See **Figure 10 - Updated Bridges Between Engineering Silos via SysSim**.

The ability to calibrate virtually is an enabler to GMPT's hardware, algorithm and calibration development. GMPT is
now able to develop hardware, algorithms and calibration simultaneously, prior to the availability of experimental hardware. This capability reduces the number of prototype iterations and increases quality resulting in the reduction of both the time and the cost of new product development.

This capability has already been enhanced by adding data measurement capability via INCA. Other potential growth areas include a closer integration of dynamometer automated calibration development (GM's RLM initiative), adding co-simulation with other plant modeling tools to increase plant model fidelity, adding distributed computing to reduce model execution time and provide a broader distribution of the capability to GMPT's production algorithm and calibration engineers.

GMPT will continue to work with ETAS to expand and enhance the calibration capabilities within the SysSim environment with the end goal to enable as much of the powertrain calibration to be completed in the virtual environment as possible.

REFERENCES


CONTACT INFORMATION

Tim Foster
ETAS Inc.
734-302-2033
tim.foster@etas.com

DEFINITIONS/ABBREVIATIONS

CAN (Controller Area Network)
A network communication bus architecture developed originally by Robert Bosch GmbH and Intel Corporation in 1983

ECU (Electronic Control Unit)
An embedded system that controls one, or more, of the electrical systems or subsystems in a vehicle
Hardware Input Output (HWIO) layer
The portion of the System Simulation that emulates the physical input and output data quantities that exist in a physical Electronic Control Unit (ECU)

MCD (Measurement, Calibration, and Diagnostic) Tool
A typically PC-based tool for used for the interrogation and manipulation of embedded controls parameters

Road-to-Lab-to-Math (RLM)
An automotive industry trend/concept to improve product quality and shorten product development cycles by moving more development to the lab and to modeling environments

SIL (Software In the Loop)
A software testing and development concept which provides a method by which compiled software code can be executed and tested in a representative environment

System Simulation (SysSym)
A Simulink-based modeling and simulation environment developed by GM Powertrain for the purposes of supporting early powertrain controls development

Transmission Control Unit (TCU)
A specific variant of an Electronic Control Unit (ECU) dedicated to the control of an automatically controlled transmission