The Pennsylvania State University

The Graduate School

Department of Mechanical and Nuclear Engineering

SCALING OF HYBRID-ELECTRIC VEHICLE POWERTRAIN COMPONENTS FOR HARDWARE-IN-THE-LOOP SIMULATION

A Thesis in

Mechanical Engineering

by

Michael D. Petersheim

© 2008 Michael D. Petersheim

Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

May 2008

The thesis of Michael D. Petersheim was reviewed and approved* by the following:

Sean N. Brennan Assistant Professor of Mechanical Engineering Thesis Advisor

Chris D. Rahn Professor of Mechanical Engineering

Karen A. Thole Professor of Mechanical Engineering Head of the Department of Mechanical Engineering

*Signatures are on file in the Graduate School

ABSTRACT

Hardware in the loop (HIL) simulation enables experimental study of prototype hardware systems or control algorithms via real-time interaction between physical hardware and virtual simulations. As a result, this method is a particularly valuable tool for hybrid vehicle powertrain analysis. In the case where novel or prototype hardware is being examined, it is often necessary to scale the signals in and out of the prototype system in order to represent production-sized components. This scaling process is usually done in an ad-hoc manner. In this work, a formal method is presented that derives appropriate input/output signal conditioning to correctly scale electric vehicle components, particularly the following subsystems: electric motor, battery pack, ultracapacitor pack, engine, and fuel cell.

TABLE OF CONTENTS

LIST OF TABLES	:
	IX
ACKNOWLEDGEMENTS	X
Chapter 1 Introduction	1
1.1 Motivation	2
1.1.1 Hybrid Electric Vehicles	2
1.1.2 Hardware-in-the-Loop	2
1.1.3 Scale Models	3
1.1.4 Dimensional Analysis	5
1.1.5 The Penn State Hardware-in-the-Loop System	6
1.1.6 Thesis Outline	/
Chapter 2 The Dimensionless Variable Method Applied to a Motor Model	9
2.1 Case 1. Electric Motor Model from the PSAT Library	9
2.2 The Dimensionless Variable Method	11
2.3 Simulation Results	15
Chapter 3 Battery Pack Model Scaling Factors	21
3.1 Case 2: Dynamic Battery Pack Model from the PSAT Library	21
3.2 Simulation Results	27
Chapter 4 Ultracapacitor Pack, Engine, and Fuel Cell Scaling	33
	22
4.1 Case 3: Ultracapacitor Pack Model from the PSAT Library	33 20
4.2 Case 4: Engine Model from the PSAT Library	38
4.5 Engine Scaling Simulation Results	42
4.4 Case 5. Fuel Cell Model IIolii ule PSAT Liolary	40
Chapter 5 Battery Scaling Experiment	52
5.1 Setup of the Experiment	53
5.1.1 Batteries	53
5.1.2 Vehicle Powertrain Models	60
5.1.3 Scaling Factors	61
5.1.4 Control Equipment	62
5.1.5 Complete Experimental System	64
5.2 Experimental Results	65
 Chapter 4 Ultracapacitor Pack, Engine, and Fuel Cell Scaling	33 33 38 42 46 52 53 60 61 62 64 65

Chapter 6 Conclusions and Future Work	70
6.1 Conditions for Use of Scaling Factors	70
6.2 Conclusion	70
6.3 Future Work	71
Bibliography	72
Appendix A Modification of PSAT Models for HIL	75
A.1 Use of PSAT models for HIL	75
A.1.1 Preparation of the PSAT model to run independently of PSAT	75
A.1.1.1 PSAT Version 5.2.	75
A.1.1.2 PSAT Version 6.1	77
A.1.2 Preparation of the Simulink model to run on an embedded system	
– xPC Target required	80
A.1.3 Addition of inputs/outputs to a Simulink model – TeraSoft library	
required for Advantech	81
A.1.4 Control of the ABC-150 from an embedded computer – requires additional PC running Windows, hereafter known as "Link,"	
equipped with serial port, with Borland C, LabView or Visual Basic	
 – requires AeroVironment serial port driver: PPSD.EXE 	82
A.2 Setup of Advantech UNO-3072 computers to boot xPC Target from	
CompactFlash	83
A.2.1 Creation of and bootup from a DOS floppy	83
A.2.2 Configuration of CompactFlash with xPC Target Boot Kernel	
(After booting with the DOS floppy)	85
A.2.3 Operation of the embedded computer, hereafter known as Target –	
requires a PC with xPC Target, hereafter known as Host	86

v

LIST OF FIGURES

Figure 1-1: Networked hybrid electric vehicle powertrain HIL system under development at the Pennsylvania State University	.6
Figure 1-2: General representation of a powertrain HIL test	.7
Figure 2-1: Comparison of sample PM and induction motor torque-speed curves	.10
Figure 2-2: Motor model inputs and outputs.	.10
Figure 2-3: Dimensionless torque-speed curve comparison	.15
Figure 2-4: Dimensionless torque-speed curves of fifteen AC induction and permanent magnet DC motors from the PSAT motor model library.	.16
Figure 2-5: Torque command input to motor models during first 100 s of simulation.	.17
Figure 2-6 : Voltage input to motor models during first 100 s of simulation	.17
Figure 2-7: Shaft speed input to motor models during first 100 s of simulation	.18
Figure 2-8 : Torque trace of motor models during first 100 s of simulation	.18
Figure 2-9 : Difference between torque traces of original and scaled motors	. 19
Figure 2-10 : Current trace of motor models during first 100 s of simulation	. 19
Figure 2-11: Difference between current traces of original and scaled motors	.20
Figure 3-1 : Time constant of two typical cells.	.22
Figure 3-2 : Dynamic resistance of two typical cells	.22
Figure 3-3 : Open circuit voltage of two typical cells.	.23
Figure 3-4 : Battery pack model signals.	.23
Figure 3-5: Scaled current loads on two NiMH battery packs in PSAT simulation of Honda Insight on US06 driving cycle.	.28
Figure 3-6 : Voltage trace comparison of PSAT simulation of Honda Insight on the first 30 s of the US06 driving cycle with two NiMH battery packs	.28
Figure 3-7 : Difference between voltages of prototype and scaled NiMH battery packs.	.29

Figure 5-7: Experiment configuration.	.61
Figure 5-8 : Experiment configuration.	.62
Figure 5-9: Aerovironment ABC-150 Power Processing System.	.63
Figure 5-10: Complete experimental system.	.64
Figure 5-11: Current load applied to batteries in the PAR simulation.	.65
Figure 5-12: Voltage response of batteries, scaled to pack size, in the PAR simulation.	.66
Figure 5-13: Difference between voltages of batteries in the PAR simulation	.66
Figure 5-14: Current load applied to batteries in the EV simulation	.67
Figure 5-15: Voltage response of batteries, scaled to full pack size, in the EV simulation.	.68
Figure 5-16 : Difference between voltage responses in the EV simulation	.68

viii

LIST OF TABLES

Table 2-1: Parameters Relevant to Motor Scaling	12
Table 2-2: Motor Scaling Pi-Groups	13
Table 2-3: Motor Scaling Equivalency	15
Table 3-1: Parameters Relevant to Battery Pack Scaling	24
Table 3-2: Battery Pack Scaling Pi-Groups	
Table 3-3: Battery Pack Scaling Equivalency	
Table 4-1: Parameters Relevant to Ultracapacitor Pack Scaling	
Table 4-2: Ultracapacitor Pack Scaling Pi-Groups	
Table 4-3: Ultracapacitor Pack Scaling Equivalency	
Table 4-4: Parameters Relevant to Engine Scaling	40
Table 4-5: Engine Scaling Pi-Groups	40
Table 4-6: Engine Scaling Equivalency	41
Table 4-7: Parameters Relevant to Fuel Cell Scaling	
Table 4-8: Fuel Cell Scaling Pi-Groups	49
Table 4-9: Fuel Cell Scaling Equivalency	
Table 5-1: Battery Testing Discharge/Charge Profile	55
Table 5-2: Battery characteristic estimation statistics	57
Table 5-3: Characteristic battery pi-parameters	
Table 5-4: Battery characteristics	59
Table 5-5: HIL scaling factors	

ACKNOWLEDGEMENTS

This work was supported in part by the U.S. Department of Energy under the Graduate Automotive Technology Education program.

Chapter 1

Introduction

Hardware in the loop (HIL) simulation enables experimental study of prototype hardware systems or control algorithms via real-time interaction between physical hardware and virtual simulations. As a result, this method is a particularly valuable tool for hybrid vehicle powertrain analysis. In the case where novel or prototype hardware is being examined, it is often necessary to scale the signals in and out of the prototype system in order to represent production-sized components. This scaling process is usually done in an ad-hoc manner. In this work, a formal method is presented that derives appropriate input/output signal conditioning to correctly scale electric vehicle components, particularly the following subsystems: electric motor, battery pack, ultracapacitor, engine, and fuel cell.

This introduction explains the motivation for the work, giving an overview of hybrid electric vehicles, hardware-in-the-loop, scale models, and dimensional analysis. A summary of the Penn State hardware-in-the-loop system follows, along with a thesis outline.

1.1 Motivation

1.1.1 Hybrid Electric Vehicles

The Toyota Prius, the first mass-produced hybrid electric vehicle (HEV), went on sale in Japan in December 1997, and was a surprising success [1]. Since then, the number of HEVs on the market increases annually. Selling points of HEVs include their reduced fuel consumption and reduced exhaust emissions. The latter point has recently grown in importance with increased concern about global warming.

The Pennsylvania State University is involved with the development of HEV technology in several ways: Student groups have constructed several HEVs for national competitions. Research involving components of HEV powertrains is performed by various faculty and research staff, some of which are members of the Advanced Energy Storage Center. Penn State hosts a Graduate Automotive Technology Education (GATE) center sponsored by the U.S. Department of Energy. The center offers several courses annually, including HEV Lab.

1.1.2 Hardware-in-the-Loop

Hardware-in-the-loop (HIL) simulation enables the interaction of virtual computer-based simulation models of a system or subsystem with actual components of the system in real-time. Because this permits the inclusion of components for which accurate computer models do not yet exist or for which intense computing resources are required, this method is finding increasing use in nearly every discipline. HIL systems have been employed for decades in numerous disciplines to evaluate novel hardware or software designs including earth-moving vehicles, ocean-going vessels, suspension systems, earthquake-proof buildings, powertrain controllers, unmanned underwater vehicles, automotive safety systems, machine tools, sonar systems, and aircraft [2, 3, 4, 5, 6]. If one assumes that the human is a subsystem central to vehicle control, then all driving simulators can also be classified as a type of HIL system.

HIL testing is increasingly useful in applications involving hybrid electric vehicle powertrains [5, 7, 8, 9, 10, 11]. The use of HIL can replace, to significant extent, the construction of expensive prototypes to test drivetrain systems. In many cases, the prototype hardware is a reduced-scale surrogate for actual size hardware, built to evaluate performance and feasibility rather than actually power a commercial vehicle. Examples include prototype fuel cells, engines, batteries, and electric motors [7, 8, 9, 10, 11, 12]. In nearly all cases, construction of a full-sized prototype is onerous and/or unnecessarily expensive.

1.1.3 Scale Models

Closed-loop HIL testing of benchtop prototypes are especially useful to understand the interaction between the highly coupled subsystems typically found in an electric or hybrid-electric vehicle. In this way, one prototype cell of a fuel cell stack may be tested in a HIL environment to estimate the performance of an entire pack of cells in a production vehicle. Or a short string of a battery pack may be used to infer the performance of a large string of batteries, etc. Scale models have been used to infer the behavior of a full-size prototype since William Froude tested ship models in water tanks [13]. Wilbur and Orville Wright built the first wind tunnel to try various configurations of scale aircraft wings, resulting in the first successful flying machine. Scale models were employed in the design of lunar rovers in the 1960s [17]. Scale models of road vehicles have been in use since 1934 [19]. More recently, the <u>Pennsylvania State University Rolling Roadway Simulator (PURRS)</u> operates a scale vehicle on a treadmill for vehicle rollover testing [19].

A key problem with comparisons is that scaling effects arise when hardware of one size is simulated by hardware of another size [12]. Doubling the number of cells in series within a fuel cell stack does not double the available electrical current. And when thermal effects are included, a production-sized pack of cells may overheat under typical environmental and packaging conditions whereas a single benchtop cell would operate without incident.

Furthermore, it is often not the intent to "scale" or operate the prototype system such that it tracks the input/output behavior of an existing system. Nor is it desirable in general to design high-gain feedback controllers that force the prototype to track a "reference" performance of existing hardware. Both ad-hoc methods negate the very intent of most prototype systems, that is, to observe differences in behavior relative to existing systems.

1.1.4 Dimensional Analysis

What is needed therefore is an understanding of how to compare dissimilarly sized components using scaling factors that are physically based, e.g. tied to experimentally measurable variations in key parameters rather than numerical methods. This understanding should be generalized and validated by comparing dissimilarly scaled systems that share common dynamic limitations. If, under the chosen scaling factors, we observe that dissimilarly sized components map to the same general model behavior in a dimension-free setting, then we have confidence that the same scaling methods might appropriately map a bench-scale prototype to the expected production-level component. The goal of this work is to apply the use of dimensionless variables, as defined by the Pi Theorem [14, 15, 16], to hybrid electric vehicle powertrain components for the purpose of taking into account the relevant scaling effects. Similar work has been conducted before [12, 17, 18, 20, 21], but not on the components mentioned herein.

Dimensional analysis has its roots in work by Euler, Newton, Fourier, Maxwell, and Rayleigh [17]. The method of dimensional scaling was formalized as the Pi Theorem by Buckingham [15]. Szirtes provided an explanation of a "painless" method for obtaining dimensionless parameters using the dimensional set matrix [14]. Brennan further developed the concepts of dimensional analysis by its application to sensitivity analysis [17]. Kittirungsi *et al* enhanced the effectiveness of the method by coupling it with activity based model reduction [12].

1.1.5 The Penn State Hardware-in-the-Loop System

The motivation for the present study is the development of a networked hybrid electric vehicle powertrain hardware-in-the-loop (HIL) system underway at the Pennsylvania State University (see Figure 1-1). In this project, HIL equipment in various laboratory settings across campus is linked via Ethernet. These include an electrical power processing machine, engine dynamometers, chassis dynamometers, and ultracapacitor and fuel cell test benches. The HIL system is used for graduate course labs, student vehicle competitions, and industry-sponsored projects. The eventual goal is to allow collaborative testing both between research laboratories at Penn State as well as off-campus industry and government laboratories.





As a basis for incorporating individual powertrain components into HIL

simulations, powertrain models from the well-known Powertrain Systems Analysis

Toolkit (PSAT) from Argonne National Laboratory [22] are utilized within a

MATLAB/Simulink/xPC TargetTM environment [23]. One or more components of the powertrain are replaced by a set of output(s) and input(s) from/to equipment which controls the individual hardware component(s) (Figure 1-2). Typically the hardware is not full-size, in which case input and output signal scaling factors must be implemented in the software environment to compare appropriately to full-sized vehicle components. Determination of these scaling factors, shown as triangles in Figure 1-2, is the focus of this work.



Figure 1-2: General representation of a powertrain HIL test.

1.1.6 Thesis Outline

The remainder of this work is organized as follows: In Chapter 2, a procedure is developed via the dimensionless variable method to derive input/output scaling factors, and is applied to a steady state motor model in the context of a vehicle powertrain simulation. In Chapter 3, the same method is applied to a dynamic battery model, also in the context of simulation. In Chapter 4, scaling factors are derived for additional powertrain components including ultracapacitors, engines, and fuel cells. In Chapter 5, a

presentation is made of the setup and results of an experiment with actual hardware.

Chapter 6 summarizes the main results and points the way for future work.

Chapter 2

The Dimensionless Variable Method Applied to a Motor Model

2.1 Case 1: Electric Motor Model from the PSAT Library

The proposed method for obtaining scaling factors and determining dynamic similarity of systems involves the formation of an equivalent system representation using dimensionless variables [17, 24]. This method will be illustrated first with an electric motor, and later with a battery.

To investigate scaling effects related to electric vehicle drive systems, the Powertrain Systems Analysis Toolkit (PSAT) [22] electric motor model library was used which includes mainly AC induction motors and large permanent magnet (PM) DC motors. In the PSAT software, one motor can be substituted for another during software prototyping of new vehicle design, hence some similarity in performance across the many motor models in this software is expected.

To investigate potential similarity of the motors, the steady-state torque-speed curves of each motor were plotted. Steady-state was chosen because transient effects of each motor are minor compared to their steady-state performance during typical driving cycles. A sample torque-speed curve comparison is shown in Figure 2-1. One can observe similarity in the curve shapes, yet little match between torque speed values themselves.



Figure 2-1: Comparison of sample PM and induction motor torque-speed curves.

The PSAT motor model takes as inputs: DC voltage V, shaft speed Ω , and a torque command signal θ with range [-1, 1], defined as desired torque T_{ref} divided by maximum torque T_{max} . Outputs are current I and torque T. Inputs and outputs are shown in Figure 2-2.



Figure 2-2: Motor model inputs and outputs.

The rest of the PSAT motor model follows: A derived quantity is power *P*.

Steady-state parameters are: maximum current I_{max} , maximum torque T_{max} , and maximum

power P_{max} . Since the application of the motor is for a traction drive, the rotational inertia of the motor is negligible in comparison to the inertia of the vehicle. Thus the dynamics of the motor are neglected and only steady-state input-output relationships are considered. The simplest relationships also neglect efficiency, as shown in Eq. 2.1.

$$T = \theta \cdot I_{max} \cdot V/\Omega$$

$$P = T \cdot \Omega$$

$$I = P/V$$
2.1

The parameters T and P are saturated by T_{max} and P_{max} , as in Eq. 2.2.

$$-T_{max} \le T \le T_{max} -P_{max} \le P \le P_{max}$$
2.2

2.2 The Dimensionless Variable Method

To apply the dimensionless variable method, let *N* be the number of system parameters, and let *M* be the number of physical dimensions required to describe all the *N* parameters in the governing equation. The motor system has N = 3 parameters, I_{max} , T_{max} , P_{max} , composed of M = 4 dimensions, length, mass, time, and current. In the SI unit system, the unit basis vector is $u = [m \text{ kg s A}]^T$. In addition to the parameters, there are signals *S*, which will also be rescaled, for example, V, Ω , θ , *I*, *T*, which represent inputs and outputs. The signals and parameters are shown with their dimensions in Table 2-1. In the PSAT environment, the torque command signal θ is dimensionless, and is thus excluded from dimensional scaling.

Variable	Symbol	Dimension
current	Ι	А
torque	Т	m ² ·kg·s ⁻²
voltage	V	$m^2 \cdot kg \cdot s^{-3} \cdot A^{-1}$
rotational speed	arOmega	s ⁻¹
maximum current	I _{max}	А
maximum power	P_{max}	$m^2 \cdot kg \cdot s^{-3}$
maximum torque	T_{max}	$m^2 \cdot kg \cdot s^{-2}$

Table 2-1: Parameters Relevant to Motor Scaling

The number of fundamental dimensions is four, but the $[m^2]$ dimension and the [kg] dimension always appear together, so the two are combined into a new composite dimension, leaving a total of M = 3 dimensions.

To transform a nondimensional representation to a dimensional (classic) representation and back again requires rescaling with respect to M independently dimensioned parameters or signals, also known as repeating parameters. These may be arbitrarily chosen, but they must among themselves contain all of the dimensions of the system. For the present example, I_{max} , T_{max} , and P_{max} , being the only parameters, must be chosen as repeating parameters. The repeating parameters, signals to be rescaled, dimensions, and dimensionless groups, also known as pi-groups, may be represented in matrix form, as in Eq. 2.3, where A_D is square and full rank. The number of pi-groups is Q = N + S - M. In this case, Q = 4.

other parametersrepeating parametersdimensions
$$B_D$$
 A_D π - groupsI C_S

With the problem thus formulated, the only unknown matrix, C_S , is determined according to Eq. 2.4. Details can be found in [14].

$$C_s = -\left(A_D^{-1} \cdot B_D\right)^T$$
 2.4

The number of repeating parameters is therefore also three, so the last three parameters are selected as the repeating parameters. The completed dimensional set matrix is given in Eq. **2.5**.

	Ι	Т	V	${\it \Omega}$	I_{max}	P_{max}	T_{max}
$m^2 \cdot kg$	0	1	1	0	0	1	1
S	0	-2	-3	-1	0	-3	-2
А	1	0	-1	0	1	0	0
$\pi_{1,mot}$	1	0	0	0	-1	0	0
$\pi_{2,mot}$	0	1	0	0	0	0	-1
$\pi_{3,mot}$	0	0	1	0	1	-1	0
$\pi_{\scriptscriptstyle 4,mot}$	0	0	0	1	0	-1	1

The resulting pi-groups, with the addition of the torque command signal θ , are

given in Table 2-2.

Table 2-2 :	Motor	Scaling	Pi-Groups

Dimensionless Variable	Variable Grouping
$\pi_{1,mot}$	$I \cdot I_{max}^{-1}$
$\pi_{2,mot}$	$T \cdot T_{max}^{-1}$
$\pi_{3,mot}$	$V \cdot I_{max} \cdot P_{max}^{-1}$
$\pi_{4,mot}$	$\Omega \cdot P_{max}^{-1} \cdot T_{max}$
$\pi_{5,mot}$	θ

The dimensionless model representation of Eq. 2.1 is given in Eq. 2.6.

$$\frac{T}{T_{max}} = \theta \cdot \frac{V \cdot I_{max}}{P_{max}} \cdot \frac{P_{max}}{\Omega \cdot T_{max}}$$

$$\frac{I}{I_{max}} = \frac{T}{T_{max}} \cdot \frac{\Omega \cdot T_{max}}{P_{max}} \cdot \frac{P_{max}}{V \cdot I_{max}}$$
2.6

In pi-variable form, the above becomes Eq. 2.7.

$$\pi_{2,mot} = \pi_{5,mot} \pi_{3,mot} / \pi_{4,mot}$$

$$\pi_{1,mot} = \pi_{2,mot} \pi_{4,mot} / \pi_{3,mot}$$

2.7

Two systems *a* and *b* are dynamically similar when their system pi-groups have the same values, respectively, i.e. $\pi_{1,a} = \pi_{1,b}$, $\pi_{2,a} = \pi_{2,b}$, etc [17]. Thus, an input $V_P(t)$ of a prototype motor model *P* may be transformed into the corresponding input $V_H(t)$ of a scaled HIL motor *H* by using an input scaling factor. Alternately, the output $T_H(t)$ of the scaled HIL motor *H* may be retransformed into the output $T_P(t)$ of the prototype motor model *P* with an output scaling factor. The scaling factors are formed by equating the relevant pi-groups and solving for the variable in question, as in Eq. 2.8. For example, to scale prototype voltage, V_P , to hardware voltage, V_H :

$$\pi_{3,mot,H} = \pi_{3,mot,P} \Longrightarrow \frac{V_H I_{max,H}}{P_{max,H}} = \frac{V_P I_{max,P}}{P_{max,P}}$$

$$\Longrightarrow V_H = V_P \frac{I_{max,P} P_{max,H}}{P_{max,P} I_{max,H}}$$
2.8

Applying this process to each variable, the resulting input-output scaling equivalency is shown in Table 2-3.

Lable 2-5. Wold Dealing Equivalence	Ι	`able	2-3	: Mc	otor	Scal	ling	Eq	uiva	lenc	y
-------------------------------------	---	-------	-----	------	------	------	------	----	------	------	---

HIL Component	Prototype Model
V_H	$V_{P} \frac{I_{max,P} \cdot P_{max,H}}{P_{max,P} \cdot I_{max,H}}$
$arOmega_H$	$arOmega_{P}rac{T_{max,P}\cdot P_{max,H}}{P_{max,P}\cdot T_{max,H}}$
$ heta_H$	$ heta_P$
$I_H \frac{I_{max,P}}{I_{max,H}}$	I_P
$T_H \; rac{T_{max,P}}{T_{max,H}}$	T_P

2.3 Simulation Results

The use of dimensionless variables to plot system characteristics is illustrated by a second look at the two motors compared earlier (Figure 2-1) in the dimensionless domain. This plot is shown in Figure 2-3.



Figure 2-3: Dimensionless torque-speed curve comparison.

The use of dimensionless variables to plot torque vs. speed for PM and AC induction motors results in visibly matching characteristic curves. For completeness, a dimensionless comparison was made of all fifteen PM and AC induction motors listed in the PSAT model library, with the results shown in Figure 2-4. Again, agreement is obvious.



Figure **2-4**: Dimensionless torque-speed curves of fifteen AC induction and permanent magnet DC motors from the PSAT motor model library.

Using this scaling method, a simulation of a Toyota Prius hybrid electric vehicle on the US06 driving cycle [29] was performed using PSAT. Details of how the Prius model was set up to run independently of PSAT are given in the Appendix, section A.1.1. Inputs to the 30 kW PM Prius motor were scaled to match a level equivalent to a 35 kW induction motor also found in the PSAT motor model library. Torque command input to both motor models is shown in Figure 2-5.



Figure 2-5: Torque command input to motor models during first 100 s of simulation.

Voltage input to both motor models is shown in Figure 2-6.



Figure 2-6: Voltage input to motor models during first 100 s of simulation.

Shaft speed input to both motors is shown in Figure 2-7.



Figure 2-7: Shaft speed input to motor models during first 100 s of simulation.

Resulting torque output traces from both motor models are shown in Figure 2-8.



Figure 2-8: Torque trace of motor models during first 100 s of simulation.

The difference between torque traces is plotted in Figure 2-9. The root mean square error for the cycle is 17.4 N-m.



Figure 2-9: Difference between torque traces of original and scaled motors.

Resulting current output traces from both motor models is shown in Figure 2-10.



Figure 2-10: Current trace of motor models during first 100 s of simulation.

The difference between current traces is plotted in Figure 2-11. The root mean square error for the cycle is 8.459 A.



Figure 2-11: Difference between current traces of original and scaled motors.

The torque trace of the prototype Prius motor in Figure 2-8 was closely predicted by the torque trace of the scaled HIL motor. In addition, the current trace of the prototype Prius motor in Figure 2-10 was closely predicted by the current trace of the scaled HIL motor. It is assumed that if motor efficiency were included in the steady-state equations, then an even closer match of the traces would result. Since, however, HIL is often performed in order to predict the efficiency of the prototype, such efficiency is generally unknown and must necessarily be left out of the scaling.

Chapter 3

Battery Pack Model Scaling Factors

3.1 Case 2: Dynamic Battery Pack Model from the PSAT Library

In the case of the battery pack as an electric vehicle component, the dynamics are relatively slow and can be neglected only by accepting significant error in the voltage prediction. The PSAT battery pack model library contains both steady-state and first-order battery models, but to maintain accuracy, only the more accurate first-order models are considered hereafter. The dynamic system equations according to the PSAT libraries are given in Eq. **3.1**, where current *I* is the input, voltage *V* is the output, *SOC* is the state of charge, V_c is a dynamic voltage, n_{cells} is the number of cells in the pack, τ is a first-order time constant, R_c is a dynamic resistance, R_{int} is the internal resistance of a cell, and V_{OC} is the steady state open circuit voltage. The parameters τ , R_c , and V_{OC} are a function of *SOC*.

$$\begin{split} \dot{SOC} &= -\frac{1}{Q} \cdot I \\ \dot{V}_c &= -\frac{1}{\tau(SOC)} \cdot V_c + \frac{R_c(SOC) \cdot n_{cells}}{\tau(SOC)} \cdot I \\ V &= V_c + n_{cells} \cdot (V_{OC}(SOC) - I \cdot R_{int}) \end{split}$$

$$\begin{aligned} \mathbf{3.1} \\ \end{split}$$

The time constant τ of two typical batteries from the PSAT library, a 6 amp-hour nickel metal hydride cell and a 14 amp-hour Li-Ion cell, is shown as a function of *SOC* in Figure 3-1.



Figure **3-1**: Time constant of two typical cells.

The dynamic resistance R_c of the same two cells is shown in Figure 3-2.



The open circuit voltage V_{OC} of the same two cells is shown in Figure 3-3.



Figure 3-3: Open circuit voltage of two typical cells.

An input-output diagram of the battery pack model is shown in Figure 3-4.



Figure **3-4**: Battery pack model signals.

Again applying the dimensionless variable method, the battery pack system has N = 5 parameters, $Q, R_c, \tau, R_{int}, V_{OC}$, composed of M = 4 dimensions, length, mass, time, and current. The number of fundamental dimensions is four, but as with the motor example, the [m²] dimension and the [kg] dimension always appear together, so the two

are combined into a new composite dimension, leaving a total of M = 3 dimensions. The number of repeating parameters is therefore also three. Whereas with the motor example, all parameters became repeating parameters, in this example a choice is necessary. It is advantageous if the repeating parameters each have a single dimension, are constant, are easily measured, etc. The only constant, R_{int} , shall therefore be selected. Q is a constant in PSAT, but is typically specified by manufacturers as a function of the magnitude of *I*, and is constant only when specified for a particular magnitude of I. The other parameters are a function of SOC. Repeating parameters need not be constants, but for convenience a nominal voltage V_{nom} is defined, as in Eq. 3.2. The time constant τ shall be selected for its single dimension. With the addition of V_{nom} , there are now a total of N = 6 parameters.

$$V_{nom} = V_{OC} \left(SOC = 0.5 \right)$$
 3.2

To the parameters shall be added S = 4 signals and state, t, V, V_c , I. The signals, state, and parameters are shown with their dimensions in Table 3-1. The number of cells n_{cells} and the state of charge SOC are already dimensionless, and are thus excluded.

Variable	Symbol	Dimension		
capacity	Q	s·A		
dynamic resistance	$\bar{R_c}$	m ² ·kg·s ⁻³ ·A ⁻²		
voltage	V	$m^2 \cdot kg \cdot s^{-3} \cdot A^{-1}$		
dynamic voltage	V_{c}	$m^2 \cdot kg \cdot s^{-3} \cdot A^{-1}$		
open circuit voltage	V_{OC}	$m^2 \cdot kg \cdot s^{-3} \cdot A^{-1}$		
current	Ι	A		
time	t	S		
time constant	τ	S		
nominal resistance	R_{int}	m ² ·kg·s ⁻³ ·A ⁻²		
nominal voltage	V_{nom}	$m^2 \cdot kg \cdot s^{-3} \cdot A^{-1}$		

The number of pi-groups is Q = N + S - M = 7. The completed dimensional set matrix is given in Eq. 3.3.

	Q	R_{c}	V	V_{c}	V_{OC}	Ι	t	τ	R _{int}	V _{nom}	
$m^2 \cdot kg$	0	1	1	1	1	0	0	0	1	1	
S	1	-3	-3	-3	-3	0	1	1	-3	-3	
А	0	-2	-1	-1	-1	1	0	0	-2	-1	
$\pi_{1,bat}$	1	0	0	0	0	0	0	-1	1	-1	
$\pi_{2,bat}$	0	1	0	0	0	0	0	0	-1	0	3.3
$\pi_{3,bat}$	0	0	1	0	0	0	0	0	0	-1	
$\pi_{4,bat}$	0	0	0	1	0	0	0	0	0	-1	
$\pi_{\scriptscriptstyle 5,bat}$	0	0	0	0	1	0	0	0	0	-1	
$\pi_{\scriptscriptstyle 6,bat}$	0	0	0	0	0	1	0	0	1	-1	
$\pi_{7,bat}$	0	0	0	0	0	0	1	-1	0	0	

The resulting pi-groups, including the number of cells n_{cells} and the state of charge

SOC, are given in Table **3-2**.

Dimensionless Variable	Variable Grouping
$\pi_{1,bat}$	$Q \cdot R_{int} \cdot \tau^{-1} \cdot V_{nom}^{-1}$
$\pi_{2,bat}$	$R_c \cdot R_{int}^{-1}$
$\pi_{3,bat}$	$V \cdot V_{nom}^{-1}$
$\pi_{4,bat}$	$V_c \cdot V_{nom}^{-1}$
$\pi_{5,bat}$	$I \cdot R_{int} \cdot V_{nom}^{-1}$
$\pi_{6,bat}$	$V_{OC} \cdot V_{nom}^{-1}$
$\pi_{7,bat}$	$t \cdot \tau^{-1}$
$\pi_{8,bat}$	n _{cells}
$\pi_{9,bat}$	SOC

Table **3-2**: Battery Pack Scaling Pi-Groups

HIL Component	Prototype Model
t_H	$t_P\cdot rac{ au_H}{ au_P}$
I_H	$I_{P} \cdot \frac{R_{int,P} \cdot V_{nom,H}}{V_{nom,P} \cdot R_{int,H}}$
$V_H \cdot \frac{V_{nom,P} \cdot n_{cells,P}}{V_H}$	V_P

The resulting input-output scaling equivalency is shown in Table 3-3.

The equations of motion are rearranged in dimensionless form in Eq. 3.4.

$$\tau(SOC) \cdot SOC = -\frac{I \cdot R_{int}}{V_{nom}} \cdot \frac{\tau(SOC) \cdot V_{nom}}{Q \cdot R_{int}}$$

$$\frac{\tau(SOC) \cdot \dot{V_c}}{V_{nom} \cdot n_{cells}} = -\frac{V_c}{V_{nom} \cdot n_{cells}} + \frac{R_c(SOC)}{R_{int}} \cdot \frac{I \cdot R_{int}}{V_{nom}}$$

$$\frac{V}{V_{nom} \cdot n_{cells}} = \frac{V_c}{V_{nom} \cdot n_{cells}} + \frac{V_{OC}(SOC)}{V_{nom}} - \frac{I \cdot R_{int}}{V_{nom}}$$
3.4

The pi-variables are substituted in Eq. 3.5, where the derivative operator is also

dimensionless, i.e. (') = $\tau(SOC) \cdot \frac{d}{dt}$.

$$\pi'_{9,bat} = \frac{\pi_{5,bat}}{\pi_{1,bat}}$$

$$\frac{\pi'_{4,bat}}{\pi_{8,bat}} = -\frac{\pi_{4,bat}}{\pi_{8,bat}} + \pi_{2,bat}\pi_{5,bat}$$

$$3.5$$

$$\frac{\pi_{3,bat}}{\pi_{8,bat}} = \frac{\pi_{4,bat}}{\pi_{8,bat}} + \pi_{6,bat} - \pi_{5,bat}$$

Again, for dynamic similarity of two systems, the system pi-groups need to have identical values [17]. By definition, any ratio of dimensionless variables can also be
defined as a pi-group, such as the ratios $\pi_{3,bat}/\pi_{8,bat}$ and $\pi_{4,bat}/\pi_{8,bat}$. These new pi-groups are equivalent between two systems by Eq. **3.5** as long as $\pi_{1,bat}$, $\pi_{2,bat}$, $\pi_{5,bat}$, $\pi_{6,bat}$, and $\pi_{7,bat}$ are equivalent. By use of an input scaling factor, $\pi_{5,bat}$ has been set equivalent in Table **3-3**, so there remain four requirements, as shown in Eq. **3.6**.

$$\pi_{1,bat,H} = \pi_{1,bat,P} \pi_{2,bat,H} = \pi_{2,bat,P} \pi_{6,bat,H} = \pi_{6,bat,P} \pi_{7,bat,H} = \pi_{7,bat,P}$$
3.6

As an option with the last requirement, to avoid using different time scales during HIL, an alternative requirement could be $\tau_H(SOC) = \tau_P(SOC)$.

3.2 Simulation Results

The dimensionless variable method was applied to battery pack scaling in a PSAT simulation with a Honda Insight vehicle model on the US06 driving cycle. Scaling was applied to input *I*, output *V*, and parameters R_c and τ . The scaled current loads on the prototype 6 amp-hour, 120 cell nickel metal hydride (NiMH) battery pack and a substitute 12.5 amp-hour, 120 cell NiMH battery pack are shown in Figure 3-5.



Figure **3-5**: Scaled current loads on two NiMH battery packs in PSAT simulation of Honda Insight on US06 driving cycle.

The resulting pack voltage traces of the battery models are shown in Figure 3-6.



Figure **3-6**: Voltage trace comparison of PSAT simulation of Honda Insight on the first 30 s of the US06 driving cycle with two NiMH battery packs.

As seen in Figure **3-6**, the voltage traces of the two battery pack models are minimal. The root mean square error for the entire 600 s cycle is 1.821 V. The difference between voltage traces is plotted in Figure **3-7**.



Figure 3-7: Difference between voltages of prototype and scaled NiMH battery packs.

In an actual HIL application, however, it is only possible to scale parameters that are not inputs or outputs by the deliberate choice or construction of the HIL component. To illustrate the need for this with a second simulation, scaling was applied to only to input *I* and output *V*, leaving parameters R_c and τ at their original values. The current load on the prototype 6 amp-hour, 120 cell NiMH battery pack and a substitute 14 amp-hour, 96 cell Li-Ion battery pack are shown in Figure **3-8**.



Figure **3-8**: Scaled current loads on two NiMH battery packs in PSAT simulation of Honda Insight on US06 driving cycle.

The resulting pack voltage of the battery models is shown in Figure 3-9.



Figure **3-9**: Voltage trace comparison of PSAT simulation of Honda Insight on the first 30 s of the US06 driving cycle with prototype NiMH and scaled Li-Ion battery packs.

The difference between pack voltage traces is plotted in Figure 3-10.



Figure **3-10**: Difference between voltages of prototype NiMH and scaled Li-Ion battery packs, compared with difference between voltages of prototype and scaled NiMH battery packs.

The root mean square error for the first 30 s of the cycle is 4.876 V, compared with 0.369 V for the first 30 s of the cycle with matched battery packs. With a bias error of 3.12 V removed, root mean square error is reduced to 2.551 V. Simulation of the entire 600 s cycle could not be completed with the degree of mismatch present in this comparison. The significant variation in predicted voltage in Figure **3-10** illustrates the need for matched battery characteristics as well as dimensionally matched input and output scaling.

Chapter 4

Ultracapacitor Pack, Engine, and Fuel Cell Scaling

In this chapter, scaling for an ultracapacitor pack, internal combustion engine, and fuel cell is performed. In addition, results are given for simulation of scaling for an engine.

4.1 Case 3: Ultracapacitor Pack Model from the PSAT Library

The dynamic system equations for an ultracapacitor pack according to the PSAT libraries [22], neglecting temperature dependence, are given in Eq. 4.1, where current *I* is the input, voltage *V* is the output, n_{cells} is the number of cells in the pack, *C* is the capacitance, *R* is the internal resistance of a cell, and V_{OC} is the steady state open circuit voltage. Capacitance and internal resistance are a function of current. Voltage *V* is saturated by V_{max} .

$$\dot{V}_{OC} = -\frac{I}{C(I)}$$

$$V = \begin{cases} n_{cells} \cdot (V_{OC} - I \cdot R(I)), -V_{max} \le V \le V_{max} \\ -V_{max}, V < -V_{max} \\ V_{max}, V > V_{max} \end{cases}$$
4.1

The internal resistance R of a typical cell from the PSAT library, the Maxwell PC2500, is shown as a function of current in Figure 4-1.



Figure 4-1: Internal resistance of an ultracapacitor as a function of current.

The capacitance C of the same ultracapacitor is shown in Figure 4-2.



Figure 4-2: Capacitance of a typical ultracapacitor as a function of current.

As seen in the above two figures, the values of R and C vary only slightly with I. As such, it will be assumed that they may be regarded as constants, and the dynamic system equations may be simplified, as in Eq. 4.2.

$$\dot{V}_{OC} = -\frac{I}{C}$$

$$V = \begin{cases} n_{cells} \cdot (V_{OC} - I \cdot R), -V_{max} \le V \le V_{max} \\ -V_{max}, V < -V_{max} \\ V_{max}, V > V_{max} \end{cases}$$
4.2

An input-output diagram of the ultracapacitor pack model is shown in Figure 4-3.



Figure 4-3: Ultracapacitor pack model signals.

Applying the dimensionless variable method, the ultracapacitor pack system has N = 3 parameters: *C*, *R*, V_{max} , composed of M = 4 dimensions: length, mass, time, and current. To the parameters shall be added S = 4 signals and states, *t*, *V*, V_{OC} , *I*. The signals, states, and parameters are shown with their dimensions in Table 4-1. The number of cells n_{cells} is already dimensionless, and is thus excluded.

Variable	Symbol	Dimension
time	t	S
voltage	V	$m^2 \cdot kg \cdot s^{-3} \cdot A^{-1}$
open circuit voltage	V_{OC}	$m^2 \cdot kg \cdot s^{-3} \cdot A^{-1}$
current	Ι	А
Capacitance	С	$m^{-1} \cdot kg^{-1} \cdot s^4 \cdot A^2$
internal resistance	R	$m^2 \cdot kg \cdot s^{-3} \cdot A^{-2}$
nominal voltage	V_{max}	$m^2 \cdot kg \cdot s^{-3} \cdot A^{-1}$

Table 4-1: Parameters Relevant to Ultracapacitor Pack Scaling

The number of fundamental dimensions is four, but the $[m^2]$ dimension and the [kg] dimension always appear together, so the two are combined into a new composite dimension, leaving a total of M = 3 dimensions. The number of repeating parameters is therefore also three, so the three parameters are selected as the repeating parameters. The number of pi-groups is Q = N + S - M = 4. The completed dimensional set matrix is given in Eq. 4.3.

	t	V	V_{OC}	Ι	С	R	V _{max}
m ² kg	0	1	1	0	-1	1	1
S	1	-3	-3	0	4	-3	-3
Α	0	-1	-1	1	2	-2	-1
$\pi_{\scriptscriptstyle 1,ult}$	1	0	0	0	-1	-1	0
$\pi_{2,ult}$	0	1	0	0	0	0	-1
$\pi_{\scriptscriptstyle 3,ult}$	0	0	1	0	0	0	-1
$\pi_{\scriptscriptstyle 4, ult}$	0	0	0	1	0	1	-1

4.3

The resulting pi-groups, including the number of cells n_{cells} , are given in Table 4-

2.

Dimensionless VariableVariable Grouping $\pi_{1,ult}$ $t \cdot R^{-l} \cdot C^{-1}$ $\pi_{2,ult}$ $V \cdot V_{max}^{-1}$ $\pi_{3,ult}$ $V_{OC} \cdot V_{max}$ $\pi_{4,ult}$ $I \cdot R \cdot V_{max}^{-1}$ $\pi_{5,ult}$ n_{cells}

The resulting input-output scaling equivalency is shown in Table 4-3.

Table 4-3: Ultracapacitor Pack Scaling Equival	lency
--	-------

Table 4-2: Ultracapacitor Pack Scaling Pi-Groups

HIL Component	Prototype Model
t_H	$t_P \cdot rac{R_H C_H}{R_P C_P}$
I_H	${I}_{P} \cdot rac{R_{P} \cdot V_{max,H}}{V_{max,P} \cdot R_{H}}$
$V_{H} \cdot rac{V_{max,P} \cdot n_{cells,P}}{V_{max,H} \cdot n_{cells,H}}$	V_P

The dynamic system equations are rearranged in dimensionless form in Eq. 4.4.

$$\frac{R \cdot C \cdot d}{dt} \left(\frac{V_{OC}}{V_{max}} \right) = -\frac{I \cdot R}{V_{max}}$$

$$\frac{V}{V_{max} \cdot n_{cells}} = \frac{V_{OC}}{V_{max}} - \frac{I \cdot R}{V_{max}}$$
4.4

The pi-parameters are substituted in Eq. 4.5, where the derivative operator is also

dimensionless, i.e. (') = $R \cdot C \cdot \frac{d}{dt}$.

$$\pi'_{3,ult} = \pi_{4,ult}$$

$$\frac{\pi_{2,ult}}{\pi_{5,ult}} = \pi_{3,ult} - \pi_{4,ult}$$
4.5

For dynamic similarity of two systems, the system pi-groups need to have identical values [17]. By definition, any ratio of dimensionless variables can also be defined as a pi-group, such as the ratio $\pi_{2,ull}/\pi_{5,ull}$. This new pi-group is equivalent between two systems by Eq. 4.5 as long as π_4 and π_1 are equivalent. By use of an input scaling factor, π_4 has been set equivalent in Table 4-3, so it remains to require that $\pi_{1,ult,H}$ $= \pi_{1,ult,P}$. As an option, to avoid using different time scales during HIL, an alternative requirement would be $R_H \cdot C_H = R_P \cdot C_P$.

4.2 Case 4: Engine Model from the PSAT Library

The equation of motion for an engine according to the PSAT libraries is given in Eq. 4.6, where throttle command θ_{cmd} is an input, with a range of [0, 1]; rotational speed ω is another input, torque *T* is the output; and T_{max} is the maximum torque as a function of speed ω .

$$T = T_{max}(\omega) \cdot (1.1 \cdot \theta_{cmd} - 0.1)$$
4.6

An input-output diagram of the engine model is shown in Figure 4-4.



Figure 4-4: Engine model inputs and output.

An empirical formula Eq. 4.7 was adopted from [25] for T_{max} , where P_{max} is maximum power, ω_{Pmax} is the speed at which maximum power occurs, and P_1 and P_2 are dimensionless coefficients. Typical values for P_1 and P_2 are 1 and 1 for spark ignition engines, and 0.6 and 1.4 for compression ignition engines.

$$T_{max} = P_1 \cdot \frac{P_{max}}{\omega_{Pmax}} + P_2 \cdot \frac{P_{max}}{\omega_{Pmax}^2} \cdot \omega - \frac{P_{max}}{\omega_{Pmax}^3} \cdot \omega^2$$
4.7

Thus, a new equation of motion for an engine is Eq. 4.8.

$$T = \left(P_1 \cdot \frac{P_{max}}{\omega_{Pmax}} + P_2 \cdot \frac{P_{max}}{\omega_{Pmax}^2} \cdot \omega - \frac{P_{max}}{\omega_{Pmax}^3} \cdot \omega^2\right) \cdot (1.1 \cdot \theta_{cmd} - 0.1)$$
4.8

Applying the dimensionless variable method, the engine system has N = 2parameters: P_{max} and ω_{Pmax} , composed of M = 3 dimensions: length, mass, and time. To the parameters shall be added S = 2 signals: T and ω . The signals and parameters are shown with their dimensions in Table 4-4. The throttle command θ_{cmd} and the coefficients P_1 and P_2 are already dimensionless, and are thus excluded.

Variable	Symbol	Dimension
torque	Т	m ² ·kg·s ⁻²
speed	ω	s^{-1}
maximum power	P_{max}	m ² ·kg·s ⁻³
nominal voltage	ω_{Pmax}	s ⁻¹

The number of fundamental dimensions is three, but the $[m^2]$ dimension and the [kg] dimension always appear together, so the two are combined into a new composite dimension, leaving a total of M = 2 dimensions. The number of repeating parameters is therefore also two, so the two parameters are selected as the repeating parameters. The number of pi-groups is Q = N + S - M = 2. The completed dimensional set matrix is given in Eq. 4.9.

The resulting pi-groups, including the throttle command θ_{cmd} and the coefficients P_1 and P_2 , are given in Table 4-5.

Dimensionless Variable	Variable Grouping
$\pi_{1,eng}$	$T \cdot \omega_{Pmax} \cdot P_{max}^{-1}$
$\pi_{2,eng}$	$\omega \cdot \omega_{Pmax}^{-1}$
$\pi_{3,eng}$	P_1
$\pi_{4,eng}$	P_2
$\pi_{5,eng}$	$ heta_{cmd}$

The resulting input-output scaling equivalency is shown in Table 4-6.

Table 4	-6 :	Engine	Scal	ing	Eq	uival	lency
		~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~					

HIL Component	Prototype Model
$ heta_{cmd,H}$	$ heta_{cmd,P}$
ω_H	$\omega_{_P} \cdot rac{\omega_{_{Pmax,H}}}{\omega_{_{Pmax,P}}}$
$T_{H} \cdot \frac{\omega_{Pmax,H} \cdot P_{max,P}}{P_{max,H} \cdot \omega_{Pmax,P}}$	T_P

The equation of motion is rearranged in dimensionless form in Eq. 4.10.

$$\frac{T \cdot \omega_{Pmax}}{P_{max}} = \left[P_1 + P_2 \cdot \frac{\omega}{\omega_{Pmax}} - \left(\frac{\omega}{\omega_{Pmax}}\right)^2\right] \cdot (1.1 \cdot \theta_{cmd} - 0.1)$$
4.10

The pi-parameters are substituted in Eq. 4.11.

$$\pi_{1,eng} = \left(\pi_{3,eng} + \pi_{4,eng} \cdot \pi_{2,eng} - \pi_{2,eng}^2\right) \cdot \left(1.1 \cdot \pi_{5,eng} - 0.1\right)$$
4.11

For dynamic similarity of two systems, the system pi-groups need to have identical values [17]. The pi-group $\pi_{1,eng}$ is equivalent between two systems by Eq. 4.11 as long as $\pi_{2,eng}$, $\pi_{3,eng}$, $\pi_{4,eng}$, and $\pi_{5,eng}$ are equivalent. By use of input scaling factors, $\pi_{2,eng}$ and $\pi_{5,eng}$ have been set equivalent in Table 4-6, so it remains to require that $\pi_{3,eng,H}$ $= \pi_{3,eng,P}$ and $\pi_{4,eng,H} = \pi_{4,eng,P}$, that is, $P_{1,H} = P_{1,P}$ and $P_{2,H} = P_{2,P}$. In other words, a gasoline engine cannot predict a diesel engine's performance, or vice versa.

Of special interest is the effect of the dimensional approach on the function noted in Eq. **4.6**. The iterations undergone by this function are developed in Eq. **4.12**.

$$T_{max} = f(\omega) \approx P_1 \cdot \frac{P_{max}}{\omega_{Pmax}} + P_2 \cdot \frac{P_{max}}{\omega_{Pmax}} \cdot \frac{\omega}{\omega_{Pmax}} - \frac{P_{max}}{\omega_{Pmax}} \cdot \left(\frac{\omega}{\omega_{Pmax}}\right)^2$$

$$\frac{T_{max} \cdot \omega_{Pmax}}{P_{max}} = \frac{\omega_{Pmax}}{P_{max}} \cdot f(\omega) \approx P_1 + P_2 \cdot \frac{\omega}{\omega_{Pmax}} - \left(\frac{\omega}{\omega_{Pmax}}\right)^2$$
4.12

The function f may be redefined as the dimensionless function d, as in Eq. 4.13.

$$d\left(\frac{\omega}{\omega_{Pmax}}\right) \equiv P_1 + P_2 \cdot \frac{\omega}{\omega_{Pmax}} - \left(\frac{\omega}{\omega_{Pmax}}\right)^2$$
4.13

A generalization may be made for functions of unspecified order. Instead of the requirement that the function coefficients of model and prototype must be equal, an equivalent requirement is that the function in its dimensionless form must be the same for both model and prototype, i.e., $d_H = d_P$, at least over the range being tested.

4.3 Engine Scaling Simulation Results

The dimensionless variable method was applied to engine scaling in a PSAT simulation. The vehicle model was a series hybrid gasoline electric, and the engines being compared were a 90 kW 1.8 L 4-cylinder Toyota and a 120 kW 4.0 L 6-cylinder Ford. To verify that both engines had similar power vs. rotational speed curves, these were plotted, along with the plot of empirical formula of Eq. 4.7, in Figure 4-5.



Figure 4-5: Engine dimensionless power.

While some similarity is evident between the model and both engines, it is necessary to proceed with the simulation in order to determine if the similarity observed is sufficient.

The vehicle was simulated on the Federal Urban Dynamometer Schedule (FUDS) driving cycle. Scaling was applied to input rotational speed ω and output torque *T* of the 120 kW engine.

The scaled shaft speed inputs of both engines are shown in Figure 4-6.





The scaled throttle inputs of both engines are shown in Figure 4-7.



Figure **4-7**: Engine throttle inputs.

The torque output, scaled to match the 90 kW engine, is plotted in Figure 4-8.



Figure 4-8: Engine torque outputs, scaled to match the 90 kW engine.

The error in torque prediction is plotted in Figure 4-9.



Figure 4-9: Difference between torques of prototype and scaled engines.

The root mean square error for the 60-second cycle is 2.51 N*m. The differences between the 120 kW engine scaled torque output and the 90 kW engine torque output evident in Figure 4-8 may be attributed to the differences visible in Figure 4-5 between the power vs. speed curves of the two engines.

4.4 Case 5: Fuel Cell Model from the PSAT Library

The equations of motion for a fuel cell according to the PSAT libraries are given in Eq. 4.14, where reference power P_{ref} is the input, power P is the output, T is dimensionless temperature ratio with range [0, 1], $P_{max,c}$ is the maximum power output when cold, $P_{max,h}$ is the maximum power output when hot, $\dot{m}_{H_2,c}$ is the mass flow rate of hydrogen as a function of power when the fuel cell is cold, $\dot{m}_{H_2,h}$ is the mass flow rate of hydrogen when the fuel cell is hot, $\dot{m}_{H_2,max}$ is the maximum mass flow rate of hydrogen, τ_h is the hot temperature time constant, τ_c is the cold temperature time constant, and τ is the power time constant.

$$P_{ref} \leq (1-T) \cdot P_{max,c} + T \cdot P_{max,h}$$

$$\dot{T} = \left(\frac{1}{\tau_h} \cdot \frac{\dot{m}_{H_2,h} - \dot{m}_{H_2,c}}{\dot{m}_{H_2,max}} - \frac{1}{\tau_c}\right) \cdot T + \frac{1}{\tau_h} \cdot \frac{\dot{m}_{H_2,c}}{\dot{m}_{H_2,max}}$$

$$\dot{P} = -\frac{1}{\tau} \cdot \left(P - P_{ref}\right)$$

$$A.14$$
Note:
$$\begin{cases} \dot{m}_{H_2,c} = f_2 \left(\frac{P}{P_{max,h}}\right) \\ \dot{m}_{H_2,h} = f_1 \left(\frac{P}{P_{max,h}}\right) \end{cases}$$



The cold H₂ mass flow rate $\dot{m}_{H_2,c}$ of a typical fuel cell from the PSAT library is shown in Figure 4-10.

Figure 4-10: Cold hydrogen mass flow rate.

The hot H₂ mass flow rate $\dot{m}_{H_{2},c}$ of a typical fuel cell is shown in Figure 4-11.



Figure 4-11: Hot hydrogen mass flow rate.

An input-output diagram of the fuel cell model is shown in Figure 4-12.



Figure 4-12: Fuel cell input and output.

Applying the dimensionless variable method, the fuel cell system has N = 8parameters: $P_{max,c}$, $P_{max,h}$, $\dot{m}_{H_2,c}$, $\dot{m}_{H_2,h}$, $\dot{m}_{H_2,max}$, τ_h , τ_c , and τ . They are composed of M = 3dimensions: length, mass, and time. To the parameters shall be added S = 3 signals: t, P_{ref} , and P. The signals and parameters are shown with their dimensions in Table 4-7. The temperature ratio T is already dimensionless, and is thus excluded.

Variable	Symbol	Dimension
time	t	S
hot temperature time constant	$ au_h$	S
cold temperature time constant	$ au_c$	S
reference power	P_{ref}	m ² ·kg·s ⁻³
power	P	m ² ·kg·s ⁻³
cold max power	$P_{max,c}$	m ² ·kg·s ⁻³
cold hydrogen mass flow rate	$\dot{m}_{H_2,c}$	kg·s ⁻¹
hot hydrogen mass flow rate	$\dot{m}_{H_2,h}$	kg·s ⁻¹
power time constant	τ	S
maximum hydrogen mass flow rate	$\dot{m}_{H_2,max}$	kg·s ⁻¹
maximum power	$P_{max,h}$	m ² ·kg·s ⁻³

Table 4-7: Parameters Relevant to Fuel Cell Scaling

There are a total of M = 3 dimensions. The number of repeating parameters is therefore also three, so the three constant parameters $P_{max,h}$, $\dot{m}_{H_2,h}$, and τ are arbitrarily selected as the repeating parameters. In this example, $P_{max,c}$ and either of the other time constants could have served as well as those actually selected. The number of pi-groups is Q = N + S - M = 8. Non-repeating parameters with the same dimensions are grouped together for simplicity. The completed dimensional set matrix is given in Eq. 4.15.

	t, τ_h, τ_c	$P_{ref}, P, P_{max,c}$	$\dot{m}_{H_2,c}, \dot{m}_{H_2,h}$	τ	$\dot{m}_{H_2,max}$	$P_{max,h}$	
m	0	2	0	0	0	2	
kg	0	1	1	0	1	1	
S	1	-3	-1	1	-1	-3	4.15
$\pi_{1,2,3,fc}$	1	0	0	-1	0	0	
$\pi_{_{4,5,6,fc}}$	0	1	0	0	0	-1	
$\pi_{_{7,8,fc}}$	0	0	1	0	-1	0	

The resulting pi-groups, including the temperature *T*, are given in Table 4-8.

Dimensionless Variable	Variable Grouping
$\pi_{1,fc}$	$t \cdot \tau^{-1}$
$\pi_{2,fc}$	$ au_h \cdot au^{-1}$
$\pi_{3,fc}$	$ au_c \cdot au^{-1}$
$\pi_{4,fc}$	$P_{ref} P_{max,h}^{-1}$
$\pi_{5,fc}$	$P \cdot P_{max,h}^{-1}$
$\pi_{6,fc}$	$P_{max,c} \cdot P_{max,h}^{-1}$
$\pi_{7,fc}$	$\dot{m}_{H_2,c}\cdot\dot{m}_{H_2,max}$
$\pi_{8,fc}$	$\dot{m}_{H_2,h}\cdot\dot{m}_{H_2,max}$
$\pi_{9 fc}$	T

Table 4-8: Fuel Cell Scaling Pi-Groups

HIL Component	Prototype Model
t_H	$t_{_P}\cdot rac{ au_{_H}}{ au_{_P}}$
$P_{ref,H}$	$P_{ref,P} \cdot rac{P_{max,h,H}}{P_{max,h,P}}$
$P_H \cdot \frac{P_{max,h,P}}{P_{max,h,II}}$	P_P

The resulting input-output scaling equivalency is shown in Table 4-9.

The equations of motion are rearranged in dimensionless form in Eq. 4.16. Note that since the functions f_1 and f_2 already take dimensionless arguments, they undergo no change of form. They only need to be scaled by the factor $\dot{m}_{H_2,max}$.

$$\frac{P_{ref}}{P_{max,h}} \leq (1-T) \cdot \frac{P_{max,c}}{P_{max,h}} + T$$

$$\tau \cdot \dot{T} = \left[\frac{\tau}{\tau_h} \cdot \left(\frac{\dot{m}_{H_2,h}}{\dot{m}_{H_2,max}} - \frac{\dot{m}_{H_2,c}}{\dot{m}_{H_2,max}} \right) - \frac{\tau}{\tau_c} \right] \cdot T + \frac{\tau}{\tau_h} \cdot \frac{\dot{m}_{H_2,c}}{\dot{m}_{H_2,max}}$$

$$\tau \cdot \frac{\dot{P}}{P_{max,h}} = -\frac{P}{P_{max,h}} + \frac{P_{ref}}{P_{max,h}}$$

$$4.16$$
Note:
$$\begin{cases} \frac{\dot{m}_{H_2,h}}{\dot{m}_{H_2,max}} = \frac{1}{\dot{m}_{H_2,max}} \cdot f_1 \left(\frac{P}{P_{max,h}} \right) \\ \frac{\dot{m}_{H_2,max}}{\dot{m}_{H_2,max}} = \frac{1}{\dot{m}_{H_2,max}} \cdot f_2 \left(\frac{P}{P_{max,h}} \right) \end{cases}$$

The pi-parameters are substituted in Eq. 4.17, where the derivative operator is

also dimensionless, i.e. (') = $\tau \cdot \frac{d}{dt}$.

$$\pi_{4,fc} \leq (1 - \pi_{9,fc}) \cdot \pi_{6,fc} + \pi_{9,fc}$$

$$\pi_{9,fc}' = \left[\frac{1}{\pi_{2,fc}} \cdot (\pi_{8,fc} - \pi_{7,fc}) - \frac{1}{\pi_{3,fc}}\right] \cdot \pi_{9,fc} + \frac{\pi_{7,fc}}{\pi_{2,fc}}$$

$$\pi_{5,fc}' = -\pi_{5,fc} + \pi_{4,fc}$$
4.17

For dynamic similarity of two systems, the system pi-groups need to have identical values [17]. The pi-groups $\pi_{5,fc}$ and $\pi_{9,fc}$ are equivalent between two systems by Eq. 4.17 as long as the other pi-groups are equivalent. By use of an input scaling factor, $\pi_{4,fc}$ has been set equivalent in Table 4-9, so the requirements are those listed in Eq. 4.18.

$$\pi_{1,fc,H} = \pi_{1,fc,P}
\pi_{2,fc,H} = \pi_{2,fc,P}
\pi_{3,fc,H} = \pi_{3,fc,P}
\pi_{6,fc,H} = \pi_{6,fc,P}
\pi_{7,fc,H} = \pi_{7,fc,P}
\pi_{8,fc,H} = \pi_{8,fc,P}$$
4.18

As an alternative, to avoid using different time scales during HIL, the

requirements for $\pi_{1,fc}$, $\pi_{2,fc}$, and $\pi_{3,fc}$ translate into the requirements that $\tau_H = \tau_P$, $\tau_{c,H} = \tau_{c,P}$, and $\tau_{h,H} = \tau_{h,P}$.

Chapter 5

Battery Scaling Experiment

In order to test and demonstrate the scaling procedures derived in this work, an experiment was developed and undertaken comparing the responses of two sealed leadacid batteries: a 13.6 Ah Enersys Odyssey PC680 battery and a 120 Ah Deka 6TAGM battery. The Deka was chosen to be the prototype, and the Odyssey was chosen as the scale model to be tested to estimate the characteristics of the prototype. This chapter will describe the components of the experiment, as shown in Figure **5-1**, in this order: battery, powertrain model and drive cycle, the "scale" and "rescale" multipliers, and the ABC-150 power system. Finally, the results of the scaling comparison will be presented.



Figure 5-1: Experiment configuration.

5.1 Setup of the Experiment

5.1.1 Batteries

The battery portion of the experiment, highlighted in Figure 5-2, is the topic of this section.



Two sealed lead-acid absorbed glass mat batteries were selected for scaling comparison: a 13.6 Ah Enersys Odyssey PC680 battery and a 120 Ah Deka 6TAGM battery. Both batteries were of similar construction and chemistry. Although the battery characteristic pi-parameters, which will be reviewed below, were assumed to be equivalent, both batteries were subjected to testing to determine if this were so. Based on techniques described in [26], a series of 10 cycles of the FreedomCar Maximum Power-Assist (50 Wh) Efficiency and Baseline Cycle Life Test profile was applied to both batteries, with measurement made of battery voltage throughout the cycle.

The profile, shown in Figure 5-3, is designed to maintain state of charge, assuming a discharge/charge efficiency of 90%. The magnitude of the profile is designed for an entire battery pack, and is designed to be scaled down when an individual battery

is being tested. The characteristics to be determined by the test are assumed to be a function of state of charge only, so the degree to which the profile is scaled is considered to be of minor importance, since the battery remains at approximately the same state of charge throughout the test. What is important in scaling the profile is that the battery voltage does not go outside its acceptable range during testing, which for the Odyssey is [7.2, 14.7] V [27], and for the Deka is [9.6, 14.75] V [28].



Figure **5-3**: FreedomCar Maximum Power-Assist (50 Wh) Efficiency and Baseline Cycle Life Power Demand Profile [26].

It was assumed that an appropriate battery pack would have a size of 30 12-volt batteries, for a nominal pack voltage of 360 V. Hybrid-electric vehicle battery packs are typically limited to 400 V for reasons of safety. Since only one battery out of each pack was tested, the power demand of the profile was divided by 30. Applying the reduced profile to the Odyssey battery, however, caused it to exceed the upper voltage limit of 14.7 V. Since none of the characteristics being determined are considered to be a function of the magnitude of current load in the current model, the particular scaling is assumed to be of minor importance. Thus, to prevent the Odyssey battery from exceeding its maximum voltage, the profile was divided by 60 instead of 30. The resulting profile for each battery is described in Table **5-1**.

Odyssey PC680	Deka 6TAGM	
0.05	0.1	36
0.4	0.8	3
0.054	0.107	49
0.35	0.7	2
	0.05 0.4 0.054 0.35	0.05 0.1 0.4 0.8 0.054 0.107 0.35 0.7

Table 5-1: Battery Testing Discharge/Charge Profile

A new variable, dynamic current I_c , is defined, as shown in Eq. 5.1, to allow the use of linear regression to determine of battery characteristics, cell dynamic resistance R_c and cell internal resistance R_{int} .

$$I_c \equiv -V_c / (R_c \cdot n_{cells})$$
 5.1

The equations of motion for the battery, previously described in Eq. 3.1, were modified by the substitution of I_c , as shown in Eq. 5.2.

$$\dot{I}_{c} = -\frac{1}{\tau} \cdot (I - I_{c})$$

$$V = (-I_{c} \cdot R_{c} - I \cdot R_{int} + V_{OC}) \cdot n_{cells}$$
5.2

For the purposes of regression analysis, an estimated voltage \hat{V} was calculated according to Eq. 5.3, with $\hat{R}_{c,bat}$ being the estimated battery dynamic resistance,

 $\hat{R}_{int, bat}$ being the estimated battery internal resistance, and $\hat{V}_{OC, bat}$ being the estimated battery open circuit voltage.

$$\hat{R}_{c,bat} \approx R_c \cdot n_{cells}$$

$$\hat{R}_{int,bat} \approx R_{int} \cdot n_{cells}$$

$$\hat{V}_{OC,bat} \approx V_{OC} \cdot n_{cells}$$

$$\hat{V} = -I_c \cdot \hat{R}_{c,bat} - I \cdot \hat{R}_{int,bat} + \hat{V}_{OC,bat}$$
5.3

Based on measured current during the profile, and an estimated time constant $\hat{\tau}$, dynamic current was calculated for each time step according to Eq. 5.4 [26].

$$I_{c,i} = \{ 1 - [1 - \exp(-\Delta t / \hat{\tau})] / (\Delta t / \hat{\tau}) \} \cdot I_i$$

+ $\{ [1 - \exp(-\Delta t / \hat{\tau})] / (\Delta t / \hat{\tau}) - \exp(-\Delta t / \hat{\tau}) \} \cdot I_{i-1}$
+ $\{ \exp(-\Delta t / \hat{\tau}) \} \cdot I_{c,i-1}$
5.4

Thus there were four estimated parameters: $\hat{R}_{c,bat}$, $\hat{R}_{int,bat}$, $\hat{V}_{OC,bat}$, and $\hat{\tau}$. The statistic used as a measure of combined estimation accuracy was the coefficient of determination, r^2 , which is calculated according to Eq. 5.5. In this equation, V is the measured voltage, \overline{V} is the mean measured voltage, and \hat{V} is the estimated voltage.

$$r^{2} = \left| \sum \left(V - \overline{V} \right)^{2} - \sum \left(V - \hat{V} \right)^{2} \right| / \sum \left(V - \overline{V} \right)^{2}$$
5.5

Each estimated parameter was varied incrementally for each battery until a maximum value of r^2 was obtained. Results for both batteries are summarized in Table 5-2. The lead-acid batteries under test each were composed of 6 cells, so the estimated parameters on a per-cell basis are also given.

Parameter	Enersys Odyssey PC680	Deka 6TAGM		
$\hat{V}_{OC, bat}(\mathbf{V})$	12.54	12.48		
$\hat{R}_{c,bat}\left(\Omega ight)$	0.0425	0.0115		
$\hat{R}_{_{int,bat}}(\Omega)$	0.0243	0.00868		
$\hat{\tau}$ (s)	14.1	17.2		
r^2	0.915	0.980		
stimated cell V_{OC} (V)	2.090	2.081		
Estimated cell $R_c(\Omega)$	0.00708	0.00192		
Estimated cell $R_{int}(\Omega)$	0.00404	0.00145		

Table 5-2: Battery characteristic estimation statistics

The measured and estimated voltage for the Deka battery are shown in Figure 5-4.



Figure 5-4: Measured and estimated voltage of Deka battery.

The measured and estimated voltage for the Odyssey battery are shown in

Figure **5-5**.



Figure 5-5: Measured and estimated voltage of the Odyssey battery.

The characteristic pi-parameters for batteries are $\pi_{1,bat}$ and $\pi_{2,bat}$, as derived in

C	hapt	er	3,	sect	tion	3.	Ι,	and	Sł	10W	n	ın	Ta	ble	e	5-	3.
---	------	----	----	------	------	----	----	-----	----	-----	---	----	----	-----	---	----	----

Table 5-3: Characteristic battery pi-parameters	
Dimensionless Variable	Variable Grouping
$\pi_{1,bat}$	$Q \cdot R_{int} \cdot \tau^{-1} \cdot V_{nom}^{-1}$
$\pi_{2,bat}$	$R_c \cdot R_{int}^{-1}$

The parameters required for calculating $\pi_{1,bat}$ and $\pi_{2,bat}$, as well as the calculated pi-parameters themselves, are shown in Table 5-4. The value for cell capacity Q is specified by the manufacturer [27, 28], and the value for nominal voltage V_{nom} for both batteries, since both have lead-acid chemistry, was arbitrarily chosen to be V_{OC} at 50% state of charge, that is, 2.083 V.

Parameter	Enersys Odyssey PC680	Deka 6TAGM
Estimated cell V_{OC} (V)	2.090	2.081
Estimated cell $R_c(\Omega)$	0.00708	0.00192
Estimated cell $R_{int}(\Omega)$	0.00404	0.00145
Estimated τ (s)	14.1	17.2
Specified $Q(s \cdot A)$	8160	72000
Specified $\tilde{V}_{nom}(\mathbf{V})$	2.083	2.083
$\pi_{1,bat}$	1.123	2.913
$\pi_{2,bat}$	1.75	1.33

As seen in Table 5-4, the values for $\pi_{1,bat}$ and $\pi_{2,bat}$ are not equal, although they are within an order of magnitude. The question is, are they close enough? The answer to this question will be determined by the experiment that follows. Note that the above measurement was performed at only one state of charge. A full comparison of both batteries' characteristics would require repetition of the same test at multiple states of charge, which tests are beyond the scope of this work.

5.1.2 Vehicle Powertrain Models

The vehicle powertrain and drive cycle portions of the experiment, highlighted in Figure **5-6**, are the topic of this section.



Figure **5-6**: Experiment configuration.

Two Simulink vehicle powertrain models were created using PSAT, both a GM EV1 electric vehicle powertrain (EV) and a Honda Insight parallel hybrid vehicle powertrain (PAR) using the US06 Supplemental Federal Test Procedure [29] as the drive cycle. The Simulink model of each vehicle was saved, along with the workspace variables. Two modifications were made to the models. First, the current input signal to the battery portion of each model was connected to a UDP Send block from the xPC Target library of Simulink. This UDP signal supplied a current command for an AeroVironment ABC-150 Power Processing System, described below. Second, the output of an ADC block providing a measurement of battery voltage replaced the voltage output signal of the battery portion of each model. Third, another ADC block was included to record measurement of current load on the battery. Using the Real Time Workshop, each simulation was compiled to run on an industrial PC with data acquisition capabilities, using the xPC Target real time operating system [23].

5.1.3 Scaling Factors

The scaling factors, labeled "scale" and "rescale", highlighted in Figure 5-7, are the topic of this section.



The label "scale" refers to the scaling applied to any signals that must be transformed from the prototype domain to the hardware domain, in this case the battery current. The label "rescale," in contrast, refers to the scaling applied to any signals that must be transformed from the hardware domain back to the prototype model domain, in this case the battery voltage. In the case of the present experiment, the prototype (P) is a pack of 30 Deka 6TAGM batteries (180 cells) connected in series. The hardware (H) is alternately a single Odyssey PC680 battery (6 cells) and a single Deka 6TAGM battery (6 cells). The "scale" and "rescale" multipliers are given in Table 5-5. They were derived in Chapter 3, Section 3.1.

	Formula	Pack of 30 Deka	Odyssey	Deka
	V (V)	2 05	<u>РС080 (П)</u> 2 05	<u>отабм (п)</u> 2.05
Parameters	$R_{int}(\Omega)$	0.00145	0.00404	0.00145
	n _{cells}	180	6	6
"scale"	$\frac{R_{int,P} \cdot V_{nom,H}}{V_{nom,P} \cdot R_{int,H}}$	N/A	0.359	1
"rescale"	$\frac{V_{nom,P} \cdot n_{cells,P}}{V_{nom,H} \cdot n_{cells,H}}$	N/A	30	30

5.1.4 Control Equipment

The ABC-150 Power System portion of the experiment, highlighted in Figure **5-8**, is the topic of this section.



Figure **5-8**: Experiment configuration.

The equipment used to control the current load applied to the batteries under test was AeroVironment's ABC-150 Power Processing System. It can source or sink up to 445 VDC, 530 ADC, or 125 kW. The ABC-150, pictured in Figure **5-9**, may be controlled either manually by controls on the front panel, or remotely, via RS232. AeroVironment provides a program that executes simple command scripts, as well as a
serial port driver that can be integrated into custom controls designed for the Windows operating system. In this experiment, the former was used for measurement of battery characteristics above, and the latter for the HIL simulation below. For HIL, a Visual Basic control was implemented that receives UDP signals transmitted across the local area network, and translates them into ABC-150 commands.



Figure 5-9: Aerovironment ABC-150 Power Processing System.

5.1.5 Complete Experimental System

The complete experimental system is pictured in Figure **5-10**. On the left is the industrial PC, with data acquisition board, running the vehicle simulation in real time. In the center is the Enersys Odyssey PC680 battery, with an ammeter clamped around one of the power cables. On the right is a computer running the Visual Basic control, which translates current commands from UDP to RS232. In the background is the ABC-150 power processing system.



Figure 5-10: Complete experimental system.

5.2 Experimental Results

Two HIL simulations were performed with both batteries, one a Honda Insight parallel hybrid vehicle powertrain (PAR) and the other a GM EV1 electric vehicle powertrain (EV). The current load applied to both batteries in the PAR simulation is shown in Figure 5-11.



Figure 5-11: Current load applied to batteries in the PAR simulation.

The voltage response of both batteries, scaled to full pack size, in the PAR simulation is shown in Figure 5-12.



Figure 5-12: Voltage response of batteries, scaled to pack size, in the PAR simulation.





Figure 5-13: Difference between voltages of batteries in the PAR simulation.

With a 2.21 V bias in pack V_{OC} removed, the root mean square error for the cycle is 1.063 V, which is 0.3% of initial V_{OC} of 372.5 V.

The current load applied to both batteries in the EV simulation is shown in Figure **5-14**.



Figure 5-14: Current load applied to batteries in the EV simulation.

The voltage response of both batteries, scaled to full pack size, in the EV simulation is shown in Figure 5-15.



Figure 5-15: Voltage response of batteries, scaled to full pack size, in the EV simulation.

The difference between voltage traces is plotted in Figure 5-16.



Figure **5-16**: Difference between voltage responses in the EV simulation.

With a 1.72 V bias in pack V_{OC} removed, the root mean square error for the cycle is 9.207 V, which is 2.5% of the initial V_{OC} of 374 V. In contrast to the PAR configuration, the prediction error in the EV configuration accumulated as the cycle progressed. This accumulation of error is due to the fact that the values for $\pi_{1,bat}$ were not equal for both batteries. The formula is repeated in Eq. 5.6.

$$\pi_{1,bat} = Q \cdot R_{int} \cdot \tau^{-1} \cdot V_{nom}^{-1}$$
5.6

If both batteries had been of exactly the same construction and chemistry, then the factors τ and V_{nom} could be expected to be equal, as well as $\pi_{1,bat}$ itself. In such a case, the ratio of Q would be the inverse of the ratio of R_{int} . In this case, however, the ratio of Q for the Odyssey and Deka was 1/8.8, while the ratio of R_{int} was 2.8. In the PAR simulation, state of charge *SOC* was roughly maintained, with equal amounts of discharge and charge currents, so that the Odyssey's disproportionately lower Q was not a factor. In the EV simulation, however, the amount of discharge current greatly exceeded the amount of charge current, which substantially changed *SOC* for both batteries. Because of the Odyssey's lower Q, the Odyssey's *SOC* changed more, causing increased error in voltage as the cycle progressed. The conclusion is that for charge sustaining cycles, such as the PAR simulation, equivalence of $\pi_{2,bat}$ may be sufficient, but for charge depleting cycles, such as the EV simulation, equivalence of $\pi_{1,bat}$ is also necessary.

Chapter 6

Conclusions and Future Work

6.1 Conditions for Use of Scaling Factors

Several conditions must be met for the successful application of scaled components in an HIL simulation. First, both the scaled component and the full-size component need to be described by the same type of system model. For example, the behavior of a second-order system with eigenvalues close to each other cannot be predicted by that of a first-order system. Second, the coefficients of the dimensionless system models for both scaled and full-size components need to be equal for the dominant dynamics, that is, the dynamics that cannot be neglected. Others have already begun investigations to determine which dynamics can be neglected, for example in [4, 12].

6.2 Conclusion

The dimensionless formulation was used in this study to rescale prototype vehicle powertrain component models to emulate full-scale counterparts. In the case of motors, batteries, and engines, input/output simulation results of the resulting scaled component models were compared with those of the full-scale models available from the PSAT component model library. The results show that the nonlinear scaling factors indeed produce generalizable vehicle component models that permit performance comparisons across very large size domains. This method was also tested with actual batteries, with a degree of similarity being observable in driving cycle tests.

6.3 Future Work

Future work to extend this study could include testing of powertrain components other than batteries, as well as testing of additional battery model and prototype pairs. Further investigation could continue the work developed in [4, 12] concerning which piparameters can be neglected. In addition, a necessary step would be the development of a mathematical condition to determine how close pi-parameters of model and prototype need to be for the results of HIL simulation to be useful.

Additional work also is possible in the work of developing Penn State's HIL system. Investigation could be done in the synchronization of networked components, as well as bumpless transfer between model and hardware, as HIL hardware is brought onor off-line during system startup and shutdown. Another opportunity for research is the development of fail-safe modes of operation for a HIL network.

Bibliography

- 1. A. Taylor III, "The Birth of the Prius," *Fortune Magazine*, March 6, 2006.
- 2. R. Zhang and A. G. Alleyne, "Dynamic emulation using an indirect control input," *Journal of* Dynamic *Systems, Measurement and Control*, vol. 127, no. 1, March 2005, pp. 114-124.
- O. Egeland and R. Skjetne, (2005, March 30). "Hardware-in-the-loop simulation for testing of DP vessels." Extended abstract. Available: <u>http://www.marinecybernetics.com/conf_papers/Egeland%20Extended%20Abstra_ ct%20ESTS05.pdf</u>
- 4. H. K. Fathy, R. Ahlawat, and J. L. Stein, "Proper powertrain modeling for enginein-the-loop simulation," in *Proc. 2005 ASME International Mechanical Engineering Congress and Exposition*, Orlando, 2005, Paper IMECE2005-81592.
- 5. H. K. Fathy, Z. S. Filipi, J. Hagena, and J. L. Stein, "Review of hardware-in-theloop simulation and its prospects in the automotive area," in *Proc.SPIE*, vol. 6228, *Modeling and Simulation for Military Applications*, 2006, p. 62880E.
- 6. W. Rulka and E. Pankiewicz, "MBS approach to generate equations of motions for HiL-simulations in vehicle dynamics," *Multibody System Dynamics*, vol. 14, no. 3-4, pp. 367-386, Nov. 2005.
- R. M. Schupbach and J. C. Balda, "A versatile laboratory test bench for developing powertrains of hybrid electric vehicles," in *Proc. 2002 Veh. Technol.Conf., VTC 2002-Fall*, vol. 3, Sept. 2002, pp. 1666–1670.
- 8. S. C. Oh, "Evaluation of motor characteristics for hybrid electric vehicles using the hardware in the loop concept," *IEEE Trans. Veh. Technol.*, vol. 54, no. 3, pp. 817-824, May 2005.
- 9. G. Lo Bianco, G. Pede, A. Puccetti, E. Rossi, and G. Mantovani, "Vehicle testing in ENEA drive-train test facility," SAE Paper 2001-01-0961, 2001.
- **10**. Z. Lemeŝ, A. Vath, T. Hartkopf, and H. Mäncher, "Dynamic fuel cell models and their application in hardware in the loop simulation," *Journal of Power Sources*, vol. 154, no. 2, pp. 386-393, March 21, 2006.
- A. Vath, Z. Lemeŝ, H. Mäncher, M. Söhn, N. Nicoloso, and T. Hartkopf, "Dynamic modelling and hardware in the loop testing of PEMFC," *Journal of Power Sources*, vol. 157, no. 2, pp. 816-827, July 3, 2006.

- B. Kittirungsi, H. K. Fathy, and J. L. Stein, "An efficient scaling methodology for dynamic models using dimensional and activity analyses," in *Proc. 2006 ASME International Mechanical Engineering Congress and Exposition*, Chicago, 2006, Paper IMECE2006-14587.
- **13**. T. A. McMahon and J. T. Bonner, *On Size and Life*, W. H. Freeman, New York, 1983.
- 14. T. Szirtes, *Applied Dimensional Analysis and Modeling*, 2nd ed., Burlington, MA: Butterworth-Heinemann, 1997.
- **15**. E. Buckingham, "On physically similar systems: illustrations of the use of dimensional equations," *Physical Review*, vol. 4, series 2, 1914, pp. 345-376.
- H. J. Palanthandalam-Madapusi, D. S. Bernstein, and R. Venugopal, "Dimensional analysis of matrices: state-space models and dimensionless units," *IEEE Control Systems Magazine*, Vol. 27, No. 6, December 2007, pp. 100-109.
- S. Brennan, "On size and control: the use of dimensional analysis in controller design," PhD. dissertation, Dept. Mech. Ind. Eng. Univ. Illinois, Urbana-Champaign, IL, 2002.
- M. Ghanekar, D. W. L. Wang, G. R. Heppler, "Scaling laws for linear controllers of flexible link manipulators characterized by nondimensional groups," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 1, February 1997, pp. 117-127.
- S. Lapapong, "Vehicle similitude modeling and validation of the Pennsylvania State University Rolling Roadway Simulator," Master's Thesis, Dept. Mech. Eng. Penn State University, University Park, PA, 2007.
- **20**. S. Brennan and A. Alleyne, "Dimensionless robust control with application to vehicles," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, July 2005, pp. 624-630.
- **21**. M. Polley and A. Alleyne, "Dimensionless analysis of tire characteristics for vehicle dynamics studies," in *Proc. 2004 Am. Ctrl. Conf.*, Boston, MA, June 30-July 2, 2004, pp. 3411-3416.
- **22**. www.anl.gov
- 23. www.mathworks.com
- 24. H. Hailu and S. Brennan, "Use of dimensional analysis to reduce the parametric space for gain-scheduling," in *Proc. ACC*, vol. 1, 2005, pp. 598-603.

- **25**. G. Genta, *Motor Vehicle Dynamics: Modeling and Simulation*, World Scientific Press, 1997.
- 26. U. S. Department of Energy, *FreedomCAR Battery Test Manual for Power-Assist Hybrid Electric Vehicles*, Idaho National Engineering & Environmental Laboratory, DOE/ID-11069, October 2003.
- 27. www.enersys.com
- 28. www.eastpenn-deka.com
- **29**. www.epa.gov

Appendix A

Modification of PSAT Models for HIL

A.1 Use of PSAT models for HIL

A.1.1 Preparation of the PSAT model to run independently of PSAT

A.1.1.1 PSAT Version 5.2

- 1. Start PSAT
 - a. Start Matlab
 - b. Change to the PSAT directory, e.g. "c:\psat_v52_p_365\root"
 - c. Enter "psat" at the Matlab command prompt and follow the instructions
- 2. Set up a vehicle configuration
 - a. Click the "Load a vehicle file" button
 - b. Select a configured vehicle, e.g. "gui_split_US_prius_in.m"
 - c. Using the PSAT "Vehicle Input" GUI, make any modifications desired to the vehicle configuration
 - d. Click "Save" to save the configuration file, "Continue" to continue
- 3. Using the PSAT "Simulation Parameters" GUI, select a drive cycle, e.g. "us06"
- 4. Click the "RUN Stored and current Simu." Button
- 5. You will see the Simulink model being built and begin to run.

- 6. As soon as the Simulink model begins to run, switch to Matlab command prompt and press Ctrl-C to stop the simulation. Matlab will give some error messages.
- Save the Simulink model in a new folder under an appropriate name, e.g. "seriesPrius.mdl"
- 8. To prevent the workspace file from being too large, in case the Simulink model execution was not halted immediately,
 - a. View the workspace in Matlab
 - Right-click on the column headings in the workspace pane, e.g. Name,
 Value, etc
 - c. Select "Size" as an additional column heading
 - d. Click on the "Size" column heading to sort the workspace variables by size. This will group together all the variables ending in "…hist"
 - e. Delete all the variables ending in "...hist"
- 9. Change the Matlab current directory to the new folder
- Save the workspace under an appropriate name, e.g. "seriesPrius.mat" by entering "save seriesPrius" at the Matlab command prompt
- 11. Enter the following commands at the Matlab command prompt. They create the drive cycle data files necessary to perform a drive cycle.
 - a. save cycle sch_cycle; save grade sch_grade; save key_on sch_key_on
- 12. If you wish to use a different drive cycle, follow these instructions
 - a. Change the Matlab current directory to the PSAT driving schedule folder,
 - e.g. "c:\psat_v52_p_365\component\init_files\sch"

- b. Open the data file corresponding to the drive cycle you want, e.g."us06.mat"
- c. Change the Matlab current directory back to the new folder
- d. Enter the following commands at the Matlab command prompt. They create the drive cycle data files necessary to perform a drive cycle.
 - i. sch_cycle = sch_cycle'; sch_grade = sch_grade'; sch_key_on = sch_key_on'; save cycle sch_cycle; save grade sch_grade; save key_on sch_key_on
- Switch to the Simulink model, go to the menu item File/Model Properties/Callbacks
- 14. Add a PreLoadFcn callback: "load seriesPrius" This will cause the workspace to be loaded automatically every time you open the model. Close the model properties dialog box and save the model.
- 15. To enable use of the model on a computer that does not have PSAT, copy to the new folder all the pictures from the PSAT picture directory, e.g. "c:\psat_v52_p_365\root\pictures" Not all of them are needed, but this way you don't have to sort through them to find the ones you need.
- The Simulink model is now ready to run independent of PSAT and may be modified as needed.

A.1.1.2 PSAT Version 6.1

17. Start PSAT – double-click the icon for PSAT

- a. Enter a user name
- b. Click the icon for Light Duty or Heavy Duty
- c. If a Matlab Version Description window pops up, click OK
- 18. Set up a vehicle configuration
 - a. Select the menu item "File/Load a vehicle file"
 - b. Select a configured vehicle, e.g. "gui_split_US_prius_in.m"
 - c. Using the "Drivetrain Configuration," "Drivetrain Components," and "Controller/Strategy" tabs, make any modifications desired to the vehicle configuration
 - d. Select the menu item "File/Save Vehicle" to save the configuration file
- 19. Using the "Simulation Setup" tab, select a drive cycle, e.g. "US06"
- 20. Using the "Run Simulations" tab, select "rerun0.m" and click "Run the Simulations..."
- 21. You will see the Simulink model being built and begin to run.
- 22. The Simulink model will be saved in a new folder under an automatically generated name, e.g.

"C:\PSATv61\users\<username>\save_simu\ser_eng_2wd_p1_au_2ess_US06_02 2708_125851"

- 23. The workspace is saved under the name "SimuWS.mat" in the folder "C:\PSATv61\users\<username>"
- 24. Close PSAT and start Matlab
- 25. To prevent the workspace file from being too large
 - a. Load the workspace in Matlab

- i. Change the current directory to "C:\PSATv61\users\<username>"
- ii. Double-click on the SimuWS.mat icon or type "load SimuWS" at the Matlab command prompt
- B. Right-click on the column headings in the workspace pane, e.g. Name, Value, etc
- c. Select "Size" as an additional column heading
- d. Click on the "Size" column heading to sort the workspace variables by size. This will group together all the variables ending in "…simu"
- e. Delete all the variables ending in "...simu"
- 26. Change the Matlab current directory to the new folder where the Simulink model is located (see above)
- 27. Save the workspace under an appropriate name, e.g. "seriesPrius.mat" by entering "save seriesPrius" at the Matlab command prompt
- 28. If you wish to use a different drive cycle, follow these instructions
 - a. Change the Matlab current directory to the PSAT driving cycle folder, e.g.
 "C:\PSATv61\component\initialization\drive_cycle"
 - b. Open the data file corresponding to the drive cycle you want, e.g.
 "us06.mat"
 - c. Change the Matlab current directory back to the new folder
 - d. Save the workspace under an appropriate name, e.g. "seriesPrius.mat" by entering "save seriesPrius" at the Matlab command prompt
- 29. To avoid error messages associated with missing pictures, copy to the new folder all the pictures from the PSAT picture directory, e.g. "C:\PSATv61\root\pictures"

Not all of them are needed, but this way you don't have to sort through them to find the ones you need.

- 30. Open the Simulink model, and delete the Main Disclosure block
- 31. Select the menu item File/Model Properties/Callbacks
- 32. Add a PreLoadFcn callback: e.g. "load seriesPrius" This will cause the workspace to be loaded automatically every time you open the model. Close the model properties dialog box and save the model.
- 33. The Simulink model is now ready to run independent of PSAT and may be modified as needed.

A.1.2 Preparation of the Simulink model to run on an embedded system – xPC Target required

- 34. With the Simulink model open, select menu item Simulation/Configuration Parameters
- 35. Under Solver, change the Type to Fixed-step.
- 36. Set the Fixed-step size to something like 0.01
- 37. Under Real-Time Workshop, for use with xPC Target, change the System target file to xpctarget.tlc
- 38. Click OK
- 39. Select menu item Tools/Real-Time Workshop/Build Model
- 40. This creates a DLM file with the same name as the Simulink model, which will run on an embedded system set up for xPC Target
- 41. The Simulink model is now ready to run on an embedded system.

A.1.3 Addition of inputs/outputs to a Simulink model – TeraSoft library required for Advantech

- 42. Open the Simulink library (enter "simulink" at the Matlab command prompt)
- 43. For i/o other than digital or analog, e.g. UDP or RS-232, or for digital or analog i/o on computers other than Advantech, expand the library xPC Target and find the required blocksets
- 44. For digital or analog i/o on Advantech computers with the PCI-1716 card, expand the library TeraSoft xPC Driver Blockset/Advantech_PCI Library/PCI-1716 Library
- 45. Drag the desired block onto the desired subsystem of the Simulink model
- 46. Change the block settings appropriately
 - a. for the PCI slot setting with Advantech computers, if you leave it as -1, be sure to check afterward whether or not you have the breakout board cable connected to the correct i/o card on the Advantech
 - b. to specify an i/o card, connect to the embedded PC using xPC Target Explorer to see what are the available PCI slots, e.g. [2, 11] or [2, 12]
- 47. Connect the block port(s) to the desired signal by clicking on the port and dragging to the signal wire
- 48. Add any desired Scope (under xPC Target/Misc)
- 49. The only known reliable way to log data with xPC Target is by the use of an outport (under Simulink/Sinks) on the highest level of the Simulink model

- a. In the subsystem containing the signal you want to log, add an outport and connect the signal to the outport
- b. If there are more than one signal to log, connect them to a Mux block (under Simulink/Signal Routing), which is then connected to an outport
- c. Exit the subsystem and add another outport in the next highest level, and connect the signal(s) from the subsystem to the new outport. Continue this all the way up to the highest level, and terminate the signal in a final outport

A.1.4 Control of the ABC-150 from an embedded computer – requires additional PC running Windows, hereafter known as "Link," equipped with serial port, with Borland C, LabView or Visual Basic – requires AeroVironment serial port driver: PPSD.EXE

- 50. In the Simulink library, open the xPC Target/UDP sublibrary
- 51. Add to the Simulink model a Send block connected to a Pack block, connected to either a Voltage or Current signal
- 52. The signal will typically be a 'double' value, so the Pack block shouldn't need to be changed
- 53. Set the IP address setting of the UDP Send Binary block to the IP address of the Link computer
- 54. The "simnotebook2" computer is already set up as a Link computer, IP

128.118.33.47

a. A Visual Basic application on the Link computer, labeled UDPReceive, receives a UDP current signal from an embedded computer using the

- i. To send a voltage command, modify UDPReceive in Visual Basic so that it commands voltage instead of current and recompile
- b. To operate, turn on the ABC-150 and go through the start-up procedures
- c. Double-click the icon on the desktop of the Link computer for UDPReceive
- d. Click the Start button and wait for an audible click from the ABC-150
- e. Start the xPC Target application on the embedded computer
- f. After the xPC Target application has stopped, click the Stop button of UDPReceive

A.2 Setup of Advantech UNO-3072 computers to boot xPC Target from CompactFlash

A.2.1 Creation of and bootup from a DOS floppy

- 1. Connect VGA monitor, PS/2 keyboard, and USB Floppy drive to the UNO-3072.
- Place CompactFlash memory card in UNO-3072 (preferably not larger than 2 GB).
- 3. Turn on the computer and hold down the key to enter BIOS setup.
- 4. Configure BIOS with the following settings:
 - a. Onboard Device -> USB Controller [Enabled]
 - b. First Boot Device [USB-FDD]

- c. Second Boot Device [HDD-0]
- d. Third Boot Device [CDROM]
- e. Boot Other Devices [Enabled]
- 5. Save BIOS changes.
- 6. Reboot computer with a FreeDOS boot disk in the floppy drive.
 - a. At boot option prompt, select option [4] Boot from floppy (no install)

VI. Creation of the xPC Target DOS Loader disk – requires xPC Target

- 7. Enter "xpcexplr" at the Matlab command prompt
- 8. Select menu item File/Add Target rename if desired
- 9. Under Configuration, set Target boot mode to DOSLoader
- Under communication, set up the communication protocol TCP/IP required for control of ABC-150
- 11. The following settings are for an Advantech computer from a PTI laptop
 - a. If Ethernet cable is a patch (crossover) cable directly from Host to Target, set the Target PC IP address to something similar to that of the Host, e.g. 128.118.33.65
 - b. Set the LAN subnet mask address to 255.255.255.0
 - c. Set the TCP/IP gateway address to 128.118.33.1
 - d. Set the TCP/IP target driver to R8139
 - e. Under Settings, set the Maximum model size to 4MB
 - f. Check the box for Enable Secondary IDE

- Insert a floppy in the computer drive if the computer has no floppy drive use a USB floppy drive
- 13. Under Configuration, click the button Create Bootdisk

A.2.2 Configuration of CompactFlash with xPC Target Boot Kernel (After booting with the DOS floppy)

- 14. Run FDISK to create a DOS partition on the CompactFlash
 - a. Create 1 FAT-16 primary partition.
- 15. Format (and create the system) partition:
 - a. FORMAT C: /S
- 16. Enter DIR to put DOS commands in memory
- 17. Remove the DOS boot disk from the floppy drive.
- Insert the xPC Target DOS Loader disk (created using xpcexplr) in the floppy drive.
- 19. Copy all files on the floppy to the C: drive:
 - a. COPY *.* C:
- 20. Shut down the computer and disconnect the floppy drive.
 - a. Try rebooting the computer. If the system boots fine proceed to then skip a-g.
 - b. If boot hangs at "Verifying DMI pool" then plug in the floppy drive and insert the boot disk.
 - c. At boot option prompt, select option [4] = Boot from floppy (no install)
 - d. Run fdisk /mbr

- e. Shut down the computer and disconnect the floppy drive.
- f. Turn on the computer and press F1 when prompted.
- g. Shut down the computer and proceed.
- 21. Turn on the computer and hold down the key to enter BIOS setup.
- 22. Configure BIOS with the following setting:
 - a. Onboard Device -> USB Controller [Disabled]
- 23. Save BIOS changes.
- 24. Reboot computer to load xPC Target kernel.

A.2.3 Operation of the embedded computer, hereafter known as Target – requires a PC with xPC Target, hereafter known as Host

- 1. Host and Target need to be connected by an Ethernet cable
- 2. Enter "xpcexplr" at the Matlab command prompt
- 3. Boot the Target
- 4. Select the desired target in the xPC Target Explorer
- 5. Select the menu item Target/Connect to Target
- 6. Change the Matlab current directory to the one containing the desired DLM file
- 7. In xPC Target Explorer, drag the DLM file onto the name of the Target
- Select menu item Application/Start Application or click the Play button to begin the simulation
- 9. Select menu item Application/Stop Application or click the Stop button to stop the simulation

- 10. To save logged data from an outport to the Matlab workspace, go to the name of the model under the name of the Target.
 - a. Under the Logging section, check the box for Output.
 - b. If the box cannot be checked, the model has no outports at the highest level.
 - c. Click the button Send to MATLAB Workspace.
- 11. On Advantech computers, if the digital or analog i/o do not appear to be working, try connecting the breakout board cable to the other i/o card on the computer
- 12. Close xPC Target Explorer or disconnect from the target
- 13. Shut down the Target