*EVS24*
*Stavanger, Norway, May 13-16, 2009*

# Evaluation of PHEVs Fuel Efficiency and Cost Using Monte Carlo Analysis

Gregoire Faron, Sylvain Pagerit, Aymeric Rousseau
*Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL, 6064, USA, arousseau@anl.gov*

## Abstract

Plug-in Hybrid Electric Vehicles (PHEVs) offer a great opportunity to significantly reduce petroleum consumption. The potential fuel displacement is influenced by several parameters, including powertrain configuration, component technology, drive cycle, distance… The objective of this paper is to evaluate the impact of component assumptions on fuel efficiency using Monte Carlo analysis. When providing simulation results, researchers agree that a single value cannot be used due to large amount of uncertainties. In previous papers, we have used triangular distribution, but assuming that all inputs were correlated lead to improper results. Monte Carlo allows users to properly evaluate uncertainties while taking dependencies into account. To do so, uncertainties are defined for several inputs, including efficiency, mass and cost. For each assumption, an uncertainty distribution will be defined to evaluate the fuel efficiency and cost of a particular vehicle with a determined probability.

*Keywords: simulation, modeling, PHEV, Monte Carlo, cost*

## 1 Introduction

Advanced powertains, including hybrid electric vehicles (HEVs) and plug-in HEVs (PHEVs), offer the potential to significantly reduce petroleum consumption. To evaluate the different options in a timely manner, the use of simulation tools has become a necessity. Argonne National Laboratory, working with automotive manufacturers, has developed the Powertrain System Analysis Toolkit (PSAT) to perform this task. Based primarily on Matlab, Simulink and StateFlow, the software allows a quick evaluation of different technologies. PSAT is the default vehicle simulation tool to support both the FreedomCAR and Fuels Parternship and 21 Century Truck Partnership (21 CTP).

PSAT current version behaves like a multi-input/multi-output deterministic non-linear algorithm; it generates a set of deterministic outputs from a set of deterministic inputs. The purpose of our study is to evaluate the benefits of PSAT handling stochastic inputs.

The initiative of the Risk Analysis Program started by the US Department of Energy motivated this study. When using PSAT as a design and decision tool, users legitimately expect the most accurate and complete results possible. When inputs have uncertainties, deterministic modeling cannot lead to correct simulation results. When dealing with uncertain inputs, the inputs must be stochastically modeled; as a result the generated outputs are also stochastic. Consequently, it becomes possible to compute quantities such as the most likely values to occur and some interval of confidence, which helps better describe and understand the simulation results.

Contrary to deterministic inputs, uncertain inputs are modeled by probability density function (PDF). Input PDF characteristics (shape, mean, variance, mode…) are established based on expert judgments and theoretical knowledge. The goal is to study how uncertainty propagates through PSAT algorithms, and to figure out how this uncertainty on the inputs finally impacts the algorithm outputs.

# 2 Monte Carlo Methodology Overview

Monte Carlo methods are families of computational algorithms that rely on repeated random sampling to compute their results. These algorithms are used when it is infeasible or impossible to compute an exact result with a deterministic algorithm. In our case, PSAT algorithms are too complex for us to compute the outputs generated by uncertain inputs. As a result, instead of simulating directly using uncertain inputs, we will randomly sample the uncertain inputs, generate sets of values from all the inputs samples, and then simulate each set separately. At the end, the simulation results obtained for each simulated set are aggregated, which gives the uncertain output values.

To perform a Monte Carlo simulation, we first need to choose and model the uncertain inputs. For this study, the uniform, Gaussian, and triangular PDF shapes have been implemented.

The second step consists in selecting the sampling method and the number of points to be used for the simulation. Each sampling method has its own convergence rate for a given problem and algorithm. As a result, we need to adapt sampling methods and number of points in order to reach the expected accuracy. After sampling the uncertain inputs, some correlations eventually can be added using either the Iman and Conover procedure or a Copula based method.

The completion of the previous steps leads to the definition of all the points to be simulated, from which we define the vehicles to be run in PSAT. The coordinates of a point in the hypercube define a vehicle’s uncertain input values.

Finally, these vehicles are simulated in PSAT, and their results are collected and plotted for analysis. The methodology is summarized in Figure 1. The main steps are described in greater details in further paragraphs.

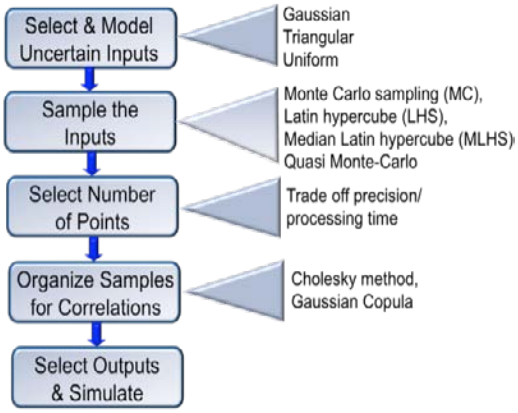

Select & Model Uncertain Inputs
Gaussian
Triangular
Uniform
Sample the Inputs
Monte Carlo sampling (MC),
Latin hypercube (LHS),
Median Latin hypercube (MLHS)
Quasi Monte-Carlo
Select Number of Points
Trade off precision/ processing time
Organize Samples for Correlations
Cholesky method,
Gaussian Copula
Select Outputs & Simulate

Figure 1: Summary of Monte Carlo simulation main steps

## 2.1 Input Sampling

A Monte Carlo simulation starts by sampling the input’s PDF. This first step has a major impact on the simulation result, because it determines the set of values representing the uncertain inputs. These samples need to represent the full range of the PDF, but also need to highlight high probability areas more than low ones. Cumulative Distribution Function (CDF) inversion is the most common method used to sample PDF.

The general idea of CDF inversion is that by inverting the uncertain input CDF over uniformly distributed points, we obtain a good sample of the PDF.

The precision of this method relies mainly on the uniform sequence inverted over the CDF. Consequently, our main concern will be to define the best sequence of point to be inversed over the joint CDF. In the next section (sampling methods), we will discuss different ways of generating these uniformly distributed points.

The main idea behind Monte Carlo simulation is to sample a K-dimensional hypercube with N points, i.e.: to generate an N-point uniform sequence into a K dimensional unit hypercube. Then, inverse the join CDF over the sequence of points from the hypercube, and as a result obtain the samples for each uncertain inputs PDF.

Figure 2 shows a sample for triangular distribution.

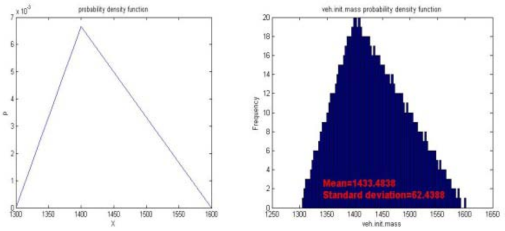


Figure 2: Triangular PDF sample values histogram

## 2.2 Sampling Methods

### 2.2.1 Monte Carlo Sampling

The most popular sampling method is called Monte Carlo sampling (MC). This method uses pseudo random numbers (between 0 and 1) to approximate a uniform distribution. Monte Carlo sampling convergence rate is on average $O(N^{-1/2})$ (central theorem consequence). This bound does not depend on the number of inputs K, unlike for the other methods. The independence in between a number of uncertain inputs and a convergence rate makes Monte Carlo sampling a very useful and efficient sampling method when dealing with large number of uncertain inputs. However, as the bound is probabilistic, there is no way to build the sequence reaching the optimal bound (Papageourgiou and Wasilkowski in [1]).

According to the literature, the convergence rate depends more on the equidistribution of the sample over [0 1], than on the randomness (Morgan and Henrion in [2]). Because uniformity is the main aspect, we will introduce other sampling methods that focus on the sample's uniformity.

### 2.2.2 Hypercube sampling (Iman and Shortencartier in [3])

One method of creating more uniform samples (i.e.: to get a faster convergence rate) is to use stratified sample methods, such as Latin Hypercube Sampling (LHS) or Median Latin hypercube sampling (MLHS). In LHS, the range of each uncertain input $X_i$ is sub-divided into non-overlapping intervals of equal probability. Then, one value from each interval is selected at random with respect to the probability distribution in the interval. In MLHS, this value is the mid-point of the interval. The N values thus obtained for $X_1$ are paired in a random manner (i.e., equally likely combinations) with N values of $X_2$. These N values are then combined with N values of $X_3$ to form N-triplets, and so on, until N k-tuplets are formed. MLHS usually gives better results than LHS. However, it fails sometimes with periodic functions with a period similar to the size of the equiprobable intervals. There are no periodical functions in PSAT, so we will generally use MLHS more than LHS.

Hypercube sampling methods only provide probabilistic bounds. Moreover, hypercube methods were designed to provide good uniformity in one dimension. Thus, it does not produce perfect random uniformity in multidimensional configurations.

As shown in [4] and [5], and assuming PSAT simulation algorithm is monotonic in each of the inputs, we can easily compare MC sampling to LHS. Considering forecast sample means, variances and percentiles as estimators, we can show that these estimator variances are lower for LHS than for MC sampling. In [5], [6] and [7] we see that this result is confirmed by the experiments. LHS converges faster than MC for a low number of inputs (up to 15), and in the worst case LHS is not worse than MC.

### 2.2.3 Quasi-Monte Carlo methods

Quasi-Monte Carlo methods are based on low-discrepancy sequences, which use optimal design schemes for placing N points on a k-dimensional hypercube. Unlike Monte Carlo Sampling and Latin Hypercube, the quasi-random sampling technique ensures that the sample sets show more uniformity of properties in multi-dimensions. There are several different low-discrepancy sequences (Hammersley, Sobol, Halton, Faure,…), well-described in the literature, that can be used for quasi-Monte Carlo simulation. Here we choose to use Hammersley and Halton sequences.

Using a quasi-Monte Carlo simulation is much more complicated than using Monte Carlo methods. These difficulties come from the lack of theoretical results that allow us to evaluate the quasi-Monte Carlo method's accuracy.

This accuracy depends mainly on the simulation algorithm's characteristics (such as it variations for example) and the number of dimensions of the problem. As a result, there are no general results concerning the convergence rate estimation. However, upper and lower bound rates of convergence can be expressed; we will try to take advantage of those bounds to get theoretical information on the precision.

As shown in [8], a lower bound rate of convergence can be derived using the Koksma-Hlwaka inequality. This inequality expresses an absolute bound on the accuracy of quasi-random integration (in our case simulation is equivalent to integration). This bound is proportional to the discrepancy of the sequence used (in our case the Hammersley sequence). Knowing the Hammersley sequence discrepancy, we can derive the lower bound of convergence: $O(\frac{\log(N)^K}{N})$.

According to Morokoff in [9] the optimal rate of convergence is faster than Monte Carlo sampling. In this case the upper bound rate of convergence is: $O(N^{-1})$.

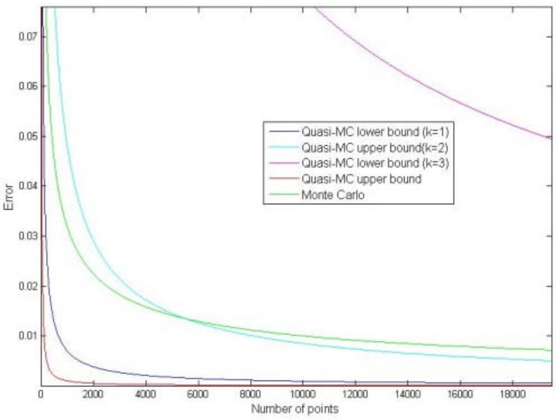


Figure 3: Error bounds comparison

Figure 3 illustrates the theoretical convergence rate differences between the Monte Carlo and quasi-Monte Carlo methods (based on our first experimental results). As one notices, the upper bound for quasi-Monte Carlo performs always better than Monte Carlo. On the other hand, the lower bound seems wide and only performs better for a few uncertain inputs and a large number of points.

As illustrated by the diagram above, using quasi-Monte Carlo method is tricky – it is a priori difficult to know whether or not it is worth using it compared to LHS. However, as expressed in [7], [8], [10] and [11], it is often worth using the quasi-Monte Carlo for low numbers of uncertain inputs.

## 2.3 Sampling Method Comparison

### 2.3.1 Convergence comparison

Our first purpose is to make sure all different sampling methods implemented in PSAT lead to the same results. We first simulate different vehicles, with different number of uncertain inputs and sampling methods. Our purpose is to verify that in some representative cases, our algorithm converges to the right output PDF. All the sampling methods were simulated for up to 1000 points, which is sufficient to get an output PDF with constant means and variances.

We then compared the results obtained with the four different sampling methods (all methods are independent from one to the other). If the four methods converge around the same value, there are great chances that this value is the right one.

Figure 4 shows the result from an uncertain input sampling (i.e.: equivalent to the Monte Carlo simulation of one input through the identity function). It is a good indicator of the sampling method efficiencies since the quality of the input sampling makes most of the Monte Carlo simulation efficiency. Moreover we can derive the input theoretical mean and variance, and then derive the exact convergence rate of each sampling method.

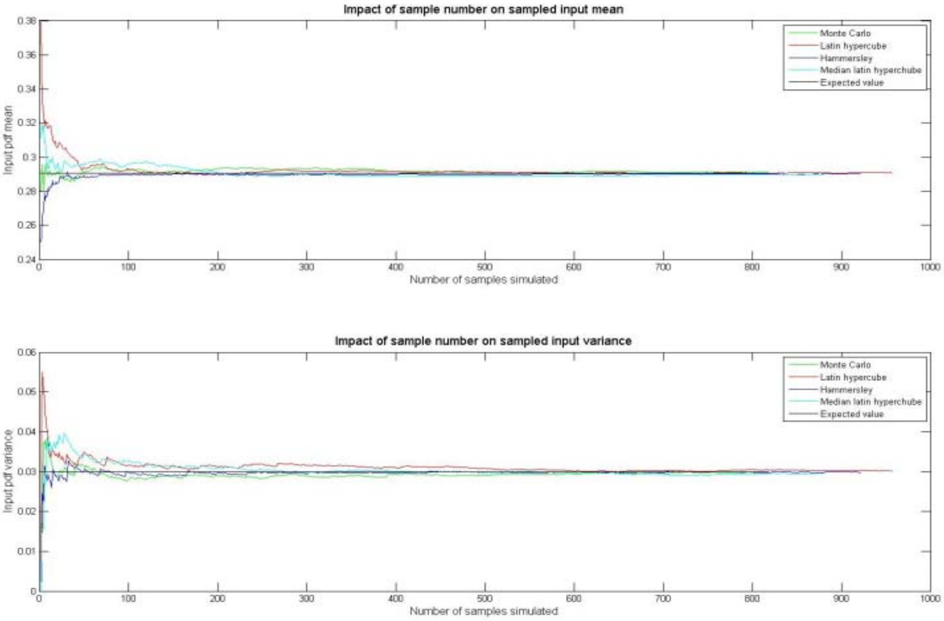


Figure 4: Mean and variance of a sampled Gaussian input PDF, for different sampling method and number of points

All the methods converge to the theoretical value (straight black line). On the second diagram, we can see that for a large number of points, approximately over 600, results from all the methods are bounded in the interval: theoretical value +/- 0.5 %. A 100-point sample gives at least a five-percent accuracy on mean and variance for each method.

It is clear that the Hammersley sequence gives the best results, a five-percent error with 30 points, less than 0.1% error for more than 600 points. The second best method is MLHS.

Figure 5 illustrates the results of the PSAT fuel consumption simulation, using Monte Carlo with 5 uncertain inputs. In this case, we derived theoretically the 95% confidence interval on the forecast mean (black dashed line).

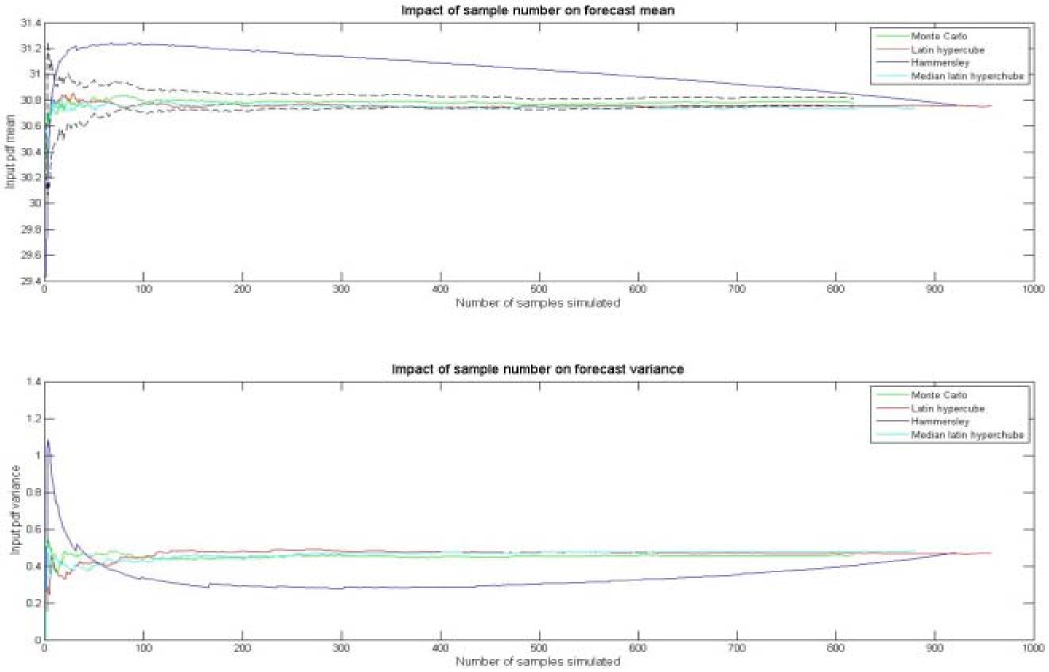


Figure 5: Mean and variance of a forecast PDF, for different sampling method and number of points

In this case, it is not possible to figure out the PDF mean and variance theoretical values. However, we can notice that all methods converge to about the same value.

Hammersley sequence results are not relevant. This is due to the fact that when generating a Hammersley sequence for N points, the N points have a uniform repartition. However, any subset of points does not. This aspect and the fact that quasi-random sequences are not stochastic make quasi-Monte Carlo hard to validate. However, we can notice that it converges to the same value as the others do.

Similar simulations and observations were carried out for different vehicles types and different numbers of uncertain inputs. In every case, the Monte Carlo simulation led to similar observations as above. Based on these results, we will assume that each sampling method converges to the right output PDF, under the following assumptions.

- Less than 35 uncertain inputs
- At least 5% accuracy with 1000 points

### 2.3.2 Sampling method convergence rate

Being sure that all the sampling methods converge to the right output PDF, we now need to study the convergence rates particular to each sampling method.

Monte Carlo simulation is a stochastic process, which means that running the algorithm multiple times with the exact same assumptions leads to different results (because the point sequences generated are different from one run to another).

As a result, we need to find the number of points leading to an acceptable error interval around the simulation results. To do so, we choose a vehicle and define a set of uncertain inputs. We then run multiple Monte Carlo simulations without modifying the predefined assumptions.

Figure 6 illustrates this method. In this example, we simulated five times a hybrid vehicle with 6 uncertain inputs, using MLHS. The tags indicate estimator variances for 50, 100, 200 and 1000 points.

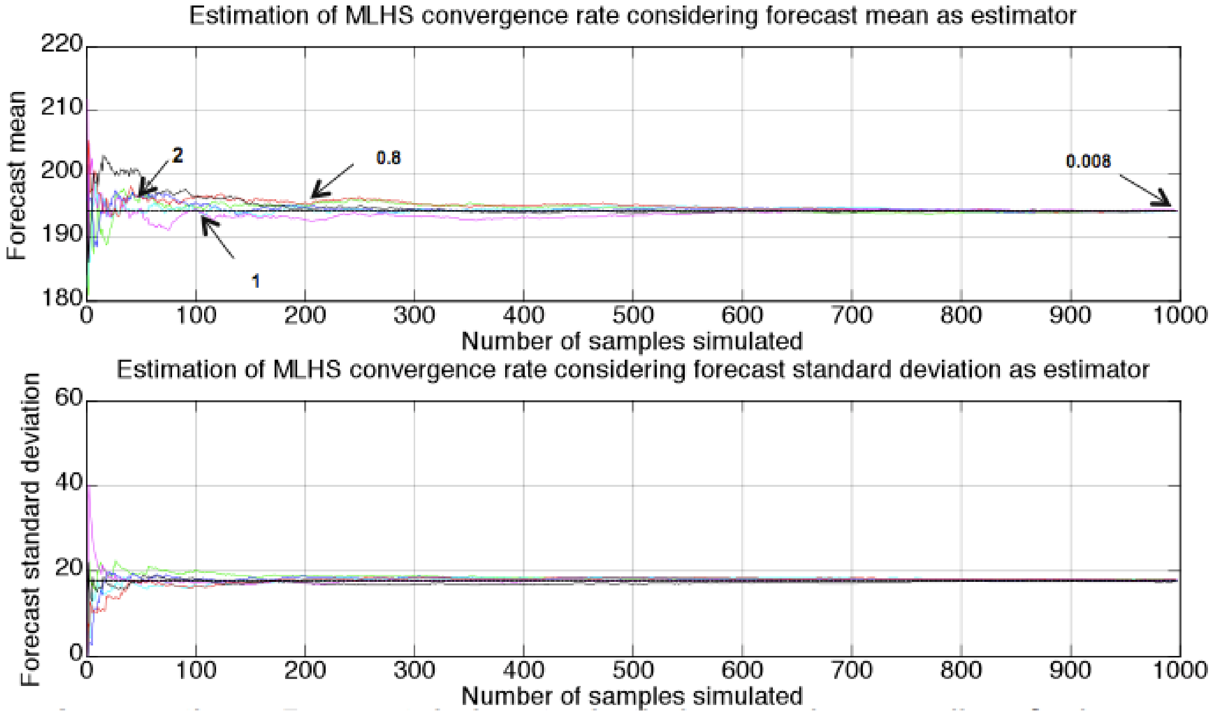


Figure 6: MLHS convergence rate study

The process described above was carried out for all the stochastic sampling methods (all the methods except the Hammersley sampling).

We then derived the estimator's variances in function of the number of points and compared the results obtained with the different sampling methods. The sampling methods in which the estimator's variances go to zero the fastest are the best. Figure 7 illustrates this comparison between the different sampling methods. This result was obtained simulating six uncertain inputs with a hybrid vehicle.

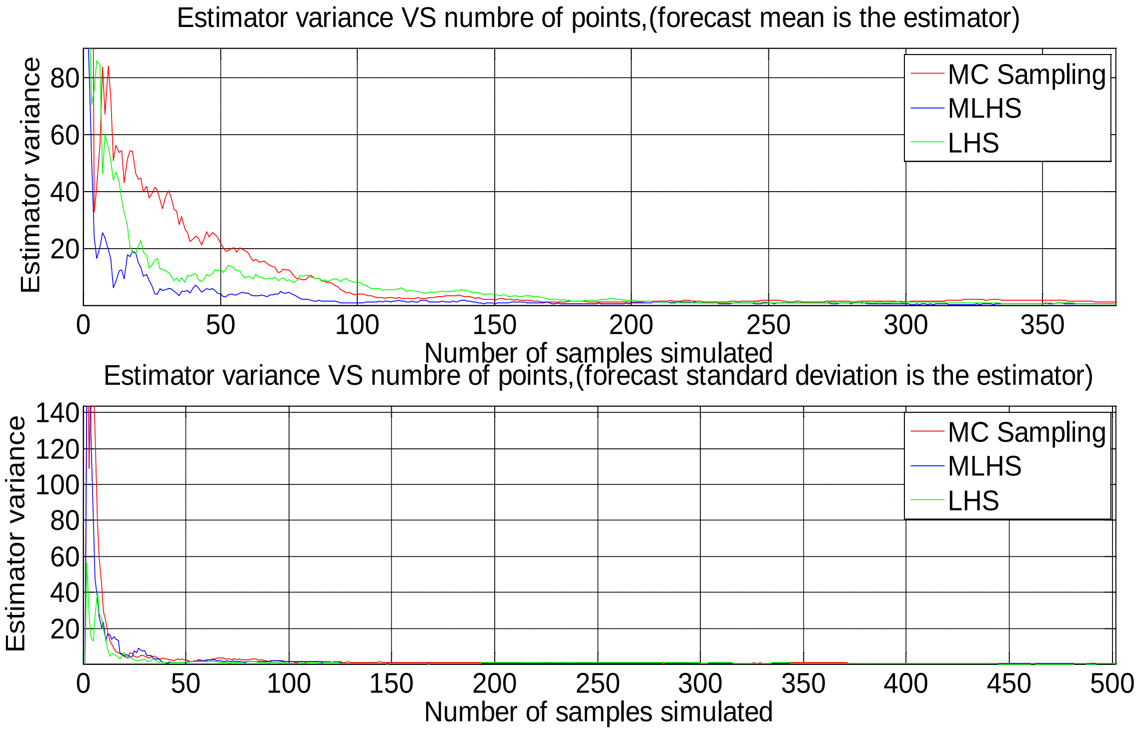


Figure 7: Sampling method convergence rate comparison

In the above example, MLHS converges the fastest for the output mean. The sampling method does not have a significant impact on the output variance.

Based on these diagrams, we can derive the number of points required for each sampling method to reach a particular accuracy. To be more specific, we define the expected precision as an estimator variance, and determine the number of points providing this variance for each sampling method.

This method cannot be applied to quasi-Monte Carlo simulation, because the point sequences used are not stochastic. However, individual simulations allow comparisons of non-stochastic methods with conventional Monte Carlo methods. According to the literature and our experimental results, we can state that quasiMonte Carlo simulations give the best results for a low number of uncertain inputs [12].

## 2.4 Determining the sampling method

This validation study gave us a better understanding of our algorithm behaviour. Table 1 summarizes how to select a pair of uncertain inputs/number of samples, depending on the number of uncertain inputs. The values given below are based on the experimentations run during the validation process. Some additional simulations need to be run in order to get a more accurate understanding of the algorithm's behaviour. The impact of aspects such as the type of vehicle simulated, the cycle considered, or the correlations structure should also be considered.

Table 1: Monte Carlo simulation assumptions to be used for different number of inputs

| Number of uncertain inputs n | Sampling method | Number of points | Correlation method |
|---|---|---|---|
| n<12 | Hammersley | 100 | Cholesky |
| 12<=n<=25 | MLHS (or LHS) | 150 | Cholesky |
| n>25 | Monte Carlo Sampling | 200 | Copula |

# 3 Simulation Assumptions

The vehicle simulated is a midsize car Plug-in Hybrid Electric Vehicle with a battery sized to follow the Urban Dynamometer Driving Schedule (UDDS) drive cycle for 10 miles in electric mode. The configuration considered is an input split, similar to the one used by Ford and Toyota. The engine power is sized to sustain a 6% grade at 65 mph (~100 km/h) without any support from the battery. The main characteristics are defined in Table 2.

Table 2: Main Vehicle Characteristics

| | |
|---|---|
| Glider mass (kg) | 990 |
| Frontal area (m2) | 2.2 |
| Coefficient of drag | 0.29 |
| Wheel radius (m) | 0.317 |
| Tire rolling resistance | 0.008 |

For each set of assumptions considered, a specific vehicle is defined to meet the vehicle technical specifications (performance, gradeability…).

The control strategy used in the simulation is based on a blended approach, where the engine is started based on a power threshold dependent upon the battery state of charge (SOC). The engine is then used close to its best efficiency curve. As a consequence, the battery is recharged and the charge depleting range increased.
The objective of the following chapters is to evaluate the benefits of using Monte Carlo approach compared to the initial method based on using only three points. In the case of the triangular distribution based on three points, a vehicle was defined for each case, with the low case being composed of all the low case assumptions, the middle case of all the middle assumptions and the high case of all the high assumptions. Using that approach, one expects to have a larger uncertainty range.

The output from Monte Carlo will be first discussed and then compared with the three points approach.

Twelve inputs were considered in the Monte Carlo simulation to assess the uncertainty of the vehicle fuel efficiency:

- Glider (body & chassis) mass
- Frontal area
- Coefficient drag
- Rolling resistance
- Electric motor controller #1 power density
- Electric motor #1 power density
- Electric motor #1 efficiency
- Electric motor controller #2 power density
- Electric motor #2 power density
- Electric motor #2 efficiency
- Engine efficiency
- Electrical power

The cost estimation of the vehicle powertain was also based on several inputs, including component powers (engine, electric machine, battery…), energy, glider mass…

Due to the large number of uncertain inputs, the MHLS was used. One thousand points were simulated to ensure proper convergence. Figure 8 shows the overall process.

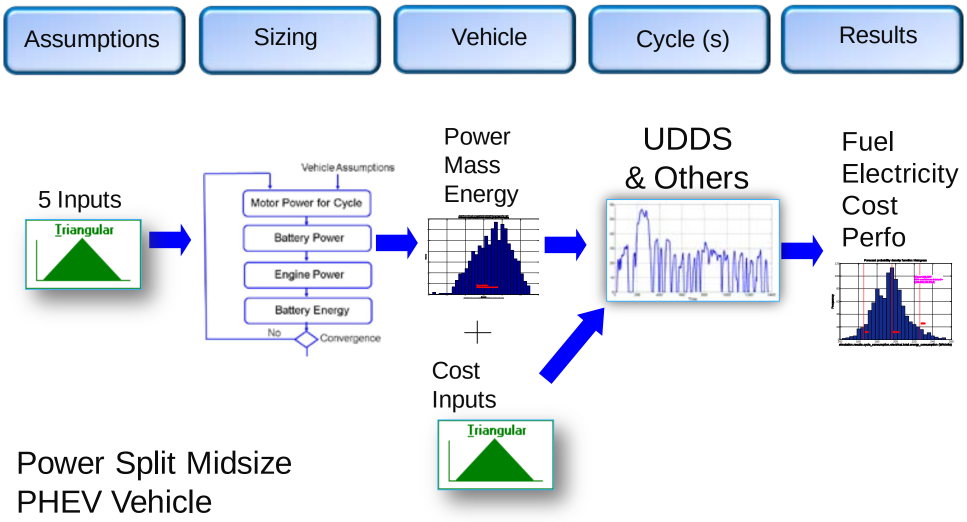


*Figure 8: Vehicle Simulation Process*

## 4 Simulation Results

After the simulation, we collected the considered outputs samples (fuel consumption and cost), and displayed their distribution. These histograms provide graphical representations of the fuel consumption and cost of PDF as well as their mode and their 80% confidence interval (10%-90%).

Based on the output samples we also computed a kernel estimation of the PDF.

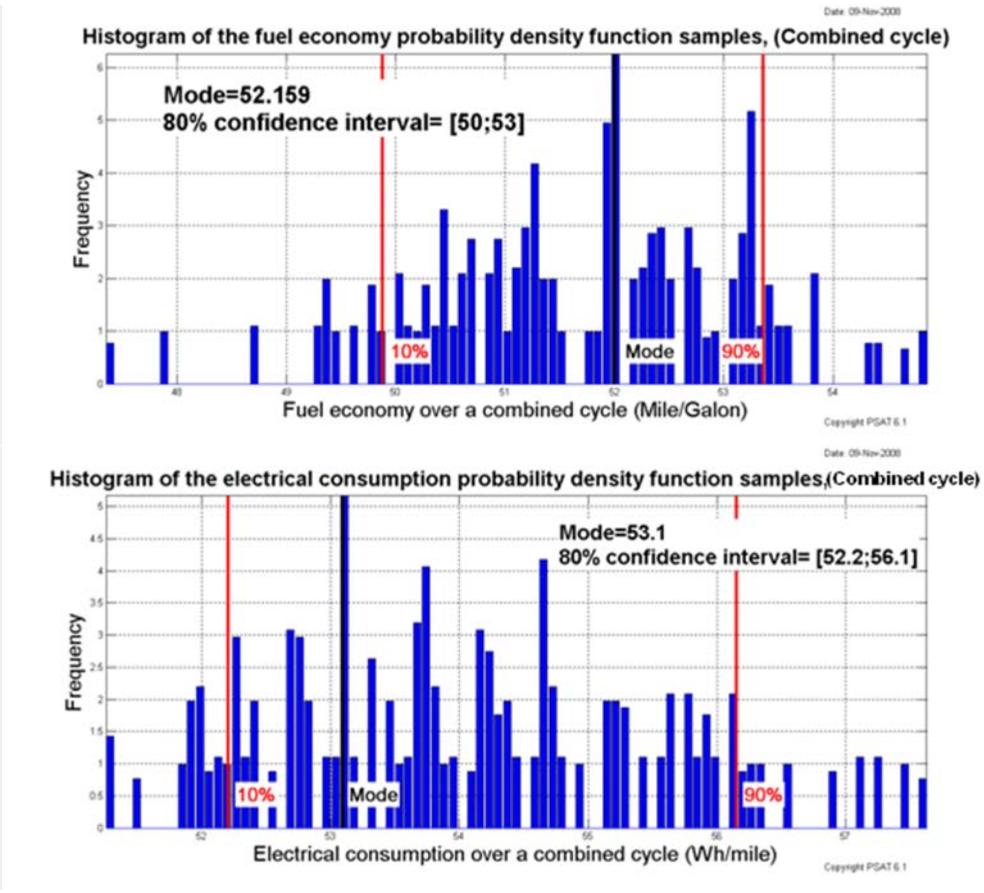


Figure 9: Histogram of the fuel economy and electrical consumption forecasts

Figure 9 illustrates the fuel economy and electrical consumption forecasts. The center line represents the mode of the output PDF, which represents the most likely value to occur considering the uncertainty of the inputs. The lines at both ends represent the 10% and 90% distribution percentiles. These 2 lines define an 80% confidence interval on the result (i.e. there is 80 % probability for the output result to occur in this interval).

The output PDF obtained here are multi-mode. This is due to the fact that for a PHEV the fuel economy and electrical consumption are non-monotonic over the uncertain inputs. Each local mode represents the most likely value to occur, for a given number of internal combustion engine (ICE) starts. This PDF provides additional information on the vehicle's general behavior. However, it requires more points to be simulated to get the expected accuracy for a given number of ICE starts. Moreover, given this multi mode shape it is very hard to get a PDF estimation of the result.

Figure 10 illustrates the cost forecast. Contrary to the forecast considered above, the cost forecast is single mode, which makes its interpretation easier.

Based on this single mode forecast we can derive a Kernel estimation of the cost PDF. Figure 11 illustrates the cost PDF estimation using Kernel estimation. This estimation provides a full and accurate description of the output PDF.

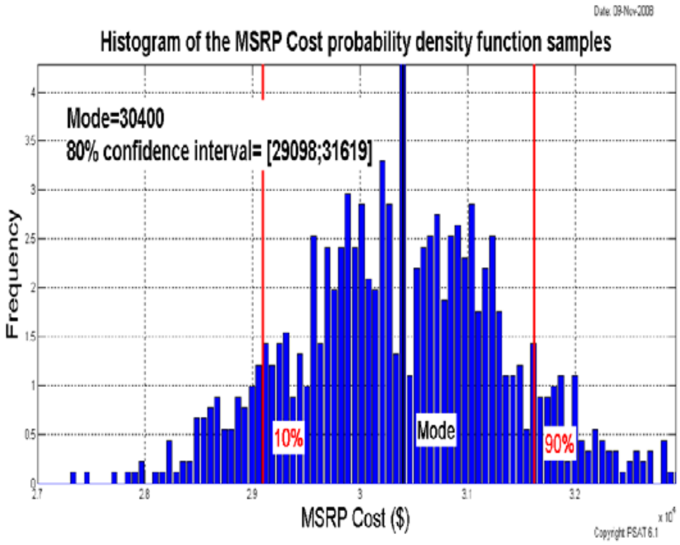


Figure 10: Cost Forecasts Histogram

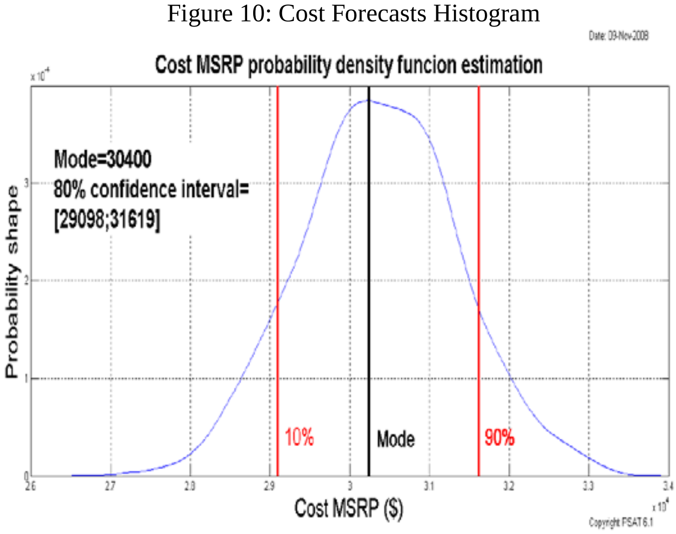


Figure 11: Cost PDF Estimation

## 5 Comparison between Monte Carlo and Three Points Approach

Figure 12 compares the fuel economy and electrical consumption obtained from the Monte Carlo analysis with the three values (all low, all medium, all high together).

As expected the range between the min and the max values is smaller using Monte Carlo. While the 3 points study fuel economy ranges from 53 to 61 mpg with a mode of 57 mpg, Monte Carlo provides a mode of 55 mpg within 54 and 57.5 mpg with an 80% confidence interval.

Similar conclusions can be drawn from the electrical consumption where a smaller range is achieved. The 3 points approach provides extreme cases

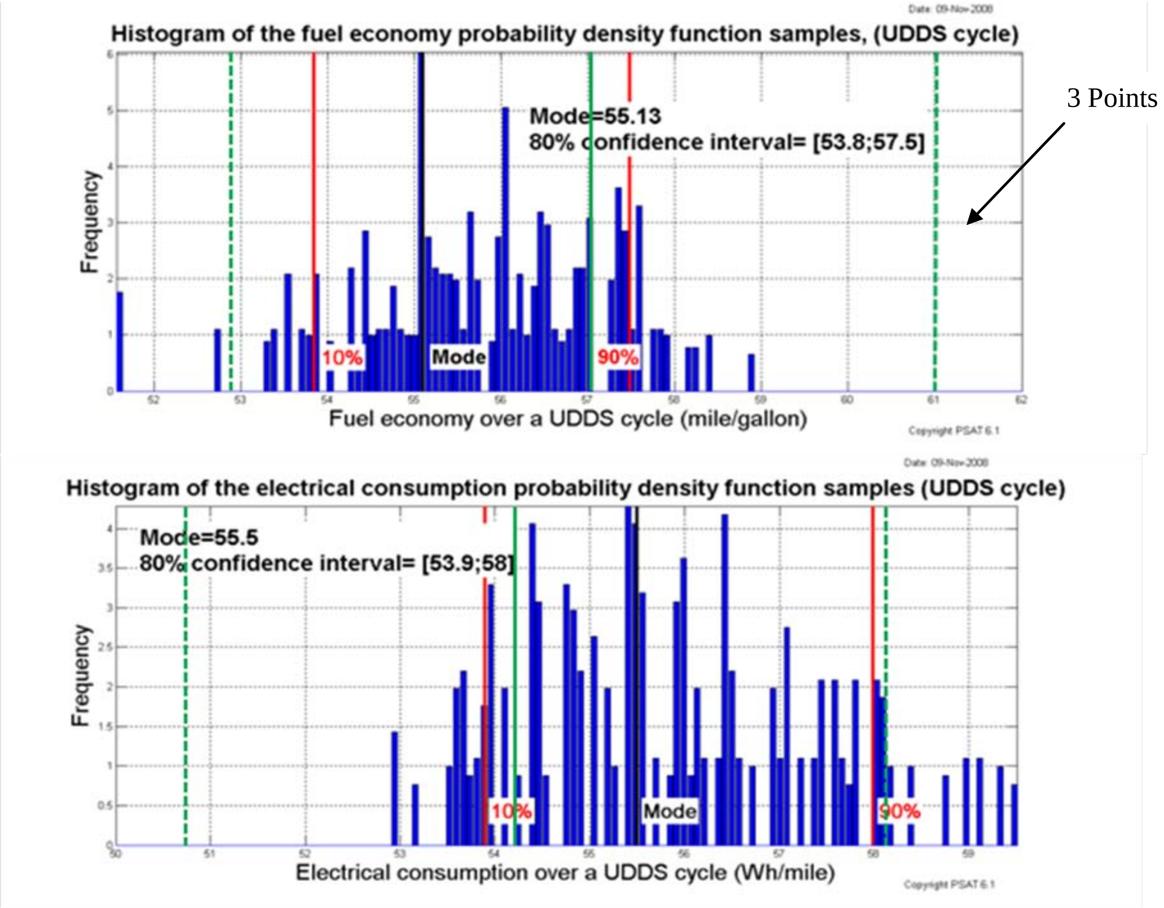


Figure 12: Comparison Between Three Points and Monte Carlo for Fuel and Electrical Consumptions

## 6 Conclusion

By allowing the introduction of uncertainty in our algorithm inputs, the Monte Carlo method increases the amount of information useful to describe a vehicle’s possible behaviors. The major improvement concerns the introduction of the risk notion associated with each result. Instead of providing a single forecast value, Monte Carlo simulation provides the probability of occurrences associated with every possible output value. As a result, forecasts are more fully and accurately described and confidence intervals can be derived for each output.

A Monte Carlo library containing all the essential features required to carry out accurate studies of the uncertainty propagation has been implemented into PSAT and validated.

Different sampling methods were compared for several powertrain configurations. For each option, the most appropriate number of samples was defined.

The results from Monte Carlo based on a midsize PHEV were defined, providing a mode for both fuel economy and cost within a certain confidence interval. The approach was then compared with the existing 3 points option. Results demonstrated that Monte Carlo provided a narrower range.

However, increasing the amount of information available on the results has a computational cost. The experiments carried out so far led us to a first evaluation of the number of points required to simulate. This number of points varies from 100 to 200 points, depending on the number of uncertain inputs considered. While computational time varies from each configuration, the average time required to simulate a PHEV on all these points is 150 minutes.

To conclude, Monte Carlo analysis provides useful insight for the uncertainty of specific technologies. Due to the requirements for the computations, this method is currently only applicable to studies with limited number of vehicles or powertrain configurations.

## Acknowledgments

This work was supported by DOE's FreedomCAR and Vehicle Technology Office under the direction of Lee Slezak. 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- A. Papageorgiou and G.W. Wasilkowski, *On average case complexity of multivariate problems,"* Journal of Complexity, vol. 6, pp. 1–6, 1990

2- Morgan, M. G., and M. Henrion. *Uncertainty -- a guide to dealing with uncertainty in quantitative risk and policy analysis*. Cambridge: Cambridge University Press, 1990.

3- Iman, R.L. and M.J. Shortencarier.. *User's Guide for the Generation of Latin Hypercube and Random Samples for Use with Computer Models*, querque, NM: Sandia National laboratories. NUREG/CR-3624, SAND83-2365. 1984

4- McKay, M.D., R.J. Beckman, and W.J. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, Technometrics. Vol. 21, no. 2, pp. 239-245. 1979.

5- J. C. Helton and F. J. Davis, *Latin Hypercube Sampling and the Propagation of uncertainty in Analyses of Complex Systems*, Sandia National laboratories, 2002.

6- L. Chrisman, M. Henrion, and R. Morgan, *Analytica user guide, release 4.0,* 2007

7- Urmila M. Diwekar, *A Novel Sampling Approach to Combinatorial Optimization Under Uncertainty,* Computational Optimization and Applications, 24, 335–371, 2003.

8- H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods,* SIAM, CBMS 63, 1992.

9- Morokoff, W.J. (1998): *Generating Quasi-Random Paths for Stochastic Processes*, SIAM Review, vol.40, n$^{o}$ 4, , pp. 765-788, 1998

10- J.R. Kalgnanam and U.M. Diwekar, *An efficient sampling technique for off-line quality control.* Technometrics, vol. 39, no. 3, pp. 308–319, 1997.

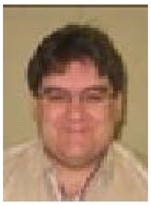
11- William C. Snyder, *Accuracy estimation for quasi-Monte Carlo simulations,* Mathematics and Computers in Simulation 54, 131–143, 2000.

12- Helton, J.C., *Uncertainty and Sensitivity Analysis Techniques for Use in performance Assessment for Radioactive Waste Disposal*, Reliability Engineering and System Safety. Vol. 42, no. 2-3, p. 327-367, 1993.

## Authors

Gregoire Faron, Research Aid, Ecole de Mines de Nantes

Gregoire is is currently completing a Master of Science in Electrical and Computer Engineering at the Georgia Institute of Technology. He lead the development and the implementation of MonteCarlo in PSAT as part of his final training period to obtain his Master of Science



Sylvain Pagerit, Research Engineer, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439-4815, USA, spagerit@anl.gov

Sylvain received a Master of Science in Industrial Engineering from the Ecole des Mines de Nantes, France, in 2000, as well as a Master of Science in

Electrical Engineering from the Georgia Institute of Technology, Atlanta, in 2001. He is currently working on ANL's Vehicle modelling and simulation group.

Aymeric Rousseau, Program Manager, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439-4815, USA, arousseau@anl.gov@anl.gov

Aymeric received his Master of Science in Industrial System from EIGSI in La Rochelle, France in 1997. He is currently leading ANL's Vehicle modelling and simulation group.

APPENDIX – Monte Carlo Main Assumptions

| | | ***DISTRIBUTION*** | | | | |
|---|---|---|---|---|---|---|
| | | ***Triangular*** | | | ***Gaussian*** | |
| ***Parameter*** | ***Unit*** | ***Min*** | ***Max*** | ***Mode*** | ***Mean*** | ***Variance*** |
| Glider Mass | Kg | 886 | 995 | 940.5 | | |
| Motor Controller #1 Power Density | W/kg | 4900 | 12600 | 6000 | | |
| Electric Motor #1 Power Density | W/kg | 1085 | 1300 | 1255 | | |
| Motor Controller #2 Power Density | W/kg | 4900 | 12600 | 6000 | | |
| Electric Motor #2 Power Density | W/kg | 1085 | 1300 | 1255 | | |
| Frontal Area | $m^2$ | | | | 2.222 | 0.0172 |
| Coefficient Drag | | 0.24 | 0.31 | 0.27 | | |
| Rolling Resistance | | 0.0072 | 0.01 | 0.0078 | | |
| IC Engine Efficiency | % | | | | 38 | 0.39 |
| Electric Drive #1 Efficiency | % | | | | 95.5 | 0.39 |
| Electric Drive #2 Efficiency | % | | | | 95.5 | 0.39 |
| Electrical Accessory Power | W | | | | 210 | 7.803 |

Wheel Radius : 0.317 m
Total Battery Energy : 8 kWh
Battery Voltage : 194.4 V