EVS28
KINTEX, Korea, May 3-6, 2015

Novel Large Scale Simulation Process to Support DOT's CAFE Modeling System

Ayman Moawad[1], Prasanna Balaprakash, Aymeric Rousseau, Stefan Wild
[1]*Argonne National Laboratory, 9700 S. Cass Avenue, Lemont, IL, USA, amoawad@anl.gov*

Abstract

This paper demonstrates a new process that has been specifically designed for the support of the U.S. Department of Transportation's (DOT's) Corporate Average Fuel Economy (CAFE) standards. In developing the standards, DOT's National Highway Traffic Safety Administration made use of the CAFE Compliance and Effects Modeling System (the "Volpe model" or the "CAFE model"), which was developed by DOT's Volpe National Transportation Systems Center for the 2005–2007 CAFE rulemaking and has been continuously updated since. The model is the primary tool used by the agency to evaluate potential CAFE stringency levels by applying technologies incrementally to each manufacturer's fleet until the requirements under consideration are met. The Volpe model relies on numerous technology-related and economic inputs, such as market forecasts, technology costs, and effectiveness estimates; these inputs are categorized by vehicle classification, technology synergies, phase-in rates, cost learning curve adjustments, and technology "decision trees." Part of the model's function is to estimate CAFE improvements that a given manufacturer could achieve by applying additional technology to specific vehicles in its product line. A significant number of inputs to the Volpe decision-tree model are related to the effectiveness (fuel consumption reduction) of each fuel-saving technology.

Argonne National Laboratory has developed a full-vehicle simulation tool named Autonomie, which has become one of the industry's standard tools for analyzing vehicle energy consumption and technology effectiveness. Full-vehicle simulation tools use physics-based mathematical equations, engineering characteristics (e.g., engine maps, transmission shift points, and hybrid vehicle control strategies), and explicit drive cycles to predict the effectiveness of individual and combined fuel-saving technologies. The Large-Scale Simulation Process accelerates and facilitates the assessment of individual technological impacts on vehicle fuel economy. This paper will show how Argonne efficiently simulates hundreds of thousands of vehicles to model anticipated future vehicle technologies.

Keywords: Large Scale Simulation, Hybrid vehicles, CAFE, Autonomie

1 Introduction

In 1975, Congress passed the Energy Policy and Conservation Act, requiring standards for Corporate Average Fuel Economy (CAFE), and charging the U.S. Department of Transportation (DOT) with the establishment and enforcement of these standards. The Secretary of Transportation has delegated these responsibilities to DOT's

National Highway Traffic Safety Administration (NHTSA). DOT's Volpe National Transportation Systems Center provides support for DOT/NHTSA's regulatory and analytical activities related to fuel economy standards, which, unlike long-standing safety and criteria pollutant emissions standards, apply to manufacturers' overall fleets rather than to individual vehicle models. In developing the standards, DOT/NHTSA made use of the CAFE Compliance and Effects Modeling System (the "Volpe model" or the "CAFE model"), which was developed by the Volpe Center for the 2005–2007 CAFE rulemaking and has been continuously updated since.

Part of the model's function is to estimate CAFE improvements that a given manufacturer could achieve by applying additional technology to specific vehicles in its product line. DOT/NHTSA has made use of vehicle simulation results to update technology effectiveness estimates used by the model. In recent rulemakings, the decision trees have been expanded so that DOT/NHTSA is better able to track the incremental and net/cumulative cost and effectiveness associated with each technology, substantially improving the "accounting" of costs and effectiveness for CAFE rulemakings.

Full vehicle simulation tools use physics-based mathematical equations, engineering characteristics (e.g., engine maps, transmission shift points, and hybrid vehicle control strategies), and explicit drive cycles to predict the effectiveness of individual fuel-saving technologies and the effectiveness of combinations of fuel-saving technologies. Argonne National Laboratory, a U.S. Department of Energy (DOE) national laboratory, has developed a full-vehicle simulation tool named Autonomie [1]. Autonomie has become one of the industry's standard tools for analyzing vehicle energy consumption and technology effectiveness.

The objective of the current project is to develop and demonstrate a process that, at a minimum, provides more robust information that can be used to calibrate inputs applicable under the Volpe model's existing structure. The project will be more fully successful if a process can be developed that minimizes the need for decision trees and replaces the synergy factors with inputs provided directly from a vehicle simulation tool. The present report provides a description of the Large-Scale Simulation Process (LSSP) that was developed by Argonne National Laboratory and implemented in Autonomie to answer this need.

The Volpe model currently relies on multiple decision trees to represent component technology options, including:

- Powertrain electrification;
- Engine ;
- Transmission;
- Light weighting;
- Aerodynamics; and
- Rolling resistance

Figure 1 shows the decision trees in the Volpe model. During the simulation, the model walks through each decision tree to find the technology that should be selected next to provide the best fuel energy consumption improvement at the lowest cost.

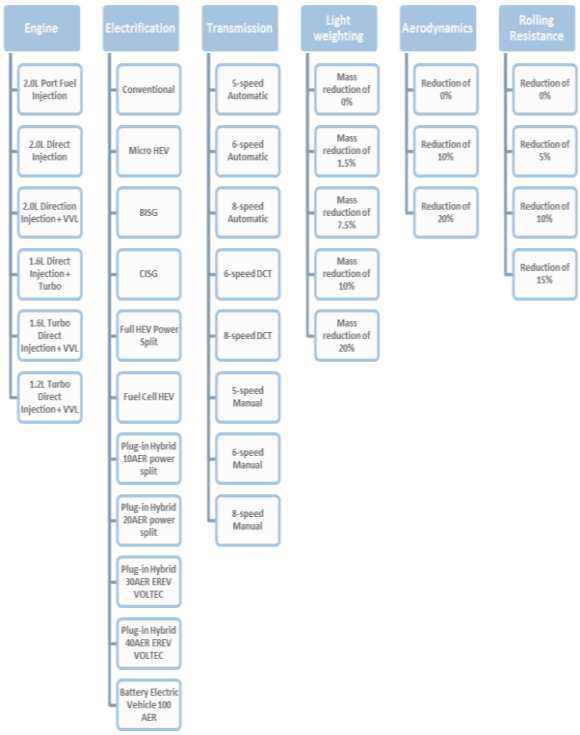


Figure 1: All technological decision trees in the Volpe model

2 Process Overview

The main objective of the present study is to provide an efficient tool for performing individual vehicle simulations. To do so, individual vehicles have to be simulated to represent every combination of vehicle, powertrain, and component technologies.

The current decision trees include

- 5 vehicle classes (Compact, Midsize, Small SUV, Midsize SUV, Pickup);
- 17 engine technologies;
- 11 electrification levels, comprising 4 no- or low-electrification levels (conventional vehicle is equivalent to no-electrification level) and 7 levels of hybridization;
- 8 transmission technologies (applied to no/low-electrification-level vehicles only);

- 5 light-weighting levels;
- 4 rolling-resistance levels; and
- 3 aerodynamic levels.

For each vehicle class,

17 engine technologies × 4 no/low-electrification levels × 8 transmission technologies × 5 light-weighting levels × 4 rolling-resistance levels × 3 aerodynamic levels = 32,640 vehicles

+

7 hybridized vehicle technologies × 5 light-weighting levels × 4 rolling-resistance levels × 3 aerodynamic levels = 420 vehicles

=

33,060 vehicles for each vehicle class.

Thus, the combination of the technologies from each decision tree leads to 33,060 simulations for a single vehicle class (or 165,300 for 5 classes) to fully populate inputs to the Volpe model.
The LSSP includes the following steps, as shown in Figure 2:

1. Collect/develop all the technology assumptions;
2. Develop a process to automatically create the vehicle models;
3. Size all the individual vehicles to meet the same vehicle technical specifications;
4. Run each vehicle model on the specified driving cycles;
5. Create a database with all the required inputs for the Volpe model; and
6. Create a post-processing tool to validate the database content.

Since this process has to be performed in an acceptable amount of time, two additional processes were developed and implemented:

- Use of distributed computing for vehicle sizing and simulation, and
- Use of statistical analysis to minimize the number of simulations that need to be performed.

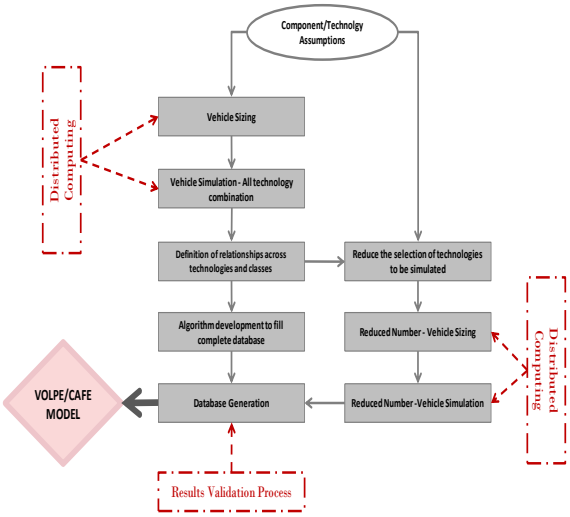


Figure 2: Overview of Large-Scale Simulation Process (LSSP)

3 Autonomie

Autonomie is a MATLAB-based software environment and framework for automotive control system design, simulation, and analysis [1] Sponsored by the DOE Vehicle Technologies Office, the tool is designed for rapid and easy integration of models with varying levels of detail (low to high fidelity) and abstraction (from subsystems to systems and entire architectures), as well as processes (calibration, validation, etc.). Developed by Argonne in collaboration with General Motors, Autonomie was designed to serve as a single tool that can be used to meet the requirements of automotive engineers throughout the development process, from modeling to control. Autonomie's ability to simulate many powertrain configurations, component technologies, and vehicle-level controls over numerous drive cycles has been used to support dozens of studies focusing on fuel efficiency, cost-benefit analysis, or greenhouse gases.

4 Individual Vehicle Setup Process

The LSSP was developed by Argonne to run a very large number of vehicles/simulations in a fast and flexible way. It allows Argonne to quickly respond to Volpe Center and DOT/NHTSA requests to be able to simulate any technology combination in any vehicle class. The following subsections describe the different steps of the process.

4.1 Vehicle Spreadsheet Definition

A template spreadsheet contains the basic information for a vehicle, such as vehicle name, vehicle class, and vehicle technology, as well as components information such as battery

technology, engine technology, and transmission type.

The template spreadsheet contains seven tabs: Vehicle Details, Parameter Values, Control Settings, Sizing Algorithm and Information, Running Procedures and Cycles, Translation for Matlab Computation, and Assumptions Details. Under each tab, columns outline vehicle configurations. Four columns refer to the four no/low-electrification-level vehicles, and 7 columns refer to the high-electrification-level vehicles.

4.2 Multi-Spreadsheet Expansion/Duplication

After the LSSP defines the spreadsheet with all the component and vehicle inputs, a multiplier code, shown in Figure 3, expands the reference/template spreadsheet into as many spreadsheets as needed to define the vehicle's technology combinations on the basis of the decision trees' input.

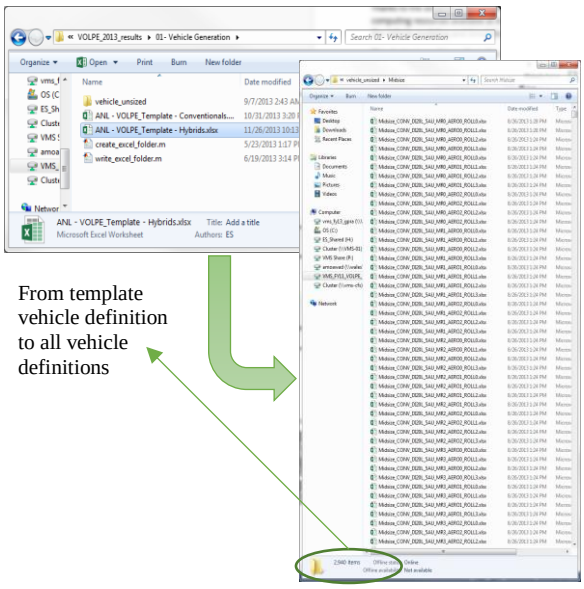


Figure 3: Multi-spreadsheet expansion/duplication

5 Distributed Computing Process

At this stage of the LSSP, all the vehicles are created and ready to be sized and simulated in Autonomie. Running 33,060 vehicles requires more than 250,000 simulations, from sizing algorithms—imposing recurrence and iteration/ looping—to vehicle simulation on cycles and combined or Plug-in Hybrid Electric Vehicle (PHEV) procedures.

With the multitude of technology combinations to simulate, the usual computing resources are no longer practical. Running all of the simulations on one computer would take several months or years before any analysis could be completed. Thanks to advances in distributed computing, simulation time can be greatly reduced. Among the computing resources available at Argonne National Laboratory is a cluster of 128 worker nodes dedicated to the System Modeling and Control Group. A larger computing facility could be used in the future to further accelerate the simulations.

5.1 Setup

The researchers of the System Modeling and Control Group use Autonomie as the simulation framework, synchronized by a cluster head node computer. The head computer extracts the data from the Excel files describing the different technology pathways and distributes it to the researchers, as diagrammed in **Error! Reference source not found.**. An algorithm optimizes the distribution of jobs for vehicle simulations and parametric studies. The total simulation time for the 33,060 vehicles was about 84 hours (3.5 days).

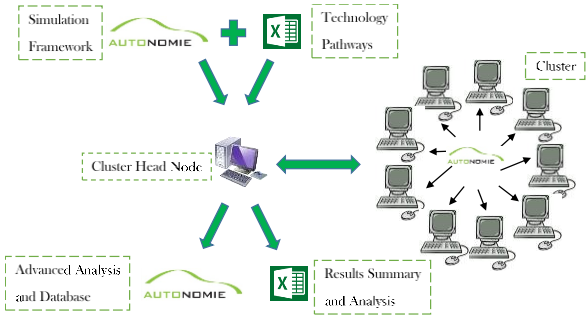


Figure 4: Diagram of distributed computing process

One of the biggest advantages of distributed computing is that it facilitates the quick rerun of simulations, which occurred many times during this study. This experience allowed Argonne to develop an ultimate LSSP that is functional, smooth, and flexible, with the ability to easily and quickly add and rerun as many vehicles and new technologies as needed. The generic process will be able to automatically handle additional technologies without any code modification. As a result, the Volpe model's future technological needs will be easily and quickly integrated at any time in order to feed the model for CAFE rulemaking analyses.

6 Vehicle Sizing Process

6.1 Vehicle Technical Specifications

To compare different vehicle technology-configuration-powertrain combinations, all vehicles to be studied are sized to meet the same requirements:

- Initial vehicle movement (IVM) to 60 mph in 9 sec ± 0.1 sec;
- Maximum grade (gradeability) of 6% at 65 mph at gross vehicle weight (GVW); and
- Maximum vehicle speed >100 mph

These requirements are a good representation of the current American automotive market and of American drivers' expectations. The relationship between curb weight and GVW for current technology-configuration-powertrain combinations was modeled and forms the basis for estimating the GVWs of future vehicle scenarios.

6.2 Component Sizing Algorithms

Owing to the impact of the component maximum torque shapes, maintaining a constant power-to-weight ratio for all configurations leads to an erroneous comparison between technologies because of different vehicle performance characteristics (i.e., time for 0–60 mph). Each vehicle should be sized independently to meet the vehicle technical specifications.

Improperly sizing the components will lead to differences in fuel consumption and will influence the results. On this basis, we developed several automated sizing algorithms to provide a fair comparison between technologies. Algorithms have been defined depending on the powertrain (e.g., conventional, power split, series, electric) and the application (e.g., HEV, PHEV). All algorithms are based on the same concept: the vehicle is built from the bottom up, meaning each component assumption (e.g., specific power, efficiency) is taken into account to define the entire set of vehicle attributes (e.g., weight). This process is always iterative in the sense that the main component characteristics (e.g., maximum power, vehicle weight) are changed until all vehicle technical specifications are met. The transmission gear span or ratios are currently not modified to be matched with specific engine technologies. On average, the algorithm takes between five and 10 iterations to converge. Figure 4 shows an example of the iterative process for a conventional vehicle.

Since each powertrain and application is different, the rules are specific:

- For HEVs, the electric-machine and battery powers are determined in order to capture all of the regenerative energy from a city cycle (UDDS). The engine and the generator are then sized to meet the gradeability and performance (time from IVM to 60 mph) requirements.
- For PHEV20s (PHEVs with 20-mi all-electric range), the electric machine and battery powers are sized to follow the city cycle in electric-only mode (this control is only used for the sizing; a blended approach is used to evaluate fuel consumption). The battery's usable energy is defined to follow the city drive cycle for 20 miles, depending on the requirements. The engine is then sized to meet both performance and gradeability requirements (usually, gradeability is the determining factor for PHEVs).

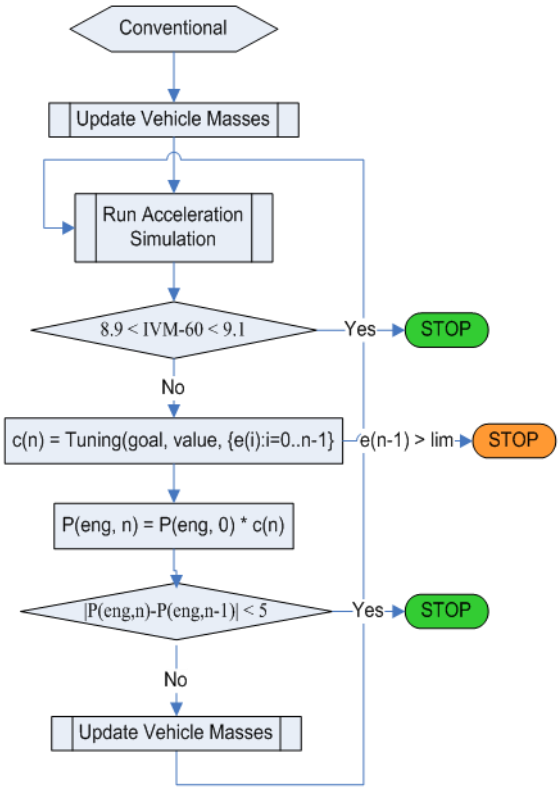


Figure 4: Conventional-powertrain sizing algorithm

- For PHEV40s, the main electric-machine and battery powers are sized to be able to follow the aggressive US06 drive cycle (duty cycle with aggressive highway driving) in electric-only mode. The battery's usable energy is defined to follow the city drive cycle for 40 miles, depending on the requirements. The genset (engine + generator) or the fuel cell systems are sized to meet the gradeability requirements.
- For Battery Electric Vehicles, the electric machine and energy storage systems are sized to meet all of the vehicle technical specifications.
- The micro-HEV, BISG (Belt-integrated starter generator), and CISG (Crank-integrated starter generator) have sizing results very similar to their conventional counterparts because they all use the same sizing rule. [2] [3]

7 Vehicle Simulation Process

Once the vehicles are sized to meet the same vehicle technical specifications, they are simulated on the appropriate standard driving cycles (33,060 vehicles or >250,000 runs). It is important to properly store individual results as structured data, because they will be reused to support database generation.

Figure 5 shows the folder organization for each individual simulation. Each folder contains the results for one combination and characterizes one branch/path of the tree. Folders can contain up to five directories, depending on the vehicle technology and the type of run performed. Results are divided into directories representing the cycle or procedure simulated. For example, the combined procedure for conventional vehicles has two parts separating the city and highway runs, and the PHEV procedure has four parts separating the city and highway runs as well as the charge-sustaining and charge-depleting modes. The last directory is the sizing structure (performance test).

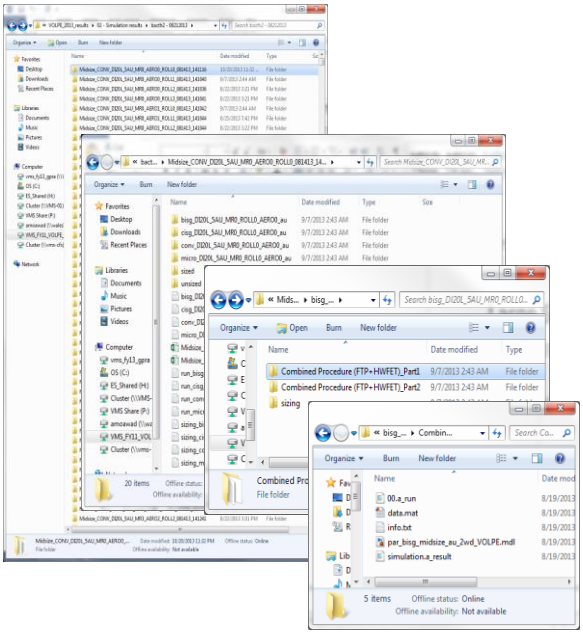


Figure 5: Results folder organization for individual simulations

7.1 Run File

As shown in Figure 6, "xx.a_run" includes all the information on the vehicle as well as a cycle/procedure. This file allows us to reproduce the simulation in the future if modifications or changes occur.

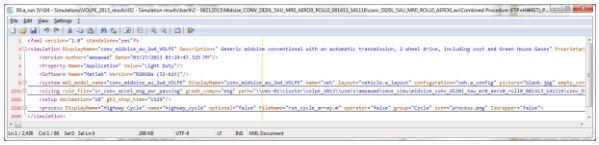


Figure 6: Autonomie run file

7.2 Data.mat File

"data.mat" is the results file containing all of the vehicle parameters and all of the time-based signals. A sample of signals and parameters included in data.mat is shown in Figure 7.

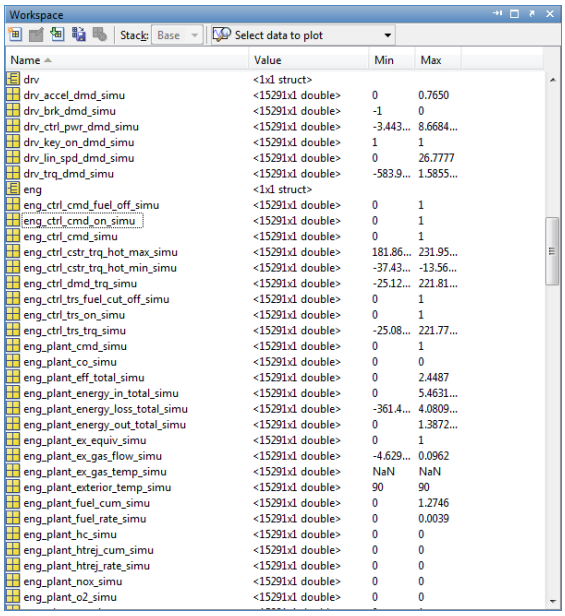


Figure 7: Autonomie data.mat file

7.3 Vehicle Model

As shown in Figure 8, "*.mdl" represents the complete vehicle model. Saving each vehicle model ensures that any simulation can be replicated at any time.

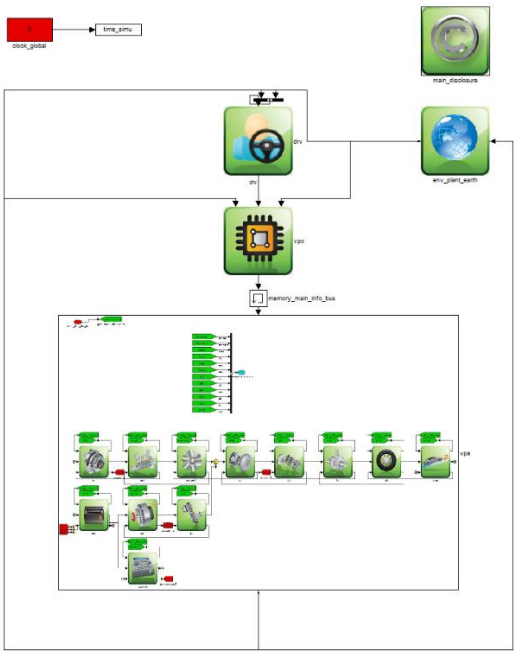


Figure 8: Autonomie representation of conventional vehicle

7.4 XML Results File

As shown in Figure 9, "simulation.a_result" is an XML version of the results file that includes the main simulation inputs and outputs. This file is later used to generate the complete MySQL database.

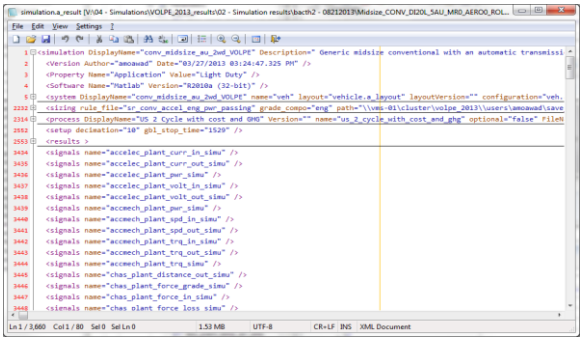


Figure 9: Autonomie XML results file

7.5 Folder Nomenclature

The MySQL database created and used by the Volpe model requires a searchable list of parameters from which to retrieve information about a particular vehicle. Because some of these parameters did not come from Autonomie, a folder nomenclature was adopted, as shown in Figure 10.

The naming conventions are similar to the acronyms currently used in the decision trees by the Volpe model.

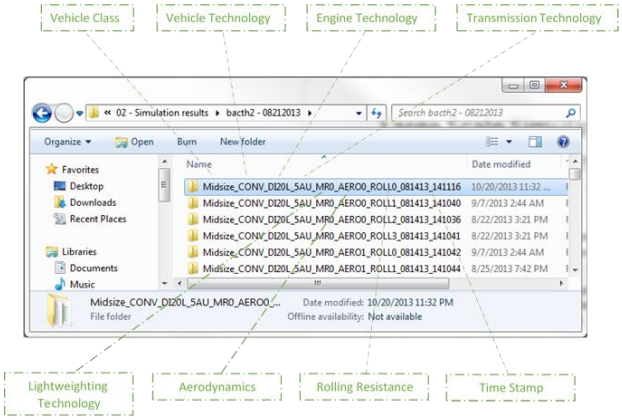
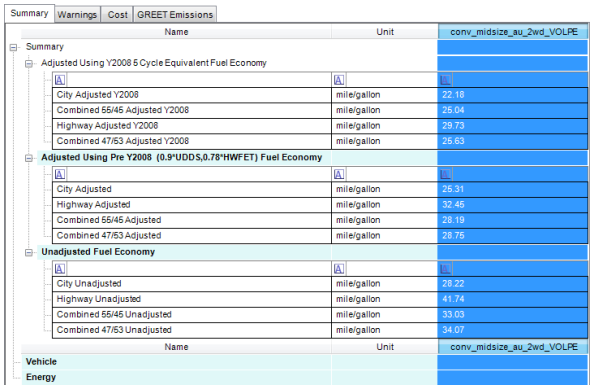


Figure 10: Folder nomenclature

7.6 Individual Vehicle Validation

Once the individual simulations are completed, Autonomie provides the ability to analyze them at both a high level (i.e., fuel economy) and a low level (i.e., time-based engine power) through its graphical user interface. An algorithm is also used to automatically flag any potential issues within a simulation (i.e., too many shifting events on a specific cycle). [5]

Figure 11 shows a sample of the parameter outputs from Autonomie provided for every vehicle among the 33,060 vehicles simulated.



Name	Unit	conv_midsize_au_2wd_VOLPE
Summary		
Adjusted Using Y2008 5 Cycle Equivalent Fuel Economy		
City Adjusted Y2008	mile/gallon	22.18
Combined 55/45 Adjusted Y2008	mile/gallon	25.04
Highway Adjusted Y2008	mile/gallon	29.73
Combined 47/53 Adjusted Y2008	mile/gallon	25.63
Adjusted Using Pre Y2008 (0.9*UDDS,0.78*HWFET) Fuel Economy		
City Adjusted	mile/gallon	25.31
Highway Adjusted	mile/gallon	32.45
Combined 55/45 Adjusted	mile/gallon	28.19
Combined 47/53 Adjusted	mile/gallon	28.75
Unadjusted Fuel Economy		
City Unadjusted	mile/gallon	28.22
Highway Unadjusted	mile/gallon	41.74
Combined 55/45 Unadjusted	mile/gallon	33.03
Combined 47/53 Unadjusted	mile/gallon	34.07
Name	Unit	conv_midsize_au_2wd_VOLPE
Vehicle		
Energy		

Figure 11: Baseline conventional vehicle outputs

Numerous predefined plots are also available to analyze any time-based parameter from the simulation. Figure 12 shows an example of engine speed, vehicle speed, and gear number for a conventional vehicle.

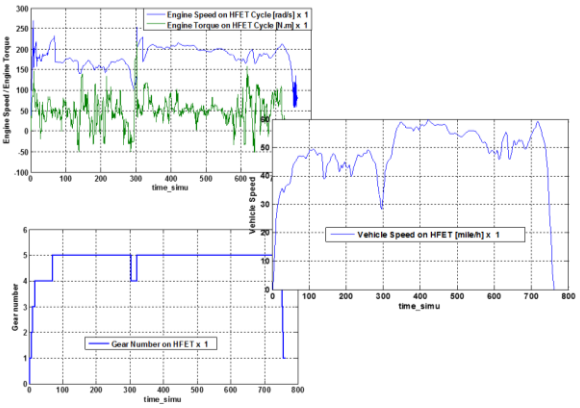


Figure 12: Vehicle detailed signals for individual check of anomalies

8 Vehicle Database

The Volpe model requires the user to tackle two complicated problems simultaneously:

1. A vehicle simulation tool must be used to quickly and properly estimate the energy consumption of extremely large numbers of specific vehicle, powertrain, and component technologies; and
2. The user must easily access and analyze information across large amounts of data.

A process for performing large-scale simulation with Autonomie is now in place. With it, a simulation can be quickly validated, or any discrepancies in the results can be examined in detail.

Autonomie was not originally designed to analyze such large sets of data. Such analyses impose data management concerns (numbers of files, disk sizes, access times); require the ability to run post-processing calculations without the time cost of rerunning all of the simulations; and involve plots, calculations, and other analytical tools for looking at high-level indicators and spotting overall trends. In response, Argonne's new process allows the detailed simulation results provided by Autonomie to either be distilled into a format that can be easily distributed and analyzed horizontally across many simulations, or examined via a deep, vertical dive into one simulation. Both aspects are critical for the full-scale vehicle analysis that the Volpe model requires.

The output of the simulations includes everything necessary for Autonomie to analyze or recreate an individual simulation, including the Simulink model representing the vehicle, a metadata file containing simulation results (*.a_result file), and a data.mat file containing all of the time-based signal data. These results can be archived for full traceability and reproducibility of all simulations. However, it is currently not feasible to share or analyze these data. For example, 36,000 simulation results resulted in 2 TB of disk space usage. It's simply not feasible to pass this much information around, much less the number of simulations required for the Volpe analysis (i.e., second-by-second fuel or electrical consumption values). Additionally, each simulation has individual files storing the results, so just managing or indexing the sheer number of files becomes an issue. Most of the information contained in those results files, however, is not necessary for the Volpe analysis. Therefore, a subset of the data is collected into a portable, user-friendly database.

8.1 Database Creation

Argonne's database creation process works from an input sheet that specifies which input and output parameters should be included in the database. The process scans all of the simulation results files, extracts the specified parameters, and stores them in a single, specialized database file. This allows us to exclude irrelevant information not needed for cross-cutting analyses, while leaving the full results archived, just in case. Figure 14 shows a list of the input and output parameters currently included in the database.

A single database file is easy to redistribute. The aforementioned 2 TB of data was compressed into 30.4 MB of data, and took only 27 minutes to generate from the original simulation results. Additionally, the database is developed using the MS SQL Express 2012 format, which is free and easily accessed by standard structured query language tools.

8.2 Database Structure

As shown in Figure 14, the database is structured to be generic, so that any simulation input parameter, result, or descriptive property can be stored. This approach allows maximum flexibility in the type of data that can be stored. The tables are structured to allow logical grouping of data, maximize retrieval speed, and minimize disk space.

Vehicles and the references to their parameters are stored separately from parameters specific to the type of simulation, because the same vehicle can be run on multiple procedures or cycles. For example, one vehicle may be subjected to an acceleration test and a fuel consumption test, such as a combined-cycle procedure. Each simulation may produce a fuel consumption result, which would then be linked to that simulation record. However, parameters common across both simulation runs, such as the coefficient of drag of the vehicle, would be linked to the vehicle record. Not all vehicles and simulations have the same

parameters; for example, motor parameters are only available for a vehicle with an electric power path (e.g., EVs, HEVs, PHEVs), and fuel consumption is only available for simulations with an engine or fuel cell, which exclude EVs.

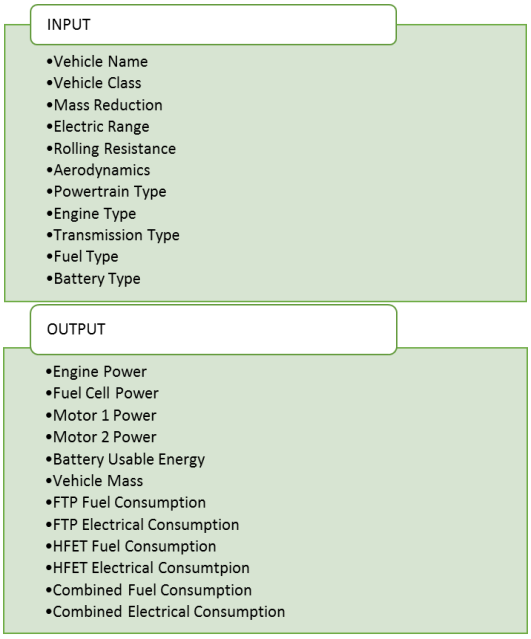


Figure 13: Inputs and outputs from simulation that can be saved to the database

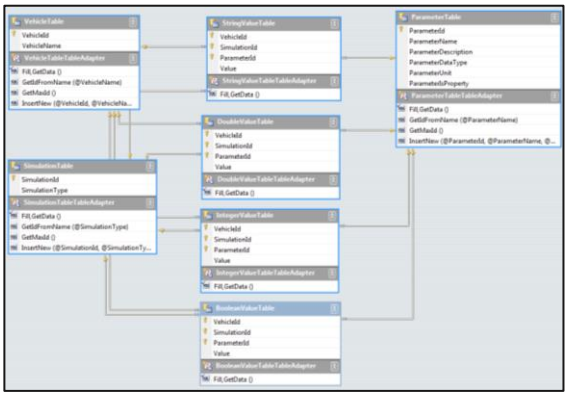


Figure 14: Database structure

Each parameter stores name, description, data type (i.e., string, double, integer, Boolean), and unit. The values themselves are organized into tables by data type for disk size optimization.

8.3 User Interface

Although the database is accessible by any tool or programming language that can interact with databases, Argonne has also developed a tool to easily visualize and analyze the data (**Error! Reference source not found.**). This tool provides a quick and intuitive way for users to quickly select subsets of simulation results of interest, select which parameters to view, modify assumptions, perform additional post-processing calculations on the data retrieved from the database, and view plots to better visualize the data.

Additionally, the user interface provides some advanced features that allow users to import their own plots and analysis functions; save "projects" of filters, parameters, and overridden assumptions; or export subsets of the data to Excel for additional data analysis or redistribution.

This tool allows users who are not familiar or comfortable with direct database access to perform the analysis necessary for Volpe modeling.

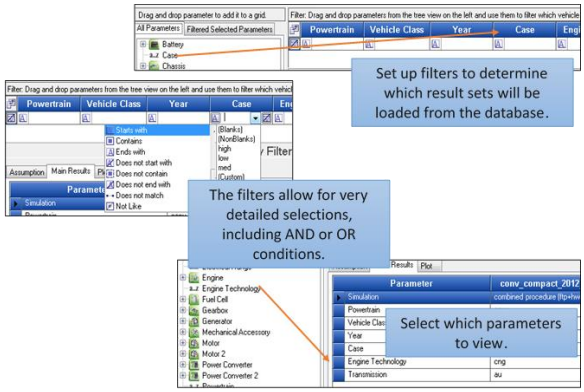


Figure 15: Database analysis tool

9 Reducing the Number of Simulations through Statistical Analysis

Distributed computing is one approach to accelerating simulation. Another approach is to use statistical analysis to downselect the number of simulations to be run and develop an algorithm to populate the complete database from a subset of simulations using statistical predictive modeling [6]. Motivated by the fact that several of the technological improvements were linear, and expecting to find apparent relationships and trends, especially linked to weight reduction, aerodynamics, and rolling resistance, the Mathematics and Computer Science Division (MCS) at Argonne collaborated with the System Modeling and Control Group to develop a method for minimizing the number of runs required to fulfill the Volpe model's demand. MCS has defined the relationships between component technologies to minimize the number of simulations.

9.1 Exploratory Data Analysis

This section presents an overview of the exploratory data analysis performed on the vehicle simulation results. The analysis comprises three

phases: correlation studies, sensitivity analysis, and predictive modeling.

In the first phase, the number of outputs that need to be modeled was reduced by eliminating outputs that are highly correlated with another output. In the second phase, for a given output that is not eliminated in the first phase, the inputs that do not have a significant impact on the output were removed by performing a nonlinear sensitivity analysis. In the third phase, a supervised machine-learning approach was used to learn the relationship between the input and the output of an unknown response function by fitting a model from relatively few representative simulation runs. When the model is accurate enough, it can predict the output from new (unseen) input combinations. This prediction provides numerous benefits and is especially valuable when the evaluation becomes expensive, such as with vehicle simulations. The preliminary analysis shows that

- For a given class of vehicles, several outputs from the vehicle simulation are highly correlated; and
- The machine-learning model only requires 50% of the simulation runs as training data to predict the outputs of the remaining simulation runs with reasonable accuracy.

Consequently, using the models, we can reduce the number of required simulations by 50%.

9.1.1 Correlation Studies

First, the dimensionality of the output space was reduced by eliminating outputs that are highly correlated with another output. Given two outputs, the linear correlation between them was computed using the Pearson product-moment correlation coefficient [6]. Using this correlation measure, when an output y_p is correlated to an output y_q, we remove y_p.

Figure 16 shows the pairwise correlations between outputs. Several outputs are highly correlated. This analysis showed that five outputs are not correlated with one another and, given these five outputs, the remaining eight outputs can be predicted (and are hence removed from further analysis).

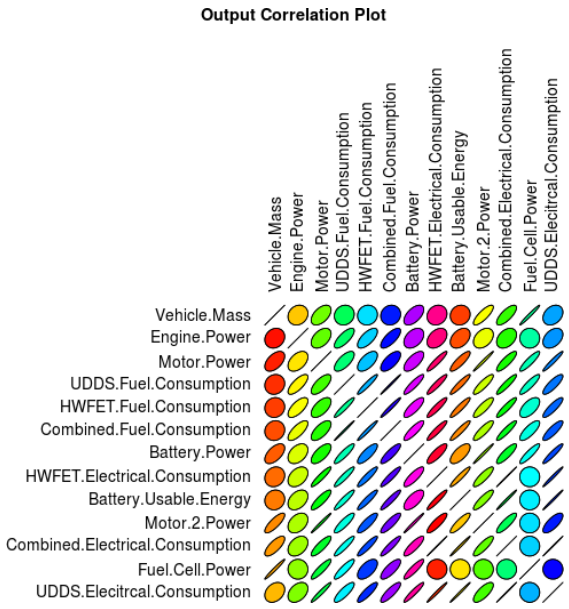


Figure 16: Pairwise correlations among outputs. Each entry in the matrix represents the correlation measure between the corresponding entries. Circles and slanting lines denote no correlations and high correlations, respectively.

9.1.2 Sensitivity Analysis

Next, we reduced the dimensionality of the input space by analyzing the impact of the input parameters on the uncorrelated outputs. For this purpose, the random forest (RF) method [7], a state-of-the-art machine learning approach for nonlinear regression, was adopted.

RF uses a decision tree-based approach that recursively partitions the multi-dimensional input space D into a number of hyper rectangles. The hyper rectangles are disjoint, so that each input configuration falls in exactly one hyper rectangle. The partitioning is done in such a way that input configurations with similar outputs fall within the same hyper rectangle. The partitioning gives rise to a set of if-else rules that can be represented as a decision tree, with each hyper rectangle corresponding to a leaf in this tree.

Over each hyper rectangle and for each output, a constant value is assigned; typically, this is an average of the values for that output for each seen configuration that falls within the hyper rectangle. Given an unseen input x^*, the algorithm uses the if-else rule to find the leaf (hyper rectangle) to which this input belongs and returns the corresponding constant value as the predicted value. RF uses a collection of decision trees, where each tree is obtained by such a partitioning approach using a different set of seen ("training") points. In particular, for each tree generation, the algorithm takes a different random subsample of points from the given master training set. Since each individual tree's prediction is thus based on a

different subsample of points, the prediction of the output at a given x^*can vary from tree to tree.
In addition to nonlinear regression (the next phase), RF can be used for analyzing the sensitivity of the inputs from the training points. This was implemented as follows: In the training phase of RF, we randomly sample 50% of the data. The random forest model is fit on this master training set. The mean squared error (MSE) on a training set is computed as follows:

$$MSE = \frac{1}{l}\sum_{i=1}^{l}\left(f(x_i) - \hat{f}(x_i)\right)^2 , \qquad (1)$$

where l is the number of training points and $f(x_i)$ and $\hat{f}(x_i)$ are the observed value (from the vehicle simulation) and predicted value (from the RF model) at the input parameter configuration x_i, respectively. To assess the impact of an input parameter m, the values of m in the training set are randomly permuted. Again, an RF model is fit on this imputed training set and the MSE is computed. This procedure is repeated several times. If a parameter m is important, permuting the values of m should affect the prediction accuracy significantly, resulting in a substantial increase in the MSE. For each parameter m, the percentage increase in MSE (%IncMSE) allows one to assess the importance of the parameter m. A permutation example is shown below. To assess the importance of the first parameter in the training set (containing 3 points and represented by the matrix T), the values of the first column are permuted (the first column corresponding to the values of the first parameter) in the matrix, which is shown in T_1':

$$\mathbf{T} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 & \cdots & \cdots \\ 3 & \ddots & \vdots \\ 4 & \cdots & \cdots \end{bmatrix} \qquad \mathbf{T'}_1 = \begin{bmatrix} 4 & \cdots & \cdots \\ 2 & \ddots & \vdots \\ 3 & \cdots & \cdots \end{bmatrix}$$

For the output *vehicle mass*, Figure 17 shows the %IncMSE for each input parameter. We observe that input parameters *glider mass*, *mass reduction*, and *powertrain* have a significant impact on *vehicle mass*, but that the *Aero* and *Rolling* input parameters are insignificant. Note that the negative %IncMSE is an artifact of over-fitting the RF model.

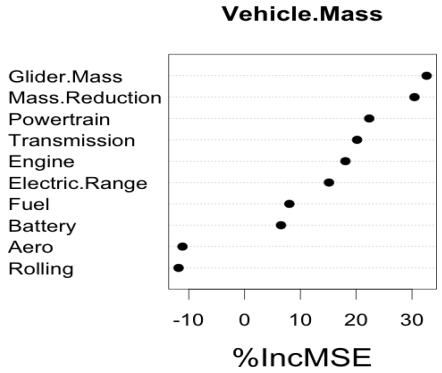


Figure 17: Impact of each input on vehicle mass measured across the testing set

Using this analysis, for each uncorrelated output we removed inputs that do not have a significant impact on the corresponding output.

9.1.3 Predictive Modeling

For each uncorrelated output, we built a predictive model using the RF method. In the experiments, we analyzed the number of training points (simulation runs) required to obtain high prediction accuracy. Given N_{y_p} observed data points for the output Y_p, we sample k% of the N_{y_p} points at random for training, and the remaining points are used for testing. As an accuracy measure, the root mean squared error (RMSE = √MSE) is computed on the test set. To reduce the effects of the randomness from the random sampling and the RF method, we repeat the experiments for a total of 10 repetitions.
For illustration, we first consider the output *vehicle mass* (where $N_{y_p} = 11{,}490$). Figure 18 (top) shows the box plots of the RMSE obtained over 10 repetitions for increasing sizes of training point sets. We observe that for a training set containing 30% of all possible points, the RF method can achieve high prediction accuracy. Further increases in the number of training points do not reduce the RMSE significantly. For a training point size of 30%, Figure 19 (bottom) shows the correlation between the observed and the predicted values, along with the error bound computed on the observed values. The results indicate that the errors in the predicted values are within ±3% of the observed values (blue lines).

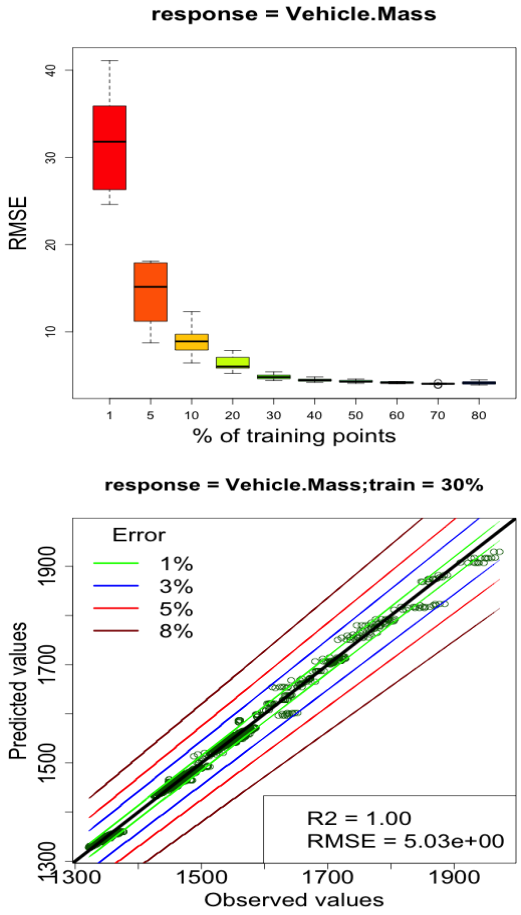
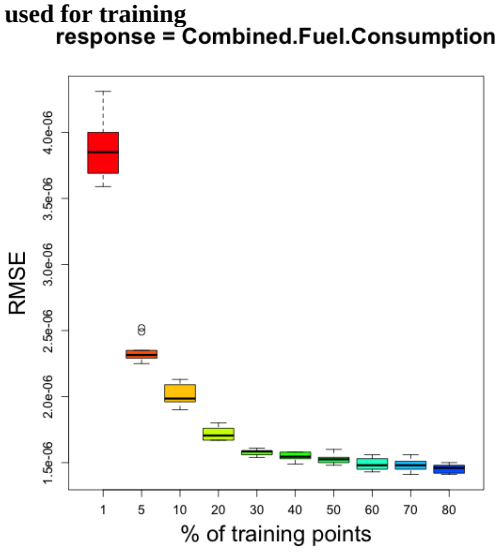


Figure 18: Predictive modeling for vehicle mass. (Top) Root mean squared error as a function of the % of training points obtained over 10 repetitions. (Bottom) Correlation between observed and predicted values when 30% of the simulation runs are used for training



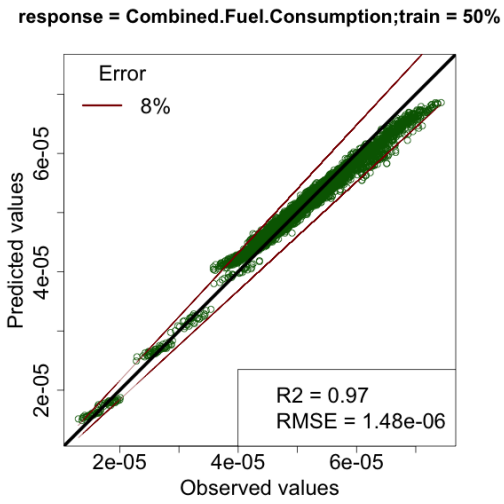


Figure 19: Predictive modeling for combined fuel consumption. (Top) Root mean squared error as a function of the % of training points obtained over 10 repetitions. (Bottom)

Acknowledgments

This work was supported by the U.S. Department of Transportation's Volpe Center under the direction of Ryan Harrington and Kevin Green. The authors would also like to thank John Whitefoot and Lixin Zhao from the U.S. Department of Transportation, NHTSA. The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE–AC02–06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

References

[1] http://www.autonomie.net

[2] A. Moawad and A. Rousseau, *Impact of transmission technology on fuel efficiency*, ANL/ESD/12-6, Argonne National Laboratory, June 2012.

[3] A. Moawad and A. Rousseau, *Impact of electric drive vehicle technologies on fuel efficiency*, ANL/ESD/12-7, Argonne National Laboratory, June 2012.

[4] http://www.nhtsa.gov/fuel-economy

[5] A. Moawad and A. Rousseau, *Light-duty vehicle fuel consumption displacement potential up to 2045*, ANL/ESD/11-4, Argonne National Laboratory, April 2014.

[6] C.M. Bishop, *Pattern recognition and machine learning, volume 1*, New York, Springer, 2006.

[7] L. Breiman, *Random forests*, Machine Learning, 45(2001), 5–32.

Authors

Ayman Moawad is a research engineer in the Vehicle Modeling and Simulation Group at Argonne National Laboratory. He graduated from the Ecole des Mines de Nantes, France, in 2009 with a Master's Degree in Science with a major in Automatics, Control Systems, and Industrial Computer Science. He focuses his research on light-duty-vehicle fuel consumption analysis as well as costs of powertrains to support the Government Performance and Results Act. He also has been a key participant in the support of DOT's CAFE standards by developing Large Scale Simulation Processes involving vehicle modeling and simulation as well as High Performance Computing.

Aymeric Rousseau is the Manager of the Vehicle Modeling and Simulation Group at Argonne National Laboratory. He received his engineering diploma at the Industrial System Engineering School in La Rochelle, France, in 1997. After working for PSA Peugeot Citroen in the Hybrid Electric Vehicle research department, he joined Argonne National Laboratory in 1999, where he is now responsible for the development of Autonomie. He received an R&D100 Award in 2004 and a 2010 Vehicle Technologies Program R&D Award in 2010. He has authored more than 40 technical papers in the area of advanced vehicle technologies.

Prasanna Balaprakash is an assistant computer scientist with a joint appointment in the Mathematics and Computer Science Division and the Leadership Computing Facility at Argonne National Laboratory. He received his Ph.D. from Universite Libre de Bruxelles, Belgium, where he was a Marie Curie and F.N.R.S. Aspirant Fellow. His research spans the areas of machine learning, numerical optimization, and performance engineering.

Stefan Wild is a Computational Mathematician in the Laboratory for Advanced Numerical Simulation at Argonne National Laboratory, and a Fellow in the Computation Institute at the University of Chicago. He received his Ph.D. in Operations Research from Cornell University in 2009, where he was a DOE Computational Science Graduate Fellow; he also holds BS and MS degrees in Applied Mathematics from the University of Colorado. Wild has been at Argonne since 2008, where his research focuses on numerical optimization and data analysis.