Bayesian Parameter Estimation for Heavy-Duty Vehicles

Eric Miller, Arnaud Konan, and Adam Duran

National Renewable Energy Laboratory

Abstract

Accurate vehicle parameters are valuable for design, modeling, and reporting. Estimating vehicle parameters can be a very time-consuming process requiring tightly-controlled experimentation. This work describes a method to estimate vehicle parameters such as mass, coefficient of drag/frontal area, and rolling resistance using data logged during standard vehicle operation. The method uses a Monte Carlo method to generate parameter sets that are fed to a variant of the road load equation. The modeled road load is then compared to the measured load to evaluate the probability of the parameter set. Acceptance of a proposed parameter set is determined using the probability ratio to the current state, so that the chain history will give a distribution of parameter sets. Compared to a single value, a distribution of possible values provides information on the quality of estimates and the range of possible parameter values. The method is demonstrated by estimating dynamometer parameters. The results confirm the method's ability to estimate reasonable parameter sets, and indicate an opportunity to increase the certainty of estimates through careful selection or generation of the test drive cycle.

Introduction

While consumer vehicle manufacturers publish parameters, commercial vehicles are a unique amalgam of well-characterized components that are tested very little after vehicle assembly. Vehicle parameters are most commonly estimated using a "coast-down" procedure, such as the test outlined in SAE standard J2263 [1]. In this test, a vehicle is driven to a set speed, the driveline is disengaged, and the vehicle coasts until it comes to a stop. Parameters are estimated by fitting the vehicle speed trace with a second-order polynomial [2]. Standard J2263 outlines controls for tests such as requiring low wind speed and a flat track, for example. This method is rigorous and well tested, but time consuming. A test that can be conducted while the vehicle fulfills its vocation will report parameters regularly throughout the life of the vehicle.

The potential benefits of good estimates for vehicle parameters are numerous. For modeling purposes, a set of basic parameters can be used to predict vehicle performance over a range of operating conditions [3]. Parameters are benchmarks for vehicle performance. More frequent measurement and improved accuracy will benefit designers and speed the development of new technologies. For researchers studying vehicle power consumption, parameters communicate national trends and the combined efficacy of new technologies [4]. An algorithm which can be applied to existing vehicle data saves the cost of further data collection.

Anticipating the advent of autonomous vehicles, work has been done to estimate parameters on-board to be used as a sort of feed-forward control [5]. A truck can increase its following distance as its mass increases, for example. Vehicles, especially commercial vehicles, convey loads of variable mass throughout the day. Working with an estimate of its own mass, a vehicle may be able to tune the engine, brakes or hybrid control strategy in real time to improve performance.

Various regression methods have been used to estimate vehicle parameters, road conditions, etc. from on-road data, often in real time, to improve vehicle performance. A least-squares with forgetting method utilizes a controller which selects suitable drive cycle segments for regression analysis [6]. Regression with forgetting has made good approximations of vehicle mass and road grade simultaneously[7]. Kalman filters have also been used to estimate vehicle parameters [8, 9, 10], and have done well when sensor noise is Gaussian and can be accurately estimated.

So-called Bayesian inference methods have been used to identify model parameters using minimally-processed real-world data [11, 12, 13]. One benefit of these methods is that their computational demand scales well as the number of parameters and the measured data increase, so they are often used with systems that have a large number of parameters. Also, bounds can be placed on the parameter search space so that non-physical parameter estimates (such as negative mass) are avoided.

Algorithm

The challenge is to find a vector of parameters $\underline{\theta}$, such that modeled and observed results are equal, $\hat{y} = y(\underline{\theta})$. The parameters space is searched using the Metropolis Hastings algorithm, which returns a posterior distribution rather than a single answer. The algorithm is sketched below.

Initialization

A random set of parameters, $^0\underline{\theta}$, is sampled from prior distributions.

1. The system model is evaluated for the parameter set. The result, $y(^i\underline{\theta})$, is obtained

2. Modeled and measured results are compared using squared error, χ^2.

$$^i\chi^2 = \sum_{t=0}^{t_f} \left(\hat{y}_t - y_t(^i\underline{\theta})\right)^2$$

(1)

3. A Gaussian probability density function centered at $\chi^2 = 0$ is used to express the likelihood of the parameter set:

$$P(^i\underline{\theta}) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{^i\chi^2}{2\sigma^2}}$$

(2)

4. The current parameter vector is perturbed by some random increments to produce a new parameter set θ^*.

$$^*\underline{\theta} = {^i\underline{\theta}} + \underline{J}$$

(3)

5. The new parameters are used in the vehicle model to generate an estimate $y(^*\underline{\theta})$.

6. The squared error of the new result is calculated:

$$^*\chi^2 = \sum_{t=0}^{t_f} \left(\hat{y}_t - y_t(^*\underline{\theta})\right)^2$$

(4)

7. The likelihood of this set of parameters is given by:

$$P(^*\underline{\theta}) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{^*\chi^2}{2\sigma^2}}$$

(5)

8. The Markov chain should visit areas of higher likelihood more frequently. To achieve this result, the acceptance parameter, α is calculated as:

$$\alpha = min\left\{\frac{P(^*\underline{\theta})}{P(^i\underline{\theta})}, 1\right\} = min\left\{e^{\frac{^i\chi^2 - ^*\chi^2}{2\sigma^2}}, 1\right\}$$

(6)

so that $\alpha = 1$ when $^*\chi^2 < {^i\chi^2}$ and the chain will move to regions of higher likelihood.

9. The proposed parameters are accepted with probability α or the parameters remain unchanged with probability 1 - α.

$$^{i+1}\underline{\theta} = \begin{cases} ^*\underline{\theta} & \texttt{with probability } \alpha \\ ^i\underline{\theta} & \texttt{with probability } 1 - \alpha \end{cases}$$

(7)

10. The counter is updated ($i = i + 1$), then the algorithm returns to step 2 if $i < i_{max}$. Otherwise the algorithm is finished.

The above description omits the calculation of the variables σ^2, $\underline{J}$. The variance, σ^2, of the model is calculated as:

$$\sigma^2 = \frac{1}{t_f - p} \sum_{t=0}^{t_f} \left(\hat{y}_t - y_t(^i\underline{\theta})\right)^2$$

(8)

where p is the length of θ, or the number of parameters to be estimated. The jump vector is formed by multiplying a vector of random numbers by a scaling matrix $\underline{\underline{R}}$, formed by first calculating the covariance matrix, $\mathcal{C}$:

$$\mathcal{C}_{kj} = \frac{\partial y(\theta_k)}{\partial \theta_k} \frac{\partial y(\theta_k)}{\partial \theta_j}$$

(9)

then:

$$\underline{\underline{R}} = Chol\left(\sigma^2 \underline{\underline{\mathcal{C}}}^{-1}\right)$$

(10)

The Cholesky decomposition, $Chol()$, gives a matrix, $\underline{\underline{R}}$, which is upper triangular with a positive diagonal as long as the matrix $\sigma^2\mathcal{C}$ is positive-definite[14]. The jump is a vector of random variables drawn from a normal distribution then scaled:

$$\underline{J} = \underline{\underline{R}}\underline{z}$$
$$\underline{z} = \mathcal{N}(\mu = 0, \sigma^2 = 1)$$

(11)

These variables were not included inside the iterative loop because they are not calculated at every iteration. Using $\underline{\underline{R}}$ to scale jump values is a method taken from [11]. Although σ^2 and some random jump are always calculated in initialization, subsequent updating of these variables is considered to be an adaptive refinement of the algorithm. Testing [15] shows that using an approximated covariance matrix to scale jump proposals speeds convergence, especially in cases where variables are highly covariant. In [16], Eydgahi et al. employed an approximate Hessian matrix to guide a Markov chain Monte Carlo algorithm. The authors note that without Hessian guidance, a stable estimate could not be attained for all 78 parameters in their model. Calculation of the covariance matrix can be computationally expensive, especially for large datasets. For this reason, the covariance matrix and variance are updated at regular intervals of 100 iterations and only after a set number of iterations called the "burn-in" or "warm-up" period. Additionally, parameter vectors are not saved to the posterior during the burn-in period, as they will include the chain segments used to walk from the random starting point, to the region of highest likelihood, which are not of interest here. Figure 1 shows the value of the burn-in period and the gradual convergence of independent chains.

Figure 1. Random walk generated for mass

Application

The method described above was applied to estimate the coefficients provided by a chassis dynamometer to test a class 8 tractor. The parameters to be estimated are:

$$\underline{\theta} = \left[A(lbf),\ C\left(\frac{lbf}{mph^2}\right),\ m(lbm) \right]$$

$$(12)$$

where A is a friction coefficient estimating rolling resistance and mechanical losses, C is a drag coefficient for aerodynamic losses, and mass is directly measurable. The power provided by the vehicle's engine is modeled as:

$$\underline{P}_{vehicle} = A\underline{v} + C\underline{v}^3 + m\underline{\dot{v}}\underline{v}$$
$$\underline{P}_{vehicle}[\underline{P}_{vehicle} < 0] = 0$$

$$(13)$$

The dynamometer measures instantaneous power supplied by the vehicle, $\hat{P}_{vehicle}$, which will be compared to the value calculated in Eq. 13. Measured data were only used if the vehicle output power was greater than 0. This was done because braking force is not accounted for in Eq. 13 and because the dynamometer has a drum only for the rear wheels and will, therefore, assist the vehicle in braking. For this reason, the rule $\underline{P}_{engine} \geq 0$ was added to Eq. 13. The ability to add logic to the model is an important feature of the Markov chain Monte Carlo method. The model is not required to be continuous or directly differentiable. The dynamometer in this study uses an electric generator to provide resistance to the vehicle, which is fixed so that its rear wheels are turning a large drum. The electronics inside the dynamometer should vary the force required to turn the drum so as to maintain the coefficients A, C, and m. However, the drum itself, along with the mechanical components that connect it to the generator will resist rotation as well. This means that the coefficients used at the generator will be lower than the coefficients measured in a coastdown test. Also, the resistance provided by various gears and axles will change as these components are warmed by testing. The result is that these components need to be tuned and checked throughout the test. The friction parameter, A, is known to be especially susceptible to drift, as the meshing friction of

the gears is temperature dependent. The dynamometer operator may change the dynamometer parameters to compensate for changes in internal loss as outlined in [17].

The tests were conducted for three separate cycles: the U.S. EPA Highway Fuel Efficiency Cycle (HWFET), The SAE j1376 Commuter Cycle (COMMUT), and the CARB Heavy Heavy-Duty Diesel Truck cycle less the low speed creep section (HHDDTNC). The combination of these cycles is meant to replicate the daily operation of the class 8 truck being tested. For each drive cycle, the Markov chain was initialized at three unique, random, locations. Each chain was run for 10^4 iterations, with a burn-in period of 5×10^3 iterations. Figure 1 shows the random walks generated for mass over all iterations of the estimator. Once the burn-in time has been reached, variance in the walks drops dramatically. At this point, it is also assumed that the walks are truly random, and independent of their starting condition. The "multi-chain" approach has been used to provide estimates for convergence [18], but was used in this case to choose a suitable number for total iterations and burn-in. If the chains do not converge to the same approximate region(s), the algorithm isn't working properly or has not run for enough iterations. The deviation in estimates between individual chains was less than 1% on average for the selected iteration and burn-in numbers.

Results

Parameter estimates obtained by the Markov-chain Monte Carlo Method and listed in table 1 are in the vicinity of the nominal parameters supplied to the dynamometer operator. When comparing estimates to the nominal parameters, however, it is worthwhile to note Figure 2, which shows modeled power traces match the measured trace more closely when the estimated, rather than nominal parameters where used in Eq. 13. To reproduce the trace, the combination of A,C, and m which provided the lowest squared error were used. The plots show that the vehicle power output can be more closely matched by estimated parameters than by the nominal parameters. Reproducing the measured power trace is a good metric for the quality of the estimate because the dynamometer is using the same model as the estimator to apply a resistive torque to the drum.

Table 1. Summary of estimated dynamometer parameters for each cycle. Mean and standard deviation are shown.

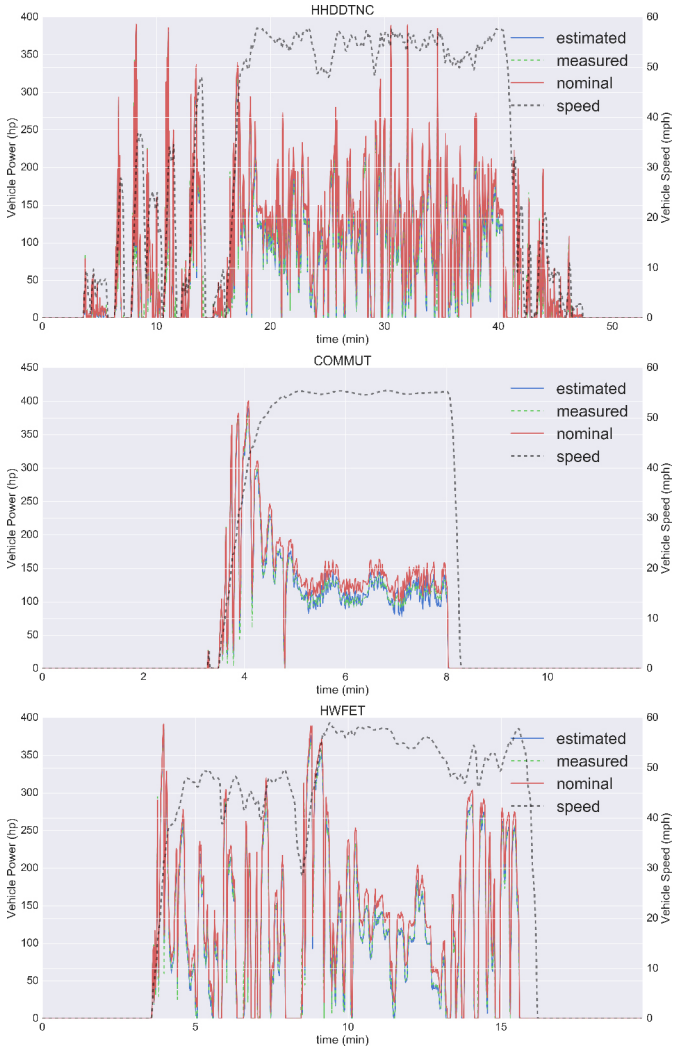| cycle	A	C	m
Nominal			
	391	0.168	1863
COMMUT			
	274 ± 101	0.164 ± 0.033	1870 ± 32.5
HHDDTNC			
	270 ± 10.5	0.164 ± 0.003	1887 ± 3.22
HWFET			
	263 ± 14.4	0.164 ± 0.004	1883 ± 4.86

Figure 2. Power trace for the three drive cycle

Each drive cycle was tested eight to ten times. The best estimate for each run is shown in Tables 2, 4, and 3.

Table 2. Estimated dynamometer parameters A, C, mass, and root mean squared error for the power trace Drive Cycle:HWFET

run	$m(lbm)$	$A(lbf)$	$C(lbf/mph^2)$	$rms(hp)$
1	1883	256.9	0.165	6.811
2	1883	261.9	0.164	6.461
3	1879	255.7	0.165	5.246
4	1888	256.6	0.166	4.835
5	1887	242.2	0.169	7.210
6	1878	253.8	0.168	5.943
7	1878	261.5	0.163	6.718
8	1892	268.9	0.164	7.400
9	1879	294.9	0.156	7.401
10	1878	275.7	0.162	7.175
avg	1883	262.8	0.164	6.520

Table 3. Estimated dynamometer parameters A, C, mass, and root mean squared error for the power trace Drive Cycle:COMMUT

run	$m(lbm)$	$A(lbf)$	$C(lbf/mph^2)$	$rms(hp)$
1	1903	178.8	0.196	6.748
2	1894	214.9	0.183	5.445
3	1824	403.3	0.122	5.629
4	1830	428.5	0.113	6.434
5	1897	183.4	0.191	6.665
6	1908	176.9	0.195	4.639
7	1841	366.6	0.132	5.909
8	1845	324.4	0.144	7.270
9	1892	178.1	0.196	6.322
10	1861	282.3	0.163	6.003
avg	1870	273.7	0.164	6.106

In examining average values, it seems that parameters were estimated adequately using the COMMUT drive cycle. However, standard deviation for the parameter estimates are an order of magnitude larger than those for the HWFET and HHDDTNC cycles. Levene's test reveals [19] that the COMMUT cycle test runs display unequal variance compared to HWFET and HHDDTNC. One possible cause of deviation is operator error, meaning that the dynamometer parameters were altered significantly between runs. Another possible cause is that the COMMUT cycle is ill-suited for parameter estimation using the current vehicle power model.

Model Evaluation

The model in Eq. 13 was evaluated by performing an eigen-decomposition on the covariance matrix (Eq. 9) associated with the best estimate for each cycle. such that:

$$\underline{\underline{C}}\,\underline{x}_i = \lambda \underline{x}_i \; for \; i \in \{1...m\}$$

(14)

An ideal model for parameter estimation is one in which the covariance matrix has no off-diagonal terms, meaning that each parameter is associated with a unique contribution and the eigenvectors are Euclidean basis vectors:

Table 4. Estimated dynamometer parameters A, C, mass, and root mean squared error for the power trace Drive Cycle:HHDDTNC

run	$m(lbm)$	$A(lbf)$	$C(lbf/mph^2)$	$rms(hp)$
1	1891	258.1	0.168	5.009
2	1883	273.0	0.164	4.475
3	1888	261.2	0.167	4.202
4	1884	259.2	0.167	4.528
5	1890	281.9	0.160	4.565
6	1886	286.8	0.159	5.099
7	1889	271.5	0.164	4.904
8	1883	270.0	0.165	4.488
avg	1887	270.2	0.164	4.659

$\underline{e}_1 = [1, 0, ..., 0]...\underline{e}_m = [0, ...0, 1]$. For each eigenvector:

$$\psi_i = \sin^{-1}(\underline{x}_i \times \underline{e}_i)$$

(15)

The angle of rotation is equal for the A and C parameters, while it is roughly zero for m. This relationship indicates that the measured data most directly conveys information about the mass and some linear combination of A and C.

Table 5. Eigen-decomposition metrics from dynamometer tests

λ_A	$\psi_A(^o)$	λ_C	$\psi_C(^o)$	λ_m	$\psi_m(^o)$
COMMUT					
13.12	19.07	5.348	19.07	3.650	0.019
HHDDTNC					
13.88	21.08	6.034	21.08	5.318	0.019
HWFET					
13.41	18.70	5.800	18.70	4.940	0.020

The probability densities in the appendix 1 show every credible parameter set for every run. These point clouds indicate strong coupling between the A and C parameters. Uncorrelated parameters would produce elliptical point clouds with the major and minor axes parallel to the A and C axes. Sivia [20] shows that a point cloud will slope downward when the system is identifiable in the dimension $A + C$, but not $A - C$, which is sensible for two parameters that both dissipate more power at higher vehicle speed. The point clouds also reveal higher uncertainty in estimates made from the COMMUT cycle. For a smaller deviation, σ^2, point clouds will encompass a smaller area, reflecting greater certainty in the estimates. Gearhart and Wang [21] propose that a good model has low rms, fewer parameters, and less covariance (or estimate scattering). Estimates for the COMMUT cycle show high rms error, and would have encompassed an even larger area, if not for the bounds placed on the estimates:

$$
\begin{aligned}
A &\in & \{0, 500\} \\
C &\in & \{.1, .2\} \\
m &\in & \{1000, 5000\}
\end{aligned}
$$

(16)

which demonstrates another valuable feature of Markov chain Monte Carlo, which is the ability to constrain the search space using any a priori knowledge (such as nominal parameters) to obtain reasonable estimates even in cases of high uncertainty.

Identifiability

Identifiability is a measure of each parameter's proclivity to estimation. There are two ways that a parameter can be unidentifiable. First, it may be that the covariance matrix(Eq. 9) is rank-deficient. Usually, this failure indicates a poor selection of model. In this case, it can be shown that the covariance matrix will not be rank deficient so long as:

$$
\underline{a} \neq \underline{v} \neq \underline{v}^3
$$

(17)

in which case the matrix would have linearly dependent rows, and

$$
\underline{a}, \underline{v} \neq 0
$$

(18)

inn which case the matrix would have a row of zeros (linearly dependent again). For parameter estimation, accuracy decreases as the covariance matrix approaches singularity. The second issue is practical identifiability, which suffers from noise present in measured signals. To test practical identifiability, a dummy case was used. The drive cycles were obtained from NREL's on-line database [22]. Using the speed traces, engine power was calculated as in Eq. 13, but with dummy coefficients:

$$
\underline{\theta}_t = [m = 2000, A = 300, C = 0.15]
$$

(19)

Adding noise:

$$
\hat{P}_{engine} = A\underline{v} + C\underline{v}^3 + m\underline{v}\dot{\underline{v}} + \Omega\underline{w}
$$

(20)

Noise is represented by $\Omega\underline{w}$, where $\underline{w}(\mu = 0, \sigma = 1.)$ is a vector of random values sampled from a normal distribution and Ω is noise magnitude. Engine power and the exact speed trace were used to estimate the coefficients in 19. Using the dummy case, drive cycle variety, measurement error, and cycle length will be tested to characterize their effect on the accuracy of parameter estimation.

To evaluate the variety in each drive cycle, the data were binned by unique speed and acceleration combinations. The number of bins with one or more data points were counted and divided by the total number of possible bins to give a coverage fraction:
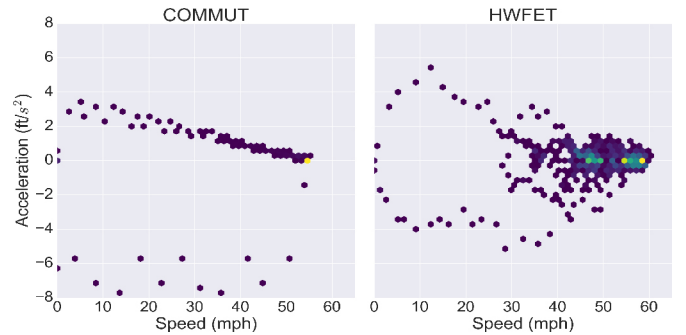

Figure 3. Coverage by COMMUT and HWFET cycles

Figure 3 shows the coverage attained by the COMMUT and HWFET cycles. As expected, the HWFET cycle shows more coverage. In figure 4, the coverage is plotted against the estimation error, calculated as:

$$
e_{est} = \left| \frac{\underline{\theta}_t - \underline{\theta}}{\underline{\theta}_t} \right|
$$

(22)

Coverage has a significant impact on estimate accuracy. In this case, the curve fit was roughly logarithmic, predicting diminished returns for higher coverage, but also very high error for cycles with the sparsest coverage. The curve predicts that the COMMUT cycle will provide worse estimates than HWFET and HHDDTNC.
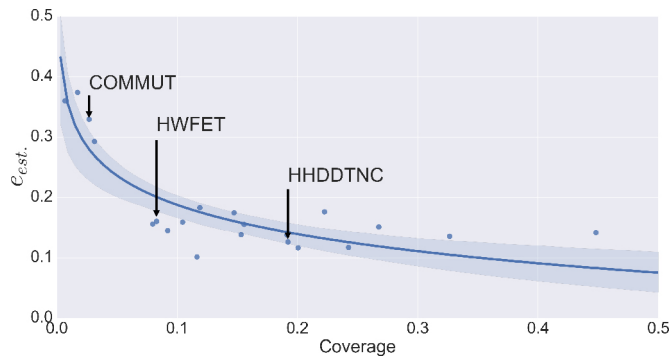
Figure 4. Impact of state variety on estimate deviation

$$\text{coverage} = \frac{\text{bins occupied}}{\text{bins available}}$$

$$(21)$$

Noise is another possible cause of mis-estimation. To test this effect, the dummy case was used again, with the noise magnitude, Ω, being gradually increased. Figure 5 shows the impact of noise on the accuracy of the parameter estimates. The RMS error increased proportional the square of Ω, along with the estimate error. Figure 5 shows the impact of noise on estimate accuracy. The results indicate that parameters A and C will be the first estimates affected by noise, and that these parameters can be mis-estimated without as much impact on the rms error.
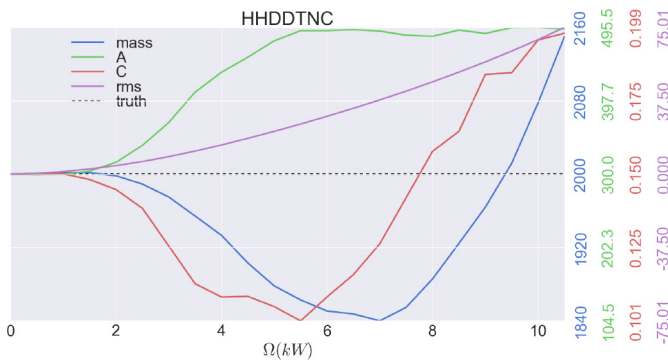
Figure 5. Effect of noise on estimated parameters
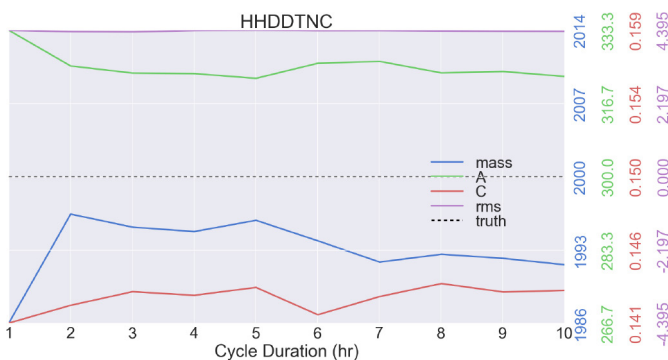
Figure 6. Effect of cycle duration on estimated parameters

Figure 6 shows the effect of cycle duration on accuracy. A fixed noise magnitude, $\Omega = 5$, was used for this test, and the HHDDTNC cycle was cloned to add duration, but not variety. The result is a nearly constant rms error, and little variation in parameter estimates. Although noise varied between cloned drive cycles, the result indicates that the cloned cycles were not much more valuable than duplicate data.

CONCLUSION

A Markov chain Monte Carlo method was used to estimate vehicle parameters for a dynamometer test case. Although the accuracy of the estimate is difficult to verify, parameters agree reasonably well with nominal parameter values. The benefits to this method are scalability and the ability to use any driving case for estimation. The method was shown to make better estimates when given a larger dataset, and when the data were measured with less noise. These findings will inform future use of the technology in on-road vehicle testing. Opportunities abound for improved implementation of the Metropolis-Hastings algorithm. Methods have been developed for the detection of outliers and for hierarchical combinations of data. As the accuracy of estimates improve, applications will be pursued more actively.

REFERENCES

1. SAE International Surface Vehicle Recommended Practice, "Road Load Measurement Using On-board Anemometry and Coastdown Techniques," SAE Standard J2263, 2008.

2. McAuliffe, B. and Chuang, D., "Track-Based Aerodynamic Testing of a Heavy-Duty Vehicle: Coast-Down Measurements," *SAE Int. J. Commer. Veh.* 9(2):381-396, 2016, doi:10.4271/2016-01-8152.

3. Brooker, A., Ward, J., and Wang, L., "Lightweighting Impacts on Fuel Economy, Cost, and Component Losses," SAE Technical Paper 2013-01-0381, 2013, doi:10.4271/2013-01-0381.

4. Carlson, R., Lohse-Busch, H., Diez, J., and Gibbs, J., "The Measured Impact of Vehicle Mass on Road Load Forces and Energy Consumption for a BEV, HEV, and ICE Vehicle," *SAE Int. J. Alt. Power.* 2(1):105-114, 2013, doi:10.4271/2013-01-1457.

5. Ryu, J., "State and Parameter Estimation For Vehicle Dynamics Control Using GPS," Ph.D. dissertation, Stanford University,

6. Zhang, D., Ivanco, A., and Filipi, Z., "Model-Based Estimation of Vehicle Aerodynamic Drag and Rolling Resistance," *SAE Int. J. Commer. Veh.* 8(2):433-439, 2015, doi:10.4271/2015-01-2776.

7. Vahidi, A., Stefanopoulou, A., and Peng, H., "Recursive Least Squares With Forgetting For Online Estimation of Vehicle Mass and Road Grade: Theory and Experiments," *Vehicle System Dynamics*, vol. 43, 2005, DOI: 10.1080/00423110412331290446.

8. Winstead, V. and Kolmanovsky, I., "Estimation of Road Grade and Vehicle Mass via Model Predictive Control," Toronto, Canada, August 28-31, pp. 1588-1593, DOI: 10.1109/CCA.2005.1507359.

9. Lingman, P. and Schmidtbauer, B., "Road Slope and Vehicle Mass Estimation Using Kalman Filtering," *Vehicle System Dynamics*, 2002, DOI: 10.1080/00423114.2002.11666217.

10. Reina, G., Paiano, M., and Blanco-Claraco, J.-L., "Vehicle parameter estimation using a model-based estimator," *Mechanical Systems and Signal Processing*, pp. -, 2016, DOI: http://dx.doi.org/10.1016/j.ymssp.2016.06.038.

11. Crews, J. H. and Smith, R. C., "Modeling and Bayesian parameter estimation for shape memory alloy bending actuators," Goulbourne, N. C. and Ounaies, Z., Eds. International Society for Optics and Photonics, apr 2012, p. 83421N, DOI: 10.1117/12.914792.

12. Wiecki, T. V., Sofer, I., and Frank, M. J., "HDDM: Hierarchical Bayesian estimation of the Drift-Diffusion Model in Python," *Frontiers in Neuroinformatics*, vol. 7, p. 14, 2013, DOI: 10.3389/fninf.2013.00014.

13. Lauret, P., Boland, J., and Ridley, B., "Bayesian Statistical Analysis Applied to Solar Radiation Modelling," *Renewable Energy*, vol. 49, pp. 124-127, 2013, DOI: 10.1016/j.renene.2012.01.049.

14. Shotwell, M. S. and Slate, E. H., "Bayesian Outlier Detection with Dirichlet Process Mixtures," *Bayesian Analysis*, vol. 6, no. 4, pp. 665-690, 2011, DOI: 10.1214/11-BA625.

15. Solonen, A., "Monte carlo methods in parameter estimation of nonlinear models," Master's thesis, Lappeenrante University of Technology, 2006.

16. Eydgahi, C.-e. Hoda, a., "Properties of Cell Death Models Calibrated and Compared using Bayesian Approaches," *Molecular systems biology*, vol. 9, no. 1, p. 644, 2013, DOI: 10.1038/msb.2012.69.

17. "40CFR.86 Subpart B; Emission Regulations for 1977 and Later Model Year New Light-Duty Vehicles and New Light-Duty Trucks and New Otto-Cycle Complete Heavy-Duty Vehicles; Test Procedures," 2016.

18. Gelman, A. and Rubin, D., "Influence from Iterative Simulation Using Multiple Sequences," *Statistical Science*, vol. 7, no. 4, pp. 457-511, 1992.

19. NIST/SEMATEC, *Handbook of Statistical Methods*. [Online]. Available: http://www.itl.nist.gov/div898/handbook/

20. Sivia, D., *Data Analysis: A Bayesian Tutorial*. Oxford University Press, 1997.

21. Gearhart, C. and Wang, B., "Bayesian metrics for comparing response surface models of data with uncertainty," *Structural and Mulitdisciplinary Optimization*, vol. 22, 2000.

22. Kelly, K., Prohaska, R., Ragatz, A., and Konan, A., "NREL DriveCAT - Chassis Dynamometer Test Cycles," www.nrel.gov/transportation/drive-cycle-tool, 2016.

CONTACT INFORMATION

Eric Miller can be reached at
Eric.Miller@NREL.gov

Adam Duran can be reached at
Adam.Duran@NREL.gov

***rms* - root mean squared**

Ken Kelly can be reached at
Kenneth.Kelly@NREL.gov

ACKNOWLEDGMENTS

DEFINITIONS/ABBREVIATIONS

i - iteration counter

i_{max} - maximum number of iterations

j - parameter index

χ^2 - squared error

t_f - drive cycle length

$\hat{y}$ - measured data

y - modeled data

$\underline{\theta}$ - parameter set

$^*\underline{\theta}$ - proposed parameter set

$P(\theta)$ - likelihood probability of θ

σ^2 - variance of the posterior

$\underline{J}$ - jump vector

α - Bayes factor

$\mathcal{N}(\mu, \sigma^2)$ - sample from normal distribution centered at μ w/ deviation σ^2.

$\underline{\underline{C}}$ - estimated covariance matrix

$\underline{\underline{R}}$ - inverse estimated covariance matrix

Chol() - Cholesky decomposition

A - dynamometer friction parameter

C - dynamometer drag parameter

m - dynamometer mass parameter

$\underline{x}_j$ - eigenvector associated with parameter j

λ_j - eigenvalue associated with parameter j
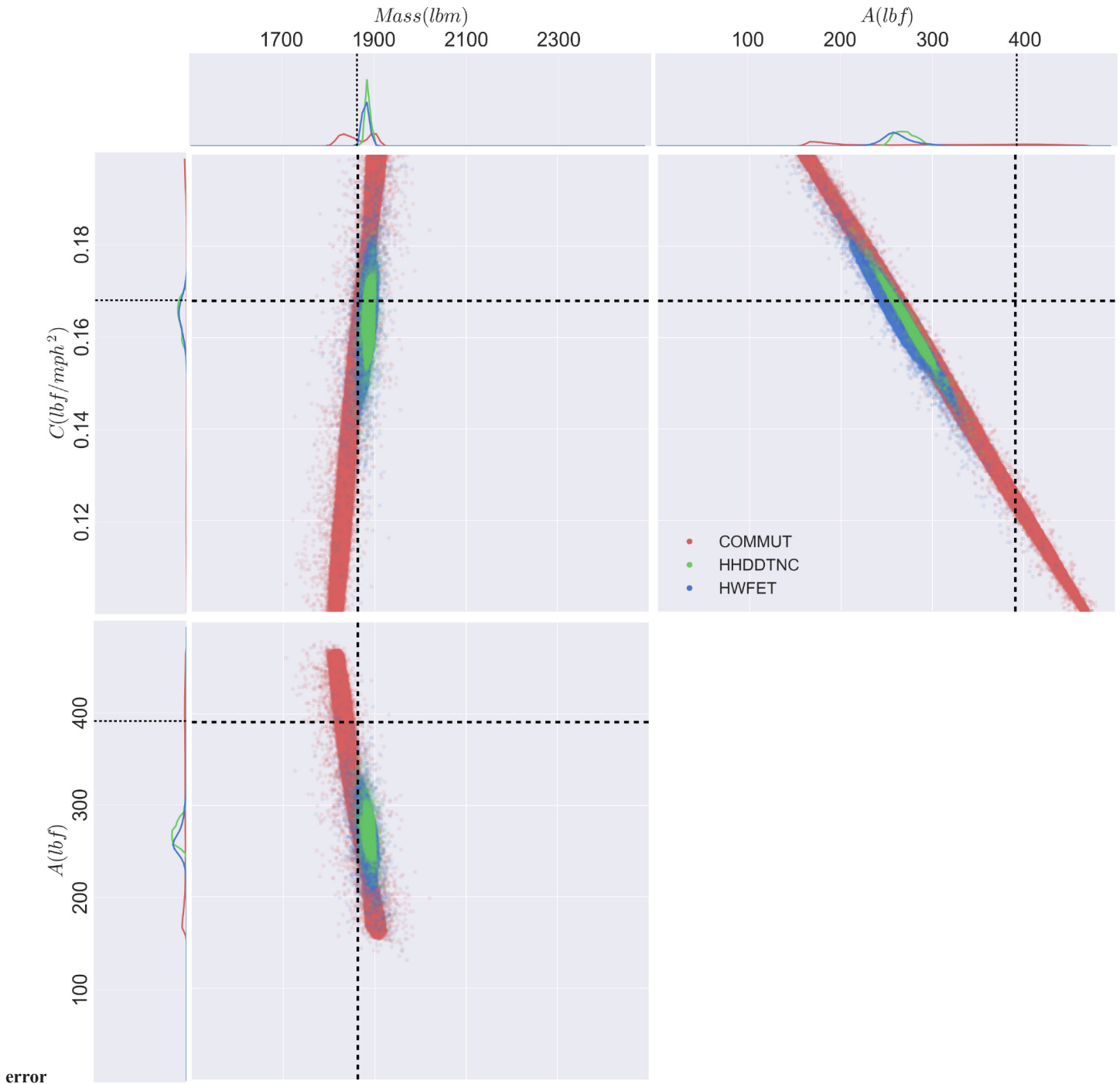
$\underline{e}_j$ - basis vector associated with parameter j

ψ_j - angle between $\underline{x}_j$ and $\underline{e}_j$

Appendix 1. Joint posteriors pictured in the center, probability densities at the boundaries. Dashed lines show nominal parameter values