2017 NDIA GROUND VEHICLE SYSTEMS ENGINEERING and TECHNOLOGY SYMPOSIUM
Modeling & Simulation, Testing and Validation (MSTV) Technical Session
August 8-10, 2017 - Novi, Michigan

THE HPCMP CREATE^{TM}-GV PROGRAM, SOFTWARE, DEVELOPMENT, AND APPLICATIONS

Thomas Skorupa
Sara Pace Boyle
Jeremy Mange, PhD
Daniel Kedziorek
Cesar Lucas
TARDEC
Warren, MI

Christopher Goodin, PhD
Jody D. Priddy
Kevin Walker
Michael Puhr
ERDC
Vicksburg, MS

Michael S. Mazzola, PhD
Mississippi State University
Starkville, MS
Jacob Brendle
Alion Science & Technology Corporation

ABSTRACT

The High Performance Computing Modernization Program (HPCMP) Computational Research and Engineering Acquisition Tools and Environments – Ground Vehicles (CREATE^{TM}-GV) Program is a software development effort to create government-owned scientific High Performance Computing (HPC) code for the next generation of mobility analysis tools. The HPCMP CREATE^{TM}-GV software consists of three main components: the Ground Vehicle Interface – a web-based interface for interacting with the HPCMP CREATE^{TM}-GV tools on the HPC; Mobility Analysis Tool (MAT) – computing tactical mobility performance of ground vehicles over broad areas of real-world terrain for mission-based performance metrics; and Mercury – a high-fidelity, multi-body physics analysis tool that runs a co-simulation of many components on the HPC, the results of which can then be fed into MAT or exported to trade space tools for further analysis.

In this paper, we provide an overview of the HPCMP CREATE^{TM}-GV program and present details about each of these components, the applications of the software to acquisition efforts, and verification and validation of the software and data involved.

UNCLASSIFIED: Distribution Statement A. Approved for public release; distribution is unlimited.

INTRODUCTION

The use of validated physics-based modeling and simulation methods during the ground vehicle design phase is a critical enabler to allow development of robust platforms that meet mission needs while minimizing vehicle development costs. The Department of Defense (DoD) High Performance Computing Modernization Program (HPCMP™), established in 1992 by a Congressional mandate to maintain large-scale super computing resources for the DoD Science and Technology communities, realized the need for specialized software to run on those computing resources. In 2007, the HPCMP™ established the HPCMP Computational Research and Engineering Acquisition Tools and Environments (CREATE™) Program designed to improve the DoD acquisition process by developing and deploying four sets of advanced computational engineering design tools to enhance military aircraft design, ship design, RF antenna design and integration, and mesh & geometry generation. The HPCMP CREATE™ Program was expanded to include a fifth set of tools to address the advanced needs of the Army's ground vehicle (GV) acquisition community, named HPCMP CREATE™-GV.

The primary goal of HPCMP CREATE™-GV is to provide a suite of government-owned, commercial-quality software tools that has maximum impact on acquisition programs by leveraging DoD HPC capabilities, with a secondary goal of improving the way the government interfaces with its industry partners regarding ground vehicle design and modeling and simulation (M&S).

At its genesis, HPCMP CREATE™-GV employed a Quality Functional Deployment (QFD) approach to identify the requirements and areas for analysis improvement within the DoD Ground Vehicle Acquisition Process, consisting of three major decision making systems: the Planning, Programming, Budgeting and Execution [1] (PPBE) system, the Joint Capabilities Integration and Development System [2] (JCIDS), and the Defense Acquisition System [3] (DAS). The results of the QFD were documented in a Capabilities and Gap Document [4] (CGD) that was endorsed by the Joint Center of Ground Vehicles and a HPCMP CREATE™-GV Board of Directors consisting of senior management with extensive experience in oversight of acquisition programs. The CGD identified six technology gaps and seven acquisition processes that HPCMP CREATE™-GV tool development would be targeted to address, provided in table 1 and table 2.

Gap 1:	Scalable Multidisciplinary physics solvers
Gap 2:	Rapid physics based concepting tool for ground vehicles
Gap 3:	Augment system engineering trade space tools with physics-based data
Gap 4:	Robustness optimization tool
Gap 5:	Soldier centric design and analysis
Gap 6:	Concept Manufacturability analysis tools

Table 1: Technology Gap Categories

GV-001	Technology assessment
GV-002	Requirements analysis (problem domain)
GV-003	Vehicle and technology virtual prototyping
GV-004	Cost and reliability & maintainability process
GV-005	Operational effectiveness
GV-006	Trade off analysis process (solution space)
GV-007	Production, quality, and manufacturing management

Table 2: Acquisition Processes Targeted by the HPCMP CREATE™-GV Project

ARCHITECTURE OF HPCMP CREATE™-GV

The HPCMP CREATE™-GV Program is a software development effort to create government-owned scientific High Performance Computing (HPC) code for the next generation of mobility analysis tools. There are three main components of the HPCMP CREATE™-GV software: the Ground Vehicle Interface (GVI), the Mobility Analysis Tool (MAT), and the Mercury co-simulation environment. The GVI is a web-based interface for interacting with the HPCMP CREATE^{TM}-GV tools on the HPC. MAT converts vehicle performance metrics and terrain information into mission-based analysis of vehicle performance over large areas of terrain. Mercury is a high-fidelity, multi-body physics analysis tool that runs a co-simulation of many components on the HPC, the results of which can then be fed into MAT or exported to trade space tools for further analysis.

Our approach in managing HPCMP CREATE^{TM}-GV's code development workflow employs the light-weight software development with the best features of Milestone-based methods. Our development process incorporates a practice-driven, agile software development approach to risk management, rather than process-driven approaches. As many other, Computational Science and Engineering (CSE) software development projects have adopted some agile software engineering practices [24], with the best features of Milestone-based methods.

GROUND VEHICLE INTERFACE PHILOSOPHY AND DESIGN

The Ground Vehicle Interface (GVI) is the web portal for the HPCMP CREATE^{TM}-GV program. GVI provides the front-end interface for user interaction in addition to the data integration for the underlying HPCMP CREATE^{TM}-GV models. The basic workflow is broken into three actions: create vehicle models, select metrics (tests) to run on those vehicle models, and execute those metrics in an HPC environment. These actions are facilitated by encapsulating them in a project wrapper.

The end goal of GVI is to provide an intuitive interface for subject matter experts to access the HPCMP CREATE^{TM}-GV suite of models. This allows GVI to reach a broader audience in the subject matter field, as a web-portal can offer a more user-friendly experience than a terminal interface.

User Perspective

Upon logging in, the user is greeted with the Home page where they are presented a listing of their projects. From the Home page, the user can navigate to one of their projects or to the Vehicle Model Builder page. The Vehicle Model Builder provides the ability to add or edit existing vehicle models that can be included in any given project. When a user first selects a project, they are presented with the Overview page, which details project metadata and a comprehensive history of the project. The user can navigate to the Metrics and Vehicles pages in the user's preferred order. The Metrics page allows the user to select the metrics of interest to ascertain about the current project's collection of vehicles. Any options associated with a particular metric will be available on the Metrics page. The Vehicles page displays all instantiated vehicle models present in the selected project, as well as associated high-level meta information about each vehicle. The user can add or edit an existing vehicle from the Vehicles page. Upon selecting a vehicle, the user has access to the editable parameters for that vehicle. When editing parameters, the user can elect to replace a single value with a range of values, allowing the creation of multiple vehicles from one instance. Once the selected project is ready to be executed, the user can proceed to the Run page. After reviewing a summary, the analysis is executed in the HPC environment, with notification sent to the user when the analysis has completed.

GVI empowers users to perform virtual prototyping of various types of ground vehicles. With GVI's intuitive interface, a user can easily expand a single vehicle into many different vehicle variations. By iterating over virtual prototypes, HPCMP CREATE^{TM}-GV helps eliminate the need for multiple, expensive physical proof of concepts. By providing editable templates of commonly used vehicles, the GVI further facilitates this simple workflow (e.g., a user has the option to take a base vehicle template and tweak parameters to expand it into many different variants in only a matter of moments).

Database

GVI utilizes a MongoDB database served by a NodeJS platform, as shown in figure 1. MongoDB is an easily adaptable, document style database that allows for irregularly shaped data. Using MongoDB in conjunction with the NodeJS Mongoose library, a schema can be enforced to allow the storage of various types of vehicle data without constant alteration. Additionally, MongoDB can grow with GVI with exceptional scalability through sharding, a type of database partitioning. Finally, MongoDB documents are stored in Binary JSON (BSON), which is designed to work naturally within web applications.

The use of a centralized database is a departure from what most users are accustomed. However, it affords the user and the community a much higher degree of flexibility when collaborating within a team. Multiple users will be able to work on a single vehicle design stored in the database. Furthermore, a group of subject matter experts can create the foundation of validated models upon which other users can build and tweak customized models. Vehicle models of current military vehicles will be available in the database for users to expand upon.

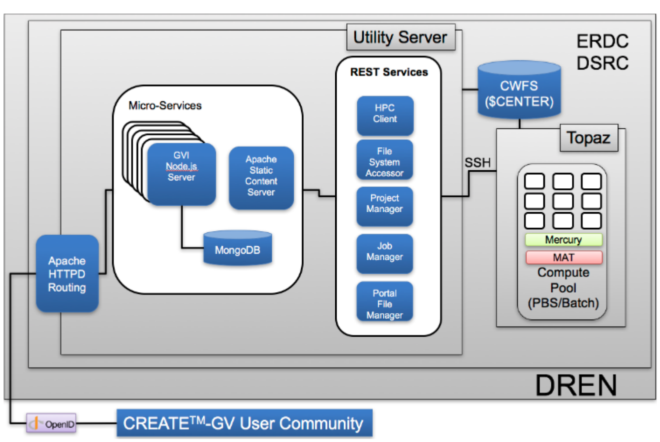

Figure 1: Overview of the GVI Architecture.

Zero Footprint Capability

Unlike traditional desktop applications, GVI provides a zero-footprint web interface, eliminating the need to download or install anything in order to use GVI. This provides users increased accessibility since the only requirements to run GVI are an internet connection and an account on the HPC Portal. Having this zero-footprint capability also allows users that have strict computing environments (e.g., users that are unable to install software on their machines) to utilize the HPCMP CREATE™-GV tools by removing any device, hardware, or software limitations.

Another benefit to the zero-footprint web interface is having a centralized codebase. This allows GVI to provide seamless maintenance and upgrades, which greatly reduces downtime and unnecessary hassle on the user. A centralized codebase also ensures that every user is using the most up to date version of GVI. This zero-footprint web interface paired with the centralized database provides the GVI users with a unique collaboration environment that will help streamline the vehicle creation and modification process.

MOBILITY ANALYSIS TOOL

The Mobility Analysis Tool (MAT) provides a flexible, modular software framework for computing tactical mobility performance of ground vehicles over broad areas of real-world terrain for mission-based performance metrics that are used as acquisition requirements and purchase specifications. MAT provides two key types of performance metrics that quantify tactical mobility in specific areas of interest around the globe, taking into account all major aspects of overall vehicle design:

- Mission Rating Speeds (MRS) – spatially averaged speeds based on the “mission profile” of a vehicle, which is a summary of the types of maneuvers the vehicle is expected to perform

- Terrain accessibility (%NOGO) – Quantifies the percentage of a terrain that is non-navigable for a vehicle

These metrics effectively extend developmental testing type metrics from Mercury into mission-level metrics that quantify the relative suitability of different vehicle designs for tactical mobility operations in different global regions based on prevailing terrain, seasonal, and weather conditions. MAT integrates proven comprehensive methods [5] that have been standardized for use within the DoD for these mission-level mobility metrics. Due to the high degree of experience and confidence with these metrics, they are often used as acquisition requirements and purchase specifications.

MAT uses force balance modeling based on summation of forces equal to zero to compute maximum steady-state speeds for all terrain conditions that exist within a global theater of interest. The theater extents considered span large areas of terrain, typically in the range of 200 to 300 square miles. Powertrain-controlled maximum tractive effort versus speed performance is computed by Mercury for use in MAT. Terrain influences are computed within Mercury and MAT to produce the powertrain/terrain-controlled tractive effort versus speed limits, as well as various resisting factors which produce impediments to motion. The sum of these resistances compared with the powertrain/terrain-controlled tractive-force versus speed performance provides a maximum possible force-controlled speed. Several other powertrain-independent limiting speed considerations are computed by Mercury (e.g., ride quality over rough terrain) or within the MAT algorithms (e.g., tire durability speed limits). The minimum of these other speeds and the force controlled speed are then compared to yield a final predicted maximum potential speed for each terrain condition. Areas of terrain where predicted speeds are zero represent %NOGO regions, and all other regions are used to compute MRS.

%NOGO metrics quantify the percentage of terrain in a region of interest that is non-navigable for a vehicle based on various types of immobilizing terrain features and conditions. They represent the amount of terrain that is inaccessible to a vehicle for tactical mobility operations within a specific theater of interest. It is an estimate of the percentage of terrain in a region that a vehicle cannot traverse, considering the various types of terrain features and conditions the vehicle may encounter that would cause immobilizations (e.g., getting stuck in deep mud) during real missions within specific theaters of operation around the globe. The %NOGO metrics can be used to quantify and compare the mobility performance of different vehicle designs based on the relative percentage of terrain that is inaccessible for tactical mobility operations. %NOGO metrics are influenced by regional terrain features and prevailing seasonal and weather conditions that occur within different regions of interest.

MRS metrics are spatially averaged omnidirectional speeds that take into account the mission profile of a vehicle, which is a summary of the types of on-road and off-road maneuvers the vehicle is expected to perform during its life-cycle. They are essentially wide-area, statistical measures of the maneuver-based mission effectiveness of a vehicle considering the various types of terrain the vehicle may encounter during real missions within specific theaters of operation around the globe. Like %NOGO metrics, MRS metrics are influenced by regional terrain features and prevailing seasonal and weather conditions that occur within different regions of interest. The MRS metrics also include off-road speed ratings for which only off-road terrain conditions are considered. The MRS metrics can be used to quantify and compare the mobility performance of different vehicle designs based on the relative speeds that can be achieved for tactical mobility operations.

For the MAT metrics, it is assumed that the vehicle may need to maneuver in any direction through any patch of terrain located within a global region of interest, constrained only by the mission profile. Considering all of the terrain features and conditions that a vehicle will encounter in different theaters of operation for particular mission profiles eliminates the need to specify all possible real missions that the vehicle will be expected to accomplish during its lifecycle. The mission profile reflects the expected distribution of major terrain types that the vehicle will be required to traverse over time, and the specific features and conditions of each major terrain type will be dependent on the specific theater of operation. For %NOGO metrics, emphasis is placed on off-road portions of the mission profile, where immobilizations are likely to occur. For MRS metrics, the major terrain types are segregated into categories associated with primary roads, secondary roads, trails, and off-road terrain.

MAT uses integrated terrain databases that provide key terrain features for various terrestrial regions of interest. Several terrain databases are available for unique and disparate locations around the globe, while new terrain databases can be developed to meet future demands. Sources of information available for developing terrain databases include data and modeling sources from the National Geospatial-Intelligence Agency (NGA), Army Geospatial Center (AGC), Air Force Weather Agency (AFWA), and National Aeronautics and Space Administration (NASA). The terrain features offered in MAT include slope grade, surface roughness, visibility conditions, and weather related effects due to snow, ice, frozen ground, and rainfall. Specific terrain features for off-road terrain include soil type, soil strength, and dimensional characteristics of linear feature obstacles (e.g., drainage ditches), vegetation, and other discrete obstacles. Specific terrain features for roads and trails include pavement type or soil type and soil strength, dimensional characteristics of traffic lanes, and radius of curvature for turns.

MAT is accessible within the GVI graphical user interface, through which MAT vehicle models will be developed. MAT makes use of HPCMP CREATE™-GV Mercury simulations for various types of physics-based vehicle-terrain interaction performance information. Direct interfacing between Mercury and MAT occurs during the generation of HPCMP CREATE™-GV performance data packages. MAT uses standard terrain models included in the GVI database, and has extensibility to include new dynamic terrain models produced by the Environmental Simulator software. Environmental Simulator is another development effort that is proceeding in parallel with the HPCMP CREATE™-GV tools.

MERCURY

Mercury is a software application for simulating the performance of wheeled and tracked ground vehicles in engineering-level performance tests such as slope climbing and max speed. Mercury combines multiple physics models for vehicle subsystems into a single, integrated simulation tool that captures the interactions of vehicle dynamics, powertrain performance, and vehicle-terrain interaction. Mercury allows users to modify vehicle parameters such as mass, suspension stiffness, wheelbase, powertrain capability, and tire shape to evaluate the influence of these parameters on overall vehicle performance. It does this as a set of several applications, each one consisting of different combinations of physics models, allowing users to select the appropriate combination of models of interest. In the initial release of the HPCMP CREATE™ -GV tools, Mercury consists of the following applications: Chrono, Powertrain Analysis Computational Environment (PACE), Vehicle-Terrain Interaction (VTI), and Driver input modules.

One of the core goals of the HPCMP CREATE™-GV project is to effectively utilize High-Performance Computing (HPC) resources for the simulation and analysis tasks. For this reason, it was recognized early in the program that commercial software would not be a viable option for most components, since licensing restrictions would severely limit the use of possibly hundreds-of-thousands of HPC cores envisioned for the software.

For this reason, the central multi-body dynamics software chosen is Chrono, an open-source middleware library that provides detailed physics modeling with a high level of control for the implementing software. In addition, other open-source or government-owned software has been chosen or created for each component of the co-simulation, many of which will be discussed in detail in the following sections.

utilizing the Chrono::Vehicle dynamics library, which provides multi-body dynamic simulation of wheeled and tracked vehicles. Vehicle-terrain interaction (VTI) is simulated with the Ground Contact Element (GCE), which provides forces to the Chrono-vehicle solver. The powertrain is modeled using the Powertrain Analysis Computational Environment, a behavior-based powertrain analysis code based on the U.S. Department of Energy's Autonomie software. The driver model implements an array of control strategies for evaluating vehicle mobility performance.

The Mercury framework enables the user to run a

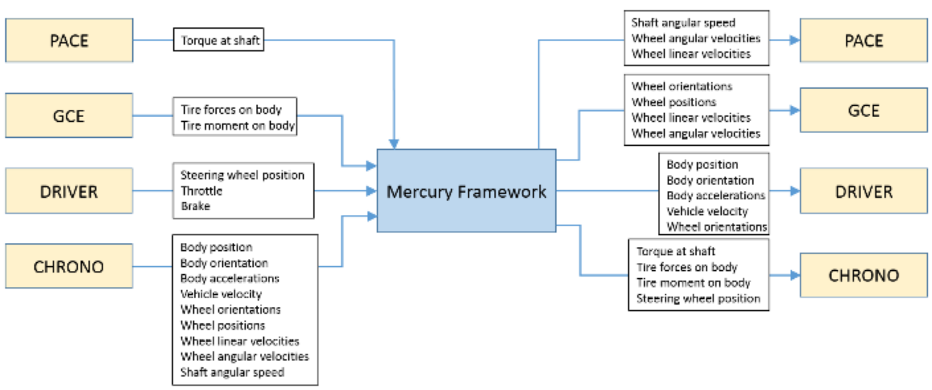

Figure 2: Illustration of Mercury co-simulation.

Architecture Overview and Design Approach

A Mercury analysis consists of three main components:

1. *Modules* represent a vehicle sub-system in the co-simulation
2. *Simulations* are a combination of several *Modules*
3. *Tests* are an instantiation of a *Simulation* with an associated terrain and a set of desired output metrics

The main *Modules* of Mercury include vehicle dynamics, vehicle-terrain interaction, powertrain, and driver models. Vehicle dynamics is simulated

co-simulation of various Mercury *Modules*. Throughout the co-simulation, these various *Modules* update the vehicle sub-system state and pass information to the other modules through the Mercury framework, as shown in figure 2. The *Modules* are implemented as C++ classes and each *Module* must have an update method that takes the timestep and other vehicle state data as input to perform the sub-system update calculation. The modular structure of Mercury allows for new tools to be implemented as various *Modules* with very little change to the existing Mercury codebase.

A *Simulation* in Mercury is a combination of *Modules* that make up a vehicle model with various sub-systems. As with the *Modules*, there is a standard structure for the methods and data associated with a *Simulation*. Not all of the *Modules* need to be implemented for every *Simulation*. For example, a tilt-table test only requires the vehicle dynamics and vehicle-terrain interaction *Modules*. The structure of the *Simulations* also allow the user to create additional *Simulations* using new *Modules* as they are developed without having to change the underlying code.

A *Test* is a specific instance of a *Simulation* paired with a terrain and output metrics that can be used to evaluate different aspects of vehicle mobility. There is not a standard format for *Tests* in Mercury. The *Tests* are designed to process the inputs of a *Simulation* and produce a desired output. To accomplish this a defined *Test* may utilize stopping conditions, such as course completion or immobilization, or it could use a set of specified maneuvers, such as a lane change maneuver. A particular *Simulation* can also be used in a variety of different tests. For example, a *Simulation* that includes the vehicle dynamics and vehicle-terrain interaction *Modules* can be used for a "drop" *Test*, which calculates the weight distribution on each axle of the vehicle, and it could also be used for the "tilt-table" *Test*, which calculates the maximum side slope angle.

Mercury is designed to run on the HPC system and has the ability to run thousands of different *Tests* with different vehicle variants simultaneously to feed tradespace analysis tools. The Mercury portion of the HPCMP CREATE™-GV program contains many complex pieces that make up the larger system.

Chrono

Chrono is an open-source physics based multi-body dynamics modeling and simulation framework implemented in C++ that is being developed by the University of Wisconsin and the University of Parma-Italy. The basic modeling elements of the Chrono::Engine have been validated against Adams models. [6-8]

The Chrono::Vehicle library is used in Mercury to model wheeled and tracked vehicles. In order to allow for flexibility and ease of modification, templates are used in Chrono::Vehicle to define a particular implementation of a sub-system. This allows for components and sub-systems to be easily reused or modified for different vehicles. These templates are defined through JavaScript Object Notation (JSON) files and arranged in a hierarchical structure that mirrors the organization of a physical vehicle. For example, the front and rear suspension sub-systems of a two-axle vehicle are often very similar, allowing for the same suspension template to be used for each, with minor modifications as appropriate to represent the differences.

These templates provide another major advantage to the program, allowing for the management of vehicle data in a central and easily accessible location. Through the GVI described earlier, users will be able to access a set of verified vehicle models from common Army ground vehicle programs, which will allow them to perform tests, analysis, and modifications. These models will be subject to appropriate data management standards and only be accessible to those users with applicable access permissions. This repository of common, verified information will be a major contribution of the HPCMP CREATE™-GV project, even aside from the actual simulation and analysis capabilities.

A full explanation of the many features of this extensive software is beyond the scope of this paper, but the interested reader may see [9-12].

Powertrain Analysis Computational Environment

The Powertrain Analysis and Computational Environment (PACE) is a high-fidelity behavioral model of a vehicle powertrain coded in C++ with appropriate wrappers to execute as a *Module* of Mercury. Each PACE executable represents a different vehicle powertrain, such as a conventional internal combustion engine connected to a transmission, a series hybrid electric, or an integrated start-stop mild hybrid powertrain.

PACE does not model the vehicle suspension or vehicle-terrain interaction, therefore, this information is provided through the Chrono vehicle dynamics *Module*, as illustrated in figure 3. The PACE *Module* outputs the drive shaft torque data to the Chrono *Module*, while the PACE *Module* receives drive shaft speed data from the Chrono *Module*. In addition, PACE receives input from the Driver *Module* of Mercury to control powertrain demand based on a drive cycle [13].

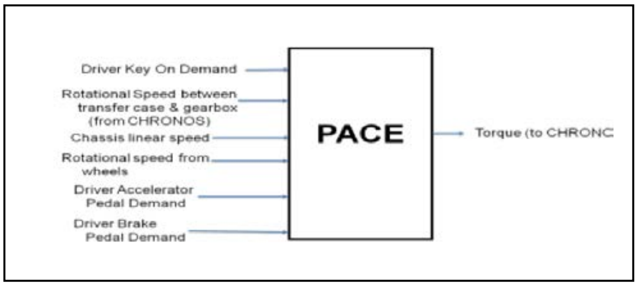

Figure 3: Illustration of inputs and output from the current version of PACE. Future versions will expand to accommodate additional inputs and outputs as needed [13].

The number of inputs and outputs is expected to expand in future revisions to accommodate increasingly comprehensive needs for full spectrum analysis of concept designs. For example, thermal modeling is being incorporated into PACE to ensure thermal limits of powertrain performance are self-consistently considered [14]. Potential outputs to the Mercury Driver *Module* include a temperature gauge that would allow the driver to react to engine temperature during aggressive drive cycles to reflect practical limits. Presently the PACE thermal model sets flags to indicate violations of powertrain thermal limits and self-consistently computes additional powertrain fuel consumption imposed by the parasitic load of the thermal management system [15].

PACE is a high-fidelity behavioral model, while each powertrain architecture has origins based on an original Simulink model translated with a custom work flow as first reported in [13]. This allows for two important features consistent with the HPCMP CREATE™-GV principles:

- Many different sources of valid powertrain models can be integrated as a PACE powertrain model. Support for third-party Simulink models is an underpinning feature of the PACE work flow.
- PACE operates with an open source license based on BSD-3 which allows unlimited cloning of the PACE executable on large cluster computers.

PACE is highly adaptable to support broad trade space evaluations. While each PACE powertrain model is fixed to a specific interconnection of powertrain components, it is a simple matter to change the type and capacity of those components through JSON format initialization files linked to the PACE source code at time of compilation. For example, the rating of an engine in the PACE conventional powertrain model can be scaled over a significant range. Transmission ratings can also be paired with the engine since each JSON initialization file is a complete specification within each instantiation of the same powertrain model. If a different powertrain is to be considered (e.g., a parallel hybrid in addition to a conventional powertrain), then additional executables are compiled based on a selection from the list of supported powertrain models. The number and types of available powertrain models supported by PACE is expected to increase as part of Mercury's continuing development.

Since the origins of any PACE powertrain model can be traced back to its originating Simulink model, model verification is a key part of the work flow. Examples of model sources range from Simulink-based powertrain modeling environment derived from established powertrain simulation environments (see [13]) to proprietary models supplied by U.S. Army contractors. As part of the PACE validation process, base cases are run using the Simulink versions, the results of which are compared to the results generated by the completely independent C++ code created from the PACE work flow. A variety of statistical measures are considered before the PACE C++ code is included for use in HPCMP CREATE™-GV. The initial test is to overlay two time series of the same variable computed for the Simulink and the PACE C++ versions of the same model to identify major inconsistent results, which are addressed first. After passing the overlay test, regression and statistical tests are applied to the two sets of results computing for the same variable to detect subtle systematic differences within the normal numerical noise associated with the completely different computational environments in which the two models are solved.

Only after the overlay and statistical tests are passed, then the resulting verified PACE C++ code is released for use within Mercury. Validation of the PACE code is based on the validity of the Simulink models that the PACE models are derived from.

Ground Contact Element and Vehicle-Terrain Interaction

The Vehicle Terrain Interaction (VTI) software performs simulations of the interaction of the track or the tire with the terrain.

The Ground Contact Element (GCE) model is part of the tire/terrain *Module* in the Mercury software that calculates vehicle-terrain interaction forces. At each time step, the state of the wheel (position, velocity, and angular velocity) is input into the GCE calculation, while the GCE returns the forces and torques on the wheel hub.

In order to perform the calculation, the coordinates of the wheel are first transformed into the system shown in Figure 4. The GCE then calculates the forces along each of these principle directions, with the local $\hat{x}$ direction also referred to as the longitudinal direction, the local $\hat{y}$ referred to as the lateral direction, and the local $\hat{z}$ referred to as the normal direction.

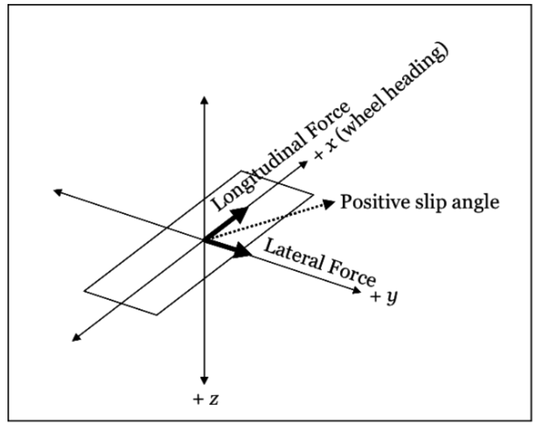

Figure 4: Coordinates used by the GCE.

Forces are calculated as a function of the wheel properties (radius, width, section height, and inflation pressure), soil properties (type, strength, and condition), tire slip, and slip angle. The tire slip is denoted s and is defined by

$$s = 1 - \frac{\omega r_{eff}}{V_v} \quad (1)$$

where V_v is the velocity of the vehicle and ω is the angular velocity of the wheel, and reff is the rolling radius. The slip angle, α is defined as

$$\alpha = \tan^{-1}\frac{V_y}{V_x} \quad (2)$$

where V_x and V_y are the longitudinal and lateral components of the vehicle velocity in the wheel's local coordinate system as defined in Figure 4.

The general goal of the VTI model is to calculate the force on the wheel in the local coordinate system with an equation of the form

$$F_{VTI} = f(s, \alpha, F_n, tireData, terrainData) \quad (3)$$

where F_n is the normal force on the tire.

The terrain properties included in GCE are the soil type as defined by the Unified Soil Classification System (USCS) [16], the soil strength as quantified by the Remold Cone Index (RCI) in units of PSI, and the soil condition (slippery or unslippery). In this article, for the sake of brevity and clarity, only the GCE equations for fine-grained soils (silts and clays with USCS soil codes ML, CL, OL, MH, CH, and OH) in the unslippery condition are listed.

In the GCE model, the VTI forces in the plane of the terrain surface are split into traction (*T*) and resistance (*R*) terms in the longitudinal and lateral directions such that

$$F_{VTI} = T_{\parallel} + T_{\perp} - R_{\parallel} - R_{\perp} \tag{4}$$

The normal force Fn (relative to the terrain surface) is calculated at each time step by the tire model and used as input to the VTI equations.

Tire Model

The tire model is used in GCE to calculate the normal force on the wheel for input into the VTI equations. The tire model is three-dimensional (3D) and can be used with a 3D triangular mesh of a surface geometry to capture the "enveloping" effects of the tire on small (relative to the contact patch of the tire) obstacles, as well as the forces from larger terrain undulations.

The GCE models the tire by dividing the tire into multiple two-dimensional (2D) cross-sections, or slices. Each slice is further divided into nodes that are used to check for contact with the terrain mesh and calculate the force on the tire patch using a linear spring model.

Much of the tire model involves geometrical calculations of which detailed equations are beyond the scope of this paper. A more extensive description of the GCE model can be found in [17]. The geometrical portion of the problem is summarized in Figure 5.

For each slice of the tire, the terrain mesh is sampled at each node position to create a 2D profile, as shown in the top of Figure 5. The area of the shaded region is calculated, as well as the corresponding section length (L_S) that would yield an equal area. From this section length, the deflection, δ is calculated. The normal force is then determined using a linear spring model for the tire with the following equations (see Figure 6 for explanation of geometry and terms).

$$\cos\theta = \frac{r_{eff}}{r} \tag{5}$$

where r is the undeflected radius of the tire and r_{eff} is the effective radius of the tire such that $r_{eff} = r - \delta$. The normal force is then given by

$$F_n = 2k(r\sin\theta - r_{eff}\cos\theta) \tag{6}$$

where k is the spring constant of the tire and can be calculated from any point on the deflection vs. load curve for the tire.

The normal force F_n is calculated for each slice of the tire and the results are summed to yield the total normal force for input into the VTI algorithms, along with the deflection δ.

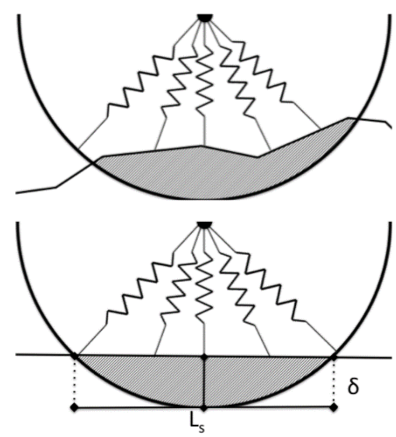

Figure 5: GCE tire model. The enveloped area (shaded region in upper figure) is used to determine an equivalent deflection such that the shaded regions in the top and bottom figure have the same area.

Fine Grained Soil Model

The traction and resistance equations discussed in this section utilize the concept of a numeric, a single value accounting for the characteristics of the tire and soil that can be used as input to a variety of equations to predict traction, resistance, and sinkage. The traction and resistance equations given in this section use the partial clay numeric, or PNC [18].

$$PNC = \frac{RCI \cdot B \cdot D}{W(1.0-\delta)^{1.5}\left(1.0+\frac{B}{D}\right)^{0.75}} \quad (7)$$

The equations presented in the next two subsections are based on a wealth of empirical data. The derivations and interpretations are beyond the scope of this paper. Additional information on the original data that were used to produce these equations can be found in [19].

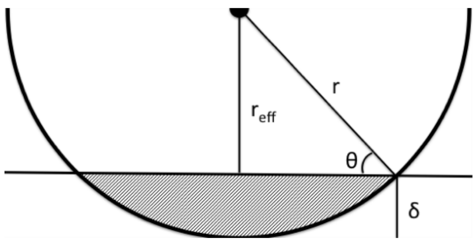

Figure 6: Geometrical definitions for the tire model calculation.

Traction

The traction model consists of the following calculations, which will be elaborated in the subsequent equations.

1. Steering adjusted clay numeric, *NC*
2. Motion resistance, *R*
3. Drawbar Pull using *NC* and s_{off}
4. Max tractive force, *T*, using coefficients from lookup table
5. Steering adjustment to the longitudinal tractive force, $T_{||}$
6. Lateral tractive force, $T_{\perp}$ with Crolla [20] model

The adjusted fine grain numeric for traction, NC_T, is calculated from the soil RCI and the partial clay numeric.

$$NC_T = RCI \cdot PNC \quad (8)$$

The motion resistance is given by

$$R/W = \frac{12.0}{NC_T^2} + 0.007 \quad (9)$$

The slip at the "self-propelled condition" is given by [18]

$$s_{sp} = \frac{21}{NC_T^{2.5}} + 0.005 \quad (10)$$

And then used to calculate the drawbar pull *(D/W)*

$$D/_W = \sqrt{\log_{10}(S/_{s_{sp}})} \quad (11)$$

The tractive force coefficient is given by the equation

$$T\left(D/_W, F_n, PNC\right) = \frac{A_1}{s-A_3} - A_2 \quad (12)$$

where A_1, A_2, and A_3 are numerical values from a lookup table that depends on soil type, soil condition (slippery or dry), vehicle contact pressure, and the drawbar pull value, D/W. The values in the lookup table are derived from the report by Priddy [19].

Steering effects are calculated in the Mercury implementation using the model of Crolla [20]. In this model, the longitudinal tractive force is set equal to the total tractive force.

$$T_{||} = T \quad (13)$$

and the lateral tractive force is calculated from the theoretical maximum tractive force T_{max}(Eq. 18 with *s* = 1.0) and the slip angle.

$$T_{\perp} = \sqrt{T_{max}^2 - T^2}\left(1.0 - e^{-\left|\frac{\alpha_{deg}}{7.0}\right|}\right) \quad (14)$$

where α_{deg} is the slip angle converted from radians to degrees.

Resistance

The following equations to determine the resistance due to steered, powered wheels comes from [21]. The steering adjustment NC_R to the fine grain numeric, PNC, is given by

$$NC_R = RCI \cdot PNC(1.0 - 2.26|\alpha|^{1.5}) \quad (15)$$

The longitudinal resistance is then given by

$$R_{||} = \frac{12.0}{NC_T^2} + 0.007 \quad (16)$$

and the lateral resistance is given by

$$R_{\perp} = \frac{15.4\alpha}{\alpha_5}(1.0 - \frac{BC}{NC_R - 7.0 - BC}) \quad (17)$$

where

$$BC = \frac{4}{\sqrt{|\alpha|}} \quad (18)$$

For a more complete discussion of VTI, see [22] and references therein.

Driver Model

The Mercury Driver *Module* provides for optimized control of complex vehicle maneuvers for a variety of other tests (including steering stability test, and the NATO lane change test), which will result in VCI, ride, and shock values. The Driver *Module* architecture allows for the later addition of dynamic control systems, such as anti-lock brakes and traction control systems.

Mercury Tests

There are a multitude of *Tests* implemented within Mercury that calculate vehicle performance characteristics, including:

1. Drop Test: calculates the tire loads and vehicle weight
2. Max Speed Test: measures the maximum speed of the vehicle either on flat terrain or at a slope
3. Ride Test: quantifies the 6-watt speed limit by simulating over 61 different courses
4. Shock Test: creates a shock curve by driving over various sizes of half rounds at eight speeds and measuring the maximum shock at the driver's seat
5. Sand Slope Test: measures the maximum slope the vehicle was able to negotiate
6. Soft Soil Test: calculates the VCI_1
7. Tilt Table Test: assessment of the vehicle weight distribution
8. Rollover Stability Test: calculates the maximum speed the vehicle can maintain a constant radius circle
9. NATO Double Lane Change Test: measures the maximum speed that a vehicle can complete the lane change maneuver
10. J-turn Test: measures the maximum speed that the vehicle can complete a J-turn maneuver
11. Drawbar Pull Test: measures the tractive force versus speed of the vehicle

Mercury with Functional Mockup Interface

The Functional Mockup Interface (FMI) is an open-standards specification that describes how a model can be created in order to perform model-exchange or co-simulation with independently developed simulation environments. The FMI specification aims to break down proprietary model formats and to utilize a unified file model that is compatible among industry, academic, and government simulation codes. There are two versions of FMI currently in use. FMI version 1.0 was released January of 2010 and FMI version 2.0 [23] was released July 2014. Version 1.0 Co-simulation is currently being used to format a vehicle model generated from Chrono for use by Mercury.

The FMI is comprised of a number of files zipped into a single file called a Functional Mockup Unit (FMU), typically using the file extension ".fmu".

Inside the FMU, a compiled dynamic library of the FMI-wrapped model implementation, along with dependent libraries, documentation, and related data files are organized in a tree structure. An XML file named "modelDescription.xml" defines the inputs and outputs of the model as continuous and discrete "states".

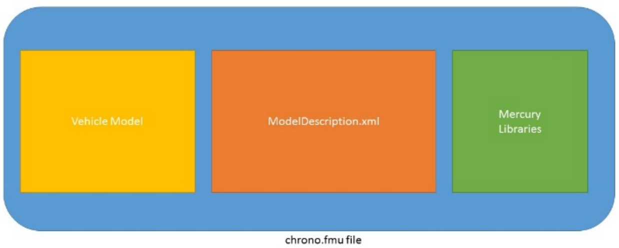

Figure 7: Example of an FMU

The FMI specification defines a standardized set of functions to be implemented either by the developer or through an FMU export tool of the simulation software. To avoid Intellectual Property (IP) issues, a compiled code within an FMU does not require the existence of source code, allowing commercial vendors to export their compiled/executable codes along with the model without exposing IP.

In Mercury, the goal is to import FMUs by each vehicle subsystem solver (Chrono vehicle dynamics, PACE powertrain, VTI, and Driver) and integrating all FMUs such that Mercury will be the communications mechanism that passes vehicle subsystem states. Thus, the FMI will provide a degree of modularity. For example, if a vehicle dynamics model from Chrono is currently being used, a different model exported from an FMI-compatible software for vehicle dynamics can be readily included without having to change the Mercury code, as long as the same states from the "modelDescription.xml" file are available.

The development plan is to first create a feasibility prototype of an FMU using Chrono as the basis of an exported model. The code for Mercury will be modified to be able to import FMU models. Once model importation is possible, other FMU models will be created by their respective solvers and integrated into the Mercury architecture. A method for the solver to export the model will be developed, thus ensuring Mercury is FMI compliant.

VERIFICATION AND VALIDATION (V&V) APPROACH AND PHILOSOPHY

Verification and validation of the HPCMP CREATE™-GV analysis tools is an essential process toward releasing a qualified software product. In order for the results to drive acquisition and engineering decisions, users must have reason to trust that the analysis answers are accurate. The following sections will present both the methods and results of a number of verification and validation efforts that have been performed on the Mercury set of tools with results of real-world physical testing. This involves detailed physics modeling of several ground vehicles exhibiting unique features and performance characteristics, as well as the vehicle subsystems corresponding to each of the Mercury major software components. These verification and validation processes helped to drive software and model development, resulting in the current version of Mercury showing impressively close matching of

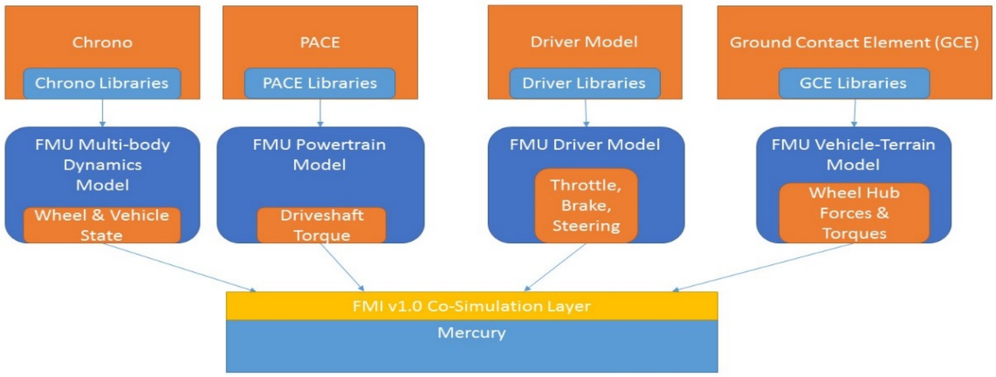

Figure 8: Mercury framework with FMI

results with both commercial and government analysis tools, and with performance data derived from physical testing.

User Stories and Scenarios

The approach HPCMP CREATE^{TM}-GV applies to driving software specifications is the practice of developing user stories and scenarios that are comprised of viewpoints from the user's perspective. User stories and scenarios are purposeful in-person engagements, usability studies targeting user's goals, performance targets and specific scenarios. These interactions capture the user role or type of user, draw out the goals or desired tasks they want to perform, and document the reason or desire to reach those goals.

The holistic view describing the user scenarios for HPCMP CREATE^{TM}-GV as they relate to supporting its intended uses with traceability to the CGD, are listed below:

1. Requirements Analysis
2. Analysis of Alternatives
3. Concept Performance Analysis
4. Trade Space Analysis
5. Technology Insertion Analysis
6. Source Selection Analysis
7. Solve Specific Problem Analysis

Verification and Validation Software Practices

The process of documenting, writing, and effectively communicating the needs of the ground vehicle community, which is captured in the CGD, formed the foundation for HPCMP CREATE^{TM}-GV. Subsequently, it was just as important to capture the user's perspective and scenarios. The Verification, Validation and Uncertainty quantification (VV&UQ) process is an integral process in releasing an engineering software solution. As we near the end of our concept development cycle we are in the process of intense software VV&UQ with the following objectives:

1. Establish simulation credibility and demonstrate its adequacy
2. Incorporate methods in quantifying uncertainty and using it as a measure of accuracy
3. Follow prescribed best practices in software VV&UQ by the HPCMP CREATE^{TM} community

Building up the trust and credibility of a physics-based software simulation is a computationally and resource intensive endeavor and is necessary to prioritize prior to the software release.

Vehicle Simulation Cases

Vehicle simulation is used to ensure that a vehicle design adequately meets performance metrics. Military vehicle ride performance is rated based on several different metrics [25]. Most of those metrics look at some aspect of the vertical acceleration at the occupant location as that information is very important to determine if a vehicle's design is a viable option. Acceptable limits for these metrics are defined based on the levels at which a human will experience discomfort or health risks, as in the 6-watt and 2.5-g vehicle tests.

There are several key vehicle design factors that affect the vehicle ride performance. Modeling and Simulation is a cost effective strategy for measuring the effects of the key vehicle design factors on the vehicle ride performance. These key vehicle design factors include suspension characteristics, tire size, and tire inflation pressure, as well as the control algorithms used for semi-active or active suspensions.

The 6-watt absorbed power metric [17] is a measure of the power delivered to a passenger due to primarily vertical vibrations of the vehicle at the occupant location. This metric is used to determine the maximum speed at which the vehicle can be utilized over a terrain of a particular "roughness", a measure estimated by calculating the Root Mean Square (RMS) over elevation data for a given terrain section. To perform the 6-watt RMS ride quality test, the vehicle simulation has several modeled terrain courses with known RMS values that the vehicle is driven over at increasing speeds. A ride quality curve is created based on the speed and terrain roughness at which the vehicle reaches the 6-watt absorbed power limit.

The half-round vertical shock test measures the peak vertical acceleration of the vehicle in response to a single excitation event, such as hitting a curb or pothole. Again based on studies of the response of the human body to various types of acceleration inputs, it has been calculated that the peak vertical acceleration that the vehicle should encounter during a single impulse event should be less than 2.5 g after low pass filtering the signal. The modeled vehicle will be driven at a variety of speeds over a set of half-rounds that vary in size to perform the half-round vertical shock test in a vehicle simulation, to determine the maximum speed at which the vehicle can be driven safely over an obstacle of specified height.

The steady-state circular steer test is designed to measure vehicle stability. Vehicle rollover is a very complex event but there are several factors that can help predict it [26]. The steady-state circular test is designed to calculate the safe operating range of a vehicle while turning to avoid rollover. The simulation test involves a modeled vehicle driving in a circle of specified radius with steadily increasing speed, and calculates the maximum speed that does not produce wheel lift-off or oversteer.

The NATO lane change test is used to analyze the handling performance and dynamic characteristics of a vehicle, by having the vehicle change from one lane to an adjacent one and back within a specified distance. Within HPCMP CREATE™-GV, a simulated vehicle drives through a course defined by automatically defined waypoints, and the speed of the maneuver is increased until the vehicle deviates from the desired path. In this way the maximum speed at which the NATO lane change can be accurately performed is determined. The simulation approach to this test in particular offers some advantages over physical testing, notably that the "skill" and reaction times of the simulated driver can be strictly defined and held constant over multiple tests, to eliminate the element of human variance that is introduced during physical testing.

Vehicle Models

One of the unique features of HPCMP CREATE™-GV is the integration of multiple non-proprietary codebases that allows for the co-simulation of a full vehicle system. To demonstrate the accuracy of the HPCMP CREATE™-GV framework, three vehicle types were modelled with the following characteristics:

1. Demonstrator A: (a) Three axles
   (b) SLA suspension
   (c) 42,000 lbs
2. Demonstrator B: (a) Two axles
   (b) SLA suspension
   (c) 32,000 lbs
3. Demonstrator C: (a) Two axles
   (b) Suspension Type
   (i) one Trailing Arm
   (ii) one SLA
   (d) 5,000 lbs
   *Short Long Arms (SLA)

Although the modelled vehicles represent a small aggregate of the various vehicle systems/platforms within the DoD's ground vehicle inventory, the varying characteristics highlight the ability to produce accurate performance output.

Vehicle Simulation Scenarios Assessment

The first simulation scenario incorporates real world test results with vehicle Demonstrator A in a ride quality assessment test scenario. The premise of this simulation is to evaluate the vehicle's suspension performance and the ride dynamics within the crew compartment across terrains with varying roughness.

The methodology to calculate ride quality of a particular vehicle is currently a two-step approach. First, the vehicle is simulated at various speeds across 61 different terrains with varying course roughness between 0.35 and 5 inch RMS. This process produces approximately 305 different data points (depicted in blue) which describe the relationship between vehicle speed and vertical absorbed power for a particular terrain – figure 9 shows the speed versus absorbed power distribution.

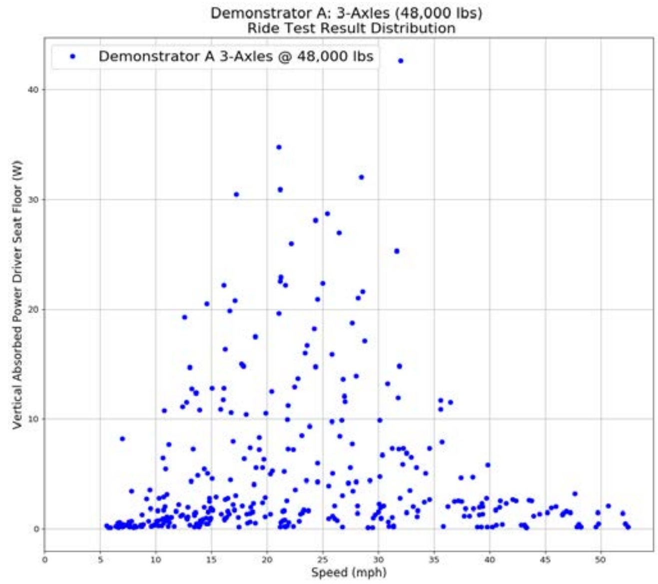

Figure 9: Vertical Absorbed Power (W) versus Speed (mph) Result Distribution

Since we are interested in determining the 6-watt speed for each of the 61 courses, we need to utilize the results attained previously (speed vs absorbed power per course) and incorporate them into the second step.

The second step utilizes the calculated speed versus absorbed power and interpolates the 6-watts speed threshold using a 2nd order polynomial as shown in figure 10. The blue dotted-line represents interpolated simulation speeds for a given terrain course at the driver's floor location. Where both the green dashed-line and the black dotted-line represent the vehicle's real world measured 6-watt threshold at the driver's seat pad and floor locations, respectively.

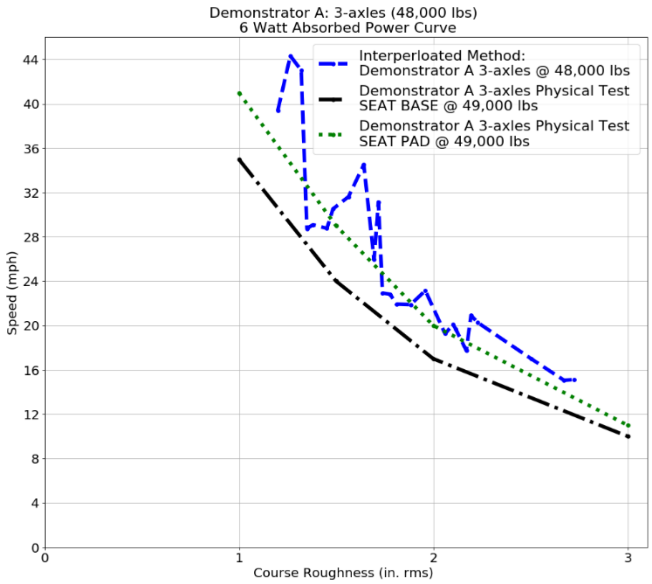

Figure 10: Vertical Absorbed Power at 6 watts versus Speed (mph) Ride Curve

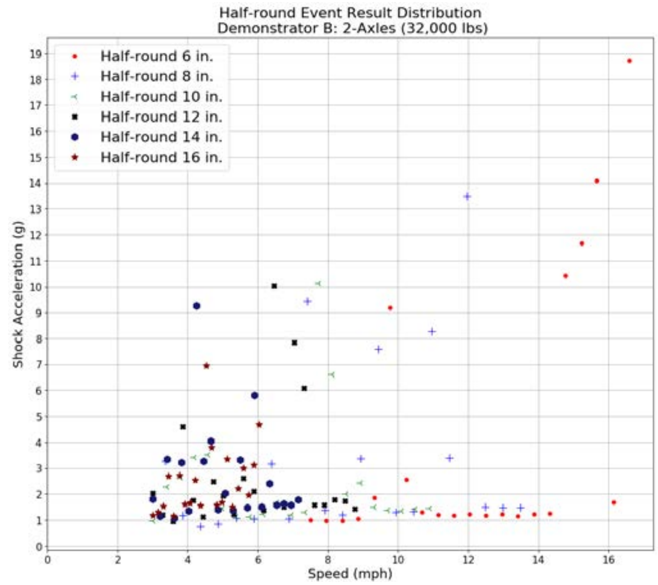

Figure 11: Vertical Acceleration (g) versus Speed (mph) Half-round Result Distribution

The two-step calculation exhibits general agreement with real world test data.

The second test case scenario incorporates vehicle Demonstrator B traversing various sized non-deformable half-rounds while measuring the vehicle's speed and vertical accelerations within the vehicle's crew compartment typically at the driver's floor location or seat. The methodology to calculate the vehicle's maximum speed threshold at the 2.5 g vertical acceleration limit utilizes the same approach as in the ride quality assessment. The first step produces a vertical acceleration versus speed distribution as shown in figure 11. In this test, the vehicle was simulated to traverse half-round obstacles from 6 to 16 inches in 2 inch increments which are designated by a particular marker type and color.

As previously mentioned, the second step interpolates the maximum speed required to achieve a 2.5 g vertical acceleration threshold at the driver's floor location. In figure 12, the blue dashed-line depicts the calculated simulation's 2.5 g acceleration threshold at a particular speed for a given half-round obstacle while the black dotted lines depict the maximum and minimum vertical accelerations for a given speed in respect to a particular half-round. In the simulation, the results were prepared for three half-round events varying in size 6 to 10 inches. In general, the trend of the simulation results correlate well with real-world tests.

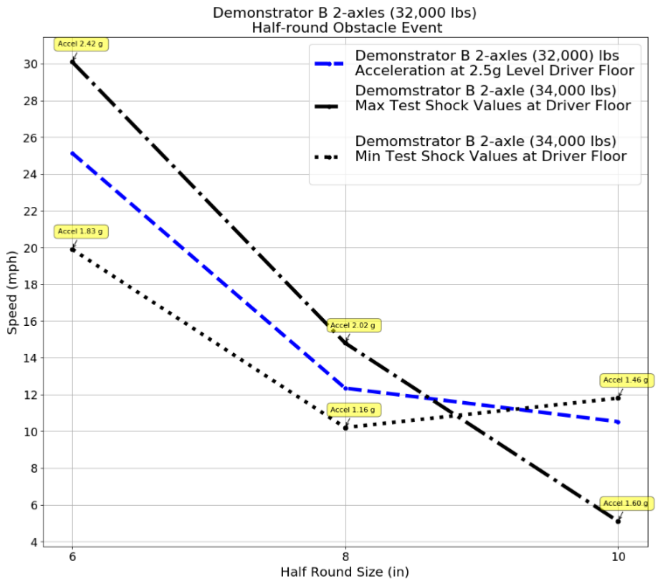

Figure 12: Shock-Limited Achieved Speeds at the 2.5g Threshold

The third simulation represents a NATO lane change maneuver. Unlike the previous two scenarios, vehicle Demonstrator C is a much lighter vehicle utilizing two different suspension designs for its front and rear axles. That is, the front axle utilizes a SLA type suspension while the rear axle is composed of a trailing arm type suspension.

The NATO lane change maneuver is administered to investigate transient response stability of a vehicle. The maneuver's gate dimensions are based on the AVTP-03-160W specification and is a performance requirement for many ground vehicle systems. The assessment criteria for the test is to successfully pass the course by transitioning from the right lane to the left lane and back again while measuring the forward velocity between entering and exiting the course's gates and recording the fastest speed achieved. Ancillary to measuring maximum speed, the vehicle will typically be instrumented with an inertia measuring unit (IMU) to record accelerations in the x, y and z directions from the vehicle's center of gravity (CG) location, as well as actual vehicle steering, roll angle, lateral acceleration and yaw rate.

The path taken by the Driver *Module* and the course dimensions are shown in figure 13. The black solid line depicts the actual path of the vehicle while the red dashed-line are the way points for the pure pursuit with enhanced lateral path tracking control. Although, very stable for course events such as the steady-state circular steer test as seen in figure 14, it has limitations, in that the pure pursuit algorithm

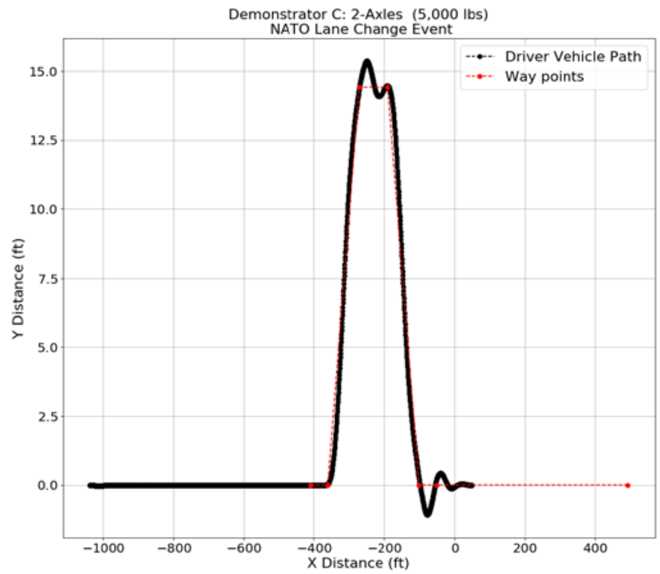

Figure 13: NATO Lane Change Maneuver Utilizing a Pure Pursuit Path Tracking Algorithm

assumes a perfect response to a curvature, hence there is a slight overshoot in the path when abrupt changes in the path occur. It is also important to mention, that physical tests of this course will produce error that is vehicle and driver dependent. Further work is needed to determine if correlation can be improved, however, the oversteering is minimal, with the deviation from desired path of approximately 12 inches – well within the tolerances of the NATO lane change specification.

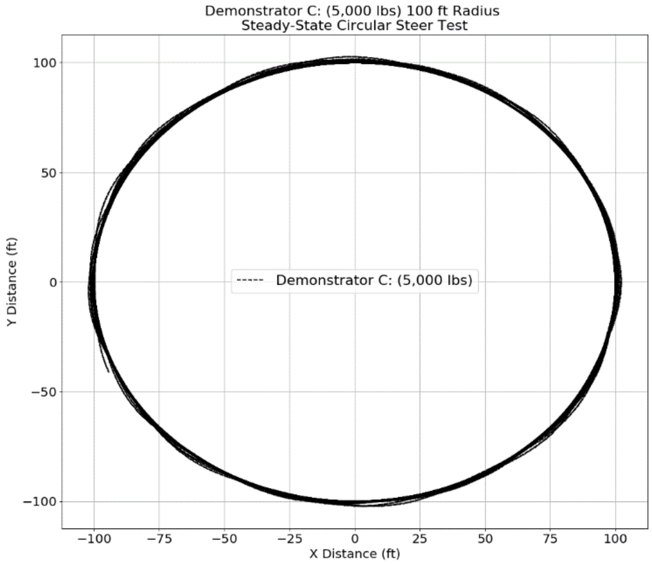

Figure 14: Path Following Control for Steady-State Circular Steer Test Example

The Driver *Module* in conjunction with the Chrono and *GCE Modules* correspond well when comparing the speed of the Demonstrator C vehicle to test data while negotiating a NATO Lane Change event at 40 mph as seen in figure 15.

One of the difficulties of the NATO Lane Change maneuver is the driver's influence (both human and robotic) on vehicle test results. The advantage our implementation has is that the Driver *Module's* control strategy will drive the vehicle the same way every time.

Subsequent figures 16 and 17 highlight agreeable simulated dynamic responses of the vehicle as compared to real world test recordings, with examples of simulated roll angle and lateral acceleration responses, respectively. Moreover, the Driver *Module's* effects on the measured response of the steering angle are noticeable but not detrimental to the overall trend of test results. The additional needed response by the Driver *Module* to maintain the desired waypoint had minimal impact on amplitude of lateral acceleration and roll angle measurements.

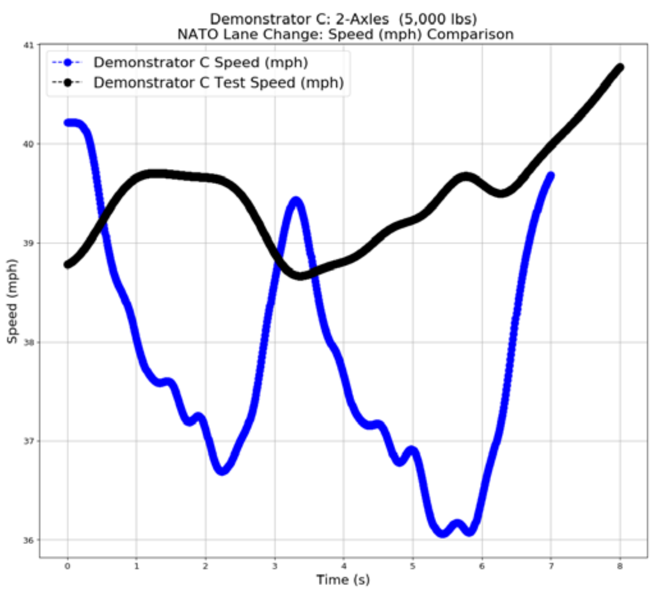

Figure 15: NATO Lane Change Maneuver Achieved Speed Comparison

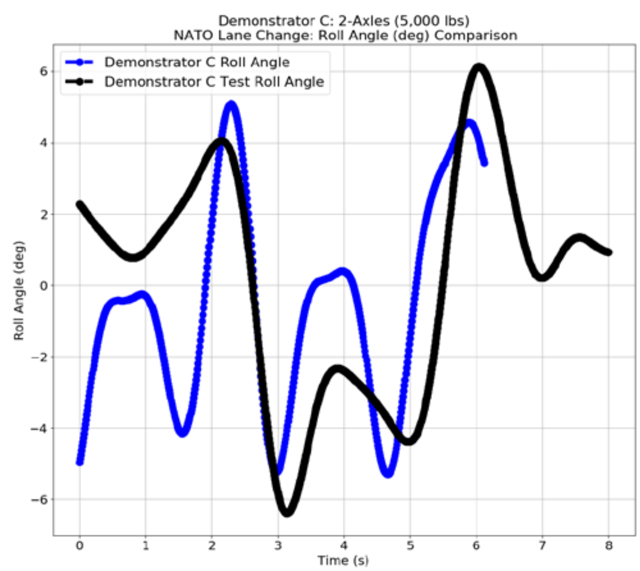

Figure 16: Vehicle's Roll Angle (deg) Response Performing the NATO Lane Change Maneuver

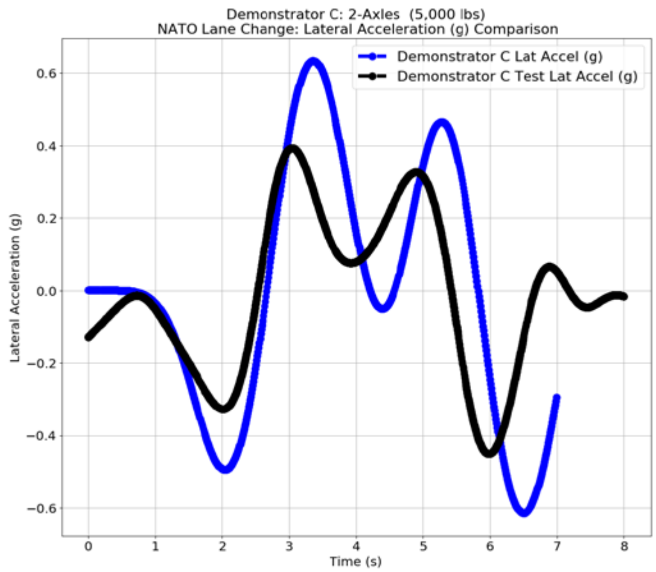

Figure 17: Vehicle's Lateral Acceleration (g) Response Performing the NATO Lane Change

CONCLUSION

The HPCMP CREATE™-GV suite of tools is positioned to improve the acquisition paradigm for U.S. Army ground vehicle mobility analysis. In particular, the use of HPC systems as developmental and execution platforms allows for massively parallel software that can enable the exploration of large trade spaces of vehicle designs, in order to provide early input to acquisition programs in addition to detailed analysis of already-established platforms. Extensive and ongoing verification and validation efforts are demonstrating the accuracy and reliability of the analysis metrics calculated by the various software components. As the program moves forward and additional models, sub-systems, and capabilities are added, the HPCMP CREATE™-GV software will continue to enable next-generation mobility analysis for all types of ground vehicle platforms.

HPCMP CREATE™-GV 1.0 Release

Version 1.0 of the HPCMP CREATE™-GV software will be released on September 30, 2017. The release will include a fully functional GVI that users can utilize to create MAT or Mercury vehicle models and launch jobs on the HPC. GVI will also have the ability to fetch and display results from the HPC once the job has been completed.

Mercury 1.0 will utilize PACE for the powertrain *Module*, Chrono for the vehicle dynamics *Module*, GCE for the vehicle-terrain interaction *Module*, and a driver model developed by the Mercury team for the driver *Module*. Mercury also has a variety of *Tests* that the user will be able to run by specifying the *Test* and the metrics through GVI. Currently, the Mercury vehicle models are limited to wheeled vehicle models.

Future Plans

The existing *Tests* in Mercury 1.0 allow for a wide variety of *Tests* to be implemented but several major new capabilities are planned for future releases, including, but not limited to:

1. *Simulation of Tracked Vehicles* – single- and double-roadwheel tracks, pin tracks, and rigid and flexible tracks
2. *Active Suspension* – adds forces and torque elements to the springs in the suspension subsystems based on active suspension algorithms
3. *Electronic Stability Control / Anti-lock braking systems* – generate additional forces and torques on the wheel hubs

ACKNOWLEDGEMENTS

The authors would like to acknowledge that the material presented in this paper is a product through the sponsorship by the U.S. Department of Defense High Performance Computing Modernization Program Office. The authors would also like to thank the guidance and support provided by management and technical staff from TARDEC and ERDC.

DISCLAIMER

Reference herein to any specific commercial company, product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply endorsement, recommendation, or favoring by the United States Government or the Department of the Army (DoA). The opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or the DoA, and shall not be used for advertising or product endorsement purposes.

REFERENCES

[1]US Department of Defense, Defense Acquisition Guidebook (Fort Belvoir, Virginia: Defense Acquisition University, Acquisition Community Connections, 2009), Para 1.2.

[2]US Army, TRADOC Regulation 71-20, October, 2009, 3-4.

[3]Col. P.W. Burden, "Acquisition Reform – What's Really Broken in Defense Acquisition", U.S. Army War College, Carlisle Barracks, PA 17013-5050.

[4]R. Smith and R. Jones, "CRES-GV: Capabilities and Gap Document for Ground System Design and Development", ERDC, November, 2012.

[5]R.B. Ahlvin and P.W. Haley, "NATO Reference Mobility Model edition II, NRMM II User's Guide." Technical Report No. GL-92-19, US Army Engineer Waterways Experiment Station, Vicksburg, Mississippi. 1992.

[6]H. Mazhar, T. Heyn, A. Pazouki, D. Melanz, A. Seidl, A. Bartholomew, A. Tasora, D. Negrut, "Chrono: a parallel multi-physics library for rigid-body, flexible-body, and fluid dynamics", *Mech. Sci. Vol. 4, No. 1*, pages 49-64, 2013.

[7]M. Taylor, R. Serban, D. Negrut, "Basic Comparison of Chrono::Vehicle and ADAMS/Car", Technical Report TR-2016-15: http://sbel.wisc.edu/documents/TR-2016-15.pdf, Simulation-Based Engineering Laboratory, University of Wisconsin-Madison, 2016.

[8]M. Taylor and R. Serban, "Validation of Basic Modeling Elements in Chrono", Technical Report TR-2015-05: http://sbel.wisc.edu/documents/TR-2015-05.pdf, Simulation-Based Engineering Laboratory, University of Wisconsin-Madison, 2016.

[9]A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut, "Chrono: An Open Source Multi-Physics Dynamics Engine," T. Kozubek (Ed.), High Performance Computing in Science and Engineering - Lecture Notes in Computer Science, Springer, pp. 19-49, 2016.

[10]A. Tasora, "Time integration in Chrono::Engine", ProjectChrono Technical Documentation, September, 2016.

[11]A. Tasora, "Kinematics of moving frames in Chrono::Engine", Chrono::Engine Technical Documentation, March, 2016.

[12]A. Tasora, "Rotations in Chrono::Engine", Chrono::Engine Technical Documentation, March, 2016.

[13]Haupt et al, SAE World Congress 2017.

[14]Gabe et al, NATO AST-265, 2017.

[15]Gabe et al, GVSETS 2017.

[16]ASTM, D. 2487-00 Standard classification of soils for engineering purposes (Unified Soil Classification System). Annual Book of ASTM Standards, Section 4.

[17]D.C. Creighton, "Revised vehicle dynamics module: user's guide for computer program VEHDYN II", US Army Engineer Waterways Experiment Station, Technical Report SL-86-9, 1986.

[18]G.W. Turnage, "Mobility Numeric System for Predicting In-the-Field Vehicle Performance, Report 1, Historical review, Planned Development", Miscellaneous paper: GL-95-12,

US Army Engineer Waterways Experiment Station, December, 1995.

[19]J.D. Priddy, "Stochastic Vehicle Mobility Forecasts Using the NATO Reference Mobility Model, Report 3, Database Development for Statistical Analysis of the NRMM II Cross-Country Traction Empirical Relationships." WES/TR/GL-95-8. Army Engineer Waterways Experiment Station, Vicksburg MS, Geotechnical Lab, 1995.

[20]D. Crolla, A.S.A. El-Razaz, "A Review of the Combined Lateral and Longitudinal Force Generation of Tyres on Deformable Surfaces", Journal of Terramechanics, Vol. 24, No. 3 pages 199-225, 1987.

[21]G.N. Durham, "Powered Wheels in the Turned Mode operating on Yielding Soils", Technical Report: M-76-9, US Army Engineer Waterways Experiment Station, September, 1976.

[22]R.A. Jones, G.B. McKinley, P.W. Richmond, D.C. Creighton, R.B. Ahlvin, P. Nunez, "A vehicle terrain interface," In Proceedings of the joint North America, Asia-Pacific ISTVS conference and annual meeting of Japanese society for terramechanics, 2007.

[23]T. Blochwitz, M. Otter, J. Akesson, et. al., "Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models", 9th International Modelica Conference 2012 (https://fmi-standard.org/literature).

[24] Boehm, B.W., "Using Risk to Balance Agile and Plan-Driven Methods," IEEE Computer Society, 2003.

[25] Test Operations Procedure (TOP) 1-1-014 Ride Dynamics, Aberdeen Proving Ground, Defense Technical Information Center (DTIC)

[26] A. Hac, "Rollover Stability Index Including Effects of Suspension Design," SAE 2002 World Congress

LIST OF SYMBOLS, ABBREVIATIONS, ACRONYMS

%NOGO	Percent No Go - Quantifies terrain accessibility
AFWA	Air Force Weather Agency
AGC	Army Geospatial Center
BSON	Binary JSON
CGD	Capabilities and Gaps Document
CREATE	Computational Research and Engineering Acquisition Tools and Environments
CSE	Computational Science and Engineering
DAS	Defense Acquisition System
DoA	Department of the Army
DoD	Department of Defense
FMI	Functional Mockup Interface
FMU	Functional Mockup Unit
GCE	Ground Contact Element
GV	Ground Vehicles
GVI	Ground Vehicle Interface
HPC	High Performance Computing
HPCMP	High Performance Computing Modernization Program
IP	Intellectual Property
JCIDS	Joint Capabilities Integration and Development System
JSON	JavaScript Object Notation
M&S	Modeling and Simulation
MAT	Mobility Analysis Tool
MRS	Mission Rating Speeds
NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organization
NC	Clay Numeric
NGA	National Geospatial-Intelligence Agency
NRMM	NATO Reference Mobility Model
PACE	Powertrain Analysis Computational Environment
PNC	Partial Clay Numeric
PPBE	Planning, Programming, Budgeting, and Execution
QFD	Quality Functional Deployment
RCI	Remold Cone Index
SLA	Short Long Arms
V&V	Verification and Validation
VTI	Vehicle-Terrain Interaction
XML	Extensible Markup Language