Journal of Power Sources 183 (2008) 275–281

Contents lists available at ScienceDirect

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

Drive-train simulator for a fuel cell hybrid vehicle

Darren Brown[a], Marcus Alexander[b], Doug Brunner[a], Suresh G. Advani[a,*], Ajay K. Prasad[a]

[a] Fuel Cell Research Laboratory, Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA
[b] Electric Power Research Institute, Palo Alto, CA 94304, USA

ARTICLE INFO

Article history:
Received 7 February 2008
Received in revised form 23 April 2008
Accepted 24 April 2008
Available online 14 May 2008

Keywords:
Vehicle simulation
ADVISOR
Fuel cell
Hybrid vehicle
LFM

ABSTRACT

The model formulation, development process, and experimental validation of a new vehicle powertrain simulator called LFM (Light, Fast, and Modifiable) are presented. The existing powertrain simulators were reviewed and it was concluded that there is a need for a new, easily modifiable simulation platform that will be flexible and sufficiently robust to address a variety of hybrid vehicle platforms. First, the structure and operating principle of the LFM simulator are presented, followed by a discussion of the subsystems and input/output parameters. Finally, a validation exercise is presented in which the simulator's inputs were specified to represent the University of Delaware's fuel cell hybrid transit vehicle and "driven" using an actual drive cycle acquired from it. Good agreement between the output of the simulator and the physical data acquired by the vehicle's on-board sensors indicates that the simulator constitutes a powerful and reliable design tool.

1. Introduction

With the introduction of hybrid vehicle technology, the relatively simple process of system level vehicle powertrain design has become more complex resulting in escalating research and development costs. To contain these costs, there is a critical need to develop and validate vehicle simulators which can predict the performance of the vehicle propulsion system under a variety of driving conditions by accurately modeling every on-board subsystem. Once the simulation tool is validated against actual vehicle data, it can be used to reliably simulate and optimize new and more advanced vehicle designs. Simulation tools are useful for the design and performance optimization of vehicles containing multiple power sources and drive systems. This need has stimulated the development of numerous simulation tools including PSAT [2], V-ELPH [3], ADVISOR [4], and others [1,5–7].

Each of these existing simulation systems can simulate the entire vehicle powertrain. Typically, the simulation model is highly advanced, and requires the input of a large number of vehicle parameters. However, these models typically embody a complex design and structure that does not permit rapid and relatively simple modifications to the fundamental vehicle design as would be required, for example, to conduct a parametric study that explores the influence of various input parameters on the vehicle performance. Therefore, a new simulation tool was developed in this study for the express purpose of providing a flexible, easily modifiable structure such that changes to the fundamental vehicle design could be easily implemented without compromising model accuracy or computational performance.

The tool developed for this purpose is called LFM (Light, Fast, and Modifiable), initiated originally by the Electric Power Research Institute (EPRI). As its name suggests, this tool is specifically designed for ease of modification and rapid execution. Hence this tool is well suited for optimizing the design of a hybrid vehicle powertrain by a process in which the simulation is executed rapidly and repeatedly with the system parameters being varied incrementally over a chosen range without user interaction, and the simulation outputs stored for subsequent analysis.

For this paper, the LFM simulation parameters were tailored specifically to match the University of Delaware's fuel cell electric hybrid bus. The simulator was exercised using actual drive cycles obtained from the vehicle's GPS system, and the simulation was validated by comparing its outputs with data acquired by the vehicle's various on-board sensors.

2. Model design

The operating environment for LFM is MATLAB/Simulink [8,9]. MATLAB is a programming environment, developed by The MathWorks, designed for rapid numerical code development. Simulink is an add-on package for MATLAB that greatly simplifies the system modeling process through a graphical programming interface.

* Corresponding author at: GW Laird Professor of Mechanical Engineering, University of Delaware, 126 Spencer Laboratory, Newark, DE 19716, USA.
Tel.: +1 302 831 8975.
E-mail address: advani@udel.edu (S.G. Advani).

0378-7753/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.jpowsour.2008.04.089

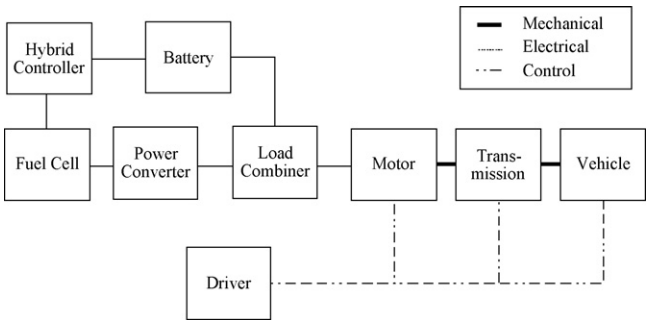

Fig. 1. LFM schematic.

The basic structure of LFM is a drive cycle-based, forward-facing model. Drive cycle-based means that the simulation is "driven" entirely by an input drive cycle. At each time step, the simulator compares the desired speed from the drive cycle with the current speed of the vehicle, and calculates driving commands to minimize their difference. The Simulink model itself is constructed from subsystems that are linked using electrical, mechanical, and control signal links. Fig. 1 shows the basic layout of the LFM simulator for the current case study.

2.1. Structure

The following sections will elaborate, in some detail, the function of each of the subsystems that constitute the overall LFM model.

2.1.1. Fuel cell subsystem

The fuel cell subsystem receives a DC current request from the power converter downstream. The fuel cell system uses this current request to determine the corresponding voltage output from the stack using a simple lookup table in the fuel cell data spreadsheet. This look-up table is created from actual performance data logged from the fuel cell stack (polarization curve) in the vehicle during operation and contains voltage values indexed by current. The other major calculation within the fuel cell subsystem is fuel usage which is obtained from another lookup table that lists the hydrogen consumption of the stack for a given current. Fuel cell balance-of-plant (BOP) loads are calculated by the accessory subsystem (Section 2.1.5) based on the power request. Fig. 2 shows a schematic of the fuel cell subsystem.

2.1.2. Hybrid controller

The hybrid controller is responsible for determining the electric power needed from the fuel cell system. This subsystem takes inputs from the batteries, the fuel cell, and the load combiner and determines the power needed based on a control algorithm. This algorithm can be easily modified so that new control strategies can be rapidly implemented and evaluated. Fig. 3 shows a schematic of the hybrid controller.

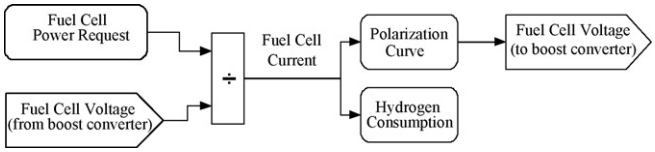

Fig. 2. Fuel cell subsystem.

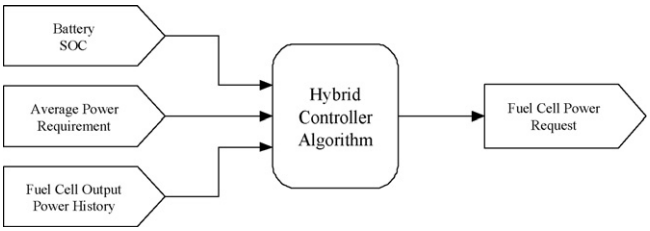

Fig. 3. Hybrid controller.

2.1.3. Battery

The battery subsystem (Fig. 4) is responsible for calculating state-of-charge (SOC), internal resistance, voltage, and current and voltage limits. The SOC is calculated by integrating the current from the battery pack with respect to time, and then subtracting this from the nominal charge capacity of the batteries. For example, if the batteries nominally hold 100 Ah, and the vehicle draws 20 A for 1 h, the SOC will be 80%.

Using the calculated value for SOC, the battery subsystem refers to a lookup table to determine open circuit voltage and internal resistance. These data are typically provided by the battery manufacturer. For this particular study, the data were provided by Ebus, Inc., the manufacturer of the vehicle. Using those values, the available battery voltage is calculated by subtracting voltage loss from the open-circuit voltage, or $V_{\text{available}} = V_{\text{OC}} - IR$ where V_{OC} is the open-circuit voltage of the battery pack, I is the current, and R is the total internal resistance of the string. This is a standard internal resistance model of a battery pack that was validated by Johnson [10]. Also, current and voltage limits are placed on the output as specified by the component configuration information in the battery spreadsheet. These data are also typically provided by the battery manufacturer. For example, manufacturers will typically specify maximum charge and discharge currents. This subsystem implements these extrema.

2.1.4. Load combiner

The load combiner subsystem is responsible for distributing the downstream load (traction system and accessories) among the different power sources. For this particular model, the basic strategy is to take any transient load needed by the motor, and accessories, directly from the batteries while the fuel cell system simply delivers the power requested by the hybrid controller. This subsystem is also responsible for scaling the power request from the hybrid controller by the boost converter efficiency and battery voltage yielding the current needed from the fuel cell stack. In this manner, the power delivered by the fuel cell system is equal to the power request from the controller (provided the request is greater than zero and less that the maximum fuel cell power). The fuel cell system power request is limited in the controller, so the request will never exceed the maximum output power. Fig. 5 schematically shows how the load combiner distributes the downstream load between the battery and the fuel cell.

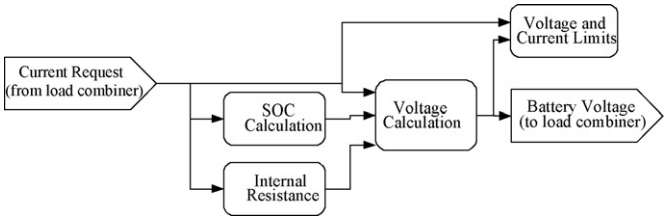

Fig. 4. Battery subsystem.

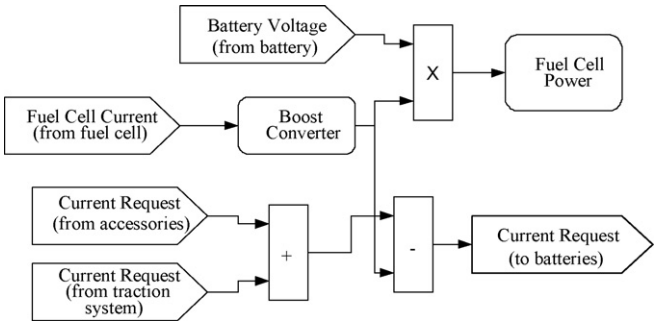

Fig. 5. Load combiner subsystem.

2.1.5. Accessory load

The accessory load subsystem calculates the current needed by each of the non-traction-related electric loads. These include: battery chiller, HVAC, air compressor, power steering, and fuel cell balance-of-plant. As shown in Fig. 6, at the top is the battery chiller whose power requirement can vary depending on the temperature of the batteries. Since this LFM model does not perform thermal modeling of the batteries, this quantity is taken as a constant load accounting for the duty cycle of the compressor. Below that is the HVAC system which can be toggled on or off by the configuration parameters that are loaded before the simulation begins. This toggle is meant to be a drive-cycle global quantity, meaning it dictates whether or not the HVAC is on for the entire drive cycle, not whether or not it is on at a particular moment in time. Similar to the battery chiller, the vehicle air compressor is modeled as a constant load since air use is not quantified by the model. Finally, the fuel cell balance-of-plant loads are modeled in a specific subsystem for each. Within the subsystem, the loads are calculated as functions of the power request. These functions can be easily modified for experimentation purposes.

2.1.6. Traction motor

The traction motor (also known as the drive motor) is one of the most important parts of the simulator (Fig. 7). In this subsystem, all of the necessary calculations are performed to determine available drive torque that eventually results in vehicle motion. The inputs to this subsystem are torque command, upstream available voltage, and downstream angular velocity. The outputs are current request and available torque.

Given an upstream available voltage, the motor calculates the necessary current to achieve the torque requested from the vehicle controller. This is done by first limiting the inputs, and then calculating losses. Torque limits are typically specified by the motor manufacturer, or they can be determined with current or power limits. Speed limits are also typically specified by the manufacturer based on balance tolerances and bearing design. The losses block (see Fig. 7) uses a two-dimensional map to determine the amount of power lost for all operating points. The map is indexed by torque and angular velocity.

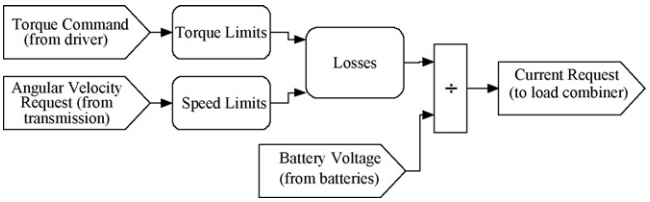

Fig. 7. Traction motor subsystem.

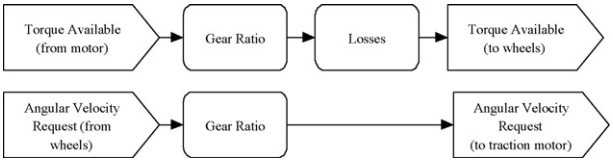

Fig. 8. Transmission.

2.1.7. Transmission

The transmission subsystem (Fig. 8) is a relatively simple system. The two most important calculations for a transmission model are input/output torques and losses. This model takes the torque from the traction motor and converts it to a torque to the wheels. The calculation is done using a constant gear ratio and torque loss vs. speed map. In addition to these calculations, the transmission subsystem calculates the upstream angular velocity based on the downstream angular velocity from the wheels.

2.1.8. Wheels/vehicle

The wheels/vehicle subsystem (Fig. 9) is the final link in the overall model. This subsystem is responsible for calculating all of the external forces on the vehicle including aerodynamic drag, gravity drag (grade), rolling resistance, and wheel torque. This subsystem balances all of these forces and determines vehicle acceleration. Aerodynamic drag is calculated by $F_{drag} = \rho V^2 C_D A/2$ where ρ is the density of air, V is the vehicle's velocity through the air, C_D is the drag coefficient of the vehicle, and A is the vehicle's frontal area.

Gravity drag, also known as grade, is calculated using $F_{grade} = mg\sin(\tan^{-1}(\text{grade}))$ where m is the mass of the vehicle,

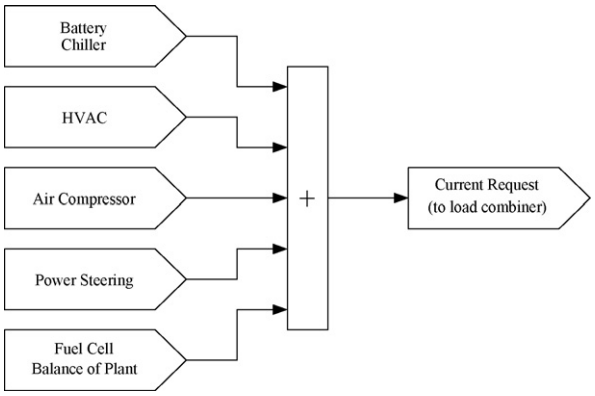

Fig. 6. Accessory subsystem.

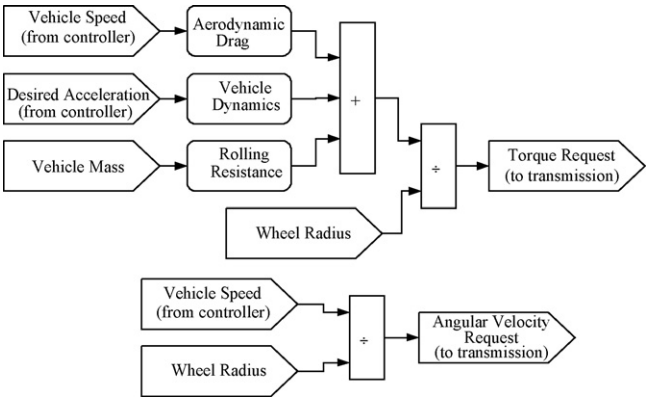

Fig. 9. Wheel/vehicle subsystem.

Table 1
Example internal resistance data

State-of-charge	Charging internal resistance (Ω)	Discharging internal resistance (Ω)
100	0.206	0.053
90	0.110	0.055
80	0.095	0.060
70	0.089	0.070
60	0.083	0.091
50	0.078	0.131
40	0.072	0.213
30	0.067	0.375
20	0.061	0.700
10	0.056	1.350
0	0.050	2.650

g is the acceleration of gravity, and grade is the current road grade (positive if climbing). Finally, rolling resistance is calculated using the formula $F_{RR} = mg(C_{RR2}V + C_{RR1})$ where C_{RR1} is the first coefficient of rolling resistance, and C_{RR2} is the second coefficient of rolling resistance (speed-dependent). Both of these values are typically provided by the tire manufacturer.

2.2. Input/output system

In contrast to most commercially available simulation packages, LFM has no user interface other than the traditional MATLAB interface. Loading of the simulation inputs is done at the script level, execution of the simulation is performed directly through Simulink, and data analysis is conducted with variables on the workspace. This design, while seemingly complex and cumbersome, is actually more powerful than other designs. First, this type of I/O system lends itself well to rapid simulation iterations. For example, the simulation and all its ancillary functions can be scripted such that the entire simulation can be run repeatedly without user intervention as part of an optimization process. Second, the simulation structure itself is designed to be easily modifiable, and hence, the I/O system must allow modifications to match the needs of the simulation structure.

2.2.1. Input spreadsheets

All of the vehicle input parameters are stored in spreadsheets, organized by subsystem, in a human-readable form. For example, the data regarding the internal resistance of the batteries are kept in a table similar to that shown in Table 1. These data can be taken directly from a manufacturer's data sheet, extrapolated from experimental data, or can be entirely theoretical. Storing these values in a spreadsheet table rather than a more typical data storage method like a text file permits users to make modifications more easily and helps prevent alignment errors.

2.2.2. Output

The default output from LFM is simply to write the desired quantities to the MATLAB workspace. These values are written during the simulation, but are typically not available for reading until the simulation is stopped. For analysis of the output data, a script was written that calculates numerous important quantities as shown in Table 2. These parameters were used for the system validation, as shown in the next section.

Fig. 10. University of Delaware's fuel cell hybrid bus.

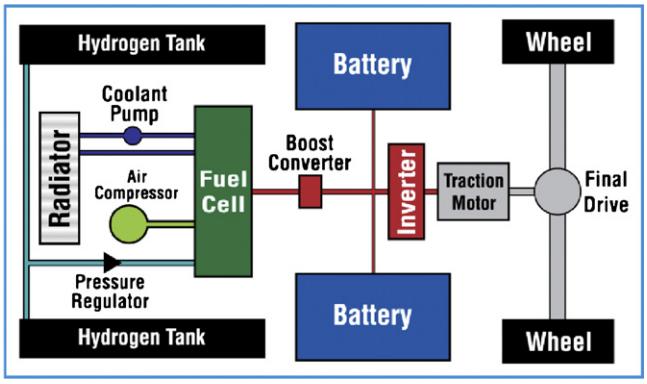

Fig. 11. Simplified power system schematic.

3. System validation

Before any design studies could be performed using the new LFM simulator, its output was compared with the physical data acquired from a real-world vehicle during test-driving. In the next three sections, the test vehicle and data acquisition system will be described, followed by qualitative and quantitative validations of the simulation.

Table 2
Important global output parameters

Parameter	Description	Expression
Total energy use	Total amount of energy output from the batteries and fuel cell	$\int_0^T \text{Power use}(t)\,dt$ for Power Use > 0 t is time, and T is drive cycle length in s
Energy recovered	Total amount of negative energy (regenerative) from the traction motor	$\int_0^T \text{Motor power}(t)\,dt$ for Motor power < 0
ΔSOC	Change in state-of-charge from beginning to end of the drive cycle	$\text{SOC}_{\text{beginning}} - \text{SOC}_{\text{end}}$
Fuel cell energy output	Gross energy output of the fuel cell stack	$\int_0^T \text{Fuel cell power}(t)\,dt$

3.1. Test vehicle

The vehicle used for this study was designed and constructed by Ebus, Inc. located in Downey, CA. Fig. 10 shows a picture of the bus. It is a fuel cell electric hybrid bus that can hold 22 seated and 10 standing passengers. The bus is driven by a single three-phase AC induction motor that is rated for 130 kW peak and 100 kW continuous. The motor is coupled to the rear drive wheels through a single speed chain drive and a differential. The motor is powered from two 300 V strings of Nickel–Cadmium batteries. Each string consists of 50 blocks, each containing five cells. The cells are rated for a nominal charge capacity of 100 Ah. The strings are connected in parallel for a total energy capacity of 60 kWh. This typically gives the vehicle approximately 70 km of all-battery range.

The fuel cell is a Ballard Mark9 SSL 110 cell 19.4 kW stack. The hydrogen is stored in two composite high-pressure tanks located on the roof of the bus. The tanks are rated for 350 bar and can hold approximately 12.8 kg of hydrogen. On average, this amount of hydrogen yields a travel range of about 260 km. Table 3 summarizes the important specifications of the bus. Fig. 11 shows how the drive train is configured as well as the fuel cell balance of plant.

Table 3
Important vehicle statistics

Parameter	Value
Length	6.7 m
Width	2.3 m
Gross weight	9300 kg
Curb weight	7030 kg
Maximum speed	72 km/h
Traction motor	130 kW AC induction
Transmission	1-Speed chain drive
Batteries	300 V 200 Ah Nominal NiCd
Fuel cell	Ballard Mark9 SSL 19.4 kW
Hydrogen storage	12.8 kg at 350 bar
Range	260 km
Fuel economy	21.4 l/100 km (gasoline equivalent)

3.2. Data acquisition system

To construct a test drive cycle, the vehicle was driven over representative routes with constant Global Positioning System (GPS) tracking. The GPS periodically collects time, speed, altitude, and position information that can be used to construct a drive cycle. The GPS used for this study was a Garmin 17HVS marine OEM tracker. This GPS uses the Wide Area Augmentation System (WAAS) which allows for an accuracy of almost 1 lateral meter. The GPS outputs all of the relevant data through an RS-232 connection in NMEA 0183 (National Marine Electronics Association) standard format to a data-logging computer.

For vehicle specific data, an interface was constructed for data transfer from the on-board vehicle computer to the data-logging computer. These data include quantities such as battery state-of-charge, battery current, motor power, fuel cell power, and configuration parameters. Many of these quantities are used in the simulator validation process. Table 4 shows all of the relevant data that are collected from the vehicle.

3.3. Vehicle to simulator comparison

The validation process presented here is similar to that presented by Syed et al. [11]. The plots shown in Figs. 12 and 13 demonstrate the validity of the LFM simulator. Each figure shows

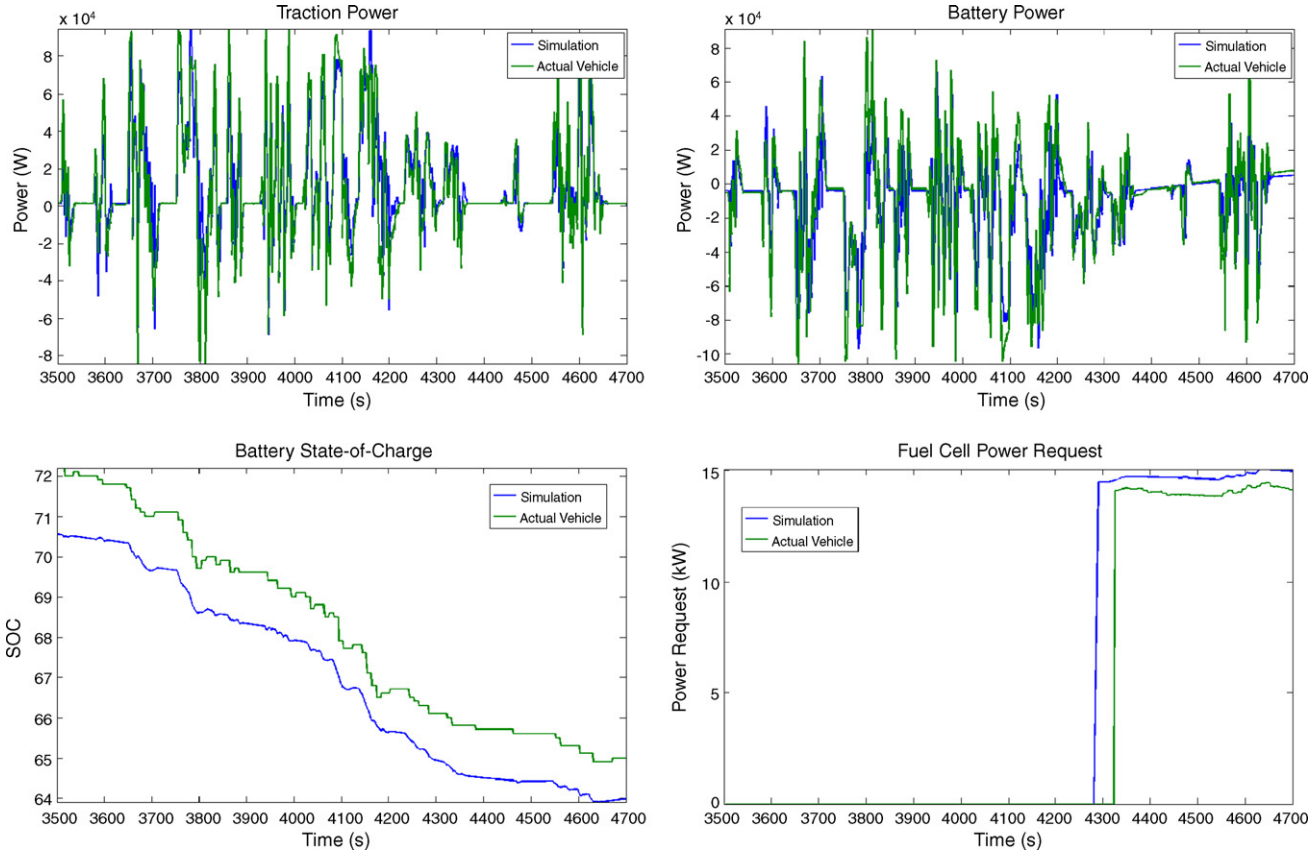

Fig. 12. Drive Cycle 1—comparison plots.

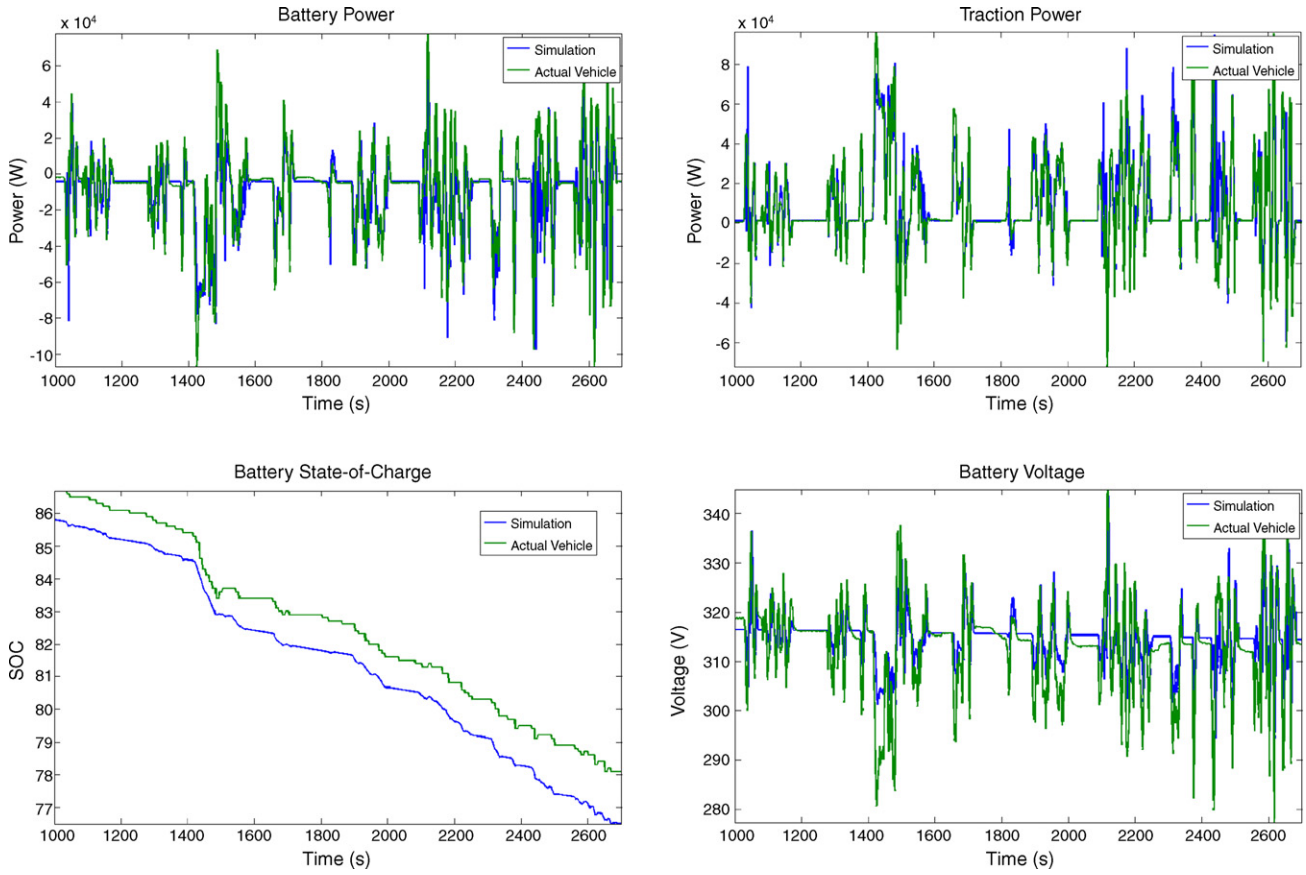

Fig. 13. Drive Cycle 2—comparison plots.

simultaneous plots of important quantities for both the simulation and physical data acquired from the vehicle during the same drive cycle. Using the speed vs. time data from the GPS, an LFM input drive cycle was generated and the simulation was executed using the same initial conditions. These initial conditions include: state-of-charge, average power requirement, and accessory loads. The output of the simulation was used to generate the comparison plots shown in Figs. 12 and 13, each for a different test drive cycle ("Drive Cycle 1" and "Drive Cycle 2", respectively).

Fig. 12 compares traction power (motor electric power usage), battery power, battery state-of-charge, and fuel cell power request. Notice that fuel cell power *request* is used and not fuel cell power. This is because for both the real vehicle and the simulation, the actual fuel cell power output becomes saturated at the maximum power output. Therefore, it is of greater interest to plot power request from the hybrid control algorithm, rather than actual power output. The purpose of Fig. 12 is to illustrate the degree of alignment between the simulator prediction and actual vehicle performance.

Fig. 13 shows the same comparison as in Fig. 12 except, instead of fuel cell power request, battery voltage is plotted. During this particular test drive cycle, the battery state-of-charge never dropped below the threshold value to activate the fuel cell system. Therefore, fuel cell power request was always zero.

To reinforce the confidence in the simulation outputs, Table 5 shows a comparison between the vehicle and simulator for several global drive cycle quantities. This comparison is more quantitative than the plots shown in Figs. 12 and 13, and helps to demonstrate the robustness of the simulation by calculating these quantities for several different drive cycles.

The majority of the error values shown in Table 5 are below 15%. However, a few error values, typically associated with state-of-charge change, are higher. For example, the error of state-of-charge change for Drive Cycle 5 is 22.6%. This percentage error is high because the error is normalized by the actual values for state-of-charge change which are usually low for both the actual vehicle and the simulation prediction. The other high error values are the result of several factors including inaccuracies in data acquisition and lack of well-quantified data for various subsystems on the real vehicle (for example, battery internal resistance). In general, a tool such as this will be used more to design a new vehicle powertrain rather than improve upon an existing design. In this respect, the acquisition of more accurate modeling parameters can vastly reduce these errors.

It is of interest to determine the source of the errors presented in Table 5. The source of error for total energy use is the most difficult to determine precisely since total energy used depends on several downstream calculations. A logical approach is to examine the

Table 4
Vehicle data acquisition parameters

Vehicle	Fuel cell system	GPS tracking
Battery SOC	Air flow rate	Latitude
Battery current	Air temperature	Longitude
Battery voltage	Air humidity	Ground speed
Battery temperature	Air pressure	MSL altitude
Traction current	Coolant flow rate	Bearing
Traction power	Coolant temperature	Position error
Motor temperature	Coolant pressure	Time of day
Motor speed	Hydrogen flow rate	
	Hydrogen temperature	
	Hydrogen humidity	
	Hydrogen pressure	
	Stack voltage	

Table 5
Quantitative comparison

Parameter	Drive Cycle 1 (8,700 s)			Drive Cycle 2 (11,330 s)		
	Vehicle	Simulation	Error (%)	Vehicle	Simulation	Error (%)
Energy use (Wh)	38,293	37,561	1.9	52,466	49,617	5.4
Energy recovered (Wh)	6,619	6,551	1.0	8,110	8,463	4.4
State-of-charge change (%)	−29.2	−30.7	5.2	−30.4	−35.0	15.2
Fuel cell energy output (Wh)	13,137	12,632	3.8	22,417	21,048	6.1
Parameter	Drive Cycle 3 (11,190 s)			Drive Cycle 4 (13,342 s)		
	Vehicle	Simulation	Error (%)	Vehicle	Simulation	Error (%)
Energy use (Wh)	49,936	49,664	0.5	53,565	58,439	9.1
Energy recovered (Wh)	7,444	6,700	9.9	8,834	7,657	13.3
State-of-charge change (%)	−31.8	−37.2	17.0	−38.0	−43.4	14.3
Fuel cell energy output (Wh)	19,092	20,473	7.2	18,749	21,543	14.9
Parameter	Drive Cycle 5 (6,000 s)			Drive Cycle 6 (9,340 s)		
	Vehicle	Simulation	Error (%)	Vehicle	Simulation	Error (%)
Energy use (Wh)	33,403	29,570	11.4	51,715	47,120	8.8
Energy recovered (Wh)	4,928	5,243	6.4	6,854	6,838	0.2
State-of-charge change (%)	−5.3	−6.5	22.6	−10.0	−11.5	15.0
Fuel cell energy output (Wh)	20,118	18,979	5.6	30,440	30,142	0.9

error contributions from each of these downstream systems. One important contributor would be energy recovered, which is a direct function of the motor power calculation. Error from this subsystem is most likely caused by inaccurate motor efficiency modeling. Again, more reliable and accurate performance data would improve this situation. The error in state-of-charge change is most likely incurred during vehicle SOC calculation since SOC is calculated by simply integrating the battery current. Consequently, the error in fuel cell energy output is caused by the error in SOC calculation because the fuel cell power request is a function of battery SOC.

4. Summary

A vehicle powertrain simulator called LFM was developed to incorporate a flexible and easily modifiable structure such that changes to the fundamental vehicle design could be rapidly implemented without compromising model accuracy or simulation computational performance. The MATLAB/Simulink environment was used to define and link various subsystems of the hybrid vehicle powertrain. A system validation study conducted by comparing time-dependent data from the simulation with physical data taken from a fuel cell hybrid transit bus demonstrates that the simulator is able to reliably predict the vehicle performance. Additional comparisons of global quantities computed from the overall drive cycle show that the LFM simulator yields reliable and robust performance results and should prove useful as a design tool for various hybrid platforms.

Acknowledgments

Funding for this work was provided by the Federal Transit Administration and the Delaware Department of Natural Resources and Environmental Control.

References

[1] H.K. Geyer, R.K. Ahluwalia, GCtool for Fuel Cell Systems Design and Analysis: User Documentation, Argonne National Laboratory Report ANL-98/8, 1998.
[2] F. Milano, IEEE Trans. Power Syst. 20 (3) (2005) 1199–1206.
[3] K.L. Butler, M. Ehsani, P. Kamath, IEEE Trans. Vehicul. Technol. 48 (6) (1999) 1770–1777.
[4] T. Markel, A. Brooker, T. Hendricks, V. Johnson, K. Kelly, B. Kramer, M. O'Keefe, S. Sprik, K. Wipke, J. Power Sources 110 (2002) 255–266.
[5] X. He, J. Hodgson, IEEE Trans. Intell. Transport. Syst. 3 (4) (2002) 235–243.
[6] X. He, J. Hodgson, IEEE Trans. Intell. Transport. Syst. 3 (4) (2002) 244–251.
[7] D.W. Gao, C. Mi, A. Emadi, Proceedings of the IEEE 95, vol. 4, 2007, pp. 729–745.
[8] C. Moler, Numerical Computing with MATLAB, Electronic edition, The MathWorks, Inc., Natick, MA, 2004. http://www.mathworks.com/moler accessed: 28 November 2007.
[9] The MathWorks, Inc., Simulink Technical Documentation, http://www.mathworks.com/products/simulink/technicalliterature.html accessed: 28 November 2007.
[10] V.H. Johnson, J. Power Sources 110 (2002) 321–329.
[11] F.U. Syed, M.L. Kuang, J. Czubay, H. Ying, IEEE Trans. Vehicul. Technol. 55 (6) (2006) 1731–1747.