Design and Hardware-in-the-Loop Testing of Optimal Controllers for Hybrid Electric Powertrains

by

Reza Sharif Razavian

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The main objective of this research is the development of a flexible test-bench for evaluation of hybrid electric powertrain controllers. As a case study, a real-time near-optimal powertrain controller for a series hybrid electric vehicle (HEV) has been designed and tests.

The designed controller, like many other optimal controllers, is based on a simple model. This control-oriented model aims to be as simple as possible in order to minimize the controller computational effort. However, a simple model may not be able to capture the vehicle's dynamics accurately, and the designed controller may fail to deliver the anticipated behavior. Therefore, it is crucial that the controller be tested in a realistic environment. To evaluate the performance of the designed model-based controller, it is first applied to a high-fidelity series HEV model that includes physics-based component models and low-level controllers. After successfully passing this *model-in-the-loop* test, the controller is programmed into a rapid-prototyping controller unit for *hardware-in-the-loop* simulations. This type of simulation is mostly intended to consider controller computational resources, as well as the communication issues between the controller and the plant (model solver). As the battery pack is one of the most critical components in a hybrid electric powertrain, the *component-in-the-loop* simulation setup is used to include a physical battery in the simulations in order to further enhance simulation accuracy. Finally, the *driver-in-the-loop* setup enables us to receive the inputs from a human driver instead of a fixed drive cycle, which allows us to study the effects of the unpredictable driver behavior.

The developed powertrain controller itself is a real-time, drive cycle-independent controller for a series HEV, and is designed using a control-oriented model and Pontryagin's Minimum Principle. Like other proposed controllers in the literature, this controller still requires some information about future driving conditions; however, the amount of information is reduced. Although the controller design procedure is based on a series HEV with NiMH battery as the electric energy storage, the same procedure can be used to obtain the supervisory controller for a series HEV with an ultra-capacitor.

By testing the designed optimal controller with the prescribed simulation setups, it is shown that the controller can ensure optimal behavior of the powertrain, as the dominant system behavior is very close to what is being predicted by the control-oriented model. It is also shown that the controller is able to handle small uncertainties in the driver behavior.

Acknowledgments

I would like to thank my supervisors, Professor Azad and Professor McPhee who guided me throughout this work. I would also like to thank Dr. Thomas Uchida for his invaluable helps.

I'm also grateful of my friends, Mr. Taghavipour, Mr. Mehrabi, and Mr. Vajedi who helped me push my research ahead whenever it was stuck.

Dedication

To my family..

Table of Contents

List of Tables

List of Figures

Nomenclature

α	Engine constant
β	Engine constant
η	Sliding mode tuning parameter
η_e	Engine thermal efficiency
η_i	Battery over-voltage
η_m	Total driveline efficiency
η_v	Volumetric efficiency
γ	Road slope
γ_a	Air heat capacity
λ	Costate
μ	State of charge inequality coefficient
ω	Engine speed
ϕ	stoichiometry fuel-air ratio
ϕ_i	Battery electromotive force

π_1	Pi group related to battery voltage
π_2	Pi group related to battery capacity
π_3	Pi group related to battery resistance
π_4	Pi group related to battery resistance
ρ	Air density
θ	Throttle angle
θ_0	Closed throttle angle
A	Vehicle frontal area
a	Filter coefficient
A_i	Battery geometry parameter
a_i	Battery geometry parameter
a_x	Vehicle acceleration
A_{thr}	Throttle area
C_D	Throttle discharge coefficient
C_d	Aerodynamic drag coefficient
D	Diameter of the output throttle vent
d	Diameter of the input throttle vent
E_n	Regenerative energy in a drive cycle
E_p	Positive energy demand in a drive cycle

E_{res}	Change in battery energy with respect to reference
F	Faraday constant
f_D	Aerodynamic drag force
f_R	Rolling resistance force
f_T	Traction force
f_{rr}	Rolling resistance coefficient
G	State of charge inequality function
g	Gravitational acceleration
$G1$	Voltage scaling gain for power supply
$G2$	Current scaling gain for power supply
$G3$	Current command scaling gain for the power supply
H	Hamiltonian
H_f	Gasoline heat of combustion
I	Current
i	Battery current
I_{BP}	Full-size battery pack current
i_{cell}	Individual battery cell current
I_{GAIA}	Battery current in CIL setup
J	Cost function

j_i	Battery specific current
l_i	Battery geometry parameter
m_v	Vehicle mass
N_{cyl}	Number of cylinders of the engine
N_{eng}	Two/four stroke number
P	Power
P_b^*	Optimal battery power
P_{gen}^*	Optimal generator power
P_0	Atmospheric pressure
P_b	Battery power
P_d	Power demand at the wheels
P_e	Electric power demand
P_m	Manifold pressure
P_{BP}	Battery power in full-size battery pack
P_{GAIA}	Battery power in CIL setup
P_{gen}	Generator output power
P_{ind}	Engine indicated power
P_{load}	Engine internal load power
P_{loss}	Engine power loss

P_{UC}	Ultra capacitor power
PRI	Subsonic/supersnic number
Q	Battery capacity
Q_{BP}	Nominal full-size battery pack capacity
Q_{COM}	Battery pack capacity in control-oriented model
Q_{GAIA}	Nominal battery capacity in CIL setup
Q_{ID}	Identified battery capacity in CIL setup
R	Battery equivalence series resistance
R_g	Gas constants
R_{BP}	Full-size battery pack resistance
R_{COM}	Battery pack resistance in the control-oriented model
R_{GAIA}	Battery resistance in CIL setup
R_{ID}	Identified battery resistance in CIL setup
R_{int}	Battery ohmic resistance
S	Sliding surface
SoC	Battery state of charge
T	Battery cell temperature
t	Characteristic time
T_0	Atmospheric temperature

T_m	Intake manifold temperature
t_p	Total time when power demand is positive
t_{BP}	Characteristic time in the full-size battery pack
t_{GAIA}	Characteristic time in CIL setup
u	Controller output
u^*	Optimal controller output
u_{max}	Maximum battery power
u_{min}	Minimum battery power
V	Voltage
V_C	Ultra capacitor voltage
V_d	Engine displacement volume
V_m	Intake manifold volume
v_x	Vehicle velocity
V_{BP}	Nominal full-size battery pack voltage
V_{cell}	Individual battery cell voltage
V_{COM}	Battery pack voltage in control-oriented model
V_{GAIA}	Nominal battery voltage in CIL setup
V_{ID}	Identified battery voltage in CIL setup
V_{oc}	Battery open-circuit voltage

x	System state (battery state of charge)
x_{max}	Maximum battery state of charge
x_{min}	Minimum battery state of charge
x_{ref}	Reference value of battery state of charge
$\dot{m}$	Engine fuel consumption rate
$\dot{m}_e$	Manifold out flow rate
$\dot{m}_{thr}$	Throttle mass flow rate
$\bar{u}$	Optimal controller constant
STS	Stop-to-stop cycle
$\mathcal{S}$	Battery power equivalence factor

Chapter 1

Introduction and Background

The automotive industry is endlessly striving to improve vehicle design. One of the main topics of interest in vehicle design is fuel economy. Although Electric Vehicles (EVs) show strong potential to become the anticipated *green* vehicles, the automotive industry is still facing difficulties in developing EVs. High price and limited drive range are the major drawbacks of these vehicles. Although the improvement of electrical storage technologies may overcome these drawbacks, we cannot expect to have so many EVs on the roads in the near future. Instead, as a short-term solution for the problems of fuel consumption and emissions, Hybrid Electric Vehicles (HEVs) are among the best candidates.

To drive a hybrid electric vehicle, the internal combustion engine (ICE) is assisted by the electric driveline. This assistance can be in various ways, as shown in Figure 1.1. Regardless of the architecture, the source of energy in HEVs is still fossil fuel, yet a reduction in fuel consumption and emissions can be achieved for the following reasons:

1. In an HEV, it is possible to use a smaller (and more efficient) engine due to the assistance of the electrical propulsion systems.

2. The electrical storage in an HEV can be charged using an external source (in plug-in HEVs), which reduces the fuel consumed.

3. The presence of an electrical path in the powertrain allows the vehicle to capture part of its kinetic energy during braking and reuse it when needed.

4. Extra degrees of freedom in the powertrain allow the engine to work at higher efficiency points.

HEVs have proven to be more fuel efficient than conventional vehicles. However, higher fuel economy cannot be achieved without an intelligent plan (the so called *supervisory controller*) to decide on the power flow in the hybrid electric powertrain. Design and testing of such optimal supervisory controllers has been an interesting research topic in the past decade. The objectives of this research fall into this category: design and evaluation of an optimal, real-time supervisory controller for a series hybrid electric vehicle.

1.1 HEV Optimal Control Problem in the Literature

There are major challenges in designing an optimal HEV powertrain controller. First, the complexity of the system under control presents a challenge, and second, the uncertainty associated with the system input (i.e., the driver commands) increases this difficulty. The powertrain controller should command each component in such a way that the fuel consumption and/or emission is minimized while the driver command is followed, and the physical constraints of the system are not violated. In the early stages of the development of HEVs, rule-based supervisory controllers were used; these plans, although being robust and simple to implement, do not necessarily result in optimal behavior and are difficult to tune.

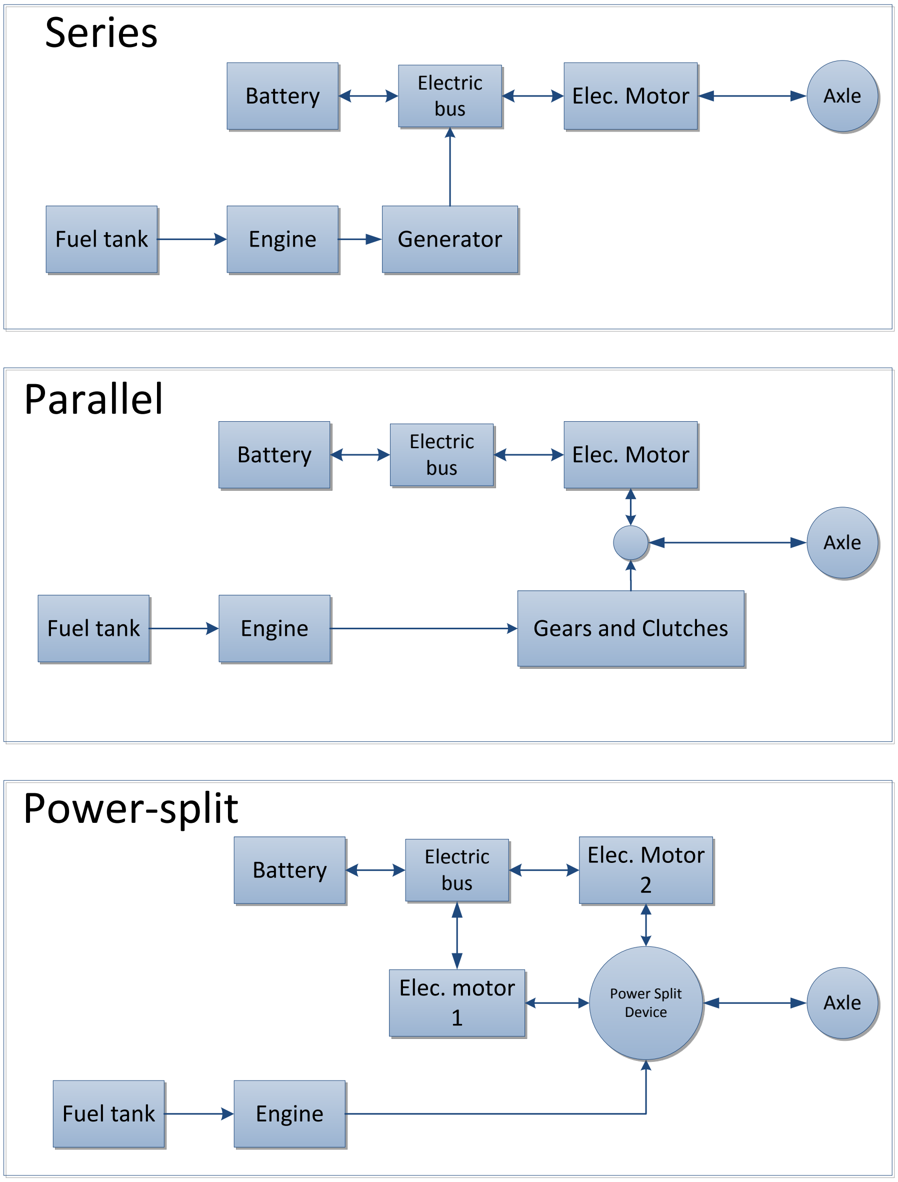


Figure 1.1: Different configurations in hybrid electric powertrains

Studies show that even a small reduction of 3% in HEVs fuel consumption will save at least 6.5 million gallons of gas annually in the United States [1]. This has been the motivation for many researchers to approach model-based controllers in recent years, as these controllers have the potential to provide much higher fuel economy compared to rule-based controllers [2]. Application of model-based controllers is widely studied in the literature, and numerous methods have been presented to find the optimal supervisory controller.

Among these model-based controllers are brute-force numerical methods that provide the global optimal solution to the problem. Dynamic Programming [3]-[7], Particle Swarm Optimization (PSO) [8], and Genetic Algorithms (GA) [9] are examples of such methods. Since these methods require exact information about the driver command for the entire trip, which is unknown for practical applications, they cannot be used in real-time applications. Moreover, these methods are computationally costly, and are optimal only for the drive cycle they are solved for. On the other hand, their guaranteed globally optimal solution can be used as a benchmark for other controllers.

To design near-optimal controllers that can be used in real-time, Stochastic Dynamic Programming (SDP) [10, 11], Game Theory (GT) [12]-[14], and Model Predictive Control (MPC) [15]-[17] are used. These controllers can provide sub-optimal yet satisfactory results [16]; however, they still rely on some information about the driving conditions, such as statistical data (SDP) or short term drive cycle (MPC). Moreover, these methods are still computationally costly and require special approaches to reduce the computational time. An example of such an approach is the combination of analytical and numerical methods [4].

Besides the numerical methods mentioned above, analytical optimal control methods are useful in HEV powertrain controllers as well. Pontryagin's Minimum Principle (PMP) is a branch of optimal control theory, in which the minimization of the integral cost (total fuel consumption) is reduced to local minimization of the Hamiltonian. The Hamiltonian is formed by augmenting the integral cost with a set of Lagrange multipliers (or the *costates* in optimal control theory) and state equations [18]. The reduction from the integral cost minimization to the instantaneous Hamiltonian minimization is the key feature in Pontryagin's Minimum Principle that makes it a useful approach for developing HEV powertrain controllers. Since the formulation of the PMP presents an analytical procedure for this reduction, the minimization process would be much faster than the numerical methods mentioned above.

Many works in the literature have shown the usefulness of the PMP in HEV applications [3], [19]-[22]; however, optimality of the solution strongly depends on the costate values. The effect of the value of the costate is shown in [19]-[23]. In fact, the costates are the parameters that must be tuned based on the drive cycle, and their value will change from one drive cycle to another. Thus, the costates can only be calculated precisely if the whole drive cycle is known in advance. The dependency of the costate on the drive cycle has been the most important drawback of this approach. Estimation of the costate without exact knowledge of the drive cycle is an ongoing research topic, and one of the contributions of this thesis falls in this category.

Lastly, the Equivalence Consumption Minimization Strategy (ECMS) is a heuristic yet promising method for HEV controller design [3], [5], [24]-[26]. In ECMS, the battery power is interpreted as an imaginary fuel consumption rate by using an equivalence factor: $\mathcal{S}$. Then the optimal control at each instant, u^*, is chosen so that the total fuel consumption rate in (1.1) is minimized.

$$u^* = \arg\min \{\dot{m} + \mathcal{S}P_{battery}\} \tag{1.1}$$

In this relation, $\dot{m}$ and $\mathcal{S}P_{battery}$ are the actual and the imaginary fuel consumption rates, respectively. It has been shown that if the value of the equivalence factor is chosen properly, the outcome of the ECMS is optimal [3], [25]. In these cases, the total fuel consumption in (1.1) is similar to the Hamiltonian introduced by the PMP, and the equivalence factor is tightly related to the costates. Thus, the optimal behavior of the controller is based on finding an optimal equivalence factor.

Regardless of the method chosen for designing the powertrain controller, the designed controller has to be evaluated in a realistic environment. It is obvious that applying a newly-designed controller to a real vehicle is unwise. Such an evaluation is extremely costly, and may be even unsafe. Therefore, controller testing platforms need to be employed to evaluate the controller, before it can be applied to the real system.

1.2 Hardware-in-the-loop Simulation Setups in the Literature

Software simulation is the most widely used test method for controllers. In software simulations, a high-fidelity model is used to emulate the real system. However, some features of the real control loop cannot be studied by just software simulations. To further investigate the details of a control loop and enhance the simulation fidelity, the completion of a hardware-in-the-loop simulation is recommended.

Unfortunately, there is no consistent definition for hardware-in-the-loop simulation among researchers. As soon as a piece of hardware (either a controller or a physical component) is included in the simulation loop, it may be called a hardware-in-the-loop simulation. In this thesis, however, it is preferred to denote different types of simulation with different names.

Within this study, the simple software simulation that only includes the controller and the high-fidelity model is referred to as the *model-in-the-loop (MIL)* simulation. In this type of test, all simulations are done in one computer; hence, real-time issues, such as controller-plant communications are ignored. The schematic of the different simulation setups are displayed in Figure 1.2.

The method of simulation in which the controller code is programmed into an individual processing unit (the so-called rapid prototyping controller) is called a *hardware-in-the-loop (HIL)* simulation. In this type of test, the high-fidelity model of the plant is solved in

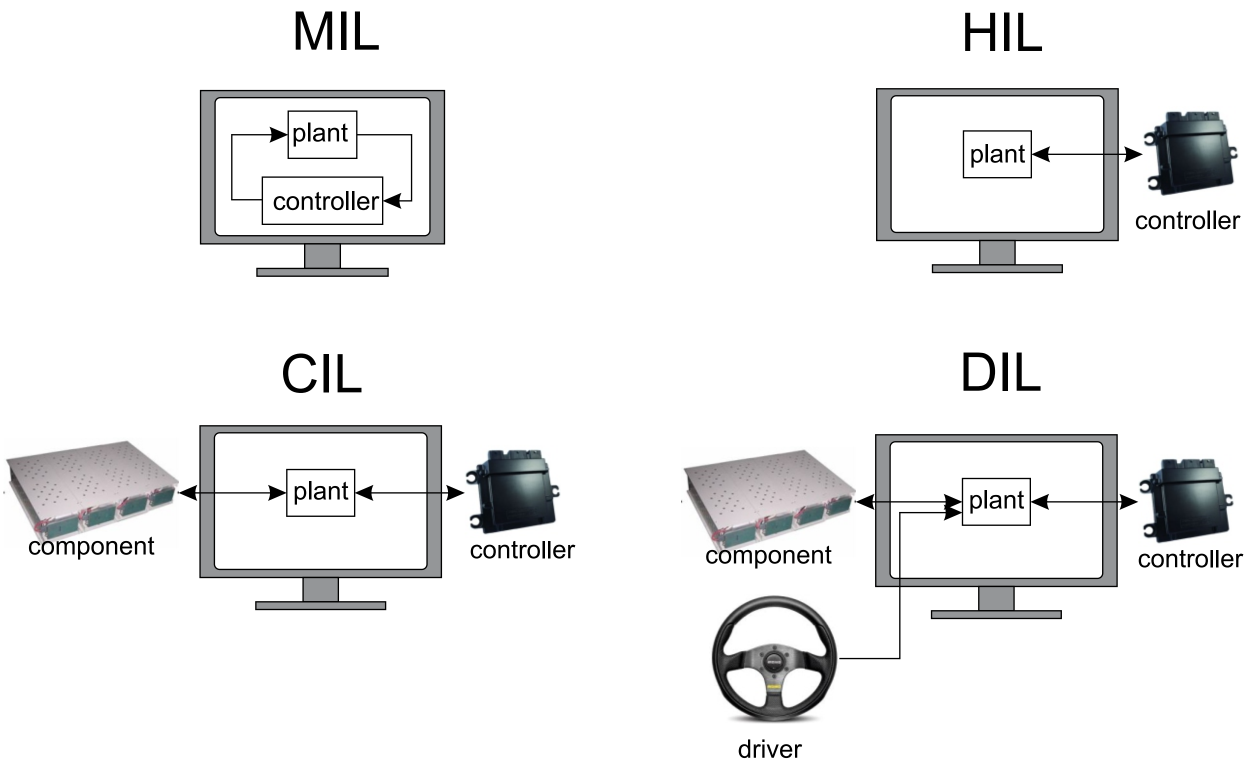


Figure 1.2: Schematic of the different simulation setups

real-time using a powerful computer. Since the simulations are done in real-time, they make it possible to see the effects of the limitations in the controller computational resources. Moreover, the effects of communication between the controller and the plant can be studied. To obtain more accurate simulation results from this simulation setup, the controller hardware and the plant-controller communication medium are most beneficial to be similar to the real-systems.

The next simulation setup that is studied in this research is the *component-in-the-loop (CIL)* test-bench. The CIL setup is intended to include physical components (such as the battery or the electric machines) in the control loop in order to enhance the fidelity of the simulations.

Lastly, *driver-in-the-loop (DIL)* is a simulation in which the human driver inputs are received by a driver interface, and the commands are sent to the simulations. With this setup, it is possible to study the effects of the human driver behavior (such as random changes in speed resulting from variations in traffic) on the performance of the powertrain controllers.

In HEV applications, hardware-in-the-loop (in general term) is a popular topic. In [27]-[29], HIL simulations have been used to evaluate designed controllers for a variety of vehicle types such as HEV, hybrid fuel-cell/battery vehicle, and plug-in HEV.

The CIL simulation setups, however, can be used in a wider range of applications. For example, in [30] and [31], the CIL simulations are used to validate the controller designed for fuel cell and electric machine, respectively, whereas in [32]-[35], the CIL simulation setup has been used for different applications such as component sizing and model verification, feasibility study, and HEV controller validation.

One of the important features of the CIL setups is the size of the components. In many of the CIL setups, the components used are not the same size as the component in the real system; therefore, the simulations have to be designed in such a way to consider the scaling of the components. In [36], the components in the HEV powertrain can be scaled to an arbitrary size using Buckingham's Pi theorem. The same procedure will be employed in chapter five for component scaling in our CIL simulation setup.

1.3 Thesis Outline

This thesis includes the remaining six chapters. In chapter 2, the design procedure and tuning method for a series HEV powertrain controller is presented. Chapters 3-6 are related to controller testing procedures. Chapter 3 discusses the model-in-the-loop simulations for the series HEV controller evaluation, and chapter 4 presents the details of the hardware-in-the-loop simulation setup. In chapter 5, the details of the component-in-the-loop setup and its application in HEV controller design and evaluation are presented. The last test, the driver-in-the-loop simulation, is discussed in chapter 6. Finally, chapter 7 presents the conclusion of this thesis and future works.

Chapter 2

Optimal Power Management Controller for a Series HEV

> Most of the materials in this chapter have been published in [37] or are submitted for publication in [38].

Due to the redundancy in hybrid electric powertrains, there has to be a supervisory controller to decide how much power each component has to produce. As each component in the HEV powertrain behaves differently in different conditions, the optimality of the system strongly depends on the amount of effort each component is applying. Model-based controllers have the potential to deliver the optimal solution, provided that the control-oriented model captures enough information about the powertrain. This chapter presents the design of an optimal model-based controller for a series HEV powertrain controller.

2.1 Control-Oriented Model

The mathematical representation of the hybrid powertrain is the core of the model-based controller design. It is essential that this *control-oriented* model be simple enough so

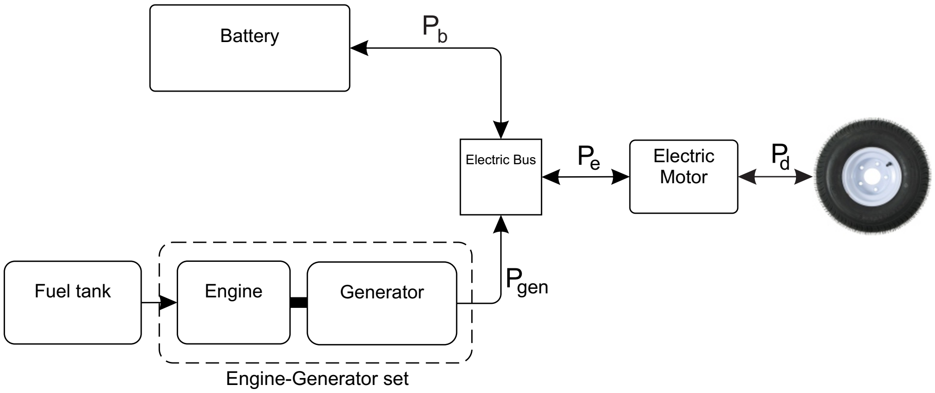


Figure 2.1: Schematic of a series HEV

that the computation time remains within real-time requirements. At the same time, this model should be able to represent the vehicle accurately enough to capture important characteristics of the powertrain.

In this work, a series HEV is studied (Figure 2.1). For designing the controller, a backward quasi-static model for the powertrain is used to calculate the required power based on the vehicle's velocity. This power is then used as the input to the optimization problem.

The longitudinal vehicle dynamics is modeled as (2.1).

$$m_v a_x = f_T - (f_D + f_R + m_v g \sin(\gamma)) \tag{2.1}$$

To find the power demand, (2.2) is used with f_D and f_R, defined in (2.3) and (2.4), respectively.

$$P_d = v_x f_T = v_x(m_v a_x + f_D + f_R + m_v g \sin(\gamma)) \tag{2.2}$$

$$f_D = \frac{1}{2}\rho v_x^2 A C_d \tag{2.3}$$

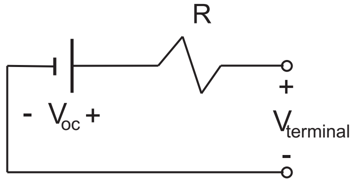


Figure 2.2: Battery circuit model

$$f_R = m_v g \cos(\gamma) f_{rr} \tag{2.4}$$

In the above equations, m_v is the vehicle mass; v_x and a_x are the longitudinal velocity and acceleration respectively; f_T is the traction force, resulting from the torque on the wheels; f_D is the aerodynamic drag force; f_R is the equivalent rolling resistance of all wheels, f_{rr} is the rolling resistance coefficient, and the term $m_v g \sin(\gamma)$ is the resistive force due to the slope of the road, γ. ρ, A, and C_D are air mass density, vehicle frontal area, and drag coefficient, respectively. Numerical values for all the parameters used in this study are given in Appendix A.

To model the hybrid powertrain, quasi-static models of each component are used [22]. The following sub-sections present the model of each component.

2.1.1 Nickel Metal Hydride Battery

As a simple realization for control purposes, a circuit model can be used for the battery modeling, Figure 2.2. In HEV applications, the battery works in a narrow window of state of charge, typically between 50% to 70%. Therefore, the change in the battery voltage (V_{oc}) is negligible, and V_{oc} can be considered constant. Figure 2.3 shows the simulation results for an accurate chemistry-based NiMH battery model [39], which justifies the assumption of constant V_{oc} for this simple model.

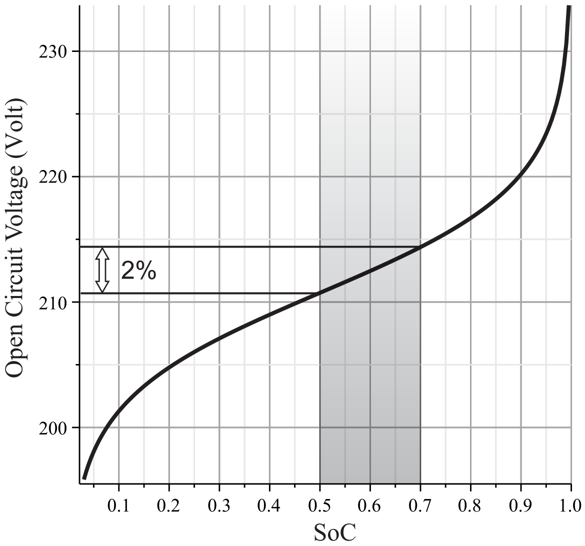


Figure 2.3: Simulation results for the open circuit voltage versus the state of charge in a NiMH battery pack [40]

Considering Q as the battery capacity and i as the current passing through it, the derivative of state of charge (SoC) can be written as:

$$S\dot{o}C = -\frac{i}{Q} \tag{2.5}$$

The negative sign in (2.5) states that a positive current discharges the battery, and a negative current charges it. For the simple model of Figure 2.2 the battery terminal power, P_b, is found using (2.6).

$$P_b = iV_{oc} - Ri^2 \tag{2.6}$$

In the above relation, V_{oc} is the battery open circuit voltage, and R is the total of internal and terminal resistances of the battery.

By substituting i from (2.6) into (2.5), the time derivative of the state of charge becomes:

$$S\dot{o}C = \frac{-V_{oc} + \sqrt{V_{oc}^2 - 4RP_b}}{2RQ} \tag{2.7}$$

In this HEV model, the only state is the battery state of charge. The control parameter is chosen to be the battery power, thus:

$$x \triangleq SoC \tag{2.8a}$$

$$u \triangleq P_b \tag{2.8b}$$

$$\dot{x} = \frac{-V_{oc} + \sqrt{V_{oc}^2 - 4Ru}}{2RQ} \tag{2.8c}$$

2.1.2 Engine-Generator

One of the major advantages of the series HEV architecture is that the engine is not mechanically connected to the driveline. Instead, it is coupled to a generator, allowing the engine speed to be chosen arbitrarily so that the engine works in the minimum Brake Specific Fuel Consumption (BSFC) point for every output power. If this minimum fuel consumption rate is plotted versus the generator output power, the outcome is a linear relation. Figure 2.4 shows the simulation results conducted on a mean-value engine model [41] coupled to a permanent magnet DC generator. In such conditions, the fuel consumption rate can be approximated as

$$\dot{m} = \alpha P_{gen} + \beta \tag{2.9}$$

with α and β being constants.

2.1.3 Electric Motor

One or more electric machines are responsible to drive the wheels. These machines can be modeled as power transducers that convert the electrical power to mechanical power and vice versa. Losses in the driveline and the motors can be modeled with a single efficiency using (2.10). It is also assumed that only a fraction of the kinetic energy (50 percent in this

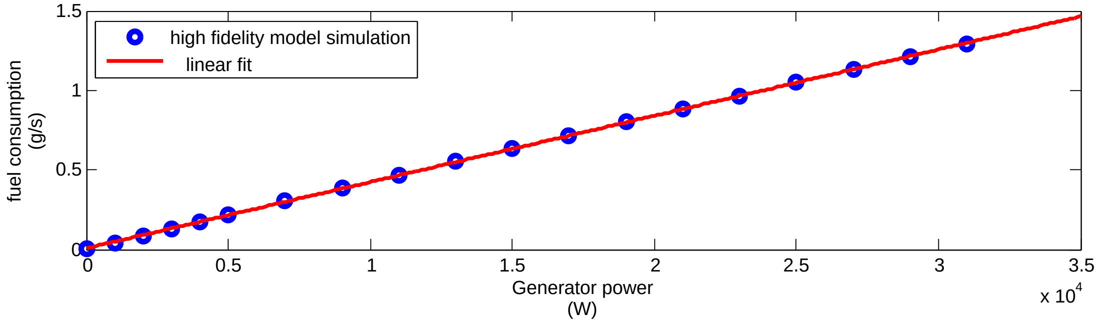


Figure 2.4: Simulation results for the minimal fuel consumption rate versus generator power

study) is restored during regenerative braking. It should be noted that this assumption does not have any effect on the general behavior of the vehicle.

$$\begin{cases} P_e = P_d\ \eta_m^{-1} & P_d > 0 \\ P_e = \frac{1}{2}\left(P_d\ \eta_m\right) & P_d < 0 \end{cases} \tag{2.10}$$

In this equation, η_m is the total efficiency of the driveline from the electric motor to the wheels. Comparisons made between this simple model and a high-fidelity HEV model showed that a constant value of η_m can be accurate enough for control purposes.

2.1.4 Electrical Bus

The electric bus consists of the power electronic drivers for the motor and the generator. Neglecting losses, the balance of the energy in the electric bus can be written as:

$$P_{gen} + P_b = P_e \tag{2.11}$$

In this relation, the positive values indicate that the power is flowing from the powertrain toward the wheels, and the negative sign shows that the power is reversed. It is obvious that the generator power cannot be negative.

Although this model is very simple, with only one state, it will be shown that this model is capable of providing enough accuracy for the purpose of the model-based control.

2.2 Optimal Control Problem

The goal of the supervisory controller in this study is to minimize fuel consumption. Therefore, a cost function of the form (2.12) is considered.

$$J = \int_0^{t_f} \dot{m}\, dt \tag{2.12}$$

By combining (2.6), (2.8), (2.9), (2.10), and (2.11) with (2.12), the cost function can be written as:

$$J = \int_0^{t_f} [\alpha(P_e - u) + \beta]\, dt \tag{2.13}$$

The physical constraints of this powertrain are

$$\begin{aligned} x_{min} &< x < x_{max} && \text{(2.14a)} \\ P_{b_{min}} &< P_b < P_{b_{max}} && \text{(2.14b)} \\ 0 < &\ \ P_{gen} < P_{gen_{max}} && \text{(2.14c)} \end{aligned}$$

It is possible to write (2.14b) and (2.14c) as a single constraint on the control by combining them with (2.11) and (2.6). Therefore, the constraints on the control would be

$$\begin{aligned} u_{min} &= \max\{P_{b_{min}}, P_e - P_{gen_{max}}\} && \text{(2.15a)} \\ u_{max} &= \min\{P_{b_{max}}, P_e, \frac{V_{oc}^2}{4R}\} && \text{(2.15b)} \\ u_{min} &< u < u_{max} && \text{(2.15c)} \end{aligned}$$

The optimal control problem can now be defined as follows: *Find the optimal control, u, such that the cost function* (2.13) *is minimized while the constraints* (2.16) *are satisfied.*

$$
\begin{aligned}
\dot{x} \quad &= \frac{-V_{oc}+\sqrt{V_{oc}^2-4Ru}}{2RQ} && \text{(2.16a)}\\
x \quad &(0) = x(t_f) = x_{ref} && \text{(2.16b)}\\
x \quad &\in \mathcal{X}\ ,\ \mathcal{X} = [x_{min}, x_{max}] && \text{(2.16c)}\\
u \quad &\in \mathcal{U}\ ,\ \mathcal{U} = [u_{min}, u_{max}] && \text{(2.16d)}
\end{aligned}
$$

This is a deterministic optimization problem, since it is assumed that the input (electrical power demand P_e) is known for all $t \in [0, t_f]$. To solve this problem, Pontryagin's Minimum Principle (PMP) [18] can be used. In the PMP formulation, the Hamiltonian is defined according to (2.17).

$$
H = \Big[\alpha(P_e - u) + \beta\Big] + \lambda\left(\frac{-V_{oc}+\sqrt{V_{oc}^2-4Ru}}{2RQ}\right) \tag{2.17}
$$

In this definition, λ is the Lagrange multiplier or the *costate*, with its dynamics defined by (2.18).

$$
\dot{\lambda} = -\frac{\partial H}{\partial x} \tag{2.18}
$$

Since none of the battery parameters (V_{oc}, R, and Q) are assumed to be a function of the state, the costate derivative is zero, and the costate holds its initial value to the end of the mission.

$$
\dot{\lambda} = 0 \tag{2.19}
$$

Pontryagin's Minimum Principle converts the integral minimization of (2.13) to an instantaneous minimization of the Hamiltonian, and states that the optimal control is the one that satisfies (2.20).

$$
u^* = \arg\min_{u\in\mathcal{U}}\{H\} \tag{2.20}
$$

The solution to this problem is the solution of a two-point-boundary-value (TPBV) problem, as the initial and final state values are specified but the initial (and constant) costate value is unknown. The shooting method is a relatively fast and simple method for this TPBV problem, in which the unknown initial conditions are *guessed* and the differential equations are integrated to the final time. If the final values found are close enough to the specified values, the solution has been reached. Otherwise, the initial guess is changed and the whole process is repeated until the final criterion is met within the desired tolerance.

For this method, a discrete-time version of dynamic equations is considered as in (2.21).

$$x[k+1] = x[k] + \frac{-V_{oc} + \sqrt{V_{oc}^2 - 4Ru[k]}}{2RQ} \tag{2.21}$$

The costate, λ, is the value to be guessed in this problem. At each step of integration, and with known values of $x[k]$ and λ, the control range $[u_{min}, u_{max}]$ is identified. In this range, the value of u that minimizes the Hamiltonian is chosen as the optimal control value, $u^*[k]$. Then with this value, (2.21) is integrated (forward Euler integrator) to the next step. This solution continues to the final time, when $x(t_f)$ is found. If $x(t_f) \neq x_{ref}$, the guess for the costate is modified and the whole process is repeated until $x(t_f)$ is close enough to x_{ref}.

Solving for the correct value of the costate is therefore a lower level optimization problem: find the costate, λ, in such a way that $(x(t_f) - x_{ref})^2$ is minimized. This problem can easily be solved using available software packages such as the optimization toolbox in Matlab.

Since at each time the control value is chosen from the range $[u_{min}, u_{max}]$, the control constraint is essentially satisfied. However, considering the state inequality constraint (2.16c) is a more challenging process. This constraint can be written as the inequality $G(x, t) < 0$, with G defined as:

$$G = (x - x_{min})(x - x_{max}) \tag{2.22}$$

Whenever this constraint is active (i.e., $G \geq 0$), the necessary conditions in the PMP must be slightly altered. In such a case, the optimal control value is identified as:

$$u^* = \arg min_{u \in \mathcal{U}} \left\{ H + \mu \dot{G} \right\}, \quad \mathcal{U} = \left\{ [u_{min}, u_{max}] \mid \dot{G} = 0 \right\} \tag{2.23}$$

where μ is a positive number, and $\dot{G}$ is the time derivative of G:

$$\dot{G} = \dot{x}(x - x_{max}) + \dot{x}(x - x_{min}) = \dot{x}(2x - x_{min} - x_{max}) \tag{2.24}$$

$$\dot{G} = 0 \Rightarrow \dot{x} = 0 \Rightarrow u = 0 \Rightarrow \mathcal{U} = \{0\} \tag{2.25}$$

Therefore, the optimal control, u^*, is zero in active constraint regions. The details and proof of this method are available in [42, § 2.5]

This method can be explained heuristically; whenever the state of charge (x) reaches its boundaries, the battery cannot be charged (or discharged), and the admissible control is zero to prevent it from being over-charged (over-discharged).

2.3 Optimal Feedback Control

At every time step, the value of the control in $[u_{min}, u_{max}]$ that minimizes the Hamiltonian is chosen as the optimal value. Since at each time step the state, the costate, and the power (x, λ and P_e in (2.17)) have certain values, the Hamiltonian takes a convex form in terms of the control, u.

$$\frac{\partial H}{\partial u} \quad = -\alpha + \frac{\lambda}{2RQ}\left[-\frac{2R}{\sqrt{V_{oc}^2 - 4Ru}}\right] \tag{2.26}$$

$$\frac{\partial^2 H}{\partial u^2} \quad = -\frac{2\lambda R}{Q}\frac{1}{(V_{oc}^2 - 4Ru)^{\frac{3}{2}}} > 0 \tag{2.27}$$

$$\frac{\partial H}{\partial u} \quad = 0 \Rightarrow \bar{u} \triangleq \frac{1}{4R}\left(V_{oc}^2 - \frac{\lambda^2}{\alpha^2 Q^2}\right) \tag{2.28}$$

In (2.28), it can be seen that the Hamiltonian has only one extremum at $\bar{u}$. For stable shooting method solutions, the costate has to be negative, thus, the second derivative in (2.27) will be positive, resulting in a convex function.

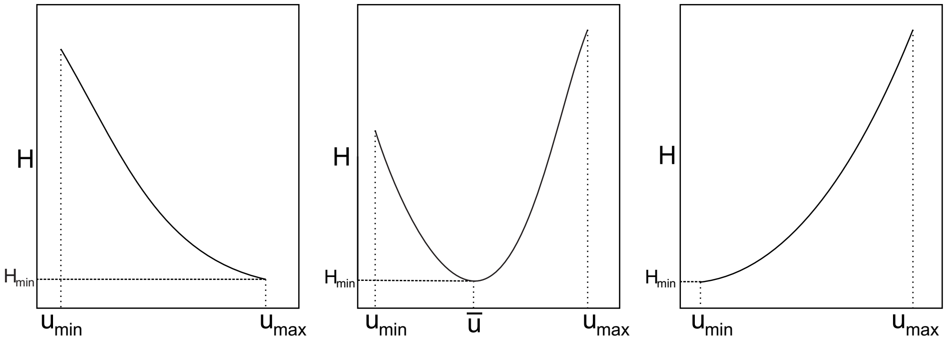
It is worth noting that by using the shooting method, the global optimality of the solution is guaranteed. That is because there is only one solution candidate, and if the choice of the costate satisfies the final boundary conditions, the solution is unique, thus globally optimal.

The quadratic form of the Hamiltonian also implies that the minimum of H happens either at a boundary value of u (namely u_{min} or u_{max}), or when $\frac{\partial H}{\partial u}$ is zero, (2.28). These conditions are shown in Figure 2.5. According to (2.29), $\bar{u}$ has a constant value throughout a mission.

$$\frac{d\bar{u}}{dt} = -\frac{1}{2R\alpha^2 Q^2}\left(2\lambda\dot{\lambda}\right) = 0 \tag{2.29}$$

In fact, $\bar{u}$ is the governing parameter in this problem, and can be found by (2.28), using only battery parameters and the costate value. Therefor, a simple yet optimal feedback controller can be defined as:

$$u^* = \begin{cases} u_{max} & u_{max} < \quad \bar{u} \\ \bar{u} & u_{min} < \quad \bar{u} < u_{max} \\ u_{min} & \qquad\quad \bar{u} < u_{min} \end{cases} \tag{2.30}$$



(a) (b) (c)

Figure 2.5: Hamiltonian versus control, u

with u_{min}, u_{max} and $\bar{u}$ defined in (2.15a), (2.15b) and (2.28), respectively. The inputs to this controller are the state (as feedback) and P_e, and the controller determines the optimal battery power.

The mechanism of the optimal control can be simply explained. When the electric power demand is negative (during braking) and the SoC is within the admissible range, the battery absorbs all the available power. When the power demand is positive, but less than a certain value ($\bar{u}$), the battery provides all the required power. These two conditions correspond to the first case in (2.30), and are shown in Figure 2.6 (a). When the power demand is more than $\bar{u}$, only a portion of the required power is delivered by the battery, and the rest is provided by the generator. This condition corresponds to the second case of (2.30), and is shown in Figure 2.6 (b). Finally, if the power demand is too high (when $u_{min} > \bar{u}$ or equivalently $P_e > P_{gen,max} + \bar{u}$), the battery provides more power than $\bar{u}$ to drive the vehicle. This corresponds to the last case in (2.30), and is shown in Figure 2.6 (c). When the SoC is less than its minimum allowable value, the battery will not provide any power, and when SoC is more than its maximum allowable value, it will not absorb electrical power.

As was mentioned earlier, the most important parameter in this controller is the costate (which in turn determines $\bar{u}$). In the next section, a simple method is presented to find

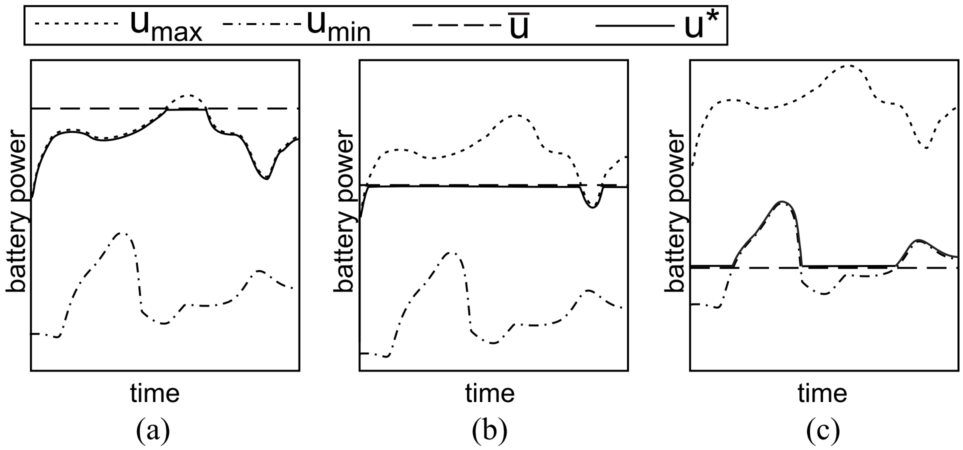


Figure 2.6: The mechanism of the optimal supervisory controller

the proper value of the costate and $\bar{u}$.

2.4 Costate Estimation Method

For optimal behavior, it is necessary to have the future driving condition. Without such information, only sub-optimal behavior is achievable [2]. In this study, it was observed that it is not necessary to consider the whole drive cycle. Instead, if only the driving condition until the next stop (*stop-to-stop* (STS) *cycle*) is known, it is possible to obtain a solution that is almost as optimal as the solution found considering the whole drive cycle. An example is presented in Figure 2.7, with the FTP75 drive cycle shown in the top plot, and the SoC shown in the bottom plot for two different control strategies. To obtain these results, the PMP was solved once for the full drive cycle, and once for successive STS cycles when the final state was required to be x_{ref} at the end of each STS cycle. A comparison of the resultant fuel consumption for various drive cycles is presented in Table 2.1, which shows negligible difference in fuel consumption between full drive cycle and STS cycle optimization.

The costate value is the only parameter that should be tuned for these STS cycles.

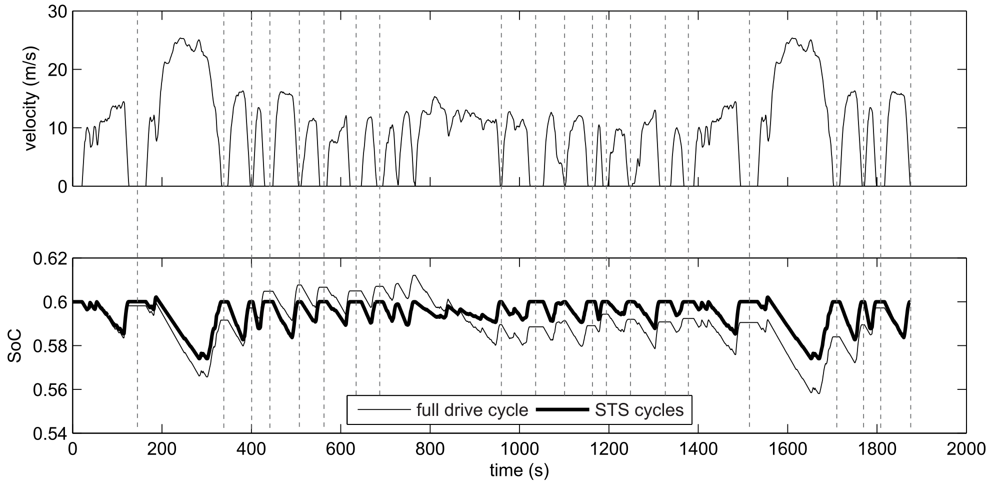


Figure 2.7: State trajectory when considering the full drive cycle, and when considering STS cycles

Since the optimal control mechanism is independent of the driving condition, it is only the charge sustenance that should be considered in tuning the costate.

The mechanism mentioned earlier follows one important concept: it tries to capture as much negative energy as possible, thus elevating the SoC. To discharge the battery to its initial charge level, the controller decides that a *certain amount* of power has to be provided by the battery during acceleration and cruising. This certain amount is $\bar{u}$ found in (2.28).

Table 2.1: Fuel consumption for the two methods

	FTP75	HUDDS	EUDC	NYCC	LA92
Full drive cycle (g/cycle)	274.8	154.4	257.5	31.69	308.5
STS cycles (g/cycle)	275.1	154.6	257.6	31.89	308.8
Increase in fuel consumption	0.10%	0.12%	0.04%	0.63%	0.10%

It should be noted that in both solutions shown in Figure 2.7, the mechanism of the optimal controller is the same, as is the amount of regenerative braking absorbed by the battery. The only cause for the change in fuel consumption is the small change in battery losses. When the whole drive cycle is considered, the optimal control adjusts $\bar{u}$ so that the battery losses are minimized over the whole mission. But when the STS cycles are considered separately, the controller changes $\bar{u}$ for each cycle, thereby increasing the charge/discharge rate of the battery, the battery loss, and finally, the total fuel consumption. It was observed that by changing battery parameters (so that the the battery efficiency is decreased), the difference in fuel consumption between the two methods becomes more noticeable. However, with reasonable battery parameters, the difference is small, as in Table 2.1.

The objective is now to estimate the costate (or equivalently $\bar{u}$) for each of the STS cycles to satisfy charge sustenance. A typical plot of electric power demand, P_e, and the corresponding battery power, u, is shown in Figure 2.8. To have the SoC at the same level at the beginning and end of the cycle, the total change in the battery energy (the integral of the shaded area) must be zero, i.e.:

$$E_n + E_p = 0 \tag{2.31}$$

where E_n represents the total negative energy absorbed by the battery, and E_p is the total energy delivered to the powertrain by the battery.

As was observed in simulations of different standard drive cycles, $\bar{u}$ has a relatively small value (usually less than 5kW). In most of the times when the power demand is positive, it is greater than $\bar{u}$; thus the optimal battery power is equal to $\bar{u}$ in a great portion of the drive time. Therefore it is reasonable to assume the battery power is equal to $\bar{u}$, which leads to

$$E_p = \int\limits_{t_p} u dt \simeq t_p \bar{u} \tag{2.32}$$

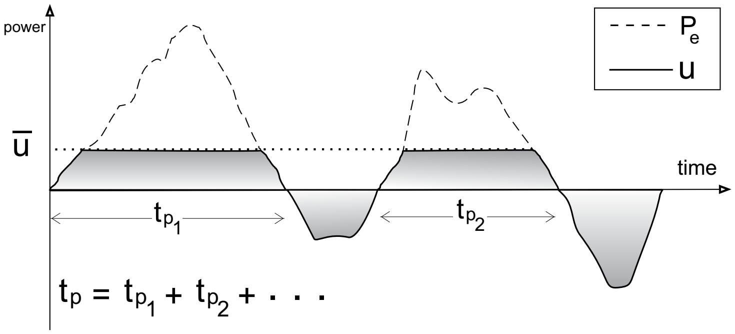


Figure 2.8: A typical plot of electric power

In (2.32), t_p is the time when power demand is positive. Combining (2.32) and (2.31) gives

$$\bar{u} = -\frac{E_n}{t_p} \tag{2.33}$$

If the relations (2.28) and (2.33) are considered together, the value of the costate that satisfies charge sustenance requirements, λ^*, can be found as in (2.34).

$$-4R\bar{u} + V_{oc}^2 = \frac{\lambda^{*2}}{a^2Q^2} \Rightarrow \lambda^* = -aQ\sqrt{V_{oc}^2 + 4R\left(\frac{E_n}{t_p}\right)} \tag{2.34}$$

Relations (2.33) and (2.34) relate the optimal values of λ and $\bar{u}$ to only two parameters of the drive cycle: the total negative energy available and the time when positive power is required. During the simulations, it was observed that the optimal value of $\bar{u}$ is independent of the order of events. For example, it is not necessary to know when the driver is going to push the brake pedal; it is only important to know how much kinetic energy is going to be transferred to electrical energy before the next stop. This behavior can be justified by (2.33), which is only a function of total energy and time.

In equation (2.31), it is assumed that the final SoC should come back to its *initial* level. If (due to any kind of error) the initial SoC has a value different from the desired

SoC_{ref}, then the feedback controller tries to bring it back to the initial value, not SoC_{ref}. To compensate for this error, the difference in the battery energy should be considered as in

$$E_n + E_p = E_{res} \tag{2.35}$$

with E_{res} being the amount of energy that the battery should absorb or release. E_{res} can be estimated using (2.38) by assuming constant V_{oc}.

$$\begin{aligned} E_{res} &= Q \int_{SoC_0}^{SoCref} [dSoC \; V_{oc}(SoC)] && (2.36) \\ E_{res} &\simeq QV_{oc} \int_{SoC_0}^{SOCref} dSoC && (2.37) \\ E_{res} &\simeq QV_{oc} \Delta SoC && (2.38) \end{aligned}$$

Therefore, $\bar{u}$ and λ can more robustly be approximated using:

$$\bar{u} = -\frac{E_n - E_{res}}{t_p} \tag{2.39}$$

$$\lambda^* = -aQ\sqrt{V_{oc}^2 - \frac{4R}{t_p}\left(E_n + Q(x_0 - x_{ref})V_{oc}\right)} \tag{2.40}$$

As mentioned in Chapter 1.1, the ECMS can be optimized using the PMP approach. In this case, the equivalence factor is tightly related to the costate. Therefore, it is possible to find the optimal value of the equivalence factor at each instant using the costate found by the method presented in this paper.

Although the costate estimation method still requires certain information about future driving conditions, it is a less demanding problem than finding the exact speed profile. It is possible to estimate cruise times using ITS and GPS systems; the available negative energy is related to vehicle kinetic energy during braking, which can also be estimated using the longitudinal vehicle dynamics [43].

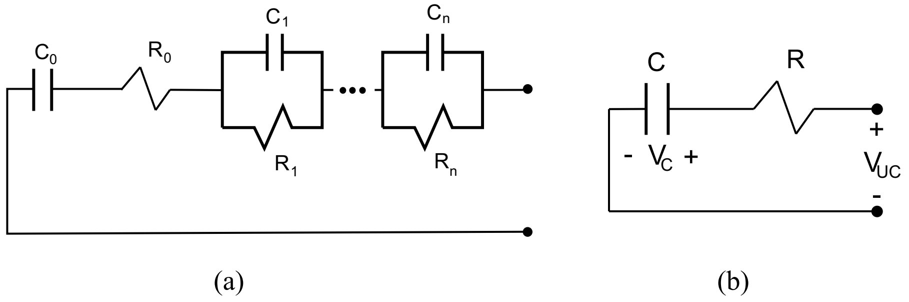


Figure 2.9: Ultra-capacitor circuit model

2.5 Comments Regarding Use of Ultra-Capacitor in HEVs

An ultra-capacitor (UC) can be a good option for an HEV electrical energy storage system, as it has higher efficiency and power-density, and much longer lifetime compared to NiMH and Li-ion batteries. In contrast, an UC's relatively low energy-density and internal energy dissipation make it inappropriate for EV and PHEV applications [44].

UCs can be modeled using RC circuits (Figure 2.9(a)), and the number of the RC branches determines the accuracy of the model [45]. Since increasing the number of the branches increases the number of the states of the system, a simple RC circuit is suitable for the control-oriented model (Figure 2.9(b)). This model can still capture enough details about the UC behavior. For such a model, the relationship between the capacitance voltage, V_C, and the current, i, can be written as

$$\dot{V}_C = -\frac{i}{C} \tag{2.41}$$

with C being the capacitance. Considering the capacitance voltage as the state of the system and the current as the control, the state equation is found to be according to (2.42).

$$\dot{x} = -\frac{u}{C} \tag{2.42}$$

Equation (2.43) defines the UC output power, P_{UC}, in terms of the state and the control.

$$P_{UC} = ux - Ru^2 \tag{2.43}$$

In the above relation, R is the UC equivalent series resistance.

Based on this modified HEV model, the new cost function and Hamiltonian can be written as in (2.44) and (2.45), respectively [37].

$$J = \int_0^{t_f} \left[\alpha(P_e - xu + Ru^2) + \beta\right] dt \tag{2.44}$$

$$H = \left[\alpha(P_e - xu + Ru^2) + \beta\right] + \lambda(-\frac{u}{C}) \tag{2.45}$$

The costate is no longer constant, and is governed by the dynamic equation:

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = \alpha u \tag{2.46}$$

When using an UC, similar to the case that a NiMH battery was used, the Hamiltonian takes a convex form (in the UC case, it is quadratic). Therefore, the arguments regarding uniqueness and global optimality of the solution are still valid. Moreover, $\bar{u}$ (the control value at which the Hamiltonian has a zero derivative) is a constant value:

$$\begin{aligned}\frac{d\bar{u}}{dt} &= \frac{1}{2R}\frac{dx}{dt} + \frac{1}{2R\alpha C}\frac{d\lambda}{dt} && (2.47)\\ &= \frac{1}{2R}(-\frac{u}{C}) + \frac{1}{2R\alpha C}(\alpha u) = 0 && (2.48)\end{aligned}$$

Therefore, the optimal supervisory controller has the same logic, except the fact that the governing parameter, $\bar{u}$, is the UC *current*, not the power. The optimal UC current is always the maximum possible current, u_{max}, except when it is higher than $\bar{u}$, in which case the optimal UC current is $\bar{u}$. Finally, the same method can be used to estimate the costate initial value and $\bar{u}$, based on the same information (i.e., cruise time and regenerative energy).

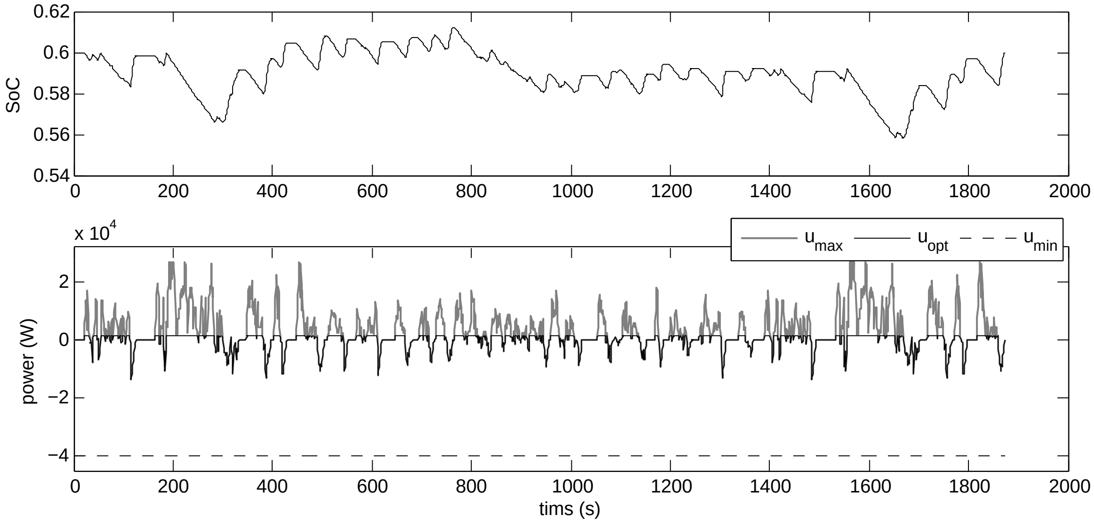


Figure 2.10: The optimal SoC and control trajectories for the FTP75 drive cycle found using the PMP solution

2.6 Controller Design Results

In this study, all simulations are conducted in the Matlab environment. The optimal control problem is solved using the PMP for the FTP75 drive cycle, and control and state trajectories are shown in Figure 2.10. Note that the optimal control value is the same as u_{max} but is limited to a constant value ($\bar{u}$).

For 68 STS cycles listed in Table 2.2, the optimal $\bar{u}$ is found, and the correlation between this parameter and the drive cycle parameters, equation (2.33), is presented in Figure 2.11. It is interesting to observe that the best linear approximation matches the relation (2.33) very well.

In Figure 2.12, the state and the control trajectories obtained by three different methods are shown for a portion of the FTP75 drive cycle. The methods are the PMP solution for the full drive cycle, the PMP solution for successive STS cycles, and the solution of the real-time feedback controller with estimated $\bar{u}$. The difference in the first two solutions is

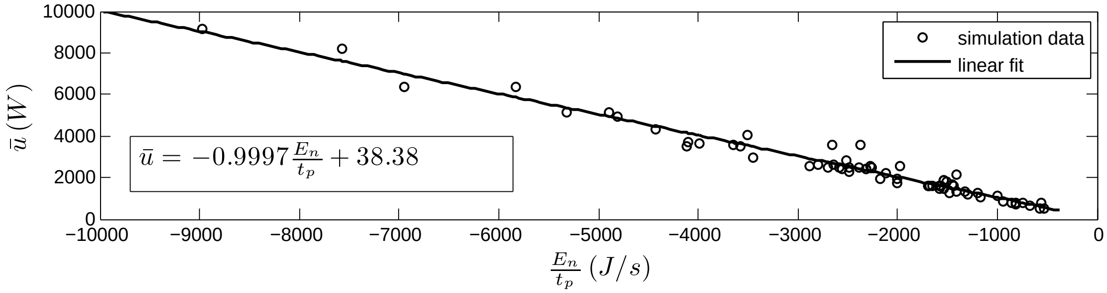


Figure 2.11: Linear fit for $\bar{u}$

natural as the constraints of the problem are different. The more interesting result is the similarity between the solution of the PMP for the STS cycles and that of the the real-time controller. Although many simplifications are made to estimate $\bar{u}$ in the real-time controller, it gives very close results to the near-optimal PMP solution.

The fuel consumption for different standard drive cycles is presented in Table 2.3. The first row of the table presents the optimal fuel consumption for the entire drive cycle, and the second row presents the fuel consumption resulted from the feedback controller with estimated $\bar{u}$ for successive STS cycles. Due to errors in $\bar{u}$ estimation, the final state of charge is not exactly the same as the reference value (the reference value is 0.6). This variation in final state of charge contributes to increase or decrease in total fuel consumption. To

Table 2.2: List of the standard drive cycles used

drive cycle	number of STS cycles	drive cycle	number of STS cycles
UDDS	17	NYCC	10
US06	5	LA92	16
EUC	5	SC03	5
HUDDS	4	JN1015	3
HWFET	1	IM240	2

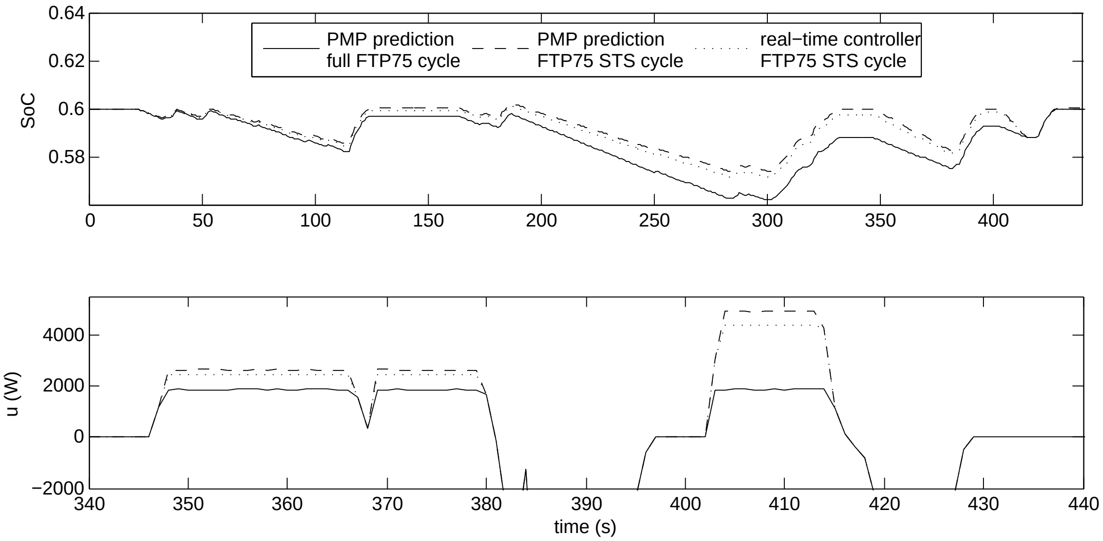


Figure 2.12: SoC and control trajectories for the three different solutions for FTP75

compensate for this error, the fuel consumption is corrected according to (2.49d).

$$\dot{m} = \alpha P_{gen} + \beta \tag{2.49a}$$

$$\int \dot{m} \; dt = \int (\alpha P_{gen} + \beta) dt \tag{2.49b}$$

$$\Delta m = \alpha E_{res} + \beta t \simeq \alpha E_{res} \qquad \text{(assuming short time)} \tag{2.49c}$$

$$\Delta m = \alpha (Q V_{oc} \Delta SoC) \tag{2.49d}$$

2.7 Chapter Summary

In this chapter, development of a near-optimal, real-time controller for a series HEV has been presented. This controller is based on the off-line solution of the optimal control problem, with known inputs. Pontryagin's Minimum Principle has been used to solve the optimal control problem, and the results are shown to be globally optimal. The real-time

Table 2.3: Comparison of fuel consumption for different drive cycles

	FTP75	NYCC	EUC	HUDDS	HWFET
PMP solution for the full drive cycle (g)	274.27	31.520	257.243	154.150	263.272
Feedback controller for the STS cycles (g)	274.68	32.379	256.080	154.368	261.757
Final state of charge	0.6005	0.6034	0.5937	0.6004	0.5925
Corrected fuel consumption (g)	274.577	31.681	257.374	154.286	263.297
Increase in fuel consumption	0.11%	0.5%	0.05%	0.09%	0.01%

controller is then designed based on the mechanism of the optimal controller. The designed controller can be tuned by adjusting only one parameter, which is related to the drive cycle. In the end, a method has been provided to effectively estimate the costate from certain information about the drive cycle.

The simulation results show that the controller in conjunction with the estimation method can be used in real-time, and the final fuel consumption is very close to the optimal value.

Chapter 3

Model-in-the-Loop (MIL) Simulation

Most of the materials in this chapter are submitted for publication in [38].

In the previous chapter, the design of an optimal controller for a series HEV was discussed, and simulation results were presented. However, the presented simulation results were found by simulating the simple control-oriented model. Since the model was simplified to be useful for control development, the results obtained might be different from actual system behavior. Therefore, to see the performance of the controller, a more complicated HEV model should be used. This chapter presents the details of the high-fidelity model-in-the-loop simulations for controller performance evaluation.

3.1 High-Fidelity Model Description

To more accurately evaluate the controller performance, the series HEV is modeled in the MapleSim environment [39]. MapleSim allows acausal modeling, and does symbolic calculations to reduce run times. The MapleSim model is presented in Figure 3.1. As can be seen, the model consists of different components, including the engine, the generator,

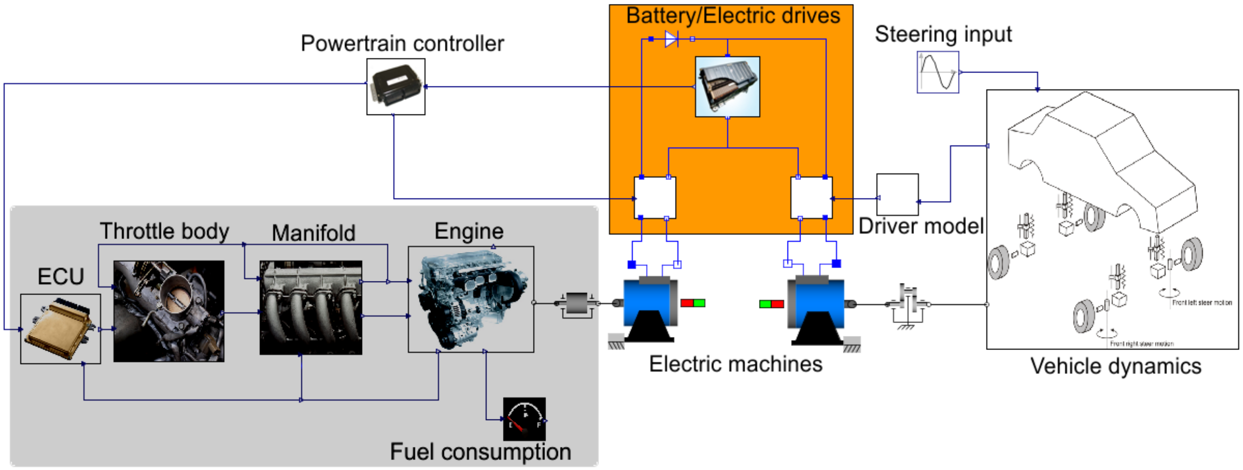


Figure 3.1: High fidelity model for a series HEV in the MapleSim environment

the battery, the electric motor, and a multi-body vehicle. The physics-based models for these components are presented in the following subsections.

3.1.1 Engine-Generator Set Model

This model uses a mean-value engine model [41], which is accurate enough for this application and also is simple enough to allow fast simulations. The engine model, as can be seen in Figure 3.1, consists of four components: the ECU, throttle, manifold, and engine body. The ECU block is responsible for adjusting the throttle angle to make the engine deliver the desired amount of torque. In the throttle and manifold blocks, the pressure and mass flow rate of the air/fuel mixture is calculated according to [46]:

$$\dot{P}_m = \frac{R_g T_m}{V_m} \left(\dot{m}_{thr} - \dot{m}_e\right) \tag{3.1}$$

where P_m is the pressure of the air/fuel mixture, T_m and V_m are the temperature and volume of the intake manifold, respectively, R_g is the gas constant, and $\dot{m}_{thr}$ and $\dot{m}_e$ are

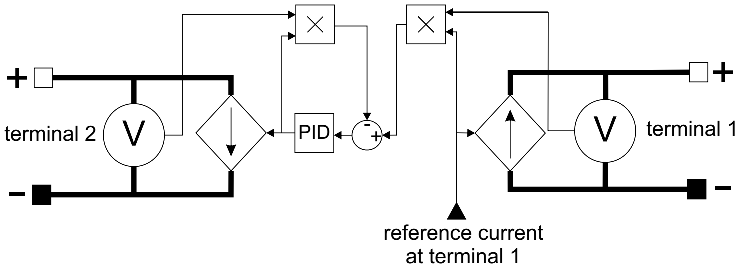


Figure 3.2: Schematic of the DC/DC converter

throttle mass flow rate and manifold out flow respectively.

Having the mass flow rate and pressure calculated, the engine body block calculates the thermal efficiency and indicated power. Finally, the engine output power is found according to:

$$P_{net} = P_{ind} - P_{loss} - P_{load} \tag{3.2}$$

with P_{ind}, P_{loss}, and P_{load} being the indicated power, lost power, and internal load power, respectively.

The engine flywheel is coupled to a permanent magnet DC generator that produces electricity to charge the battery. The effects of power electronics in the generator electric drive are neglected in this model. It is assumed that the electric drive is a DC-DC converter with an efficiency of 100%, [39]. The schematic of the power converter is shown in Figure 3.2. In this system, the power drawn from (or delivered to) terminal 2 is equal to the amount of power at terminal 1. The PID controller ensures that this power is independent of variations in terminal voltage due to changes in generator conditions.

3.1.2 Battery Model

One of the most important components in a hybrid powertrain is the electrical storage. In the model developed here, a chemistry-based NiMH battery [40] is used to further enhance the accuracy of the simulations. In this model, cell terminal voltage is calculated using:

$$v_{cell} = (\phi_c + \eta_c) - (\phi_a + \eta_a) + i_{cell} R_{int} \tag{3.3}$$

where v_{cell} is the terminal voltage, and ϕ_i and η_i are the electromotive force and overvoltage, respectively, which are found based on chemical reactions and ion concentrations in the anode (a) and cathode (c). R_{int} is the ohmic internal resistance, and i_{cell} is the cell current found using:

$$i_{cell} = A_i a_i l_i j_i \tag{3.4}$$

where A_i, a_i and l_i are the geometry parameters. j_i is defined in (3.5), with F and T being the Faraday constant and temperature, respectively. $i_{0,i}$ is also calculated based on the ion concentration at electrodes.

$$j_i = i_{0,i} \left(e^{\frac{0.5F}{RT}\eta_i} - e^{\frac{-0.5F}{RT}\eta_i} \right) \tag{3.5}$$

In (3.4) and (3.5), subscript i should be replaced with c for cathode and a for anode.

3.1.3 Vehicle Dynamics Model

For the electric motor, which is responsible for driving the wheels, a permanent magnet DC machine is used. Similar to the generator, the effects of power electronics in the motor electric drive are neglected. The output shaft of this motor is connected to the wheels via a constant gear ratio.

The vehicle dynamics is simulated by a 14 degree-of-freedom (DOF) multi-body model with Fiala tires and aerodynamic drag force. The degrees of freedom include six DOFs for the body, four DOFs for vertical displacement of the unsprung mass, and four DOFs for rotation of the wheels. This level of complexity of the model is sufficient to predict the steering and longitudinal dynamics of the vehicle. The aerodynamic drag force is simulated using an external load acting on the vehicle's center of mass, and is calculated using (2.3) and the vehicle's longitudinal velocity.

To couple the designed supervisory controller to this high-fidelity model, different components of the model are converted to the Matlab/Simulink environment as S-functions, and are connected together by Simulink signal links. The converted model in Simulink is no longer an acausal model, since different powertrain components are connected together by causal Simulink signal links. The next section presents the overview of the control loop and the way the supervisory controller interacts with other components.

3.2 Low-Level Controllers

The schematic of the converted model can be seen in Figure 3.3. The vehicle dynamics and the electric motor components from MapleSim are converted together as the new vehicle dynamics block in Figure 3.3. The driver model is a simple PID controller that adjusts the motor current to ensure that the vehicle follows the desired speed profile. The outputs of the vehicle dynamics block are the electric power required to follow the speed profile, P_e, and the vehicle speed, which is monitored and used as a feedback for the PID controller. This electric power (consumed by the electric motor or generated during braking) is used as an input to the supervisory controller. It is also used as a signal to determine the charge/discharge rate of the battery.

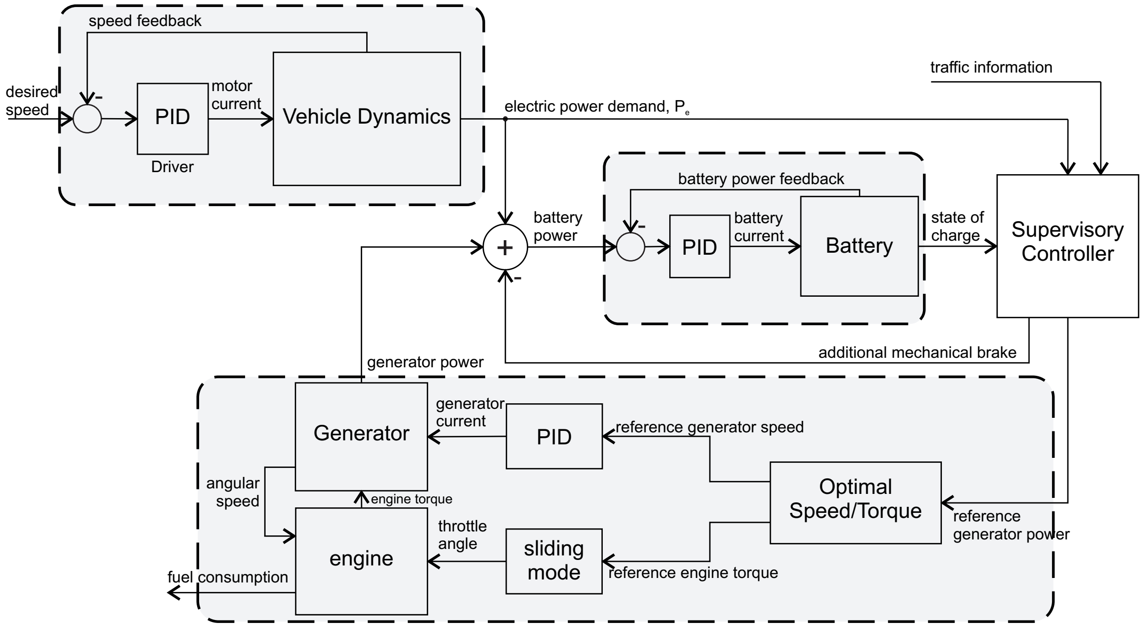


Figure 3.3: Schematic of the causal HEV model used in the control loop

3.2.1 Engine-Generator Set Controllers

The supervisory controller uses the future traffic information as an input to calculate the optimal battery power, and in this study, it is assumed that such information is available in advance for the current STS cycle. Based on the traffic information, instantaneous electric power demand, P_e, and SoC, the supervisory controller can determine the optimal battery power using the logic presented in previous chapter. Then the controller uses (3.6) to calculate the amount of power that the engine-generator set should produce. As long as the low-level controllers for the engine-generator set provide appropriate tracking of the set points, tracking of the optimal battery power is guaranteed:

$$P^*_{gen} = P_e - P^*_b \tag{3.6}$$

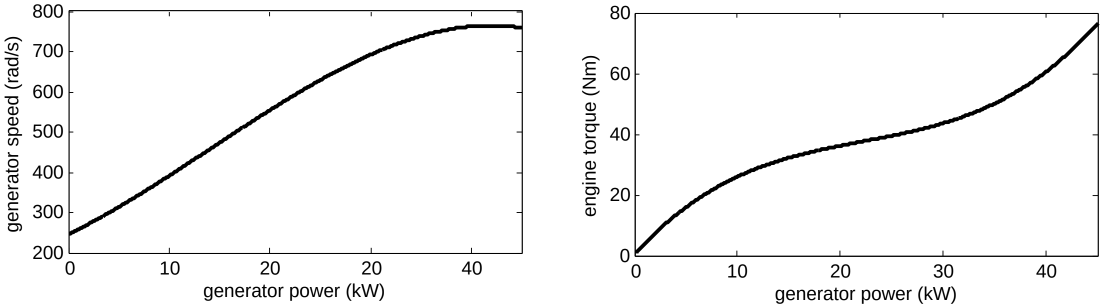


Figure 3.4: Optimal engine torque and generator speed versus generator output power

Based on the optimal generator power, P^*_{gen}, calculated by the supervisory controller, a pair consisting of the torque and speed is identified to give the maximum efficiency of the engine-generator set in that output power. To find the optimal torque/speed pairs, the high-fidelity engine-generator model was simulated a number of times with different pairs of speed/torque that produce a specific output power. Then the pair that consumed the lowest fuel was chosen as the optimal pair for that particular power. Figure 3.4 shows the optimal speed and torque for different output powers.

To make the engine and the generator follow the reference torque and speed, two controllers have been used. Because of the engine's non-linear behavior, a sliding mode controller is used to control the engine torque by adjusting the throttle angle. Moreover, a PI controller is used to adjust the generator current for controlling the speed.

Sliding mode control has proven to be a reliable method for engine torque management in practical cases. This is a model-based method, and is capable of handling the model uncertainties. The main control input for the engine is throttle angle. Other inputs, like air fuel ratio and ignition timing, highly affect the transient behavior of the engine. According to legislation to have the best fuel economy and emission for the engine, we are not allowed to change these two parameters away from their optimum values for a long time [47]. Therefore, throttle angle is generally more reliable and dominating engine input to change the steady state response in this case.

Engine torque management can be done in two ways. The first approach is using an engine torque sensor to measure crankshaft torque for feedback control. This measurement includes the effects of the engine combustion torque, friction torque, pumping torque, and all accessory loads. In this derivation, spark timing and air fuel ratio are used to control the engine torque. Using these two variables as the inputs, forces them away from the optimal ranges, as mentioned above. Thus, torque control is not a legitimate option for the torque management strategy, but it would reduce uncertainties especially in case of engine aging.

The second approach is to measure and control manifold pressure. Engine combustion torque is a function of cylinder air flow which is a function of the manifold pressure. Assuming constant air to fuel ratio and ignition timing, the control goal can be changed to make the manifold pressure follow the desired value. If the throttle is used to control manifold pressure, the effect of ignition timing and air to fuel ratio on combustion torque does affect the throttle control. Thus, the disadvantage of this approach is a larger amount of calibration required to get a proper conversion from desired torque to desired manifold pressure for all engine operating conditions. But, use of this strategy will not require a torque sensor [47].

Here we use the manifold pressure control approach and ignore the effect of transient torque change for the engine set points so we keep the spark timing and air fuel ratio unchanged throughout the simulation.

To accomplish this goal, we need a simple model of the engine to be used for designing the sliding mode controller. A simplified model of the engine is a single input single output model. The manifold pressure, P_m, is the only state variable and engine indicated torque is the output:

$$\dot{P}_m = \underbrace{-\frac{\eta_v N_{cyl} V_d \omega}{60 N_{eng} V_m}}_{A_{eng}} P_m + \underbrace{\frac{R_g T_m}{V_m} \left(C_D \times MA \times PRI\right)}_{B_{eng}} A_{th} \tag{3.7a}$$

$$T_{ind} = \underbrace{-\frac{\phi \eta_e \eta_v N_{cyl} V_d H_f}{60 N_{eng} R_g T_m}}_{C_{eng}} P_m \tag{3.7b}$$

where A_{th} is the throttle area, the control input. Throttle area can be found from the throttle angle, θ, and geometry (d and D are diameter values for input and output vents and θ_0 is the angle when throttle is totally closed):

$$\begin{aligned} A_{th} = \; & - \; \frac{d.D}{2}\sqrt{1-\left(\frac{d}{D}\right)^2} + \frac{d.D}{2}\sqrt{1-\left(\frac{d\cos\theta_0}{D\cos\theta}\right)^2} \\ & + \; \frac{D^2}{2}\sin^{-1}\left(\sqrt{1-\left(\frac{d}{D}\right)^2}\right) - \frac{D^2}{2}.\frac{\cos\theta}{\cos\theta_0}\sin^{-1}\left(\sqrt{1-\left(\frac{d\cos\theta_0}{D\cos\theta}\right)^2}\right) \end{aligned} \tag{3.8}$$

In (3.7), η_v is the volumetric efficiency which is a function of manifold pressure and engine speed, ω. N_{cyl} is the number of cylinders (four in this case). V_d and V_m are the engine displacement and air manifold volume respectively. N_{eng} is 2 for a four-stroke engine. C_D is the throttle discharge coefficient.

Also, $MA = P_0/\sqrt{R_g T_0}$ where P_0 and T_0 are atmosphere pressure and temperature, respectively, and PRI is a non-dimensional value to consider subsonic and supersonic air flow (γ_a is air heat capacity ratio):

$$PRI = \begin{cases} \left(\frac{P_m}{P_0}\right)^{\frac{1}{\gamma_a}} \sqrt{\frac{2\gamma_a}{\gamma_a - 1}\left(1 - \left(\frac{P_m}{P_0}\right)^{\frac{\gamma_a - 1}{\gamma_a}}\right)} & for \quad \frac{P_m}{P_0} > \left(\frac{2}{\gamma_a+1}\right)^{\frac{\gamma_a}{\gamma_a-1}} \\ \sqrt{\gamma_a \left(\frac{2}{\gamma_a+1}\right)^{\frac{\gamma_a-1}{\gamma_a+1}}} & for \quad \frac{P_m}{P_0} \leq \left(\frac{2}{\gamma_a+1}\right)^{\frac{\gamma_a}{\gamma_a-1}} \end{cases} \tag{3.9}$$

Lastly, η_e, ϕ and H_f are engine thermal efficiency (approximately a function of engine speed and manifold pressure), stoichiometry fuel-air ratio and gasoline heat of combustion, respectively. Readers are referred to [41] for numeric values of the parameters used in this engine model.

According to [48] we can define a sliding surface, S

$$S = P_m - P_{m,desired} \Rightarrow \dot{S} = \dot{P}_m - \dot{P}_{m,desired} \tag{3.10}$$

where $P_{m,desired}$ is the desired manifold pressure which leads to the desired engine indicated torque. Therefore, $\dot{S}$ can be found as in (3.11).

$$\dot{S} = A_{eng}P_m + B_{eng}A_{th} - \frac{\dot{T}_{ind,desired}}{C_{eng}} \tag{3.11}$$

where A_{eng}, B_{eng}, and C_{eng} are defined in (3.7).

In order to satisfy reachability condition, we use the signum (sgn) function:

$$S\dot{S} < 0 \Rightarrow \dot{S} = -\eta sgn(S) \tag{3.12}$$

Now all we have to do is to tune η according to the model uncertainty and operating conditions. Then the control input will be:

$$A_{th} = \frac{1}{B_{eng}}\left(-\eta sgn(P_m - P_{m,desired}) - A_{eng}P_m + \frac{\dot{T}_{ind,desired}}{C_{eng}}\right) \tag{3.13}$$

On the generator side, a manually tuned PID controller ensures that the generator speed follows the reference value by adjusting the generator current. These two controllers provide power tracking, and at the same time, move the engine operating point to the minimum BSFC point.

3.2.2 Battery Controller

The algebraic sum of the power generated by the engine-generator set and the power consumed (or generated) by the electric motor is used to charge/discharge the battery. A PID controller is responsible for making the battery follow the reference power by adjusting the battery current. The NiMH battery model calculates the SoC, and the SoC is sent back to the supervisory controller as a feedback.

During braking, the battery absorbs part of the kinetic energy, and the SoC increases. Although not likely with the model-based supervisory controller, the SoC may reach its upper limit during braking. In such cases, the supervisory controller turns off the regenerative braking, and to compensate for that, it increases the mechanical braking effort. In the model, the extra mechanical braking power is subtracted from the electric power demand to consider such loss of regenerative braking.

3.3 High-fidelity Model Simulation Results

A model-in-the-loop simulation is done using the presented model to evaluate the designed controller performance. The input to this simulation is the desired speed profile. The controller follows the logic in (2.30), and for the calculation of $\bar{u}$, the linear fit in Figure 2.11 is used. In the present work, it is assumed that the necessary information is available to the controller to estimate $\bar{u}$. The information includes the cruise time (t_p) and available negative energy (E_n), as is required in the calculation of $\bar{u}$ in Figure 2.11.

To compare the behaviors of the control-oriented model and the high-fidelity model, simulations shown in Figure 3.5 are conducted. The input to both models is the desired speed profile, and output is the state of charge. The comparison of the two SoC trajectories for the first 440s of the FTP75 drive cycle is shown in Figure 3.6. It can be seen that the controller can predict the actual vehicle behavior very well based on the simple control-oriented model, in spite of the extensive number of simplifications made in its design

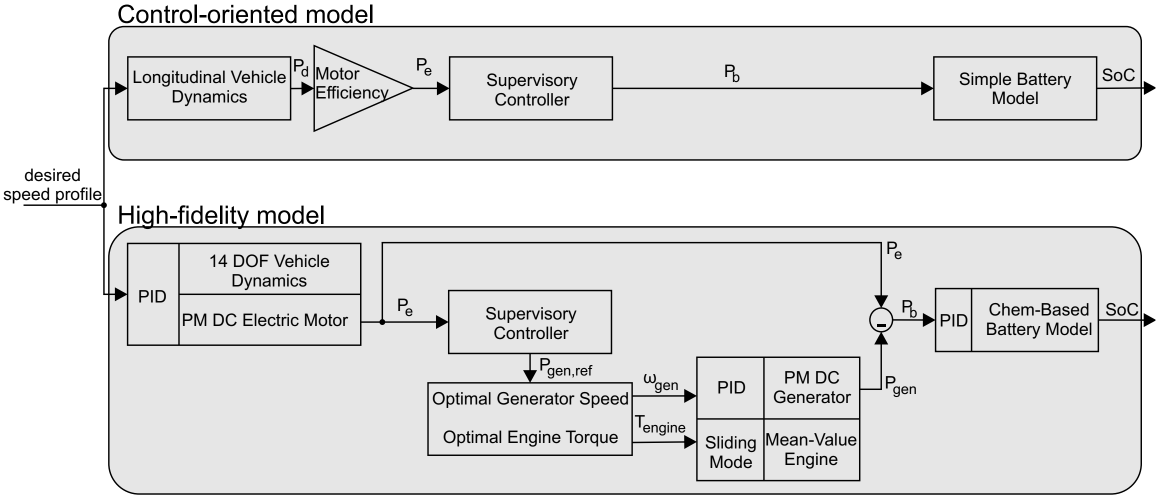


Figure 3.5: Simplified schematic of the simulations to validate controller-oriented model

process.

The similarity in SoC trajectories can be explained as follows: the battery state of charge (as the dominant dynamic behavior of the powertrain) is affected by the battery power and its internal dynamics. The battery power is the difference of the power demand and the generator power, Figure 3.5. Therefore, as long as the longitudinal vehicle dynamics in the control-oriented model is accurate enough, and the engine-generator set follows the optimal set-points properly, the battery power will be the same as that of the control-oriented model. The battery dynamics is also accurately approximated by the well-tuned circuit model. Thus, both models (control-oriented and high-fidelity models) will show similar behavior, which in turn, means that the outcome of the high-fidelity model is optimal too.

To see the optimality of the control strategy, the performance of this controller was compared with that of a PID controller. The idea behind this PID controller is to preserve the health of the battery as long as possible by minimizing variation of the SoC. The PID controller keeps the SoC near the reference value by controlling the engine-generator power. The PID controller is tuned in such a way that the battery is charged in an

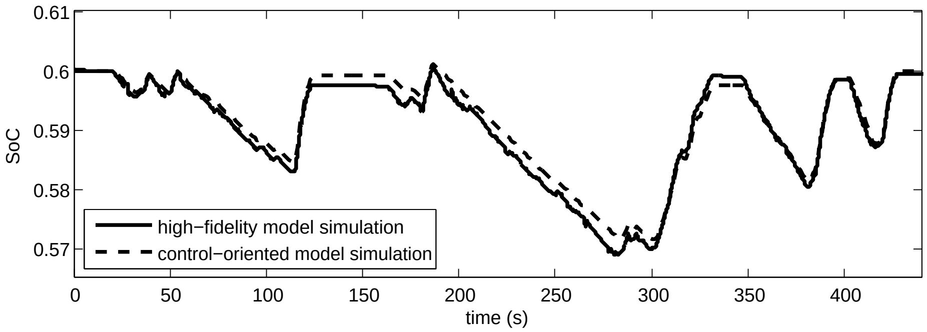


Figure 3.6: State of charge trajectory for the first 440s of FTP75 drive cycle

appropriate time. Besides, the proper tuning of the PID controller allows the battery to capture and reuse all the available regenerative braking energy. Thus, the only difference in the PID controller and the optimal controller is the higher battery efficiency in charging and discharging that affects the fuel consumption. This makes the PID controller a good candidate for the powertrain controller in terms of fuel efficiency too.

As can be seen in Figure 3.7, the final SoC with the PID controller is higher than the reference value, because the battery captures the regenerative braking energy at the end of the drive cycle, when the SoC is close to the reference value. This increased SoC causes a noticeable increase in fuel consumption, and hinders comparison between this controller and the optimal controller (which is charge sustaining). To better compare the fuel consumptions, five successive FTP75 cycles were used to approximate infinite driving pattern. The results show that the optimal controller gives a fuel consumption of 1504.2 grams, whereas the PID controller resulted in a total fuel consumption of 1607.3 grams. Thus, fuel consumption is reduced by 6.4% with our model-based controller.

Another way to compare the fuel consumptions is to change $\bar{u}$ so that the final SoC resulting from the optimal controller is equal to the final SoC resulting from PID controller. For the FTP75 drive cycle, this comparison shows that the optimal controller has a fuel consumption of 304 grams versus the 324 of the PID controller. In this case, the optimal controller shows a 6.5% reduction in fuel consumption compared to the PID controller.

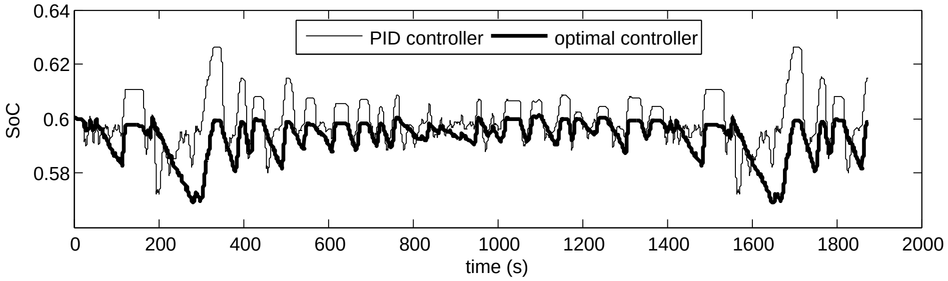


Figure 3.7: Comparison of SoC trajectories for the FTP75 drive cycle for the two controllers

3.4 Chapter Summary

In this chapter, the details of the model-in-the-loop simulation have been presented. For evaluating the performance of the designed controller, a high-fidelity model is developed in the MapleSim environment, and the model is exported to the Matlab/Simulink environment for controller applications. Low-level controllers are used for each component to ensure that the components follow the powertrain controller set-points with little error.

The simulation results shows that the powertrain controller can effectively predict the system behavior, and the high-fidelity model simulations matches the controller prediction (results of the control-oriented model simulation) very well. This shows that the controller is able to provide the optimal solution for minimum fuel consumption.

Chapter 4

Hardware-in-the-Loop (HIL) Simulation

Hardware-in-the-Loop (HIL) simulation is a well-established method to simulate physical systems and control loops with higher levels of accuracy. An HIL simulation setup provides a more realistic environment for controller evaluation purposes, as it can take into consideration different aspects of the control loop that are neglected in model-in-the-loop simulations, such as communication issues and controller computational limitations.

Figure 4.1 shows the HIL simulation setup that was developed for HEV powertrain controller evaluation. In this chapter, the details of the HIL simulation setup will be discussed.

4.1 Hardware Description

The two core components in an HIL setup are: 1) an independent processing unit to run the controller procedure, and 2) a powerful real-time processing unit to run the plant model.

For our HIL simulation, the designed controller is programmed into an Electronic Con-

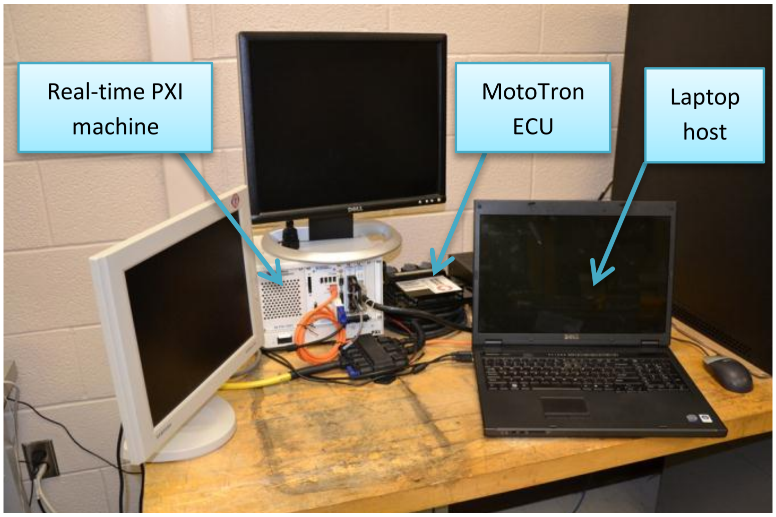


Figure 4.1: The HIL simulation setup developed

trol Unit (ECU), and the high-fidelity powertrain model is solved by a real-time target to provide the accurate sampling which the controller requires. The communication channel between the ECU and the plant (real-time target) is the Control Area Network (CAN) bus. The following sections contain details of the hardware used in this setup.

4.1.1 MotoTron ECU

HIL simulation results are more reliable when the controller prototype is the same as the controller used in the real plant. For this application (HEV powertrain controller), a MotoTron ECU is used to serve as the powertrain controller. This ECU is from the ECM-5554-112 family of controllers from Woodward that uses an 80MHz Motorola MPC5554 processor. The commercial version of this controller is used in automotive and marine applications. The automotive-based design of this ECU makes it an ideal choice for HIL simulations.

To program the controller code into the ECU, the code needs to be compiled by the MotoHawk Green Hill compiler. The generated code can then be programmed into the ECU by MotoTune software. The controller used in this setup is a calibratable version which provides controller tuning features in real-time using MotoTune. This feature is specifically useful in tuning controller parameters without encountering the need to reprogram the controller itself.

The easiest way to program the controller is to download the generated code into the controller via CAN bus using MotoTune software. For this purpose, a USB-to-CAN adapter is provided by MotoTune to facilitate the programming procedure. The controller code itself can be easily complied using Woodward's Green Hill compiler, which compiles the required code directly from a Simulink model.

4.1.2 Real-Time Target

In HIL simulations, the way the plant model is solved is critical. If the controller cannot access the measurements in the right time, or the simulation runs faster or slower than real-time, the results will not be acceptable. Only when the plant model is solved in the exact time steps, and controller obtains the readings on time, can we be sure that the simulation is representing the real control loop.

To satisfy real-time requirements, and achieve enhanced accuracy of the simulations, it is necessary to use a real-time computer to solve the plant model deterministically. To serve this purpose, a PXI platform from National Instrument (NI) is used as the real-time target. The processing unit of this computer is PXI-8110, which is powered by a 2.26GHx quad-core CPU and has 2GB of RAM. Lastly, this PXI platform runs the LabVIEW Real-Time operating system.

In real-time simulations, it is essential to run a real-time operating system on the computer. In non-real-time operating systems (such as Microsoft Windows), there is no specific deadline for computational processes, and tasks are prioritized based on different criteria such as maintaining the hardware/software functionality or user preferences. On the other hand, in real-time operating systems (such as the NI LabVIEW Real-Time operating system), the goal is to respond to an interrupt, or perform a task before a specified deadline. Therefore, by making use of such real-time operating systems, the model can be solved with greater consistency, and the communication delay can be minimized.

Our real-time target (NI PXI computer) runs LabVIEW Real-Time 2011 operating system. To run a program on this platform, a LabVIEW program must be deployed. LabVIEW is a graphical programming language that expedites the development of multi-threaded applications and facilitates communication with external hardware. LabVIEW programs are made in Virtual Instrument (VI) files. These VI programs are made in the host computer which runs a windows version of NI LabVIEW. The VI programs are then deployed into the real-time target via Ethernet. Once the program is successfully deployed, the real-time target begins to run the program, and the user can see the outputs or send commands using the host computer.

To use this platform for solving the powertrain high-fidelity model in the real-time, the model has to be converted into a C-code and then into a Digital Link Library (DLL) in order to be used in the NI LabVIEW environment. The MapleSim Connector toolbox provides a seamless process for this purpose. The model developed in MapleSim can be easily converted to such DLL files, which in turn, can be used in a *simulation loop* in LabVIEW.

Major responsibilities of the real-time target are depicted in Figure 4.2. Each of the four cores of the real-time target CPU runs a different application. The first core is responsible for running the application for PXI-host communication. This application is solely used to send and receive variables to and from the laptop host via Ethernet connection. The second CPU core runs the CAN communication application. The third core runs the battery cycler control application, which will be discussed in detail in the following chapter. Finally, the last core is responsible for solving the high-fidelity vehicle model.

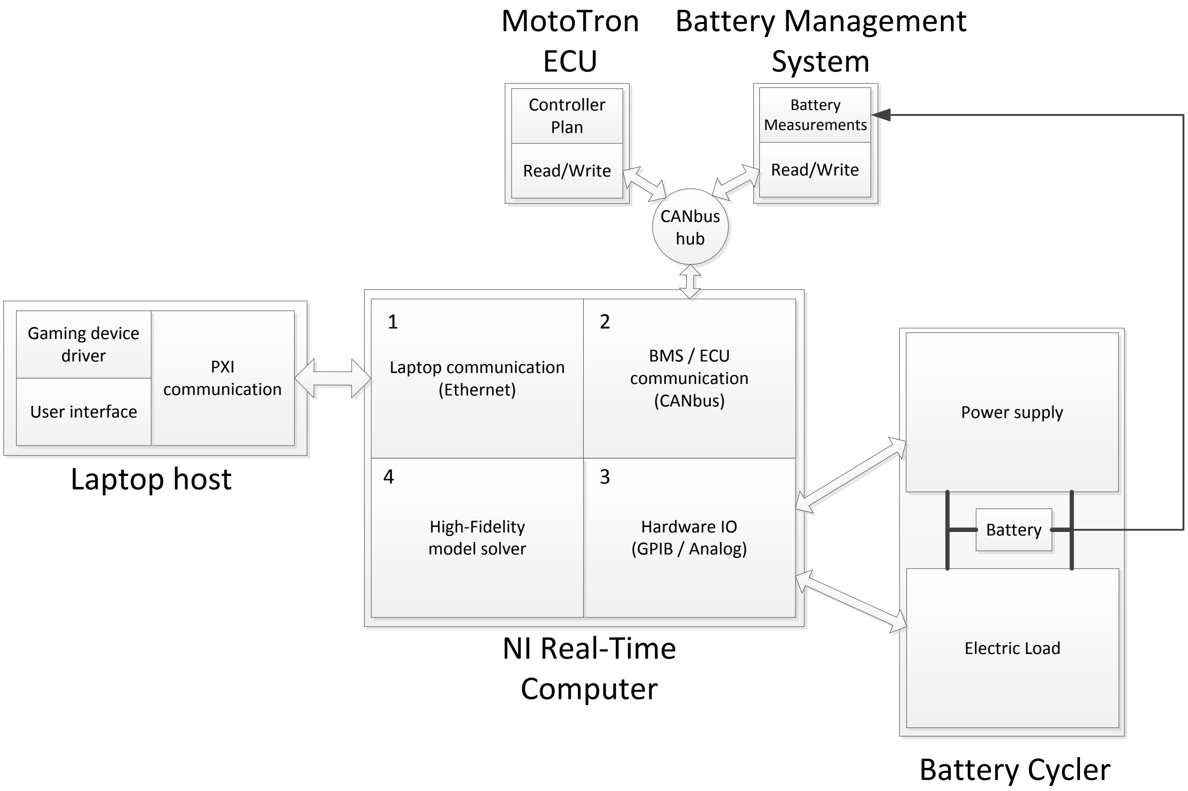


Figure 4.2: Schematic of the HIL setup

4.1.3 CAN bus

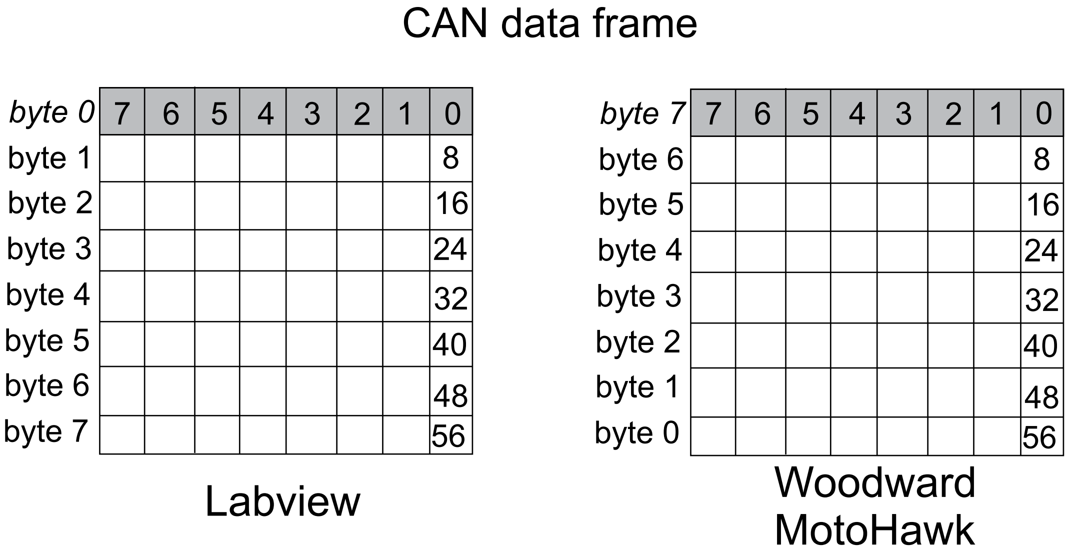
The Control Area Network (CAN) is a standard message-based protocol, which was initially developed for in-vehicle communications; however, because of its robustness and ease of operation, it is often used in automation applications too [49].

On a CAN bus, each of the nodes are directly connected to the bus, and there is no central control unit to regulate the communications. Instead, CAN bus is a serial message-based protocol, where each node can send and receive messages when the bus is free. When two nodes start to send messages simultaneously, the message with higher priority prevails, and the lower-priority message waits until the bus is free. The priority of each message is identified by an *Arbitration ID*, where lower IDs have the higher priority. The arbitration ID also serves as the name tag for each message. When a node transmits a message on the CAN bus, the message is received by every node on the bus. Each node can then ignore the message, or do a specific task based on the ID and the contents of the message.

The other part of a CAN message is the data frame. A CAN data frame is defined byte-wise, i.e., the message consists of groups of bytes that contain an integer number. Thus, to send a variable, it should be scaled to an integer number, based on its range and required accuracy. When the variable is transmitted and received, it is scaled back to its original format.

When a variable requires more than one data byte to be transmitted (when its range exceeds [0,255]), it is divided into a number of bytes. Careful attention is required during the processes of turning the variable into separate bytes and the ordering of bits in each byte.

In a CAN message, the bits are sent one by one as a serial signal. When the whole message is sent, it is interpreted as a number of bytes. However, the way the bits are grouped into bytes shows inconsistency between different devices and software packages. For example, the Woodward compiler takes the first eight bits of the message as the byte 7, but LabVIEW considers the same order of bits as byte 0 (see Figure 4.3). There are also



CAN data frame

Labview

byte 0	7	6	5	4	3	2	1	0
byte 1								8
byte 2								16
byte 3								24
byte 4								32
byte 5								40
byte 6								48
byte 7								56

Woodward MotoHawk

byte 7	7	6	5	4	3	2	1	0
byte 6								8
byte 5								16
byte 4								24
byte 3								32
byte 2								40
byte 1								48
byte 0								56

Figure 4.3: Different byte allocation methods in CAN data frame

various ways to interpret the multi-byte numbers. This is referred to as the *endianness*. In *little endian* format (Intel), the least significant byte is first and has lower index, whereas in *big endian* format (Motorola), the least significant byte is sent last and has the highest index. The difference in endianness is illustrated in Figure 4.4. In our setup, the little endian mode is used to interpret data bytes.

The optimal powertrain controller discussed in chapter 2 requires two readings from the system: the state of charge and the electric power demand (P_e). The two measurements are calculated by the real-time target by solving the high-fidelity model. The real-time target then sends these two pieces of information, along with the estimated $\bar{P}$, in a single CAN message to the ECU. The controller processes the information and calculates the optimal generator power. The optimal generator power and corresponding optimal battery power are sent back to the real-time target in another CAN message.

Table 4.1 shows the variables, and the position of the variable in the CAN messages for ECU-PXI communication.

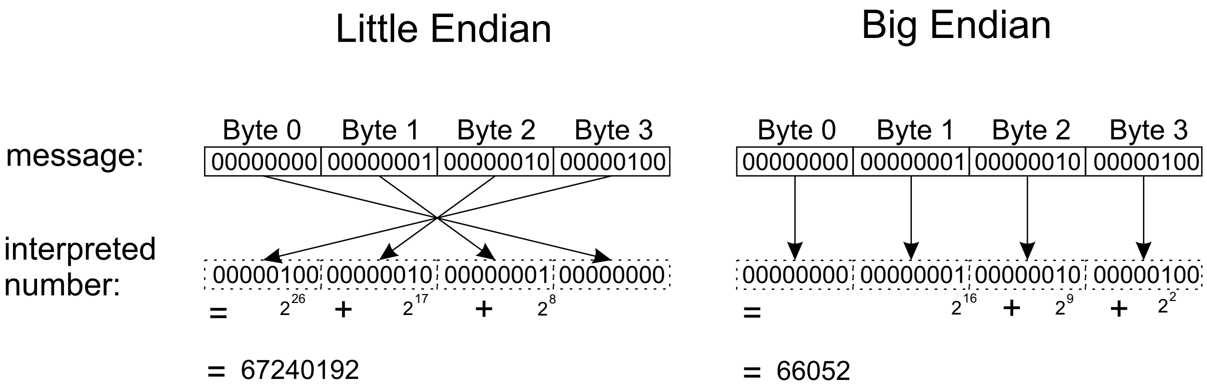


Figure 4.4: An example for different endianness definitions

Table 4.1: CAN message definition for the HIL simulations

CAN message				MotoHawk		LabVIEW	
message name	arbitration ID	message length	variable name	start bit	bit length	start bit	bit length
ECU to PXI	2	4 bytes	Pgen_ref	56	16	0	16
			Pbat	40	16	16	16
PXI to ECU	1	7 bytes	Pe	56	24	0	24
			P_bar	32	16	24	16
			SOC	16	16	40	16

In the base CAN frame format (CAN 2.0 A protocol), the identifier portion of the message (arbitration ID) contains 11 bits following the start bit. The main data frame can contain up to 8 bytes (64 bits). Combined with all other regulatory bits, a CAN message is comprised of up to 108 bits. Depending on the bit-rate of the CAN channel, a limited number of messages can be sent on a CAN bus. In this HIL setup, the CAN channels work with a bit rate of 500 kbps (kilo-bits per second); therefore, the maximum capacity of each CAN channel is roughly 4600 messages per second. The communication program on the real-time target runs at every 1ms and sends a message (PXI to ECU) in each run of the loop. The controller program also runs every 5ms and sends one message (ECU to PXI). Thus, 1200 messages are sent each second, and this load occupies 26% of the CAN channel capacity.

4.2 Numerical Convergence Study

In every numerical simulation, the process of convergence study is of great importance. It is essential that the simulation results be free of numerical errors such as integral error and discretization of simulation time. On the other hand, reducing time steps and integration tolerances increases the computational time, and it is possible that the simulation could fall behind real-time requirements.

To solve the high-fidelity model in LabVIEW, the explicit third order Runge-Kutta integrator was used. The result of such an explicit method converges to the correct solution by reducing the time step. When the solution changes negligibly with reducing the time step, it can, therefore, be inferred that the solution has converged. Figure 4.5 shows the result of the convergence study conducted for solving the high-fidelity model in LabVIEW. It can be seen that the time step of 2ms gives satisfactory results, hence is used in this simulation.

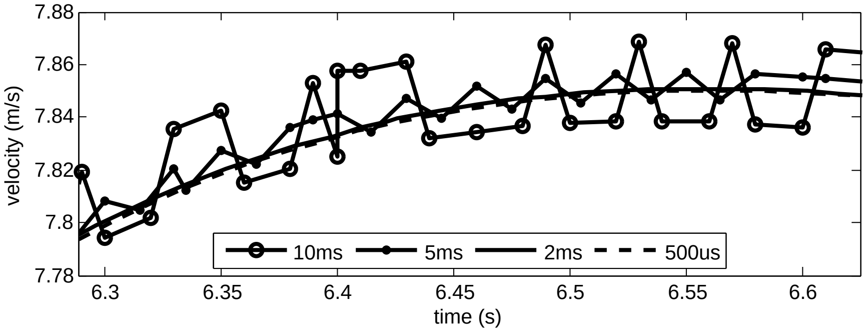


Figure 4.5: The simulation results for different step sizes

In real-time applications, it is also important to choose a simulation step size greater than the time required to solve one iteration. This ensures that all of the computations required for one time step will be completed before the next iteration begins. Figure 4.6 shows the number of iterations that finish late in relation to different time step sizes. For example, at 200μs, 91 % of iterations finish late (the model takes more than 200μs to be solved). In comparison, only 2% of iterations finish late if a step size of 210μs is chosen. It can also be seen that 250μs is the *real-time requirement* for this model as each of the iterations finish in time. Lastly, one can infer that the time required to solve one step of the model is roughly between 200μs and 210μs – where the number of finished-late iterations jumps.

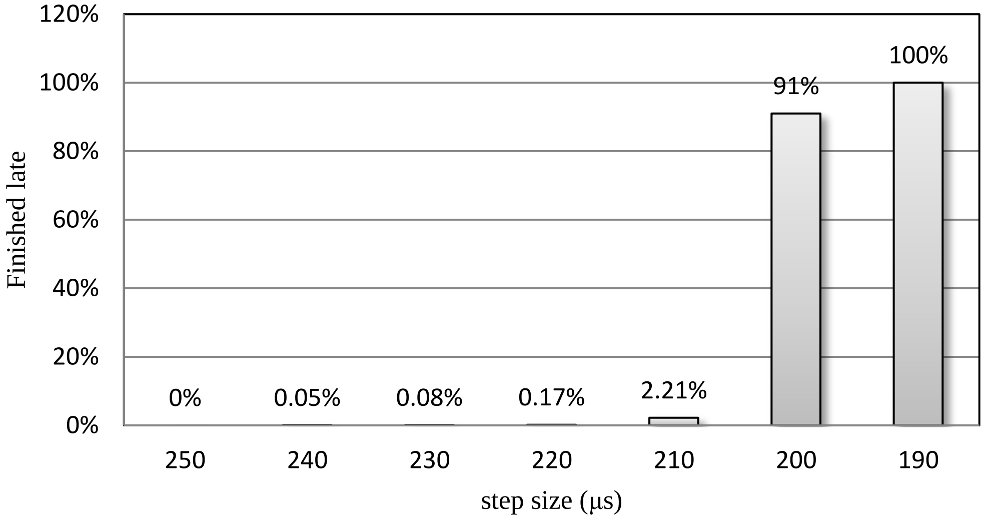


Figure 4.6: Number of iterations finished late for different step sizes

4.3 HIL Simulation Results

With the setup described in the previous sections, a full HIL simulation can be done. As shown in Figure 4.7, despite the simplifications brought forth in controller design and neglect of communication delays, the controller can successfully predict the behavior of the HEV powertrain and can maintain optimal behavior.

Figure 4.8 shows the tracking performance of the engine-generator set. Thus, as can be seen in Figure 4.9, the power that the battery has to deliver or absorb follows the optimal trajectory that the optimal controller had considered.

As the battery – the most critical component of the powertrain – behaves as predicted, one can examine such results and conclude that the optimal controller is indeed able to predict the optimal behavior of the system. Since the lower level controllers can cause the

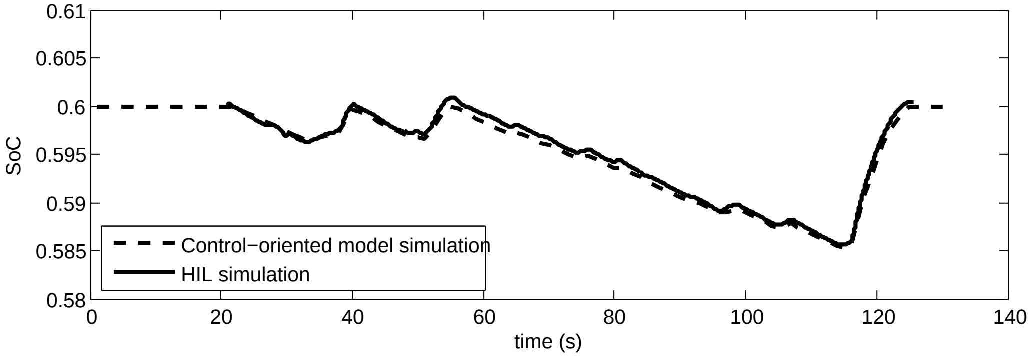


Figure 4.7: The comparison of SoC between control-oriented model simulation and HIL simulation

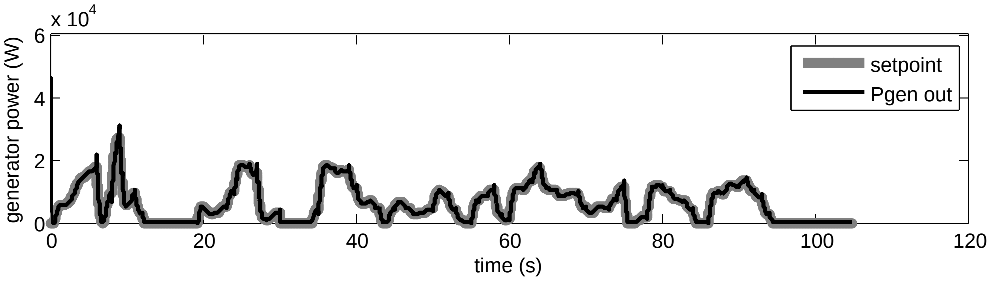


Figure 4.8: Tracking performance of the lower level controllers of the engine-generator set

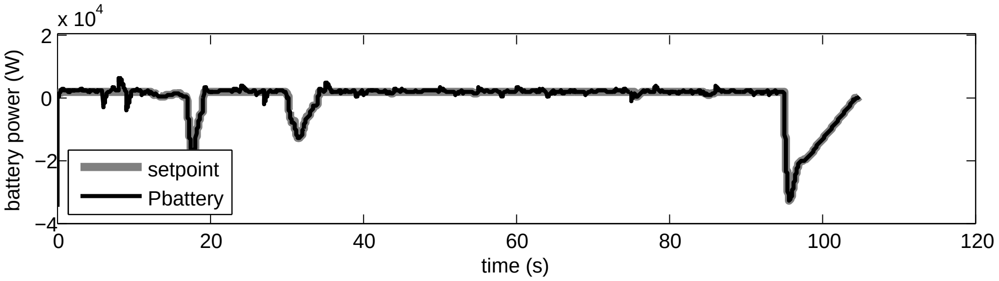


Figure 4.9: The comparison of optimal battery power, defined by the controller, and the real battery power

system to follow the optimal controller set-points, the behavior of the system with the use of the optimal controller is, therefore, optimal.

4.4 Chapter Summary

This chapter discussed the details of the hardware-in-the-loop simulation of a series HEV powertrain with the optimal control. To examine the hardware issues in the control loop, the designed controller was programmed into a MotoTron ECU, and the high-fidelity model was solved by a real-time PXI target.

The simulations showed that the designed controller was able to provide satisfactory prediction of the system, and the control loop delays and communication issues were negligible.

Chapter 5

Component-in-the-Loop (CIL) Simulation

In a hybrid electric powertrain, the battery is the component that has the most influence on the system behavior – especially in the design of optimal controllers. Therefore, to evaluate the designed powertrain controllers, it is very useful to include a physical battery in the control loop. This component-in-the-loop (CIL) simulation greatly enhances the accuracy of the simulations and provides a better benchmark for the controllers.

In CIL simulations, the physical components replace their mathematical models and are driven according to the simulation requirements. In this CIL setup, the battery will replace its mathematical model, and the simulated charge/discharge power is used to drive the battery. The replacement of a mathematical model with a physical component presents a challenge to include a component in an otherwise mathematical simulation. To achieve this functionality, a real-time battery cycler is used. The real-time target solves the models of other components in the powertrain and indicates a power that the battery should be charged or discharged with – just as it would be in the actual vehicle. This power is then used as the set-point for the real-time battery cycler. The battery cycler drives the battery with the required power, and the battery management system (BMS) monitors the changes in the battery. These measurements are then sent back to the simulation and/or the controller. Figure 5.1 shows the schematic for the HIL simulation (without a physical battery) and the CIL simulation (with the physical battery in the loop) for a series HEV.

In this chapter, the CIL setup that was constructed for the evaluation of HEV powertrain controllers will be discussed.

5.1 Hardware Description

The CIL setup is built upon the existing hardware-in-the-loop setup, through the addition of a real-time battery cycler, as displayed in Figure 5.2. The battery cycler is a device which receives the required battery charge/discharge power and charges/discharges the battery accordingly.

The battery cycler consists of two major components: a power supply and an electric load, which are responsible for charging and discharging the battery, respectively. The schematic of the battery cycler is displayed in Figure 5.3.

This CIL setup employs a scaled-down battery test bench – the battery under testing is not the same size as the battery pack in vehicles. Although in such a scaled simulation there is a small amount of error due to component scaling, the greater flexibility of the setup makes it ideal for development purposes. On the other hand, when the full-size battery pack is used, the results are only accurate for that particular battery pack, and simulating other battery sizes is not possible without the same scaling error. Therefore, working with a full-size battery test bench is not only more costly, but it also lacks the flexibility required for research and development purposes. In the following sections, the details of the real-time battery cycler are described.

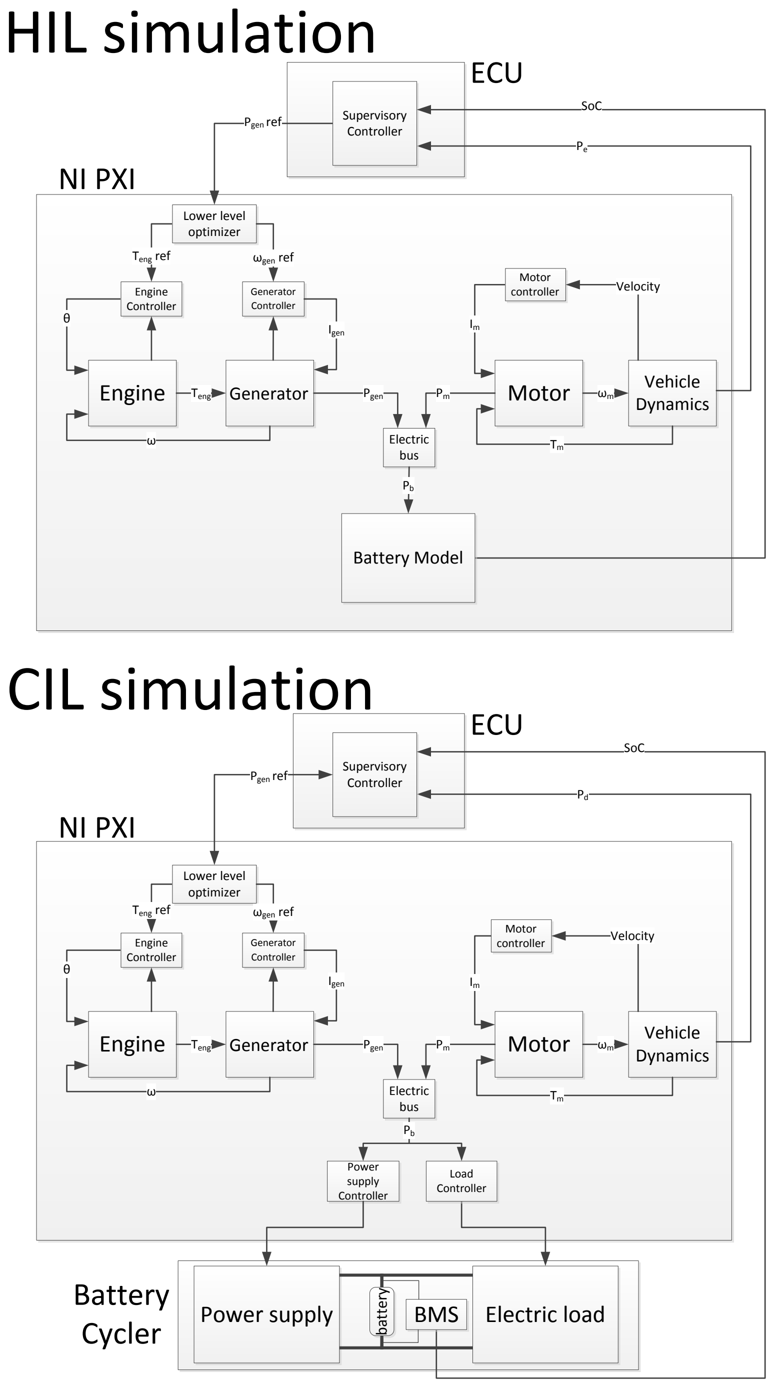


Figure 5.1: HIL and CIL simulation setup

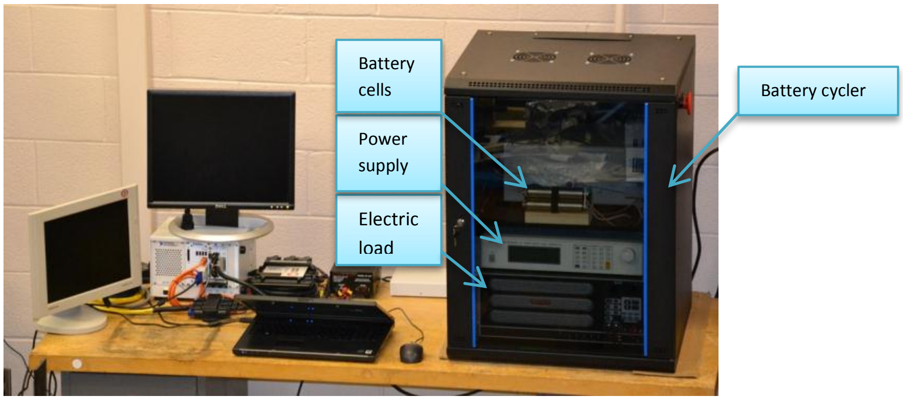


Figure 5.2: The component–in-the-loop setup

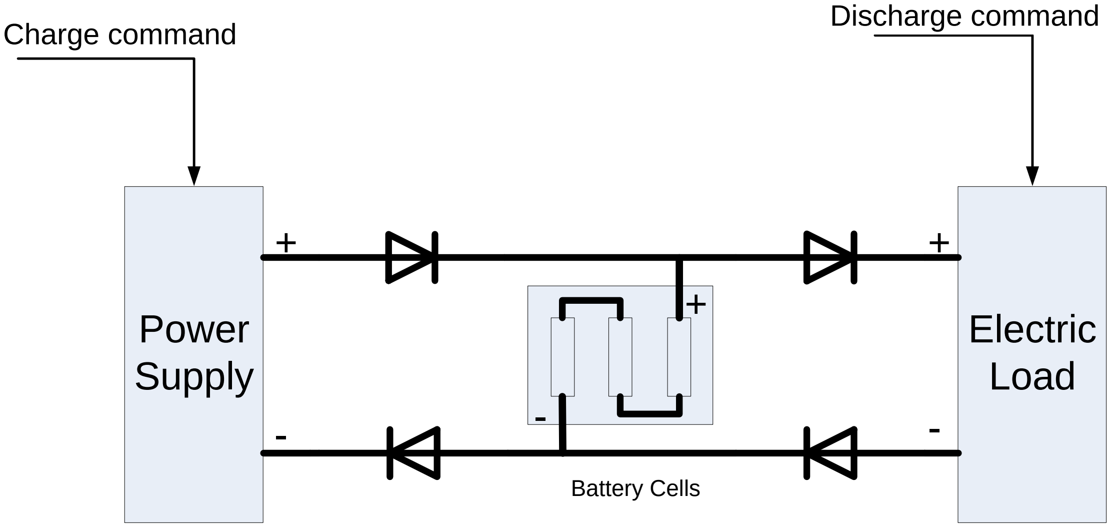


Figure 5.3: The schematic of the battery cycler

Table 5.1: Comparison of three types of batteries [50]

Characteristics	Lead-Acid	NiMH	Li-ion
Specific energy (Wh/kg)	35	70	180
Energy density (Wh/l)	70	140	180
Energy/consumer-price (Wh/US$)	7	2.75	2.8
Electrical Efficiency	90%	66%	85%
Self-discharge rate (%/month)	20	30	5
Durability (cycles)	800	1000	1200

5.1.1 Battery and Battery Management System (BMS)

There is a broad range of technologies for the sources of electrical energy in electric and hybrid electric powertrains. Such sources include but are not limited to fuel cells, batteries, ultra-capacitors, and solar cells. Among these energy sources, batteries are one of the most important sources. The battery itself branches into different categories such as Nickel Metal Hydride (NiMH), Lithium ion (Li-ion), and Lead-acid batteries [50].

In the automotive industry, especially in electric and hybrid electric vehicles, NiMH and Li-ion batteries are the most widely used technologies. NiMH batteries are known to be a stable and safe solution; however, compared to Li-ion batteries, they have much lower specific energy and electrical efficiency, and have higher self-discharge rate. In return for better performance, Li-ion batteries require more complicated battery management systems due to their unstable nature. A comparison of the battery technologies in given in Table 5.1, taken from [50].

Li-ion batteries are growing in market share, especially in plug-in HEVs. They are expected to replace NiMH batteries due to their superior performance, and they seem to be the main technology of the electrical storage system in the electric and hybrid electric vehicle of the near future.

As a result of the nature of the batteries, it is highly recommended to use battery

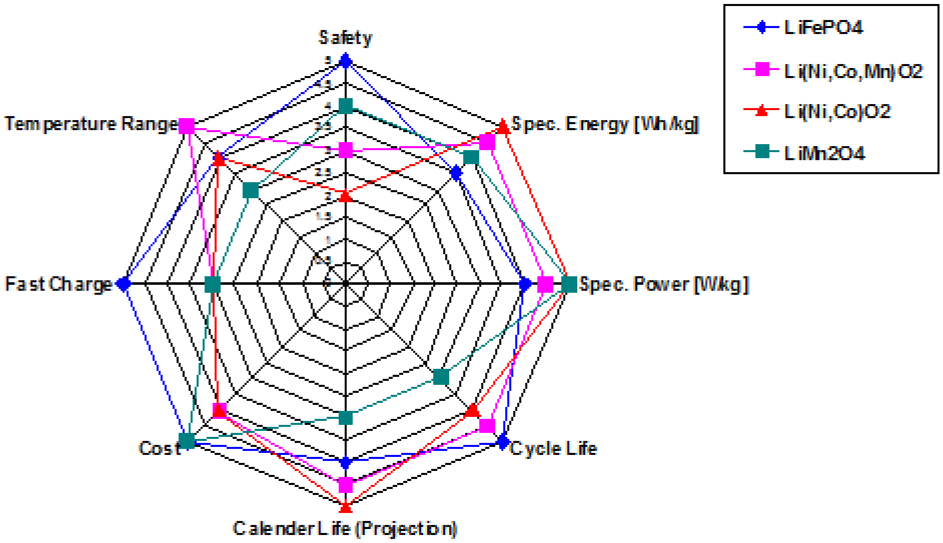


Figure 5.4: Comparison of different technologies for Li-ion battery [51]

management systems (BMS) to monitor and regulate the cells in a battery pack. A BMS monitors the individual cell voltage, temperature, and current while also calculating the battery state of charge and state of health and providing protection circuits to ensure that the batteries are working safely. As the requirements of the battery packs change with battery size and chemistry, no BMS can be used for all battery packs. Therefore, for the CIL simulation, it is best to have a versatile battery/BMS module that allows us to test for a variety of applications.

Figure 5.4 shows a comparison of different common cathode chemistry in Li-ion batteries. It can be seen that Nickel Cobalt Oxide (Li(Ni,Co)O2 or briefly NCA) cells show longer calendar life and higher energy and power density; however, they are relatively unstable. On the other hand, LiFePO4 cells are inherently more stable, but are more costly and short on storage capacity.

For our powertrain CIL setup, the right choice of battery size and chemistry is important. The battery chemistry affects the current rating of the cells (the maximum amount of current a cell can be driven with), as well as the charge/discharge dynamics of the cells; the effect of the chemistry on the latter is, however, negligible. Battery size, on the other hand, then identifies the size of the battery cycler; larger battery cells require larger and more expensive battery cyclers.

Battery size and chemistry affects the current rating. However, comparison of current ratings for different batteries may be misleading as the larger cells will have higher current ratings. It may even be possible to place a number of cells in parallel to get the same current rating. Therefore, a *per-unit* scale is usually used to compare battery current ratings. This per-unit scale (denoted by C) is the ratio of the battery current per its capacity. For example, a current of 50A passing through a 10Ahr cell is $\frac{50A}{10Ahr} = 5C$. In this way, the per-unit current of a battery pack, which consists of a number of cells in series or parallel, is the same as the per-unit current of each individual cell. This idea is further illustrated in Figure 5.5. It can be seen that the per-unit current of any configuration of cells is the same as the per-unit current of the individual cells.

During the simulations described in chapter 3 and 4, the battery current momentarily reached 100A. Considering the battery size (6.5 Ahr), the per-unit current is roughly 15C. To simulate such a full-size battery pack with the CIL setup, the test battery cells should be able to provide the same charge/discharge rate.

For this CIL setup, NCA Li-ion cells from GAIA are used (see Appendix D). These cells are available in large capacities (7.5 Ahr) and show superior charge and discharge rates – up to 16C pulse charge and up to 40C pulse discharge – which meets the power rating requirements.

The BMS to work with these cells are provided by i+ME ACTIA [52], a German manufacturer. This BMS consists of two parts: a master board and a number of slave boards. The slave boards are solely responsible for measuring the individual cell voltage and temperature in a multi-cell module. Each slave board monitors up to 10 cells and remains in sleep mode until it receives a request from the master board. The slave board starts the measurements, and when it is finished, it passes the information to the master

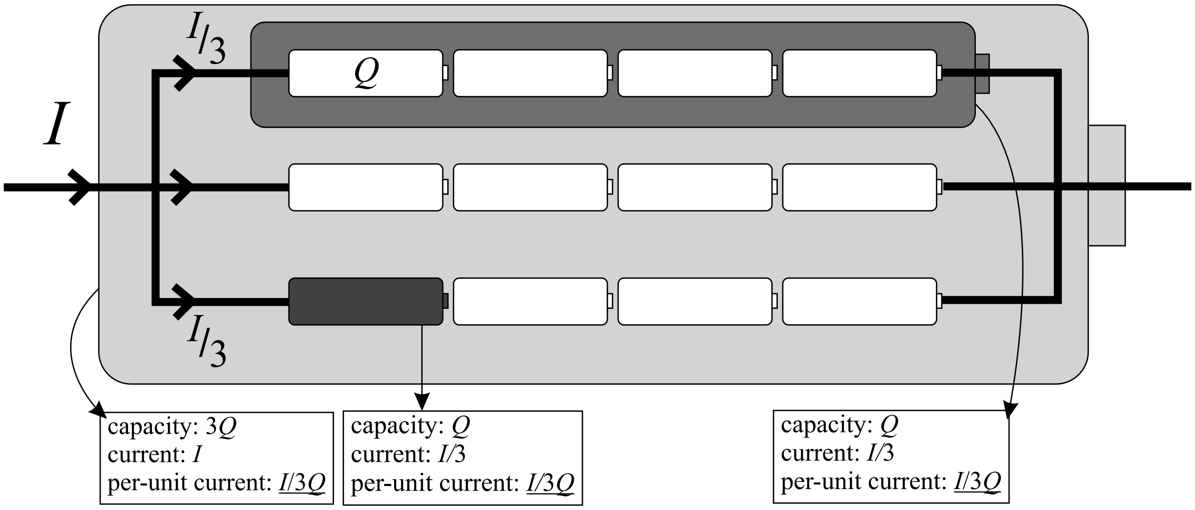


Figure 5.5: Per-unit current for different cells in a battery pack

board and goes to sleep mode again. For our CIL setup, only one slave board is used to monitor the three cells.

The slave board is connected to the master board via an RS-485 interface. The master board is responsible for collecting measurements form the slave board and current sensor, monitoring the state of charge calculation, assessing battery protection (over voltage, under voltage, over current, etc.), and maintaining communications (via CAN bus and RS 232 interface).

Once the master board is turned on (either by the CAN wake-up frame or by activating the power switch), it begins the wake-up procedure by performing an internal hardware check, retrieving statistical data, conducting initial measurements, and (if there is no error) activating the main relay. When the wake-up phase is completed, the master board sends requests to slave boards for individual cell voltage and temperature and measures the current from the current sensor every 40ms.

The BMS calculates the state of charge using the following procedure. When the battery current is low (less than 200mA), the master board acquires individual cell voltages and estimates the state of charge based on a look-up table and average battery voltage. When the current exceeds 200mA, the master board uses the estimated state of charge as the initial condition, and uses the current sensor to measure the amount of charge that passes through the battery.

The preferred method of communication with the BMS is via CAN bus. Initially the BMS is in sleep mode and is awakened by a wake-up CAN message. Once the BMS is active, it is possible to send the command to BMS using CAN messages. The BMS interprets the message and sends back an answer. For example, to retrieve the state of charge, the message with ID 0x100, containing number 3, should be sent to the BMS, and it returns a message with ID 0x101, containing a 3-byte number that will show the state of charge in 0.1% steps.

5.1.2 Power Supply

The power supply in this setup is responsible for charging the battery, according to the charge requirements found from simulating the powertrain model. After the battery size and type are selected for this setup, the power supply and electric load specifications have to be identified accordingly. The selected batteries are 7.5Ahr in capacity and can be charged with 30A continuously and up to 120A momentarily. Therefore, to fully use the battery capabilities – which are also required for simulations – the power supply should be able to deliver up to 120A.

Another important specification for power supplies is the response time – the time the power supply requires to change from no-power to full-power. To identify the required response time, the role of the power supply in the CIL setup must be considered. In the powertrain CIL simulation setup, the battery is charged either during regenerative braking or when the engine runs the generator. In the former case, the frequency range of battery power is the same as that of the power demand, which is usually less than 1 Hz (the driver

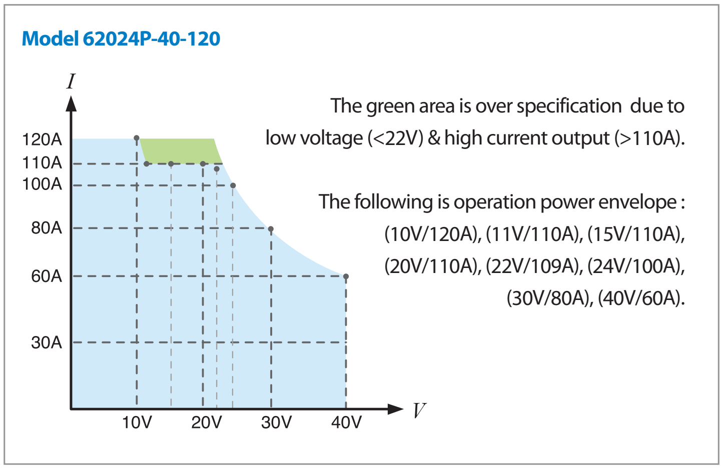


Figure 5.6: Operating range of Chroma 62024P-40-120 [53]

commands do not change faster than 1Hz). In the latter case, when the engine-generator set charges the battery, the battery power variation is limited by the engine dynamics and the engine set-points, which once again are related to the power demand.

For the power supply to track the required power trajectory, it should have a rise time that is 10 times faster than the required trajectory. Considering the maximum of 10Hz for power request variations, the power supply should have a rise time of less than 10ms. This is a constraining factor as most power supplies have longer rise time.

One power supply that could meet both specifications (current rating and rise time) is the Chroma 62024P-40-120 DC power supply. This power supply, which can deliver up to 120A (Figure 5.6), is used in this setup. More details about this power supply can be found in Appendix E.

The power supply can be controlled remotely via either a GPIB (General Purpose Interface Bus) terminal, or an analog interface. The control of the power supply is further discussed in section 5.2.

The power supply can work in two modes: constant voltage and constant current. In constant voltage mode, the power supply maintains a constant voltage across the load, regardless of the current. In constant current mode, the power supply tries to maintain the output current at a constant level by changing the terminal voltage. In this mode, when the connected load changes rapidly, the power supply output current takes some time to recover from the disturbance. This time is referred to as the *transient time* and is 3ms for the selected power supply.

5.1.3 Electric Load

Similar to the power supply, the electric load is responsible for discharging the battery to simulate the power drawn from the battery by the electric motor(s). For the electric load, similar current rating and rise time specification is required. The battery cells are capable of continuously delivering power at 150A (20C); therefore, it is best to have an electric load capable of drawing the same current. The problem surrounding rise time is much less important in electric loads, as they experience very fast transient time – typically less than 1ms.

For this CIL setup, an Ametek Sorensen SLH 1800-60-240 is chosen and meets the required specifications (see Appendix F. The SLH load has a similar constant power curve as the power supply and is able to draw up to 240A, limited by 1800W power. Therefore, the load can draw up to 163A (21C) from the battery cells at 11V (total voltage of the three battery cells). The operating curve for this SLH load is shown in Figure 5.7.

The electric load has four different modes of operation: constant current, constant voltage, constant resistance, and constant power. In constant current, the load draws a constant current regardless of the terminal voltage. In constant voltage mode, the load adjusts the current to achieve constant voltage across its terminals.

The constant resistance mode adjusts the current based on the terminal voltage according to Ohm's Law:

$$I = \frac{V}{R} \tag{5.1}$$

For this operation mode, the resistance, R, can be chosen from range $10^{-2}\Omega$ to $10^{3}\Omega$.

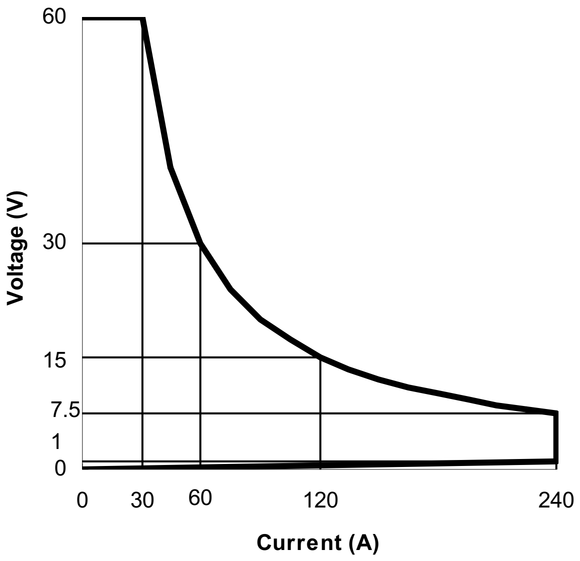


Figure 5.7: The operating curve for the Sorensen SLH 1800-60-240 electric load, [54]

Finally, in the constant power mode, the current is adjusted in such a way to maintain the power constant.

The load can also be controlled via GPIB and analog interface. Control of the electric load will be discussed in section 5.2

5.1.4 Safety Features

The battery cycler is a high-current device, which requires careful safety considerations. The battery itself is a potentially dangerous device, and if operated inappropriately, it may overheat, leak, or even explode. The inclusion of safety features in the CIL setup is, therefore, very important. Figure 5.8 shows important safety features of the battery CIL setup.

Due to very high current levels, the cables (the heavy lines in Figure 5.8) have to be chosen to sustain the largest current that the system can produce. As the highest current

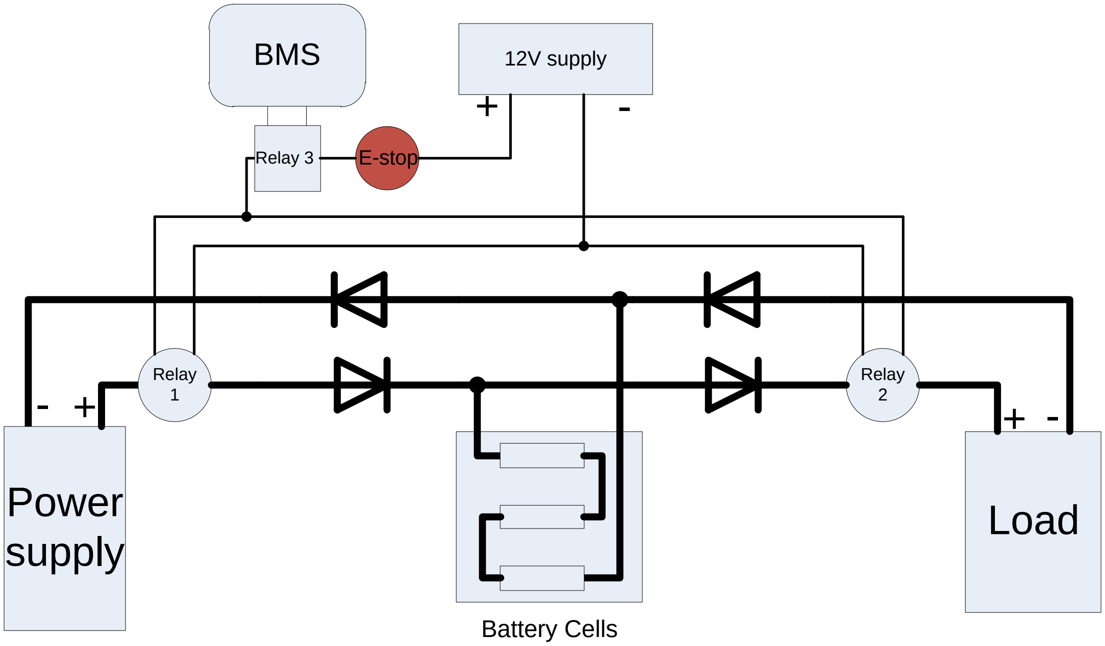


Figure 5.8: Schematic of the electrical circuits

rating belongs to the electric load (240A), 0-AGW (American Wire Gage 0) cables are chosen to connect the battery to the power supply and the electric load.

The four high-current diodes are used to direct power only in the right direction, and prevent back-flow. The two high-current relays (relay 1 and 2) are used to break the circuit whenever a problem occurs. The relays are activated by a separate 12V power supply and are only closed when both the emergency stop switch (a normally closed switch) and the relay connected to the BMS (relay 3) are closed. In a case where there is an error detected by the BMS (such as over/under voltage, over temperature, etc.), the BMS opens the relay, which in turn, breaks the 12V circuit and, therefore, the whole circuit.

5.2 Hardware Control

The real-time target used for solving the high-fidelity model in the previous chapter is a National Instruments PXI platform. National Instruments is well-known for its products for hardware control and automation. The PXI platform provides a compact solution for controlling different pieces of hardware. On our PXI computer, as was mentioned in the previous chapter, there is a two channel CAN bus card as well as an M-series multifunction DAQ card (NI PXI 6289). The main platform itself (NI PXI 1031) has a GPIB, an RS-232, an Ethernet, and four USB terminals. To control the battery cycler, the analog interface and the multifunction DAQ card is used. The NI PXI 6289 DAQ has four analog outputs and 16 analog inputs, as well as 32 digital I/O ports.

The following sections provide the details of controlling the BMS and the battery cycler.

5.2.1 Control of the BMS

As was mentioned earlier, the BMS can be controlled by sending CAN messages. The BMS responds to each CAN message by sending an answer containing the required information.

Before using the BMS, it has to be activated. This can be done by sending it a wake-up frame to the BMS. Once the BMS receives a wake-up message, it waits for a maximum of 200ms for another wake-up message. If the BMS does not receive such message, it goes into sleep mode again. As the CAN bus may be occupied with other messages, it may be required to send the wake-up frame a number of times to successfully activate the BMS. Once the BMS is awakened, it sends back a confirmation CAN message.

When the BMS is active, inquiry CAN messages can be sent to the BMS and it will return another CAN message containing the inquired information. The details of CAN message formats and message descriptions are provided in appendix C. Finally, to turn off the BMS, a shut-down message has to be sent.

5.2.2 Control of the Battery Cycler

The goal of using a battery in the simulation loop is to achieve higher accuracy. Thus, the battery cycler is expected to drive the battery cells according to the reference power with negligible error.

The two actuators – the power supply and the electric load – can be controlled by a remote computer using their analog interfaces. For this purpose, the multifunction DAQ card on the PXI platform is used to write and read analog voltages on the devices' analog interface.

Controlling the Power Supply

The power supply output voltage and current can be controlled by a 0-10V voltage applied to the analog interface terminals. To charge the battery with a specified power, it is easier to adjust the current in constant current mode. Therefore, a PI controller is used to control the power supply current to achieve power tracking. The schematic of the control loop for the power supply is shown in Figure 5.9.

In Figure 5.9, the two measurements of the system are the power supply output current and voltage. Once the output voltage and current are measured, the battery power can be found as the product of the voltage and current. This power is used as the feedback to find the error, which is the input to the PI controller. Finally, the output of the PI controller is a current command to be sent to the power supply.

The measurements for the power supply voltage and current are done through the power supply analog interface and are in forms of analog voltages from 0 to 10V. These analog voltage readings should be scaled to provide meaningful parameters. The 0-10V analog voltages maps into zero to full-scale value of the measured parameter, i.e.:

$$\frac{V_{output}}{V_{full\ scale}} = \frac{V_{analog,V}}{10V} \Rightarrow V_{output} = V_{analog,V} \times \frac{40V}{10V} \tag{5.2}$$

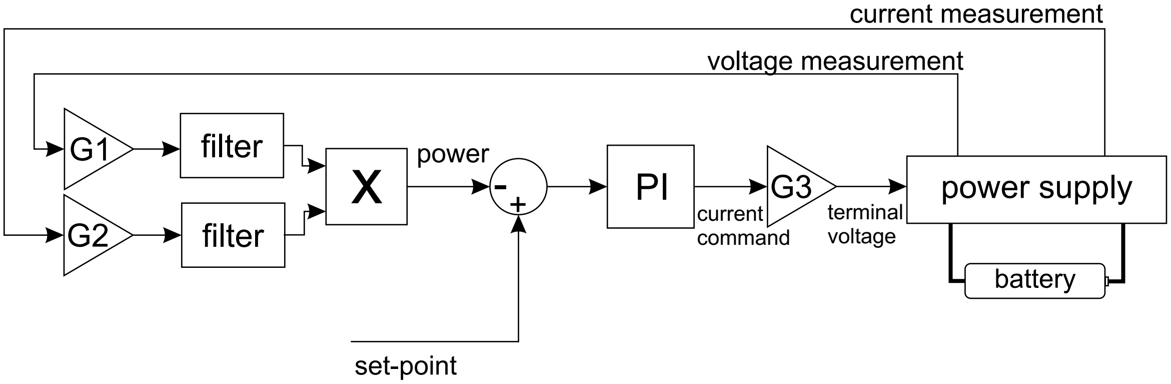


Figure 5.9: The control loop for power supply

$$\frac{I_{output}}{I_{full\ scale}} = \frac{V_{analog,I}}{10V} \Rightarrow I_{output} = V_{analog,I} \times \frac{120A}{10V} \tag{5.3}$$

In these relations, V_{analog} is the voltage measured at the analog interface (0-10V), and subscripts V and I represent the measurements for the power supply voltage and current, respectively. V_{output} and I_{output} are the voltage and current of the battery, and $V_{full\ scale}$ and $I_{full\ scale}$ are the maximum voltage and current ratings of the power supply, which are 40V and 120A, respectively.

In Figure 5.9, the gains G1 and G2 are the gains in (5.2) and (5.3), and are used to scale the analog readings to output voltage and current. G3 is used to scale the PI output (current) to an analog 0-10V voltage ($G3 = G2^{-1}$).

The readings from the power supply are highly affected by environmental noise and should be filtered before use. For this application, first order discrete filters are used.

$$y[n] = ay[n-1] + (1-a)x[n] \tag{5.4}$$

In (5.4), $x[n]$ is the reading from the analog interface, $y[n]$ is the filtered signal, and

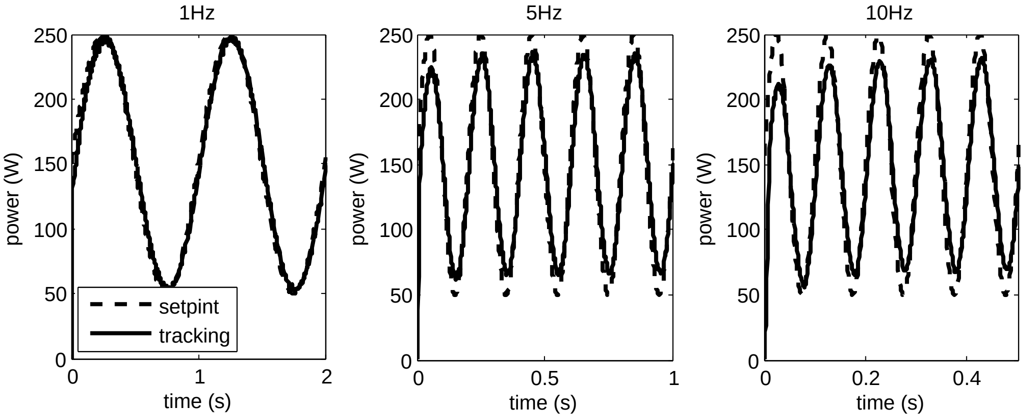


Figure 5.10: Tracking performance of the power supply

$y[n-1]$ is the signal at the previous time step. In this filter, a is the filter coefficient and is chosen to be 0.05 in this setup for both voltage and current measurements.

Once the voltage to be applied to the analog terminal is identified, the LabVIEW DAQmx package can be used to assign the voltage to one of the analog outputs of the multifunction DAQ card.

The tracking performance of the power supply in following positive battery power trajectories (i.e. charging) is shown in Figure 5.10. It can be seen that the power supply can track the required power trajectory up to 5Hz, but it cannot keep up with more rapidly changing set-points. This problem is addressed later in this section.

Controlling the Electric Load

Similar to the power supply, the electric load can be controlled and monitored through the analog interface. The same principles apply here too. The analog voltage of 0-10V maps directly into a 0-full scale parameter.

The electric load, unlike the power supply, can be used in constant power mode. This

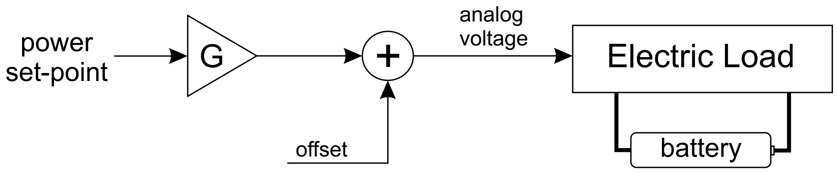


Figure 5.11: Control of the electric load

mode is especially useful for our application of controlling the power drawn from the battery. All that needs to be done in this case is to scale the desired power (0-1800W) to an analog voltage (0-10V) and apply this analog voltage to the electric load using the DAQmx package and the multifunction DAQ card.

The electric load, however, needs calibration, as a simple scaling cannot provide satisfactory results. To achieve better performance, the analog voltage has to be off-set by a small amount, as seen in (5.5).

$$V_{analog} = \text{desired power} \times \frac{10V}{P_{full\ scale}} + \text{offset} \tag{5.5}$$

For this control application, 1822W and 0.0022V (found by trial and error) are chosen for the full-scale power, $P_{full\ scale}$ and off-set voltage, respectively, in order to achieve lower tracking error. The schematic of the electric load controller is shown in Figure 5.11.

Figure 5.12 shows the tracking performance of the electric load in driving the battery power with negative (i.e. discharging) power.

Coupled Control strategy

As can be seen in Figure 5.12, the electric load has a very fast programming time; thus, the battery current can follow the set-point trajectory of up to 20Hz with little error. The

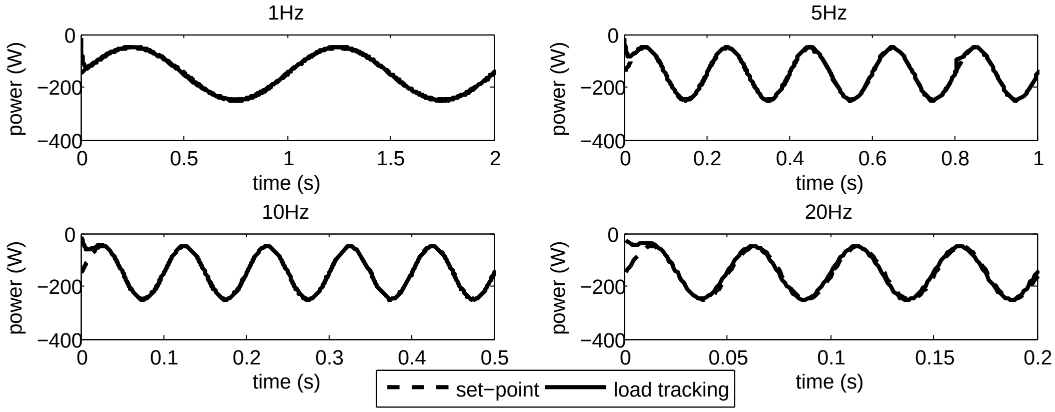


Figure 5.12: Tracking performance of the electric load

power supply on the other hand, cannot keep up with fast changing power trajectories. Fortunately, the power supply has a relatively fast rise time and is slow only in fall time. The difference in the rise and fall times of the power supply is visible in Figure 5.13.

When the power set-point falls faster than power supply capabilities, there is excessive current delivered to the battery (the shaded area in Figure 5.13). To compensate for this slow power supply behavior, the electric load can be used to effectively draw the excessive current that is being delivered by the power supply. Thus, the extra power being delivered (the difference between the actual battery power and the set-point) is added to the electric load command according to (5.6).

$$P_{electricload} = P_{set\ point} + P_{extra} \tag{5.6}$$

With this coupled control strategy, the tracking performance of the whole battery cycler is improved. This improvement is shown in Figure 5.14. As it can be seen, the battery cycler can follow the *fall* of the set-point with less error using the coupled controllers, but still there is an error in *rising* part due to power supply dynamics.

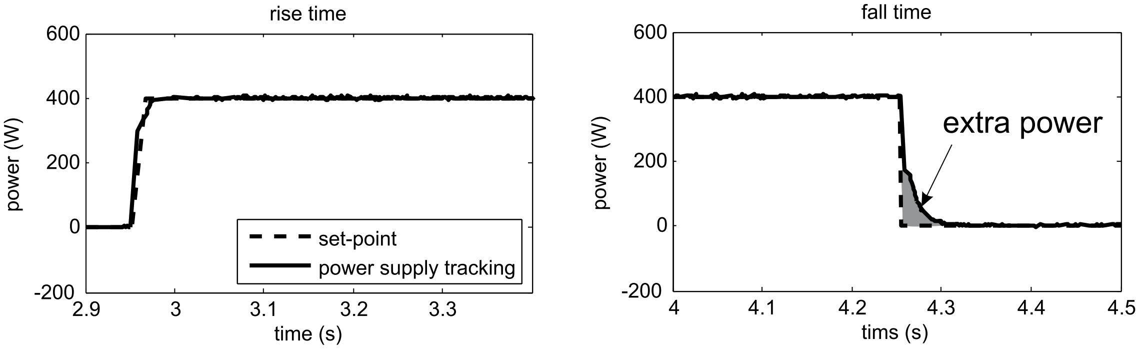


Figure 5.13: Rise time and fall time of the power supply

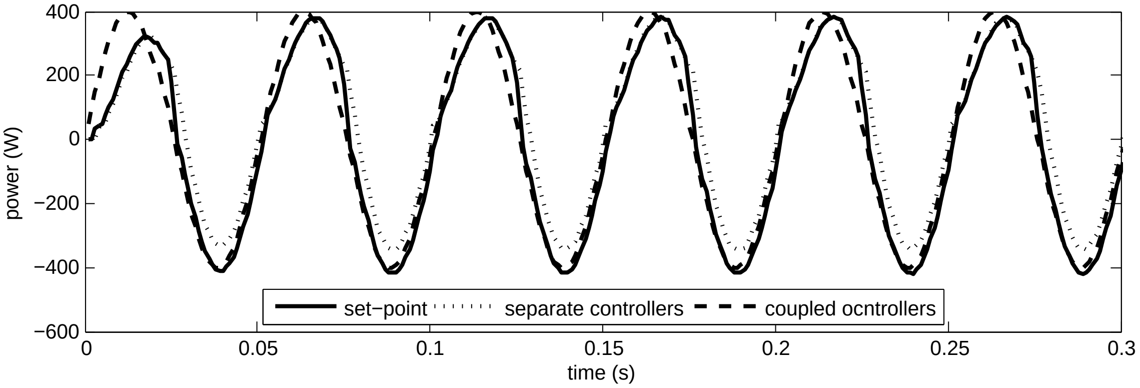


Figure 5.14: Improvement in tracking performance of the battery cycler at 20Hz

5.3 Battery Identification

The CIL simulation setup is used as a part of a hybrid electric powertrain simulation to evaluate the controller performance. As the controllers are designed based on specific system parameters, it is important to consider the GAIA battery cells as the vehicle electrical energy storage systems.

Therefore, to design a better controller, an accurate control-oriented model that is tailored for the GAIA battery cells is essential. To identify the parameters that give the best representation of the cells, a parameter identification study has to be done on this battery.

In most identification procedures, the physical system is excited by an input, and the outputs of the system are compared against the outputs of a mathematical model. Then the parameters in the model are changed in such a way that the output of the model and the physical system become as close as possible.

There are various methods with which the parameters can be identified. Some methods can be used to identify the parameters on-line – such as recursive least-square methods. These methods are especially useful in estimating time-varying parameters.

In off-line methods, the system is excited, and the outputs are stored as a series of timed signals. The stored data is later compared with the output of the model. *Controller-relevant parameter estimation* methods usually follow the same logic to identify the model parameters, which are later used in controller development.

The input by which the physical system is excited is important. If the system is excited with only one frequency, only the response of the system to that particular frequency appear in the output; whereas, if the system is shaken with a larger range of frequencies, more characteristics of the system can be extracted.

To identify the battery cells, the offline method is used. The batteries are excited with a known power, and the battery state of charge is recorded as the output of the system. The model for which the parameters should be identified is the simple model that was

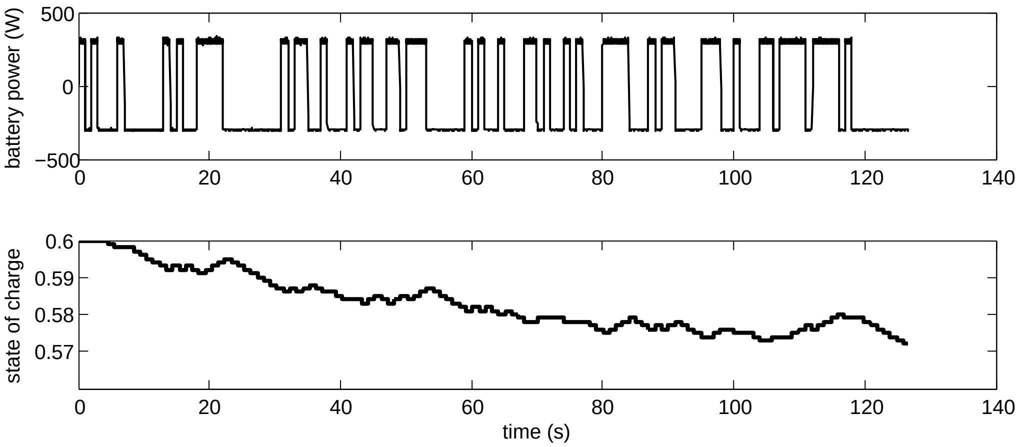


Figure 5.15: The excitation input and the resulting output of battery

previously used to design the optimal controller in chapter 2. The model is repeated in (5.7).

$$\dot{SoC} = \frac{-V_{oc} + \sqrt{V_{oc}^2 - 4RP_b}}{2RQ} \tag{5.7}$$

The parameters to be identified are the open circuit voltage (V_{oc}), the equivalent series resistance (R), and the battery capacity (Q). As was mentioned earlier, the excitation input is the battery power (P_b), and the output is the state of charge.

The excitation power input is chosen as a pseudo-random binary Sequence (PRBS), which contains a broad range of frequencies. The PRBS power input to the battery cells and the change in their state of charge are shown in Figure 5.15.

Finally, Matlab's optimization toolbox is used to find the set of parameters that make the model in (5.7) give close results to the real system. Among the optimization algorithms in Matlab, the Genetic Algorithm (GA) is one of the global optimization methods that can solve constrained optimization problems, and it is used in this parameter identification

Table 5.2: Parameters in the identification problem

parameter	lower boundary	identified value	nominal value
R	1mΩ	21mΩ	19.5mΩ
V_{oc}	10 V	10.699 V	10.80 V
Q	100 As	29729.2 As	27000 As

process. In this optimization problem, the objective function to be minimized is the sum of the square of error in each time step:

$$error = \sum \left(SoC_{model} - SoC_{experiment}\right)^2 \tag{5.8}$$

To calculate the error, the state of charge data in Figure 5.15 is interpolated in each time step, and the difference between the model output and the interpolated data is squared and summed to form the error.

Since the parameters of the model in (5.7) have physical meaning, they cannot assume any number. For example, the open circuit voltage has to be close to the terminal voltage of the cells. Therefore, the lower limits presented in Table 5.2 are specified for the parameters in the optimization problem. Table 5.2 also presents the solution of the GA algorithm, with the initial population $V_{oc} = 10.6V$, $R = 0.01\Omega$, $Q = 30000As$, population size of 100, and 100 generations.

To validate the identified model, the batteries are excited with a different input (a chirp signal). The input power and the comparison of the state of charge between the identified battery model and the experimental data are shown in Figure 5.16.

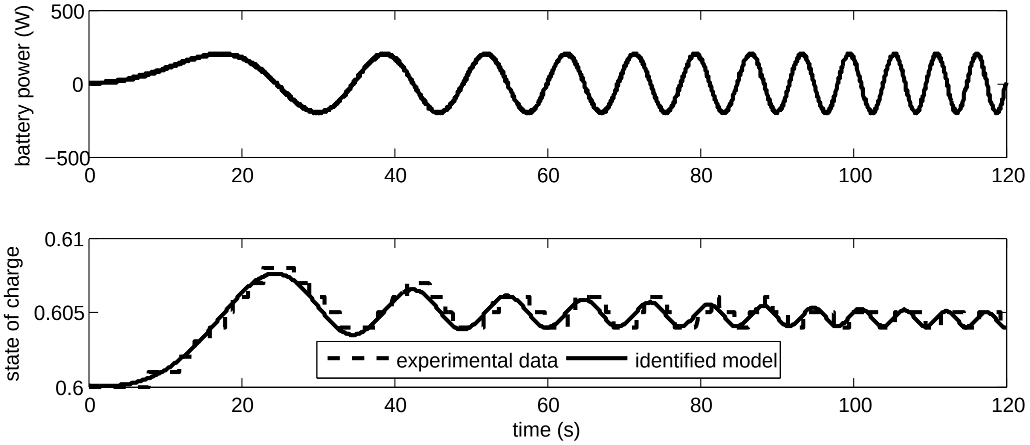


Figure 5.16: Comparison of the state of charge trajectories between the identified model and the physical battery

5.4 Component Scaling Using Buckingham's Pi Theorem

When a full-size system needs to be simulated with a smaller prototype, a meaningful scaling has to be conducted to achieve acceptable results. Dimensional analysis, especially in fluid and thermal systems, are widely used to relate the phenomena that are similar in behavior but different in size.

Similarly, the developed CIL setup uses a scaled-down battery to simulate the full size battery pack in HEVs. To accurately simulate the HEV battery with the cells in the CIL setup, the batteries must be scaled properly.

In this setup, the scaling of the battery cells occurs at the inputs and outputs of the battery cycler. When two systems follow the same principles and only differ in the value of the parameters, Buckingham's Pi Theorem can be used to map one system to the other. The formal statement for the Pi Theorem is as follows [55]:

> If an equation in n arguments is dimensionally homogeneous with respect to m fundamental units, it can be expressed as a relation between $n-m$ independent dimensionless arguments.

In the battery analysis, the following variables (or arguments) need be considered:

- SoC: battery state of charge
- P: battery power;
- V: battery voltage;
- I: battery current;
- Q: battery capacity;
- R: equivalence series resistance; and
- t: a characteristic time.

The battery state of charge itself is a dimensionless parameter, and we consider it as the output of the system. Therefore, as long as other dimensionless groups of the systems are the same, the state of charge of the two systems will also prove equivalent.

The battery power is the input to the battery cycler, and it is the parameter that must be scaled properly before being used to drive the battery. The final goal of this dimensional analysis is to identify such a scaling factor for the battery power.

The parameters mentioned above are made from four fundamental units: [M]: mass, [L]: length, [T]: time, and [A]: current. Thus the parameters can be written as functions of the four fundamental units, as in Table 5.3.

Since the dimensional bundle of $[M][L]^2$ appears together, it can be considered as one fundamental unit; therefore, the Pi Theorem states that the system (battery) can be presented by the $6-3=3$ dimensionless groups. There is no unique set of dimensionless

Table 5.3: Important parameters in battery analysis and their dimensions

parameter	dimension
P	$[M][L]^2[T]^{-3}$
V	$[M][L]^2[T]^{-3}[A]^{-1}$
I	$[A]$
t	$[T]$
Q	$[A][T]$
R	$[M][L]^2[T]^{-3}[A]^{-2}$

Table 5.4: Groups of dimensionless parameters in battery analysis

dimensionless group	parameter	relation
π_1	voltage	$\frac{P}{VI}$
π_2	capacity	$\frac{It}{Q}$
π_2	resistance	$\frac{RI^2}{P}$

groups, and in this analysis, t, I, and P, are chosen as the primary parameters. For the remaining parameters, dimensionless groups of Table 5.4 is formed.

In this experimental setup, a number of battery cells are to represent the full battery pack. As both systems have the same chemistry, the dynamics of the two systems are similar, and the characteristic time was chosen to be the discharge time, which is related to the battery power and capacity. Since the battery pack and the cells in the CIL setup should behave similarly, the following relations has to be satisfied:

$$\pi_{1BP} = \pi_{1GAIA} \tag{5.9}$$

$$\pi_{2BP} = \pi_{2GAIA} \tag{5.10}$$

$$\pi_{2BP} = \pi_{2GAIA} \tag{5.11}$$

In the above relations, the battery pack and the cells are denoted by the subscripts $_{\mathrm{BP}}$ and $_{\mathrm{GAIA}}$, respectively. Substituting the Pi relations in Table 5.4 leads to:

$$\left[\frac{P}{VI}\right]_{BP} = \left[\frac{P}{VI}\right]_{GAIA} \Rightarrow P_{CIL} = \frac{V_{GAIA}}{V_{BP}}\frac{I_{GAIA}}{I_{BP}}P_{BP} \tag{5.12}$$

$$\left[\frac{It}{Q}\right]_{BP} = \left[\frac{It}{Q}\right]_{GAIA} \Rightarrow \frac{I_{GAIA}}{I_{BP}} = \frac{Q_{GAIA}}{Q_{BP}}\frac{t_{BP}}{t_{GAIA}} \tag{5.13}$$

$$\left[\frac{RI^2}{P}\right]_{BP} = \left[\frac{RI^2}{P}\right]_{GAIA} \Rightarrow \frac{P_{GAIA}}{I_{GAIA}^2} = \frac{P_{BP}}{I_{BP}^2}\frac{R_{GAIA}}{R_{BP}} \tag{5.14}$$

By combining (5.12) and (5.13), one relation for power and capacity can be found:

$$P_{GAIA} = \left[\frac{V_{GAIA}}{V_{BP}}\frac{Q_{GAIA}}{Q_{BP}}\frac{t_{BP}}{t_{GAIA}}\right]P_{BP} \tag{5.15}$$

As the simulations have to be in real-time, the characteristic times of both systems are equal, and the scaling factor is reduced to:

$$\frac{P_{GAIA}}{P_{BP}} = \frac{V_{GAIA}}{V_{BP}}\frac{Q_{GAIA}}{Q_{BP}} \tag{5.16}$$

Therefore, the battery power has to be scaled according to (5.16) before it is sent to the battery cycler to drive the battery cells.

It is important to notice that it may not be possible to map one system to the other by just a simple scaling. In this case, once the power is scaled according to (5.16), the last Pi relation, (5.14), may or may not be satisfied. This is because the internal resistance of the battery is an independent parameter and may not be scalable. To better understand this situation, assume two battery cells with the same capacity and voltage, but different internal resistances. The difference may be due to build effects, battery wear, etc. As all of the parameters but the resistance are the same, the first two Pi groups, (5.12) and (5.13),

Table 5.5: Nominal battery parameters used for scaling

parameter	RX400-h battery pack	GAIA cells
voltage(V)	288.0	10.8
capacity (Ahr)	6.5	7.5

are essentially the same for the two batteries, but nothing can be done to make (5.14) equal.

This apparent inconsistency with battery Pi groups can be solved by involving more parameters, such as an electro-chemical parameter; however, this type of analysis is out of the scope of this thesis, and the sole power scaling meets the requirements of this work.

The GAIA Li-ion cells in the CIL setup are used to simulate HEV battery packs. The nominal values of the GAIA battery parameters and the nominal values of a full size battery pack (Lexus RX400-h) are presented in Table 5.5. With these parameters, the scaling factor can be calculated according to (5.17).

$$\frac{P_{GAIA}}{P_{BP}} = \frac{V_{GAIA}}{V_{BP}}\frac{Q_{GAIA}}{Q_{BP}} = \frac{10.8V}{288.0V} \times \frac{7.5Ahr}{6.5Ahr} = 43.27 \times 10^{-3} \tag{5.17}$$

Identified Battery Model as Control-Oriented Model

The optimal powertrain controller was previously applied to the high fidelity model that included a chemistry-based battery model; therefore, the controller had been designed for that battery. For the HIL simulations, the same high-fidelity model was used; therefore, the controller was not changed. However, for the CIL simulations, the controller is going to be applied to a model that contains the physical batteries, which is different from the battery model that was previously used. Thus, the controller has to be tailored for the GAIA battery cells, instead of the chemistry-based model.

The previously identified battery parameters can be used in conjunction with the scaling method to find a simple model to be used as the control-oriented battery model.

The target battery is a battery the same size as the Lexus RX400-h battery pack with the nominal values specified in Table 5.5. The identified capacity and voltage of the battery cells should, therefore, be scaled according to (5.18) and (5.19), respectively, to be used in the control-oriented model.

$$\frac{Q_{GAIA}}{Q_{BP}} = \frac{Q_{ID}}{Q_{COM}} \Rightarrow Q_{COM} = Q_{ID} \times \frac{Q_{BP}}{Q_{GAIA}} = 8.26Ahr \times \frac{6.5Ahr}{7.5Ahr} = 7.16Ahr \tag{5.18}$$

$$\frac{V_{GAIA}}{V_{BP}} = \frac{V_{ID}}{V_{COM}} \Rightarrow V_{COM} = V_{ID} \times \frac{V_{BP}}{V_{GAIA}} = 10.699V \times \frac{288V}{10.8V} = 285.3V \tag{5.19}$$

In these relations, the nominal GAIA cell parameters are denoted by the subscript $_{GAIA}$, nominal full-size battery pack parameters by the subscript $_{BP}$, identified GAIA parameters by the subscript $_{ID}$, and control-oriented model parameters by the subscript $_{COM}$.

To properly scale the resistance, the new dimensionless parameter in (5.20) can be used to relate the identified parameters to the scaled-up control-oriented model.

$$\pi_4 = \frac{RQ}{Vt} \tag{5.20}$$

Again, as the simulations should have the same time scale, the characteristic times are equal, and the resistor can be scaled according to (5.21).

$$R_{COM} = \frac{V_{GAIA}}{V_{ID}} \frac{Q_{ID}}{Q_{GAIA}} R_{ID} = \frac{285.3V}{10.699V} \frac{8.26Ahr}{7.16Ahr} \times 21m\Omega = 646.6m\Omega \tag{5.21}$$

With these up-scaled identified parameters in the control-oriented model, the series HEV powertrain controller of chapter 2 is re-tuned. The CIL simulation results for this controller concludes this chapter.

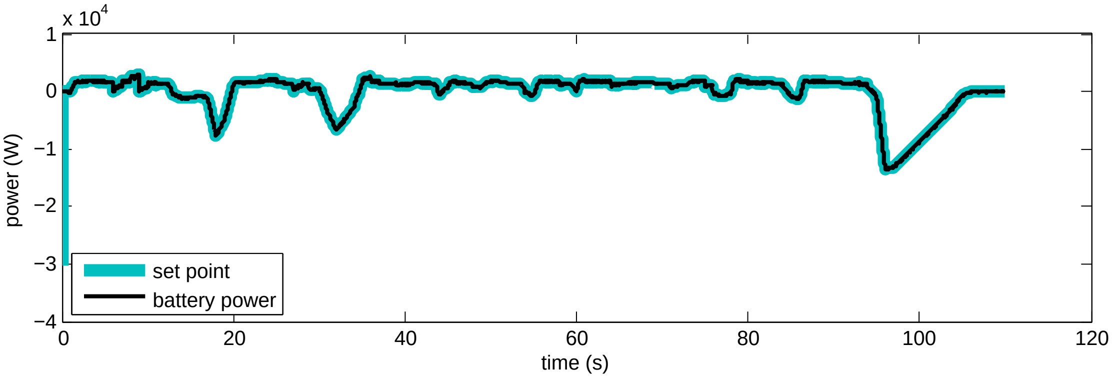


Figure 5.17: Tracking performance of the battery cycler

5.5 CIL Simulation Results

Figure 5.17 shows the perfect tracking performance of the battery cycler in tracking the battery power set-points, resulting from the simulation of the FTP75 drive cycle.

In the CIL simulation of the series HEV powertrain, all components except the battery are the same as the HIL simulation. Therefore, the only different result would be the state of charge trajectory of the battery cells, driven by the battery cycler. Figure 5.18 shows the state of charge trajectory of these cells, and what the controller had predicted based on the new control-oriented model, for the first part of the FTP75 drive cycle. As can be seen, the controller can successfully predict the system's behavior, using the new control-oriented model.

It should be noted that the Li-ion battery parameters, unlike NiMH batteries, change with variations of state of charge. However, in this simulation, and in every HEV operation, the variation of state of charge is small; thus the battery parameters remain very close to the identified parameters. This assumption was also made in the controller design process, and now can be justified by the CIL simulation results.

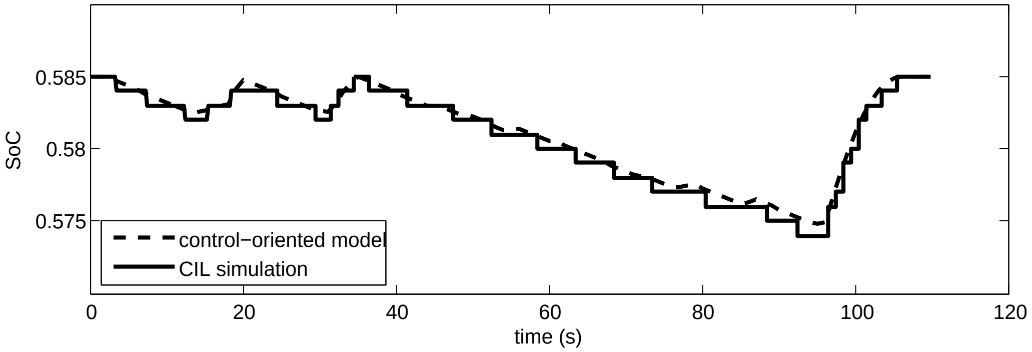


Figure 5.18: CIL simulation results for the state of charge trajectory for the first part of the FTP75 drive cycle

5.6 Chapter Summary

In this chapter, the description of the battery CIL simulation setup was presented. It was mentioned that under proper scaling of the battery parameters, the CIL simulation setup can be used to simulate full-size battery packs in HEVs.

The battery CIL setup was shown to be useful in identifying battery parameters, which can later be used in control-oriented models, and the simulation results showed that such a method can result in a successful controller development.

Chapter 6

Driver-in-the-Loop (DIL) Simulation

The driver is the last part of the control loop in an HEV simulation that has to be considered. In fact, the driver is the most critical part of the simulations that needs to be realistically involved in the control loop – as the erratic behavior of the driver is the most unpredictable phenomenon in an HEV powertrain. A successful optimal controller is not the one that gives the lowest fuel consumption for a specific drive cycle, but it is the one that can handle the uncertainty of the driver's behavior while giving *near-optimal* fuel consumption. In this chapter, the development of a Driver-in-the-Loop (DIL) simulation setup will be discussed. The DIL setup provides a way to gather the inputs from a human driver in a fairly realistic environment using a set of steering wheel/shifter/pedals.

6.1 Hardware Description

The DIL setup is constructed upon the existing setup by addition of a set of gaming devices, including a steering wheel, a gear shifter, and a set of pedals. The completed HEV controller evaluation setup that includes the DIL test bench is shown in Figure 6.1.

The gaming device is connected to the host computer (the laptop) by USB connection. To use this device in LabVIEW environments, a driver for receiving and interpreting the

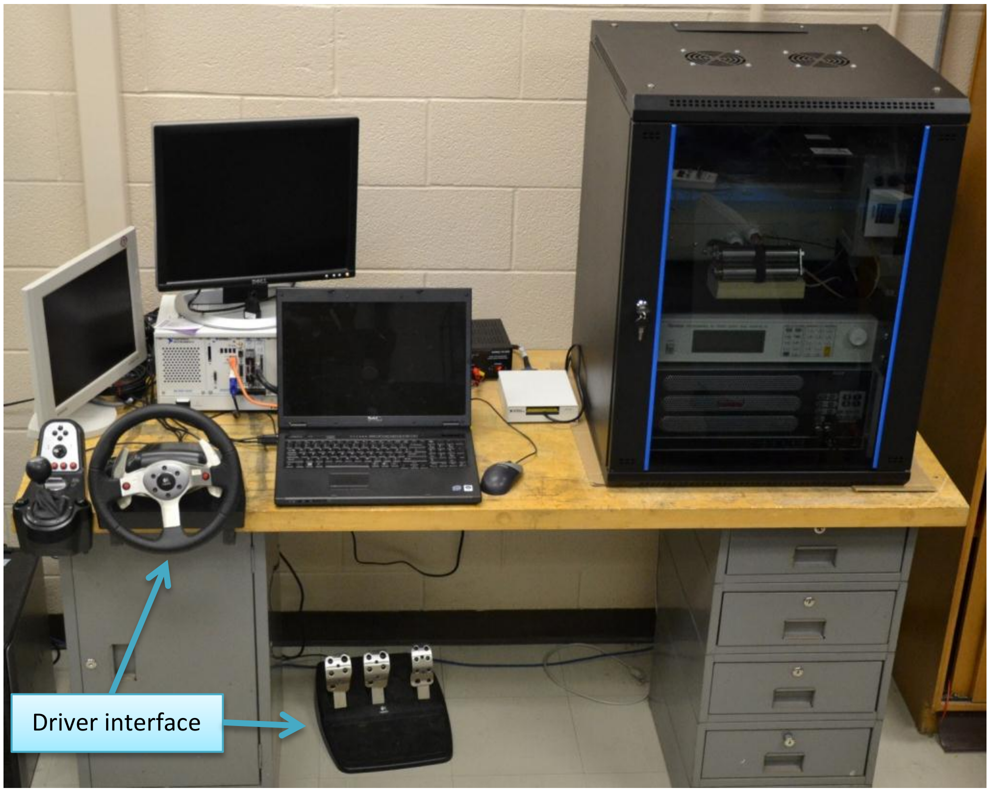


Figure 6.1: The driver interface included in the setup

serial data coming from the gaming device was developed by Dr. Thomas Uchida. The laptop that runs the Windows version of LabVIEW obtains the inputs from the gaming device, processes them as the driver commands, and sends them to the real-time target as the input of the vehicle dynamics block in the high-fidelity model.

The two inputs from the gas and brake pedals are interpreted as the electric motor current by multiplying the position value by a gain. In this case, a positive current command is produced by pushing the gas pedal, and a negative current is produced by pushing the brake pedal. The steering wheel of the gaming device is used to steer the front wheels in the vehicle dynamics model. The steering system in the vehicle model is a simple rotation of the wheels about the vertical axis, and both wheels are steered by the same amount.

Table 6.1: Effect of lane-change maneuvers on the fuel consumption

simulation	fuel consumption (grams)	final SoC
fixed steering	12.02	0.6001
lane change	12.21	0.5998

The pedals and steering wheel gains are tuned to provide a close-to-reality response to the driver.

6.2 DIL Simulation Results

The DIL simulations are used to identify the effects of the driver's behavior on the performance of the designed controllers. In the first test, two simulations were run: once with the steering wheel fixed and once with random lane-change maneuvers, while the speed profiles followed the FTP75 drive cycle (reference) exactly. The fuel consumption for the two tests are presented in Table 6.1. It can be seen that the fuel consumption is increased by 1.6% when there are lateral maneuvers. The results are acceptable, since the lane change maneuvers require extra energy that has to come from the engine. However, despite the increase in power request and fuel consumption, the controller can still keep the final state of charge at the reference value. Since the optimal controller mechanism is independent of the power demand and as the system remains charge-sustaining, it can be inferred that the controller gives the optimal solution even when there are unpredictable lateral maneuvers.

In the previous test, the speed profiles were the same, and the vehicle had followed the drive cycle precisely. However, a more challenging test for the controller would be random deviations from the reference drive cycle. To study the effect of the driving pattern, three simulations were done. In the first simulation, a PID controller was used as the driver to follow the reference drive cycle (FTP75). In the other two rounds, a human driver was in

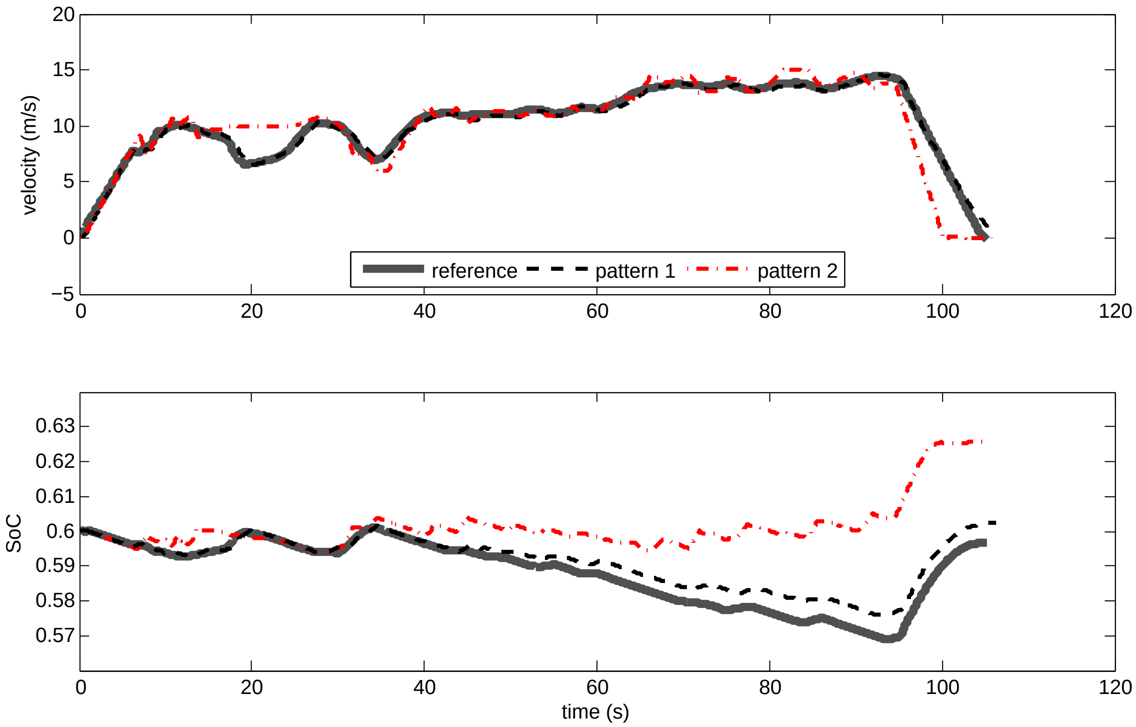


Figure 6.2: Velocity profile and state of charge trajectory for different driving patterns

charge of driving the vehicle by sending power request signals to the simulation. In the first driver-in-the-loop simulation, the driver goal was to follow the reference drive cycle with small deviations. In the second DIL simulation, the driver employed a more aggressive driving pattern with faster acceleration and deceleration rates. In the latter case, even some parts of the drive cycle were altered. The simulation results for these three driving patterns are shown in Figure 6.2. Moreover, the fuel consumption for the three simulations is presented in Table 6.2.

Simulation results show that the driving pattern has much greater effect on the fuel consumption and the performance of the controller, than the lateral maneuvers. In driving the first pattern, the driver has tried to follow the drive cycle with little error, and the results (state of charge and fuel consumption) are very similar to the reference. In this case, the increase in fuel consumption is mostly due to the bigger fluctuations in the electric power demand, which causes less use of the battery and more use of the engine, thus increasing the final state of charge and fuel consumption. From this simulation, it is evident that the controller can handle small variations in speed profile fairly well. However, in the case of the aggressive driving pattern, the controller fails to keep the final state of charge at the reference level, and due to higher acceleration, the fuel consumption is increased. This system behavior can be explained as follows. The controller is expecting gentle acceleration and deceleration rates; therefore, it limits the battery power, and most of the required energy for the fast accelerations comes from the engine, which increases the fuel consumption. Moreover, the estimated regenerative braking energy is less than what is really available. This makes the battery become charged more than expected, which increases the state of charge. The increased state of charge roughly means that the engine has put extra charge in the battery. If the controller had predicted the regenerative braking energy correctly, the battery could be used more, and the fuel consumption would be reduced.

6.3 Chapter Summary

In this chapter, the development of a driver-in-the-loop simulation setup was discussed. The DIL setup was used to evaluate the performance of the designed optimal controller for a

Table 6.2: Fuel consumption for three different driving pattern

pattern	fuel consumption (grams)	final state of charge
reference	11.31	0.596
pattern 1	12.83	0.606
pattern 2	21.43	0.625

series HEV, and the results revealed that the designed controller can handle small variations in the driving pattern (such as lane changing maneuvers and small speed variation) very well. However, it failed to keep the system charge sustaining (hence optimal) in the case of noticeable deviation from the reference drive cycle for which the controller had been tuned. Therefore, the controller has to be modified to enable it to consider such variations in advance, for example by using GPS and onboard radar systems.

Chapter 7

Conclusions

7.1 Summary

In this work, the problem of optimal control of hybrid electric vehicles was explored. In the first part of this research, a real-time, near-optimal controller for a series HEV was developed. This controller was proven to be optimal under certain conditions, which were later shown to be valid assumptions in this problem. Aside from this development, based on the drive cycle ahead of the vehicle, a method was proposed that could be used to tune the controller.

The remainder of this thesis was concerned with the testing of the designed controller. First, the controller performance was evaluated by the use of a high-fidelity model in a model-in-the-loop simulation. Using this simulation, the controller and the estimation method were shown to be successful in providing close-to-optimal behavior. Once the controller strategy had successfully passed the MIL simulation, it was coded into a real electronic control unit for HIL simulations. In the HIL simulation, hidden aspects of the control loop, such as the limited computational resources in the controller and communication delays, could be revealed. The results revealed that the controller performance was not affected significantly by such issues.

A major portion of this work was related to the component-in-the-loop simulation. In this simulation, the controller was applied to a system that included a physical battery instead of its mathematical model. The CIL simulation setup consisted of a real-time battery cycler, which was in constant communication with the model solver. As the controller had to be tailored for the battery cells, they were identified in the form of a simple model and were scaled to the size of an HEV battery pack. The results showed that the updated controller could successfully capture the system's dynamics – as the controller prediction and CIL simulation results were very close.

Finally, a human driver was included in the simulations by a driver-in-the-loop simulation setup. It was shown that the controller was able to handle small variations in the driving pattern, but it required more information to be able to provide optimal results for significantly different patterns.

In the end, it is worth noting that although the presented test bench was used for evaluating the designed controller, it can also be used for evaluating future controllers, as the test platforms are designed to be flexible.

7.2 Future Work

This worked involved the designing and testing of a series HEV controller. Although the results showed strong potential for the designed controller, there is still room for improvement in both design and testing parts.

Controller design:

In the controller design, one of the crucial areas for future research is the combination of the tuning method (presented in chapter 2) with a route estimation method that can predict future driving conditions with acceptable accuracy. This can be done by using route information such as speed limits, road grade, and traffic information, as well as onboard vehicle sensors such as Radar and GPS.

Another area of interest would be the integration of this controller with an adaptive cruise controller to reduce the deviation from the reference speed profile – thereby improving the controller performance and fuel economy.

The optimal controller was designed to minimize the fuel consumption only – the emissions (such as NOx, CO2, and HC) were not considered. Therefore, the emissions need to be included in the controller design as well. This, in turn, requires the development of a simple yet realistic model of the engine that can estimate emissions.

MIL simulations:

In software simulations, the high-fidelity model can be improved by including more accurate models. One major improvement would be the modeling of the power electronics in the electric drives. Likewise, the engine model can be further improved by using a more complex model such as the two-zone engine model, in order to provide a better estimation of fuel consumption.

CIL simulations:

Including more components (such as an internal combustion engine and electric machines) will definitely enhance the simulation fidelity. Including an internal combustion engine into the simulation loop can especially be beneficial, as the engine has a very complex dynamics, and it is difficult to model.

DIL simulations:

The developed driver-in-the-loop simulation setup can be further expanded by adding a traffic simulator with a graphical user interface. This addition will enhance the driver feel and simulation validity.

References

[1] J. D. Gonder, “Route-based control of hybrid electric vehicles,” in *SAE 2008 World Congress*, 2008.

[2] A. Sciarretta and L. Guzzella, “Control of hybrid electric vehicles,” *Control Systems, IEEE*, vol. 27, no. 2, pp. 60 –70, april 2007.

[3] L. Serrao, S. Onori, and G. Rizzoni, “A comparative analysis of energy management strategies for hybrid electric vehicles,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, no. 3, p. 031012, 2011.

[4] D. V. Ngo, T. Hofman, M. Steinbuch, and A. Serrarens, “An optimal control-based algorithm for hybrid electric vehicle using preview route information,” in *American Control Conference (ACC), 2010*, June 30- July 2 2010, pp. 5818 –5823.

[5] C. Musardo, G. Rizzoni, and B. Staccia, “A-ecms: An adaptive algorithm for hybrid electric vehicle energy management,” in *Conference on Decision and Control and European Control Conference. CDC-ECC*, Dec. 2005, pp. 1816 – 1823.

[6] C.-C. Lin, H. Peng, J. Grizzle, and J.-M. Kang, “Power management strategy for a parallel hybrid electric truck,” *Control Systems Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 839 – 849, nov. 2003.

[7] A. Boyali and L. Guvenc, “Real-time controller design for a parallel hybrid electric vehicle using neuro-dynamic programming method,” in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, oct. 2010, pp. 4318 –4324.

[8] X. Lin, H. Banvait, S. Anwar, and Y. Chen, “Optimal energy management for a plug-in hybrid electric vehicle: Real-time controller,” in *American Control Conference (ACC), 2010*, june 30-july 2 2010, pp. 5037 –5042.

[9] A. Piccolo, L. Ippolito, V. zo Galdi, and A. Vaccaro, “Optimisation of energy flow management in hybrid electric vehicles via genetic algorithms,” in *Advanced Intelligent Mechatronics, 2001. Proceedings. 2001 IEEE/ASME International Conference on*, vol. 1, 2001, pp. 434 –439.

[10] W. Li, G. Xu, H. Tong, and Y. Xu, “Design of optimal, robust energy management strategy for a parallel hev,” in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, dec. 2007, pp. 1894 –1899.

[11] C. C. Lin, H. Peng, and J. Grizzle, “A stochastic control strategy for hybrid electric vehicles,” in *American Control Conference (ACC), 2004*, vol. 5, june 30-july 2 2004, pp. 4710 –4715.

[12] H. Chin and A. Jafari, “Design of power controller for hybrid vehicle,” in *System Theory (SSST), 2010 42nd Southeastern Symposium on*, march 2010, pp. 165 –170.

[13] C. Dextreit, F. Assadian, I. V. Kolmanovsky, J. Mahtani, and K. Burnham, “Hybrid electric vehicle energy management using gametheory,” SAE, Technical Paper, 2008.

[14] A. Soltis and X. Chen, “A new control strategy for hybrid electric vehicles,” in *American Control Conference (ACC), 2003*, vol. 2, July, 2003, pp. 1398 – 1403.

[15] H. Borhan, C. Zhang, A. Vahidi, A. Phillips, M. Kuang, and S. Di Cairano, “Nonlinear model predictive control for power-split hybrid electric vehicles,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*, dec. 2010, pp. 4890 –4895.

[16] A. Taghavipour, N. L. Azad, and J. McPhee, “An optimal power management strategy for power split plug-in hybrid electric vehicles,” *International Journal of Vehicle Design*, in press.

[17] B. Sampathnarayanan, L. Serrao, S. Onori, G. Rizzoni, and S. Yurkovich, “Model predictive control as an energy management strategy for hybrid electric vehicles,” *ASME Conference Proceedings*, vol. 2009, no. 48937, pp. 249–256, 2009. [Online]. Available: http://link.aip.org/link/abstract/ASMECP/v2009/i48937/p249/s1

[18] D. E. Kirk, *Optimal Control Theory: An Introduction.* Dover Publications, Inc., 2004.

[19] N. Kim, S. Cha, and H. Peng, “Optimal control of hybrid electric vehicles based on pontryagin’s minimum principle,” *Control Systems Technology, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –9, 2011.

[20] N. Kim, S. W. Cha, and H. Peng, “Optimal equivalent fuel consumption for hybrid electric vehicles,” *Control Systems Technology, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –9, 2011.

[21] R. Cipollone and A. Sciarretta, “Analysis of the potential performance of a combined hybrid vehicle with optimal supervisory control,” in *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, oct. 2006, pp. 2802 –2807.

[22] L. Serrao and G. Rizzoni, “Optimal control of power split for a hybrid electric refuse vehicle,” in *American Control Conference (ACC), 2008*, june 2008, pp. 4498 –4503.

[23] S. Stockar, V. Marano, G. Rizzoni, and L. Guzzella, “Optimal control for plug-in hybrid electric vehicle applications,” in *American Control Conference (ACC), 2010*, june 30-july 2 2010, pp. 5024 –5030.

[24] D. Ambuhl and L. Guzzella, "Predictive reference signal generator for hybrid electric vehicles," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 9, pp. 4730 –4740, nov. 2009.

[25] L. Serrao, S. Onori, and G. Rizzoni, "Ecms as a realization of pontryagin's minimum principle for hev control," in *American Control Conference (ACC), 2009*, june 2009, pp. 3964 –3969.

[26] A. Sciarretta, M. Back, and L. Guzzella, "Optimal control of parallel hybrid electric vehicles," *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 3, pp. 352 – 363, may 2004.

[27] Z. Xiaowei, H. Hongwen, and X. Rui, "Hardware in loop simulation for vehicle controller in hev based on dspace," in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, vol. 2, aug. 2010, pp. 489–492.

[28] L. F. Xu, J. Q. Li, J. F. Hua, X. J. Li, and M. G. Ouyang, "Hardware in the loop simulation of vehicle controller unit for fuel cell/battery hybrid bus," in *Vehicle Power and Propulsion Conference, 2009. VPPC '09. IEEE*, sept. 2009, pp. 1777 –1782.

[29] Y.-H. Hung, C.-H. Wu, S.-M. Lo, B.-R. Chen, E.-I. Wu, and P.-Y. Chen, "Development of a hardware in-the-loop platform for plug-in hybrid electric vehicles," in *Computer Communication Control and Automation (3CA), 2010 International Symposium on*, vol. 1, may 2010, pp. 45 –48.

[30] L. Gauchia and J. Sanz, "A per-unit hardware-in-the-loop simulation of a fuel cell/battery hybrid energy system," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 4, pp. 1186 –1194, april 2010.

[31] C. Dufour, T. Ishikawa, S. Abourida, and J. Belanger, "Modern hardware-in-the-loop simulation technology for fuel cell hybrid electric vehicles," in *Vehicle Power and Propulsion Conference (VPPC), 2007*, sept. 2007, pp. 432 –439.

[32] Y. Xiao-kun, H. Hong-wen, P. Lian-yun, and Z. Xiaolin, "Hardware-in-the-loop simulation on a hybrid power system," in *Power Electronics Systems and Applications (PESA), 2011 4th International Conference on*, june 2011, pp. 1 –5.

[33] Z. Hui, L. Cheng, and Z. Guojiang, "Design of a versatile test bench for hybrid electric vehicles," in *Vehicle Power and Propulsion Conference (VPPC), 2008*, sept. 2008, pp. 1 –4.

[34] J. Lee, N. Jeon, and H. Lee, "Hil simulation approach for feasibility study of a tram with an onboard hybrid energy storage system," in *ICCAS-SICE, 2009*, aug. 2009, pp. 5305 –5312.

[35] J.-M. Timmermans, J. Van Mierlo, P. Lataire, F. Van Mulders, and Z. McCaffree, "Test platform for hybrid electric power systems: Development of a hil test platform," in *Power Electronics and Applications, 2007 European Conference on*, sept. 2007, pp. 1 –7.

[36] M. D. Petersheim and S. N. Brennan, "Scaling of hybrid-electric vehicle powertrain components for hardware-in-the-loop simulation," *Mechatronics*, vol. 19, no. 7, pp. 1078 – 1090, 2009, special Issue on Hardware-in-the-loop simulation. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957415809001330

[37] R. S. Razavian, N. L. Azad, and J. McPhee, "On real-time optimal control of a series hybrid electric vehicle with an ultra-capacitors," in *American Control Conference (ACC), 2012*, june 2012.

[38] R. S. Razavian, A. Taghavipour, N. L. Azad, and J. McPhee, "Design and evaluation of a real-time optimal control system for series hybrid electric vehicles," *International Journal of Electric and Hybrid Vehicles*, *under review*.

[39] T.-S. Dao, A. Seaman, and J. McPhee, "Mathematics-based modeling of a series-hybrid electric vehicle," in *5th Asian Conference on Multibody Dynamics*, August 2010.

[40] T.-S. Dao and J. McPhee, "Dynamic modeling of electrochemical systems using linear graph theory," *Journal of Power Sources*, vol. 196, no. 23, pp. 10 442 – 10 454, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378775311016089

[41] M. Saeedi, "A mean value internal combustion engine model in maplesim," Master's thesis, University of Waterloo, Department of Mechanical and Mechatronics Engineering, 2010.

[42] H. P. Geering, *Optimal Control with Engineering Applications.* Springer, 2007.

[43] M. Barth, G. Scora, and T. Younglove, "Intelligent off-board management of vehicle operating parameters," in *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, vol. 1, oct. 2003, pp. 352 – 357 vol.1.

[44] A. Burke, "Batteries and ultracapacitors for electric, hybrid, and fuel cell vehicles," *Proceedings of the IEEE*, vol. 95, no. 4, pp. 806 –820, april 2007.

[45] L. Shi and M. Crow, "Comparison of ultracapacitor electric circuit models," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, july 2008, pp. 1 –6.

[46] J. B. Heywood, *Internal Combustion Engine Fundamentals.* McGraw-Hill Science/Engineering/Math, 1988.

[47] D. M. Lamberson, "Torque management of gasoline engines," Master's thesis, University of California at Berkeley, 2003.

[48] J. Slotine and W. Li, *Applied nonlinear control.* Prentice Hall, 1991.

[49] *Bosch Automotive Handbook.* Robert Bosch GmbH, 8 edition, 2011.

[50] X. Zhang and C. Mi, *Vehicle Power Management, Modeling, Control, and Optimization.* Springer, 1988.

[51] Lithium Thechnology Coprporation. (2011) GAIA: LFP/NCA. [Online]. Available: http://www.gaia-akku.com/en/technology/lfpnca.html

[52] I+ME ACTIA GmbH, *User Manual Batery Control System rev. A*, 2008.

[53] Chroma ATE INC, *Programmable DC Power Supply 62000P Series, Operating and Programming Manual*, 2009.

[54] SORENSEN Power Supplies, Elgar Electronics Corporation, *SLH-Series DC High Power Electronic Load Operation and Programming Manual*, 2005.

[55] L. Brand, “The pi theorem of dimensional analysis,” *Archive for Rational Mechanics and Analysis*, vol. 1, pp. 35–45, 1957, 10.1007/BF00297994. [Online]. Available: http://dx.doi.org/10.1007/BF00297994

APPENDICES

Appendix A

Parameters

Below is the list of parameters used in this work.

Table A.1: Parameters used in the control-oriented model

Parameter	Description	Value
m_v	Vehicle mass	1600 kg
α	Engine constant	4.16e-5 g/s/W
β	Engine constant	0.007g
f_{rr}	Tire rolling resistance	0.01
ρ	Air density	1.15 kg/m^3
A	Vehicle frontal area	2.31m^2
C_d	Aerodynamic drag coefficient	0.32
η_m	DC machine efficiency	0.96
g	Gravitational acceleration	9.8 m/s^2
$P_{gen,max}$	Gen-set max power	50kW
SoC_{max}	Maximum allowable SOC	0.7
SoC_{min}	Minimum allowable SOC	0.5
SoC_{ref}	Reference (and initial) SOC	0.6
initial model (designed for the high-fidelity model)		
P_{bmax}	Maximum discharging current	27kW
P_{bmin}	Maximum charging current	-40kW
R	Battery resistance	399mΩ
V_{oc}	Battery voltage	212.6V
Q	Battery capacity	23.18×10^3 As
updated model (designed for the GAIA cells)		
P_{bmax}	Maximum discharging current	27kW
P_{bmin}	Maximum charging current	-20kW
R	Battery resistance	646mΩ
V_{oc}	Battery voltage	285.3V
Q	Battery capacity	25.78×10^3 As

Table A.2: Parameters in the series HEV high-fidelity model

Parameter	Description	Value
	Chassis	
m$_v$	Chassis mass	1380 kg
m$_u$	Unsprung mass	10 kg
m$_t$	Tire mass	28 kg
w_f	Front track	1.524 m
w_r	Rear track	1.519 m
l	Wheelbase	2.7 m
r_t	Tire radius	0.32 m
K_{stiff}	Tire stiffness	304000 N/m
K_{damp}	Tire damping	500 Ns/m
C_{long}	Tire longitudinal force coefficient	115000 N
C_{lat}	Tire Lateral force coefficient	117000 N
C_{rr}	Tire rolling resistance	0.003
	NiMH Battery	
N	Number of cells	168
V_n	Nominal voltage	201.6 V
Q_n	Nominal capacity	6.8 Ah
	DC permanent magnet machines	
ω_n	Nominal motor speed	1420 rpm
V_n	Nominal armature voltage	100 V
I_n	Nominal armature current	100 A
R_a	Armature resistance	0.05 Ω
L_a	Armature inductance	1.5 mH
J_r	Motor inertia	0.15 kgm^2

Table A.2 (continued): Parameters in the series HEV high-fidelity model

Parameter	Description	Value
	Engine	
N_{cyl}	Number of cylinders	4
J_{eng}	Engine Inertia	0.43 kgm^2
S	Stoke	0.1 m
B	Bore	0.085 m
λ_a	Air/fuel ratio	1.1

Table A.3: Parameters used in the HIL/CIL/DIL simulations

Parameter	Value
HIL simulation parameters	
Generator PI controller coefficient (K)	-2
Generator PI controller coefficient (I)	1e-3 min
Engine torque controller coefficient	6e4
Motor PI controller coefficient (K)	200
Motor PI controller coefficient (I)	5e-3 min
HIL simulation time step	2 ms
CIL simulation parameters	
Power supply filter coefficient	0.05
Power supply PI controller coefficient (K)	0.3
Power supply PI controller coefficient (I)	1e-4 min
Power supply voltage measurement gain	4 V/V
Power supply current measurement gain	12 A/V
Load power command gain	182.2 W/V
Load power command offset	0.0022 V
Load current measurement gain	24 A/V
DIL simulation parameters	
Gas pedal gain	0.5 A/step
Brake pedal gain	-0.5 A/step
Steering wheel gain	3000 rad/angle

Appendix B

Software and Hardware List

In this section, the list of all pieces of hardware and software used in this work is given.

Table B.1: List of hardware in the HIL simulation setup

Hardware name	manufacturer	part number	function
PXI chassis	National Instrument	NI-PXI 1031	Real-time computer platform
PXI Embedded controller	National Instrument	NI-PXI 8110	Real-time processing unit
PXI CAN card	National Instruments	PXI 8461	Two-channel CAN card
Data acquisition card	National Instruments	PXI 6289	Multifunction digital/analog data acquisition card
I/O connector	National Instruments	SCB-68	68-pin shielded connector block for DAQ devices
ECU	Woodward	ECM5554-112	Powertrian controller ECU
SmartCraft CAN hub	Mercury Marine		6-port CAN hub
USB to CAN cable	Woodward	ASMINTR00600	Two channel CAN to USB adapter
Boot key	Woodward	1635-1800	ECM calibration tool
4-port USB hub	B&B Electronics	UISOHUB4	4-port USB topical isolation
ECU power supply	Pyramid	PS-4KX	13.8V power supply for ECU

Table B.2: List of hardware in the CIL and DIL simulation setups

Hardware name	manufacturer	part number	function
Power Supply	Chroma Systems Solution	62024p-40-120	Charging the battery
Electric load	Ametek/Sorensen	SLH 60-240-1800	Discharging the battery
Battery cell	GAIA/Lithuim Technology	UHP 341440 NCA	Battery cells for CIL
BMS master	i+ME ACTIA	Master board rev. A	Battery Management System, master board
BMS slave	i+ME ACTIA	Slave board rev. A	Battery Management System, slave board
CAN isolator	B&B Electronics	CANOP	CAN optical isolator
High-current contactor	TE Connectivity	LEV200A4NAF	500A relay in power lines
High-current diode	Microsemi Power Products Group	CPT50060	250A diode in power lines
12V power supply	TDK-Lambda Americas Inc	LS100-12	Power supply for safety contactors
BMS Relay	TE Connectivity	1-1721081-2	BMS safety relay
G25 gamin device	Logitech	G25	Pedal/steering wheel/shifter for driver interface

Table B.3: List of software packages used

Software package	Version	Purpose
MathWorks Matlab	R2011a	Controller design, MIL simulation
Maple Soft MapleSim	5	High-fidelity modeling
WoodWard MotoTune	8.13.7.87	ECU programming and calibration
WoodWard MotoHawk	2011a_SP0.184	ECU code generation from Simulink
Kvasr CANKing	4.0.8.142	CAN bus monitoring
NI LabVIEW	2011	Hardware control, HIL/CIL simulation
Microsoft Visual C++	6.0	Compiling custom code to DLL code
Microsoft Visual C++	9.0	Compiling DLL code from MapleSim

Appendix C

CAN Message Description

The following pages, borrowed from BMS user manual [52, p. 39-41], present the details of the CAN messages used for BMS control.

3.6 USING THE CAN INTERFACE

For details on hard wiring the master board to an existing CAN network see chapter 2.2.2.

3.6.1 Little Endian

The little-endian system is used in all CAN Frames.

In a little-endian (LE) CAN frame system, the low-significant byte (LSB) of the data is placed in the lowest CAN - byte.

Example: Can Frame Id: 0x101 Data byte 0:0x44 byte 1:0x33 byte 2:0x22 byte 3:0x11

BCS TX ID: 0x101; TxDATA										
No.	Name	Length	Data							
			Byte 0 Sub-Id	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
1	I-Bat	8	1	Current				SOC		
				0x44	0x33	0x22	0x11			

Current = 0x11223344

3.6.2 Baudrate

The CAN baudrate is set to 250 kbps (default value)

3.6.3 CAN Frames

3.6.3.1 CAN WAKEUP Frame

BCS RX ID: 0x010; RxWakeup										
No.		Length	Byte 0 Sub-Id	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
1	Wake Up	1	Cmd							

Cmd: 0x01: Wake Up BCS
0x02: Shut down the BCS (go to sleep mode)
0x03: Shut down the BCS (go to sleep mode) with Auto-Balance-System disabled !
This is valid until the next RTC-Wakeup occurs (10 minutes)
0x04: CAN WatchDog Reset Command, if feature is enabled
0x50: CAN Reboot command, if feature is enabled
0x51: CAN Force KV-ON command, if feature is enabled
0x52: CAN Force KV-OFF command, if feature is enabled

Note: for the CAN-WatchdDog command : there is no acknowledge answer

BCS TX ID: 0x011; TxWakeup										
No.	Name	Length	Data							
			Byte 0 Sub-Id	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
1	Wake up	1	Cmd							

Cmd: 0x01: Wake Up BCS
0x02: Shut down the BCS (go to sleep mode)
0x03: Shut down the BCS (go to sleep mode) with Auto-Balance-System disabled !
0x50: CAN Reboot command received

0x51: CAN Force KV-ON command received

0x52: CAN Force KV-OFF command received

3.6.3.2 CAN DATA Frame

Send to BCS:

BCS RX ID: 0x100; RxDATA										
No.		Length	Byte 0 Sub-Id	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
1	I-Bat	>=1	0x01							
2	U-Bat	>=1	0x02							
3	SOC	>=1	0x03							
4	Temperature	>=1	0x04							
5	Temperature Master	>=1	0x05							
6	Warning-Error-status	=1 or 3	0x06	0xAA55 (option)						
7	Get Short Info	>=1	0x07							
8	Get power forecast I & P - Charge	>=1	0x08							
9	Get power forecast I & P - Discharge	>=1	0x09							
10	Get Short Info 2	>=1	0x0A							

The Warning-error-status: should have '0xAA55' (word‼ – byte order: 0x55, 0xAA) to reset the Error Flag otherwise only a read Warning-error-status will be performed..

Answers from BCS:

BCS TX ID: 0x101; TxDATA										
No.	Name	Length	Data							
			Byte 0 Sub-Id	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
1	I-Bat	8	0x01	Current				SOC		
2	U-Bat	8	0x02	Battery voltage				SOC		
3	SOC	8	0x03	SOC						
4	Temperature	8	0x04	Average		Min		Max		
5	Temperature Master	8	0x05	Temp Master						
6	Warning-Error-status	8	0x06	Error-cause		Actual-error		Warning		
7	Get Short Info	8	0x07	Ubat 16bit Word 0 ... 655,00 V step = 10 mV		IBat 16bit Int 0 ...[-] 320,00A step = 10 mA		T-Avg 8-Bit Int 0..[-]128° step = 1°C	SOC 8Bit Int ..100% step=1 %	Delta Ucell 8Bit Max-Min mV ..255
8	Get power forecast I & P - Charge	8	0x08	Predictor I Charge WORD 0 … 655,35 A Step 10 mA		Predictor P Charge DWORD 0 … 2^32 -1 [W] Step 1 W				
9	Get power forecast I & P - Discharge	8	0x09	Predictor I Discharge WORD 0 … 655,35 A Step 10 mA		Predictor P Discharge DWORD 0 … 2^32 -1 [W] Step 1 W				

10	Get Short Info 2	8	0x0A	Ubat 16bit Word 0 ... 6550,0 V step = 100 mV	IBat 16bit Int 0 ...[-] 3200,0A step = 100 mA	T-Avg 8-Bit Int 0..[-]128° step = 1°C	SOC 8Bit Int ..100% step=1 %	Delta Ucell 8Bit Max-Min mV ..255

Current: signed long: mA

Battery voltage: unsigned long: mV

SoC: 0.1 %

All Temperature: signed short: 0.1 °Celsius

Warn-Err: unsigned short:

Appendix D

GAIA Battery Cell Datasheet

Advanced Lithium Battery Systems

Ultra High Power

UHP 341440 NCA
7.5 Ah/ 27 Wh
Lithium Ion Cell

Physical and mechanical characteristics	
Diameter	34 mm
Height	174 mm (144 mm without terminals)
Terminals	Positive terminal Al M8 L: 10 mm Negative terminal Cu M8 L: 10 mm
Weight	approx. 320 g
Volume without terminals	0.13 l
Case material	Stainless Steel

Chemical characteristics	
Positve electrode	Lithium nickel cobalt oxide
Negative electrode	Graphite

Electrical characteristics*	
Nominal voltage	3.6 V
Nominal capacity at 0.2 C	7.5 Ah
Minimum capacity	7.1 Ah
AC Impedance (1 kHz)	≤ 1.2 mOhm
DC Resistance (ESR) (2 s pulse discharge @ 20 C/ 50% SOC)	≤ 6.5 mOhm
Specific energy at 0.2 C	84 Wh/kg
Energy density at 0.2 C	207 Wh/l
Specific power (2 s pulse discharge @ 40 C/ 100% SOC)	2340 W/kg
Power density (2 s pulse discharge @ 40 C/ 100% SOC)	5730 W/l

Operating conditions*	
Recommended charge method	Constant current - constant voltage
End of Charge	I ≤ C/100
Maximum charge voltage	4.2 V
Recommended charge current	up to 7.5 A (1 C)
Continuous charge current	up to 30 A (4 C)
Maximum pulse charge current (15 s) (Max. SOC 80 %, average current < 30 A)	120 A (16 C)
Recommended voltage limit for discharge	3 V
Lower voltage limit for discharge	2.7 V
Lower voltage limit for pulse discharge	2 V
Recommended discharge current	up to 15 A (2 C)
Maximum discharge current	up to 150 A (20 C)
Maximum pulse discharge current (2 s)	up to 300 A (40 C)
Operating temperature	- 30℃ to + 60℃
Recommended charge temperature	0℃ to + 40℃
Storage and transport temperature	- 40℃ to + 60℃
Cycle life at 20℃ and 100% DOD (0.5C charge; 0.5 C discharge)	> 1000 cycles to 80% nominal capacity > 2000 cycles to 60% nominal capacity

* Reference temperature 20℃

Doc UHP 341440 NCA - 2009-06

Data in this document are subject to change without notice and are not binding.

Advanced Lithium Battery Systems

GAIA Akkumulatorenwerke GmbH
Montaniastr. 17
99734 Nordhausen, Germany
www.gaia-akku.com

Technology Corporation

Lithium Technology Corporation
5115 Campus Drive
Plymouth Meeting, PA 19462
www.lithiumtech.com

Appendix E

Power Supply Datasheet

PROGRAMMABLE DC POWER SUPPLY MODEL 62000P SERIES

Chroma's new 62000P Series of programmable DC power supplies offer many unique advantages for ATE integration and testing. These advantage include a constant power operating envelope, precision readback of output current and voltage, output trigger signals as well as the ability to create complex DC transients waveforms to test device behavior to spikes, drops, and other voltage deviations.Designed for automated testing DC-DC converters and similar products, the 62000P sets a new standard for high accuracy programmable DC supplies.

The 62000P Series includes 8 different models ranging from 600W to 5000W, up to 100A and up to 600V. Due to their constant power operating envelope a single instrument can provide both high voltage/low current AND low voltage/high current thereby reducing the number of supplies needed in typical ATE applications.

The 62000P Series also includes 16 bit readback capability for accurate voltage and current readings. This means systems no longer need complex shunt/multiplexers to make accurate readings of the UUT's input parameters. The instruments also include I/O ports providing 8 bit TTLs, DC-ON, fault output signal and remote inhibit as well as a output trigger signal for system timing measurements.

Another unique capability of the 62000P Series supplies is their ability to create complex DC transient waveforms. This capability allows devices to be tested to DC voltage dropouts, spikes and other voltage variations making them an ideal choice for airborne device testing, inverter testing and other devices which will experience voltage interrupts. Applications include DC/DC Converter & Inverter voltage drop test, engine start-up simulation, battery automated charging, electronic product life cycle test, and etc.

RS-232C GPIB PFC

MODEL 62000P SERIES

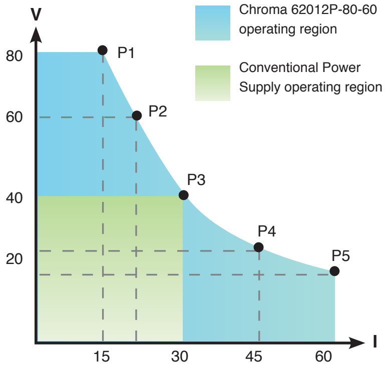
Key Features:

- Eight models : 62006P-100-25
  62006P-300-8
  62012P-80-60
  62012P-100-50
  62012P-600-8
  62024P-80-60
  62024P-100-50
  62050P-100-100
- Wide range of voltage & current combinations with constant power
- Voltage range : 0 ~ 600V
  Current range : 0 ~ 100A
  Power range : 600W, 1200W, 2400W, 5000W
- Digital encoder knobs, keypad and function keys
- Power Factor Correction (0.95)
- High-speed Programming
- Precision V&I Measurements
- Current sharing for parallel operation with Master/Slave Control
- Auto Sequencing Programming: 10 Programs / 100 Sequences / 8 bit TTL
- Voltage & Current Slew Rate Control
- OVP, Current Limit, Thermal protection
- Remote sense, 5V line loss compensation
- APG (Analog Programmable Interface) with Isolated Analog Interface Card
- Optional GPIB control with SCPI
- Standard RS-232 interface
- LabView and Labwindows
- CE Certified
- Standard USB interface (available for Model 62024P-80-60, 62024P-100-50,62050P-100-100)

Chroma

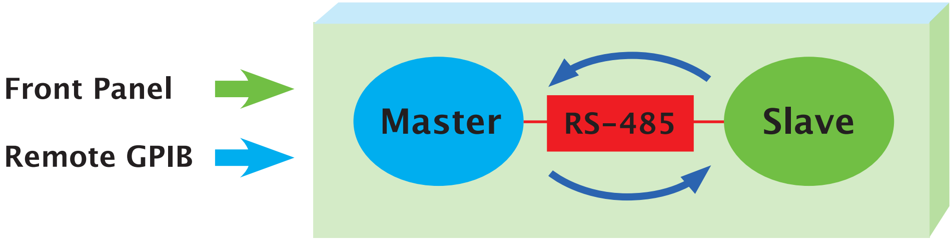
WIDE OPERATING REGION WITH CONSTANT POWER

The 62000P Series supplies offer a wide operating region. For example, the output specification for model 62012P-80-60 is 1200W/80V/60A, it allows operating flexibly in various combinations as shown in the figure at the right. As shown conventional power supplies provide the same rated current at all output voltages, however, the 62000P provides greater current at lower output voltages. This means both low voltage/high current and high voltage/low current UUTs can be tested using a single supply avoiding the for multiple supplies saving cost and space within typical ATE systems.



MASTER/SLAVE PARALLEL & SERIAL CONTROL

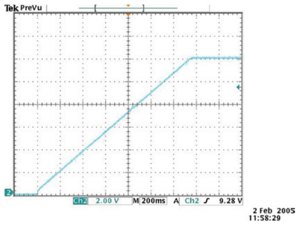
When high power is required, it is common to connect two or more power supplies in parallel or series. The 62000P Series supplies have a smart Master / Slave control mode making series/parallel operation fast and simple. In this mode the master scales values and downloads data to slave units so programming is simple and current sharing automatic.



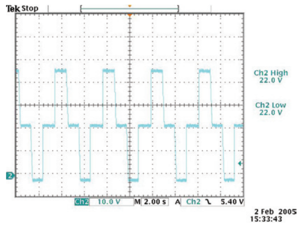
PROGRAMMING SEQUENCES APPLICATIONS

The 62000P Series supplies allow for 100 user programmable sequences with time settings ranging from 10ms to 10000s, voltage /current slew rate control and 8 bit TTL output for automated test applications. Applications include DC/DC Converter & Inverter voltage dropout testing, engine start-up simulation, battery automated charging, product life cycle testing and airborne avionics testing.

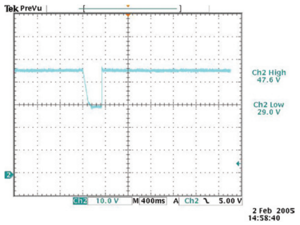
Soft Start Testing



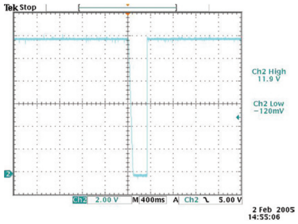
Voltage Step Waveform



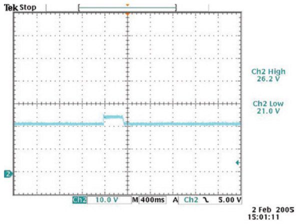
D/D Converter Sag Testing



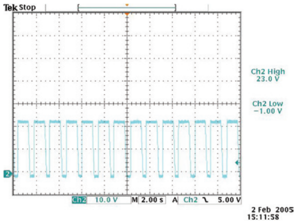
D/D Converter Cycle drop Testing



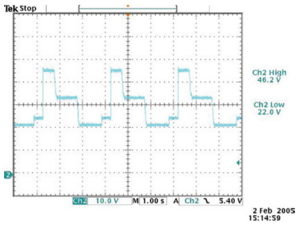
D/D Converter Surge Testing



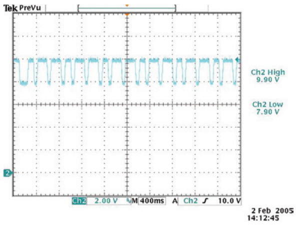
Pulse Charge of Battery



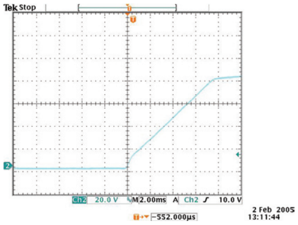
Life Cycle Testing



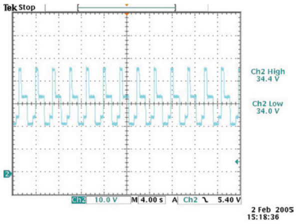
Line Regulation Testing



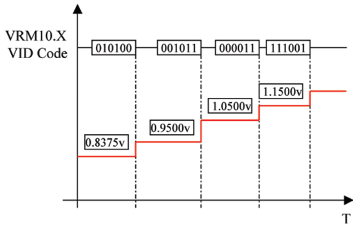
Turn on Time of Setting 80V



Voltage Sequence Program



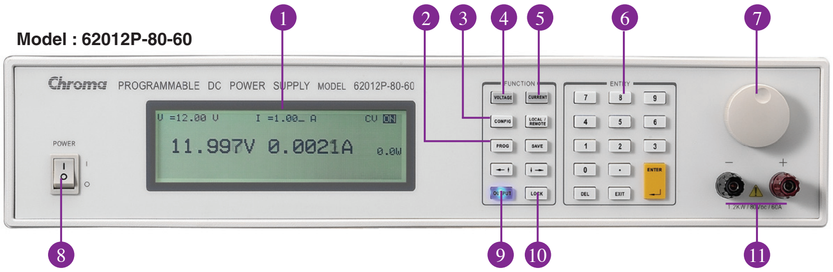
VID code Simulation for VRM/VRD



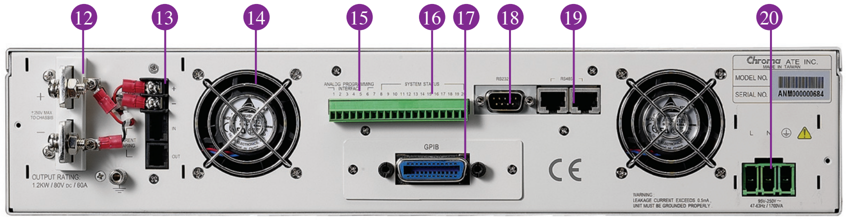
The 62000P Supplies provide 8 output TTL bits with timing control. These control lines can be used for VID control of VRMS or to control other discrete signals.

PANEL DESCRIPTION

Model : 62012P-80-60



1. LCD Display	Display setting, readings and operating status
2. PROG Key	Program the sequence
3. CONFIG Key	Set the system configuration
4. VOLTAGE Key	Set the output voltage
5. CURRENT Key	Set the output current limit
6. NUMERIC Key	Set the data
7. ROTARY Key	Adjust the V&I and set the parameter
8. POWER Switch	
9. OUTPUT Key	Enable or disable the output
10. LOCK Key	Lock all settings
11. OUTPUT Terminal	Connect the output cable to a UUT



Model : 62012P-80-60

12. OUTPUT Terminal	Connect the output cable to a UUT
13. Sense Terminal	Connect the UUT for voltage compensation
14. System Fan	
15. Analog programming interface	For analog level to program and monitor output voltage & current
16. System I/O port	Send 8 bit TTL, DC-ON, fault output signal and remote inhibit and trigger input signal
17. GPIB Connector(Optional)	
18. RS-232 Connector	
19. RS-485 Connector	For master/slave control
20. AC Input Terminal	

Model	**62006P-100-25**	**62006P-300-8**	**62012P-80-60**	**62012P-100-50**	**62012P-600-8**	**62024P-80-60**	**62024P-100-50**	**62050P-100-100**
Output Ratings								
Output Voltage	0~100V	0~300V	0~80V	0~100V	0~600V	0~80V	0~100V	0~100V
Output Current	0~25A	0~8A	0~60A	0~50A	0~8A	0~60A	0~50A	0~100A
Output Power	600W	600W	1200W	1200W	1200W	2400W	2400W	5000W
Line Regulation								
Voltage	0.01%+6mV	0.01%+18mV	0.01%+8mV	0.01%+10mV	0.01%+18mV	0.01%+8mV	0.01%+10mV	0.01%+10mV
Current	0.01%+5mA	0.03%+20mA	0.01%+10mA	0.01%+12mA	0.03%+20mA	0.01%+10mA	0.01%+12mA	0.01%+12mA
Load Regulation								
Voltage	0.01%+10mV	0.01%+50mV	0.01%+12mV	0.01%+18mV	0.01%+50mV	0.01%+12mV	0.01%+18mV	0.01%+18mV
Current	0.01%+5mA	0.03%+40mA	0.01%+20mA	0.01%+28mA	0.03%+40mA	0.01%+20mA	0.01%+28mA	0.01%+28mA
Voltage Measurement								
Range	20V/100V	60V/300V	16V/80V	20V/100V	120V/600V	16V/80V	20V/100V	20V/100V
Accuracy	0.05% + 0.05%F.S.							
Current Measurement								
Range	5A/25A	1.6A/8A	12A/60A	10A/50A	1.6A/8A	12A/60A	10A/50A	20A/100A
Accuracy	0.1% + 0.2%F.S.	0.1% + 0.1%F.S.						
Output Noise (0 ~ 20MHz)								
Voltage Ripple (P-P)	85 mV	180 mV	100 mV	100 mV	180 mV	100 mV	100 mV	125 mV
Voltage Ripple (rms)	10 mV	90 mV	10 mV	15 mV	90 mV	10 mV	15 mV	20 mV
Current Ripple (rms)	10 mA	60 mA	30 mA	20 mA	60 mA	30 mA	20 mA	30 mA
OVP Adjustment Range	110% of Vset to 110% of Vmax							
Efficiency	0.75	0.75	0.8	0.8	0.8	0.85	0.85	0.85
Drift (8 hours)								
Voltage	0.02% of Vmax							
Current	0.04% of Imax							
Temperature Coefficient								
Voltage	0.02% of Vmax/ °C							
Current	0.04% of Imax/ °C							
Transient Response Time	3 mS	3mS	3 mS	3 mS	3mS	3mS	3mS	3mS
10 % step change	180 mV	600 mV	250 mV	250 mV	600 mV	250 mV	250 mV	250 mV
AC Input Voltage	95 to 250Vac					190 to 250Vac (Single phase)	190 to 250Vac (Single phase)	190 to 250Vac (3phase 4 wire, Delta connection) or 342 to 440Vac (3phase 5 wire, Y connection)
Weight	13kg	13kg	13kg	13kg	13kg	13kg	13kg	25kg
Operating Temperature	0~40°C	0~40°C	0~40°C	0~40°C	0~40°C	0~40°C	0~40°C	0~40°C
Dimensions (HxWxD) mm	88 x 428 x 425	88 x 428 x 425	88 x 428 x 425	88 x 428 x 425	88 x 428 x 425	88 x 428 x 425	88 x 428 x 425	177 x 428 x 425

All specifications are subject to change without notice.

ORDERING INFORMATION

62006P-100-25 : Programmable DC Power Supply, 100V / 25A / 600W
62006P-300-8 : Programmable DC Power Supply, 300V / 8A / 600W
62012P-80-60 : Programmable DC Power Supply, 80V / 60A / 1200W
62012P-100-50 : Programmable DC Power Supply, 100V / 50A / 1200W
62012P-600-8 : Programmable DC Power Supply, 600V / 8A / 1200W
62024P-80-60 : Programmable DC Power Supply, 80V / 60A / 2400W
62024P-100-50 : Programmable DC Power Supply, 100V / 50A / 2400W
62050P-100-100 : Programmable DC Power Supply, 100V / 100A / 5000W
A620004 : GPIB Interface for Model 62000P Series
A620006 : Rack Mounting Kit for Model 62000P Series
A620009 : Softpanel for 62000P Series

OTHER SPECIFICATIONS

Programming &Measurement Resolution	
Voltage (Front Panel)	10 mV
Current (Front Panel)	10 mA
Voltage (Remote Interface)	0.003% of Vmax
Current (Remote Interface)	0.002% of Imax
Voltage (Analog Programming Interface)	0.04% of Vmax
Current (Analog Programming Interface)	0.04% of Imax
Programming Accuracy	
Voltage Programming (Front Panel and Remote Interface)	0.1% of Vmax
Voltage Programming (Analog Programming Interface)	0.2% of Vmax
Current Programming (Front Panel and Remote Interface)	0.3% of Imax
Current Programming (Analog Programming Interface)	0.3% of Imax
Programming Response Time	
Rise Time: For a programmed 5% to 95% step of rated voltage. (Full Load)	10 ms
Rise Time: For a programmed 5% to 95% step of rated voltage. (No Load)	10 ms
Fall Time: For a programmed 95% to 5% step of rated voltage. (Full Load)	60 ms
Fall Time: For a programmed 95% to 5% step of rated voltage. (No Load)	840 ms (max.) / 4S for 600V models
Vout setting (GPIB send command to DC Power Supply receiver)	20 ms
?Volt, ? Current (under GPIB command using Fetch)	25 ms
?Volt, ? Current (under GPIB command using Measure)	70 ms
Analog Programming Interface	
Voltage and Current Programming inputs	0~10Vdc or 0~5Vdc of F.S.
Voltage and Current monitor	0~10Vdc or 0~5Vdc of F.S.
Isolation: Maximum working voltage of any analog programming signal with respect to chassis potential	70 Vdc
Auxiliary Power Supply	
Output Voltage	12 Vdc
Maximum current source capability	10 mA
Remote Inhibit Function	
Use to disable the output of DC Power Supply; Active Low	TTL
DC-ON Output Signal	
Indicate the output status, Active High	TTL
Fault Output Signal	
Indicate if there is a fault/protection occurred, Active Low	TTL
Series & Parallel operation function with Master / Slave control	
Voltage limit @ Series Mode. (Model 62012P-600-8)	800 Volt
Voltage limit @ Series Mode (Refer to Ground)	240 Volt
Number of DC Power Supplies allowed @ master / slave control mode	5
Auto Sequencing Programmable Function	
Number of program	10
Number of sequence	100
Time Range	5 ms ~ 15000 S
TTL signal out	8 bits
TTL source capability	7 mA
Slew Rate Control Function	
Voltage slew rate range (The fall rate will be affected by the discharge rate of the output capacitors especially under no load condition.)	0.01V ~ 10V/ms
Current slew rate range of current	0.01A ~ 1A/ms
Minimum transition time	0.5 ms
Remote Sense	
Line loss compensation	5V

All specifications are subject to change without notice.

Developed and Manufactured by :
CHROMA ATE INC.
致茂電子股份有限公司

HEADQUARTERS
66, Hwa-Ya 1st Rd., Hwa-Ya
Technology Park, Kuei-Shan Hsiang,
Taoyuan Hsien 33383, Taiwan
Tel: +886-3-327-9999
Fax: +886-3-327-8898
http://www.chromaate.com
E-mail: chroma@chroma.com.tw

U.S.A.
CHROMA ATE INC. (U.S.A.)
7 Chrysler Irvine, CA 92618
Tel: +1-949-421-0355
Fax: +1-949-421-0353
Toll Free: +1-800-478-2026

EUROPE
CHROMA ATE EUROPE B.V.
Morsestraat 32, 6716 AH Ede,
The Netherlands
Tel: +31-318-648282
Fax: +31-318-648288

CHINA
CHROMA ELECTRONICS
(SHENZHEN) CO., LTD.
8F, No.4, Nanyou Tian An
Industrial Estate, Shenzhen,
China PC: 518054
Tel: +86-755-2664-4598
Fax: +86-755-2641-9620

Distributed by:

Worldwide Distribution and Service Network
62000P-200711-PDF

Appendix F

Electric Load Datasheet

Sorensen SL Series

75 W–14.4 kW

DC and AC/DC Electronic Loads

60–500 V

1–720 A

∿	100	115	230

GPIB RS232

- Flexible Product Line
  - Low power DC modules
  - Low power AC modules
  - High power DC,
- Remote: GPIB, RS-232, Analog
- DC Modes: CC, CR, CV, CP
- AC Modes: CR, CC with crest factor control
- Dynamic mode with slew rate control
- Flexible Data Feedback
- Current monitor output (SLM DC only)

The Sorensen SL series electronic loads offer the best value with the most flexible platform. A wide range of loads are available from 75-1800W with both DC and AC input in benchtop, modular and standalone form factors.

SLM Mainframe
The SLM mainframe choices include a convenient single-bay configuration for benchtop/desktop applications or a four bay configuration for multichannel and ATE requirements. Either chassis is compatible with SLM- and SLD- loads. Each chassis contains non-volatile memory capable of storing up to 150 module setups and nine 16-step sequences for automated, standalone testing. Or for more complex test sequences, the chassis come with GPIB (optional on SLM-1) and RS-232 as standard interfaces.

SLM Family
The SLM family includes nine models of fully programmable, single input AC or DC modular electronic loads. DC models are offered to test power supplies, battery chargers, battery discharge, power supply transient response and integration into ATE systems. AC models are ideal to test low power inverters.

The DC models support operation in Constant Current (CC), Constant Voltage (CV), Constant Resistance (CR) or Constant Power (CP) mode as well as a short simulation. Engineers have ultimate control of current waveforms by using either the analog input or CC dynamic mode. An analog input (single input DC models) allows arbitrary current waveforms up to 20kHz with an external 0-10V signal. In dynamic mode, the pulse generator allows fast state switching between two programmed current levels with programmed slew rate and dwell times.

SLD Family
The SLD family offers six models of fully programmable, dual input modular electronic loads. These DC modules are specifically designed for low power, high channel count testing and provide the highest channel density available.

SLH Family
Fully programmable, high power AC or DC electronic loads. The 500V models are for PFC testing, power transformers and various other AC or DC power sources. The 300V models are used for testing of UPSs, automatic voltage regulators (AVR), and batteries.

- High current, 60V DC models for general purpose power supply testing
- High voltage,AC/DCmodels are intended for inverter test,
- Power Factor Correction (PFC) circuit testing (500V) and UPS testing (300V)

SL Series

Electronic Load Selection

Often the selection of programmable power supplies is based upon volts and amps capability. However when selecting an electronic load, it is important to account for volts, amps and power. The power limit is displayed on a constant power curve. A load must be selected so that the operating points are within the Power Curve (see Figure 1). For many applications in which different power sources are tested, there may be high voltage, low current requirements as well as low voltage, high current requirements. A single load may be able to handle both with good programming resolution. In cases where a single load may not work, the broad range of current, power and voltage available in the SL series allows optimum selection depending upon the voltage, current, power required.

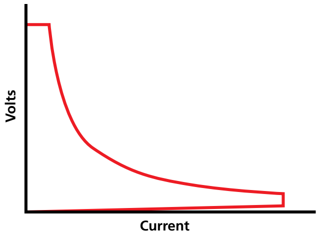
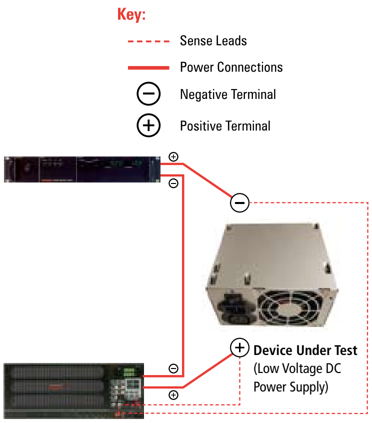


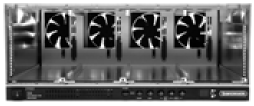
Figure 1 - Power Curve

Applications

Low Voltage Operation

All SL series loads operate well below 1V. However in many applications, such as fuel cell research and microprocessor voltage regulator modules (VRM), the voltage at the load inputs can be 0.1 to 0.2V. This low voltage does not allow the load transistors to fully turn-on (bottom right corner of the power contour). To utilize the full rated current of an electronic load, a boost supply can be placed in series to increase the voltage. While a fixed voltage DC-DC converter can be used as the boost supply, a programmable power supply is preferred to keep the load voltage at the minimum to draw full current as the device under test ramps up in voltage.





SLM-4: Chassis

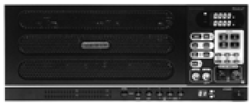
SLM: DC Module

SLM: AC Module

SLD: Dual Input DC Module

SLD: Dual Input DC Module

SLM-1 Chassis



SLH: DC Electronic Load

SLH: AC Electronic Load

SL Series : Specifications

75 W–14.4 kW

SLH - Standalone AC Loads

Model	SLH-500-4-1200	SLH-500-6-1800	SLH-300-12-1200	SLH-300-12-1800	SLH-300-18-1800
Input Ratings					
Power:	1200VA	1800VA	1200VA	1800VA	1800VA
Current:	4Arms	6Arms	12Arms	12Arms	18Arms
Voltage:	300Vrms / 500Vdc	300Vrms / 500Vdc	300Vrms	300Vrms	300Vrms
Frequency:	DC, 40 - 70Hz (CC Mode) ; DC - 70Hz (CR Mode)				
CC Mode					
Range:	0-2 / 2-4A	0-3 / 3-6A	0-6 / 6-12A	0-6 / 6-12A	0-9 / 9-18A
Resolution:	0.5 / 1mA	0.75 / 1.5mA	1.5 / 3mA	1.5 / 3mA	2.25 / 4.5mA
Accuracy:	±0.5% of (setting + range)				
Low Current:	0 - 0.2A	0 - 0.3A	0 - 0.6A	0 - 0.6A	0 - 0.9A
Accuracy:	±(0.5% of reading + 0.2% of range)				
Maximum Peak Current:	8A	12A	24A	24A	18A
CR Mode					
Range 1: (I>0.5% of rating)	50 - 200,000Ω	33.33 - 133,000Ω	20 - 80,000Ω	20 - 80,000Ω	13.3 - 53,333Ω
Range 2: (I>50% of rating)	12.5 - 50Ω	8.33 - 33.33Ω	5 - 20Ω	5 - 20Ω	3.33 - 13.33Ω
4 1/2 DVM					
Range:	0-500V	0-500V	300V	300V	300V
Resolution:	0.1V	0.1V	0.1V	0.1V	0.1V
Accuracy:	±(0.5% of reading + 0.2% of range)				
4 1/2 DAM					
Range:	0-4A	0-6A	0-12A	0-12A	0-18A
Resolution:	1mA	1mA	1mA	1mA	1mA
Accuracy:	±(0.5% of reading + 2% of range) ; ±0.5% of (reading + range) @ 50/60Hz				
4 1/2 Watt Meter					
Range:	0-1200W	0-1800W	0-1200W	0-1800W	0-1800W
Resolution:	0.1W				
Accuracy:	± (0.5% of reading)±3W				
VA / Power Meter:	Vrms × Arms				
Weight	18.5kgs/40.7lbs	21.5kgs/47.3lbs	18.5kgs/40.7lbs	21.5kgs/47.3lbs	21.5kgs/47.3lbs

SLM - AC Modules

Model	SLM-60-20-300	SLM-150-8-300	SLM-300-4-300	SLM-500-1-300
Input Ratings				
Power:	300VA	300VA	300VA	300VA
Current:	20Arms	8Arms	4Arms	1Arms
Voltage:	60Vrms	150Vrms	300Vrms	300Vrms / 500Vdc
Frequency:	DC, 40 - 70Hz (CC Mode) ; DC - 70Hz (CR Mode)			
CC Mode				
Range:	0-10 / 10-20A	0-4 / 4-8A	0-2 / 2-4A	0-0.5 / 0.5-1A
Resolution:	2.5 / 5mA	1 / 2mA	0.5 / 1mA	0.125 / 0.25mA
Accuracy:	±0.5% of (setting + range)			
Low Current:	0 - 1A	0 - 0.4A	0 - 0.2A	0 - 0.05A
Accuracy:	±2% of (setting + range)			
Maxium Peak Current:	40A	16A	8A	2A
CR Mode (1)				
Range 1: (I>0.5% of rating)	1.2-4,800Ω	7.5-30,000Ω	30 - 120,000Ω	200 - 800000Ω
Range 2: (I>50% of rating)	0.3 - 1.2Ω	1.875 - 7.5Ω	7.5 - 30Ω	50 - 200Ω
4 1/2 DVM				
Range:	60V	150V	300V	500V
Resolution:	0.01V	0.01V	0.1V	0.1V
Accuracy:	±(0.5% of reading + 0.2% of range)			
4 1/2 DAM				
Range:	20A	8A	4A	1A
Resolution:	0.01A	0.001A	0.001A	0.001A
Accuracy:	±(0.5% of reading + 2% of range); ±0.5% of (reading + range) @ 50/60Hz			
4 1/2 Watt Meter				
Range:	300W			
Resolution:	0.1W			
Accuracy:	±(0.5% of reading)±3W			
VA / Power Meter:	Vrms × Arms			
Weight	3.5kgs/7.7lbs			

SL Series : Specifications

SLM - DC Modules

Model	SLM-60-30-150		SLM-60-60-300		SLM-250-10-300		SLM-500-10-300		SLM-60-15-75	
Input Ratings										
Voltage:	60V		60V		250V		500V		60V	
Current:	30A		60A		10A		10A		15A	
Power:	150W		300W		300W		300W		75W	
Minimum Voltage: (Full Current)	0.6V @ 30A		0.5V @ 60A		0.8V @ 10A		4.5V @ 10A		0.3V @ 15A	
CC Mode										
Range 1: | Range 2:	0-3A	0-30A	0-6A	0-60A	0-1A	0-10A	0-1A	0-10A	0-1.5A	0-15A
Resolution:	0.8mA	8.0mA	1.6mA	16.0mA	0.268mA	2.68mA	0.268mA	2.68mA	0.4mA	4.0mA
Accuracy:	± 0.2% of (Setting + Range)									
CR Mode										
Range 1: (I > 0.02% of RATING)	2-7.5KΩ		1-3.75KΩ		25-18.75KΩ		50-18.75KΩ		4-15KΩ	
Range 2: (I > 0.2% of RATING)	0.1067-2Ω		0.0534-1Ω		1.333-25Ω		2.67-50Ω		0.213-4Ω	
CV Mode										
Range:	0-60V		0-60V		0-250V		0-500V		0-60V	
Resolution:	0.016V		0.016V		0.067V		0.133V		0.016V	
Accuracy:	± 0.1% of (Setting + Range)									
CP Mode										
Range:	0-150W		0-300W		0-300W		0-300W		0-75W	
Resolution:	0.04W		0.08W		0.08W		0.08W		0.02W	
Accuracy:	± 0.5% of (Setting + Range)									
Short Mode:										
Resistance:	0.02Ω		8mΩ		0.08Ω		0.45Ω		0.02Ω	
Current:	30A		60A		10A		10A		15A	
Dynamic:										
T High & T Low:	50μs to 9.999s									
Rise/Fall of Range 1:	2.0-125mA/μs		4-250mA/μs		0.8-50mA/μs		0.8-50mA/μs		1.0-62.5mA/μs	
Rise/Fall of Range 2:	0.2-1.2A/μs		0.04-2.5A/μs		8.0-500mA/μs		8.0-500mA/μs		10-625mA/μs	
Accuracy:	± 10% of Setting									
4 1/2 DVM:										
Range:	15.0V	60.0V	15.0V	60.0V	30.0V	250.0V	199.99V	500.0V	15.0V	60.0V
Resolution:	0.001V	0.002V	0.001V	0.002V	0.001V	0.01V	0.01V	0.1V	0.001V	0.002V
Accuracy:	± 0.05% of (Reading + Range)									
4 1/2 DAM:										
Range:	3.0A	30.0A	6.0A	60.0A	1.0A	10.0A		10.0A	1.5A	15.0A
Resolution:	0.001A	0.01A	0.001A	0.01A	0.0001A	0.001A		0.001A	0.0001A	0.001A
Accuracy:	± 0.2% of (Reading + Range)									
Current Monitor:	3.0A/V		6.0A/V		N/A		N/A		1.5A/V	
Load ON Volt:										
Range:	0.1-25V				0.2-50V		0.4-100V		0.1-25V	
Resolution:	0.1V				0.2V		0.4V		0.1V	
Accuracy:	1% of Setting + 0.25V				1% + 0.5V		1% of Setting + 1V		1% of Setting + 0.25V	
Load OFF Volt:										
Range:	0-25V				0-50V		0-100V		0-25V	
Resolution:	0.01V									
Accuracy:	1% of Setting + 0.25V				1% + 0.5V		1% of Setting + 1V		1% of Setting + 0.25V	
Weight:	3.5kgs/7.7lbs									

SLD - Dual Input DC Modules												
Model:	SLD-60-505-255		SLD-61-505-255		SLD-80-20-102		SLD-61-5-752		SLD-62-5-752		SLD-60-105-550	
Input Rating:												
Channel	A	B	A	B	A	B	A	B	A	B	A	B
Voltage (Volt)	+60V	+60V	+60V	-60V	+80V	+80V	+60V	-60V	-60V	-60V	+60V	+60V
Current (Ampere)	50A	5A	50A	5A	20A	20A	5A	5A	5A	5A	100A	5A
Power (VA)	250W	50W	250W	50W	100W	100W	75W	75W	75W	75W	500W	50W
Minimum Voltage (Full Current)	0.4V @ 50A	0.4V @ 5A	0.4V @ 50A	0.9V @ 5A	0.4V @ 20A	0.4V @ 20A	0.4V @ 5A	0.4V @ 5A	0.4V @ 5A	0.4V @ 5A	0.4V @ 100A	0.4V @ 5A
CC Mode:												
Range	0 - 5A / 50A	0 - 0.5A / 5A	0 - 5A / 50A	0 - 0.5A / 5A	0 - 2.0A / 20A	0 - 2.0A / 20A	0 - 0.5A / 5A	0 - 0.5A / 5A	0 - 0.5A / 5A	0 - 0.5A / 5A	0 - 10A / 100A	0 - 0.5A / 5A
Resolution	1.34 / 13.4mA	0.134 / 1.34mA	1.34 / 13.4mA	0.134 / 1.34mA	0.533 / 5.33mA	0.533 / 5.33mA	0.134 / 1.34mA	0.134 / 1.34mA	0.134 / 1.34mA	0.134 / 1.34mA	2.66 / 26.6mA	0.134 / 1.34mA
Accuracy	±0.2% of (Setting + Range)											
CR Mode:												
Range 1: (Ω) (I>0.02% of rating)	1.2 - 4500	12 - 45000	1.2 - 4500	12 - 45000	4 - 15000	4 - 15000	12 - 45000	12 - 45000	12 - 45000	12 - 45000	0.6 - 2250	12 - 45000
Range 2: (Ω) (I>0.2% of rating)	0.04-1.2	0.4-12	0.04-1.2	0.4-12	0.133-4	0.133-4	0.4-12	0.4-12	0.4-12	0.4-12	0.02-0.6	0.4-12
CV Mode												
Range	0 – 60V			0 – (-60)V	0 – 60V			0 – (-60)V			0 – 60V	
Resolution	16mV				21.3mV		16mV					
Accuracy	±0.2% of (Setting + Range)											
Short Mode												
Resistance	8mΩ	0.08Ω	8mΩ	0.18Ω	0.02Ω	0.02Ω	0.02Ω	0.06Ω	0.06Ω	0.06Ω	4mΩ	0.08Ω
Current	50A	5A	50A	5A	20A	20A	5A	5A	5A	5A	100A	5 A
Dynamic Mode												
T High / T Low	50μs to 9.999s											
Slew Rate (mA/μs)	4-200 40-2000	0.4-20 4-200	4-200 40-2000	0.4-20 4-200	1.6-80 16-800	1.6-80 16-800	0.4-20 4-200	0.4-20 4-200	0.4-20 4-200	0.4-20 4-200	8-400 80-4000	0.4-20 4-200
Resolution (mA/μs)	0.8 8	0.08 0.8	0.8 8	0.08 0.8	0.32 3.2	0.32 3.2	0.08 0.8	0.08 0.8	0.08 0.8	0.08 0.8	1.6 16	0.08 0.8
Accuracy	±(10% +10μs)											
4 1/2 DVM:												
Range	15V / 60.00V				20V / 80V		15V / 60.00V					
Resolution	0.001 V / 0.01 V											
Accuracy	±0.05% of (Reading + Range)											
4 1/2 DAM:												
Range	15A / 50A	1.5A / 5A	15A / 50A	1.5A / 5A	2.0A / 20A	2.0A / 20A	1.5A / 5A	1.5A / 5A	1.5A / 5A	1.5A / 5A	10 / 100A	1.5A / 5A
Resolution	1mA / 10mA	0.1mA / 1mA	1mA / 10mA	0.1mA / 1mA	0.1mA / 1mA	0.1mA / 1mA	0.1mA / 1mA	0.1mA / 1mA	0.1mA / 1mA	0.1mA / 1mA	1 / 10mA	0.1mA / 1mA
Accuracy	±0.2% of (Reading + Range)											
Load ON Voltage												
Range	0.1-25V											
Resolution	0.1V											
Accuracy	1% of Setting +0.25V											
Load OFF Voltage												
Range	0-25V											
Resolution	1mV											
Accuracy	1% of Setting +0.25V											

SL Series : Specifications

SLH - Standalone DC Loads							
Model	**SLH-60-120-600**	**SLH-60-120-1200**	**SLH-60-120-1800**	**SLH-60-240-1200**	**SLH-60-240-1800**	**SLH-60-360-1800**	**SLH-500-60-1800**
Input Ratings							
Voltage	60V						500 V
Current	120A			240A		360A	60 A
Power	600W	1200W	1800W	1200W	1800W	1800W	1800 W
Minimum Voltage (Full Current)	0.5V @ 120A	0.4V @ 120A	0.3V @ 120A	0.5V @ 240A	0.5V @ 240A	0.4 @ 360A	6V @ 60A
CC Mode							
Range	0-12 / 0-120A			0-24 / 0-240A		0 - 36 / 360A	0 - 6/60 A
Resolution	3.2 / 32mA			6.4 / 64mA		9.6 / 96mA	1.6/16 mA
Accuracy	±0.2% OF (SETTING + RANGE)						
CR Mode							
Range 1 (I>0.05% of rating)	0.5 - 1875Ω			0.25 - 937.50Ω		0.167 - 624.9Ω	8.33 - 18750Ω
Range 2 (I>0.5% of rating)	0.027 - 0.5Ω			0.0133 - 0.25Ω		8.3 - 167mΩ	0.444 - 8.33Ω
CV Mode							
Range	0 - 60V						0 - 500 V
Resolution	0.016V						0.133V
Accuracy	±0.1% OF (SETTING + RANGE)						
CP Mode							
Range	0 - 600W	0 - 1200W	0 - 1800W	0 - 1200W	0 - 1800W	0 - 1800W	0-1800W
Resolution	0.16W	0.32W	0.48W	0.32W	0.48W	0.48W	0.48W
Accuracy	±0.5% OF (SETTING + RANGE)						
Short Mode							
Maximum Resistance	4.2mΩ	3.3mΩ	2.5mΩ	2.1mΩ		1.1mΩ	0.1 Ω
Current	120A			240A		360A	60A
Dynamic Mode							
T High / T Low	50µs to 9.999s						
Slew Rate Low	8mA - 500mA/µs			16mA - 1A/µs		24mA - 1.5A/µs	4.8-300 mA/µs
Slew Rate High	80mA - 5A/µs			0.160A - 10A/µs		0.24A - 15A/µs	0.048-3.0 A/µs
Accuracy	±(10% OF SETTING +10µs)						
4 1/2 DVM							
Range	0 - 20.00 / 60.00V						0 - 60.00/600.0
Resolution	0.001 / 0.01V						0.01/0.1V
Accuracy	±0.05% OF (READING + RANGE)						
4 1/2 DAM							
Range	0 - 12A / 0 - 120A			0 - 24A / 0 - 240A		0 - 36A / 0 - 360A	0 - 6/60 A
Resolution	1mA / 4mA			1mA / 10mA		1.2mA / 12mA	0.001A/0.01A
Accuracy	±0.5% OF (READING + RANGE)						
Current Monitor	12A/V			24A/V		36A/V	N/A
Load ON Volt							
Range	0.1 - 25V						0.4 - 100V
Resolution	0.1V						0.4V
Accuracy	1% of SETTING +0.25V						
Load OFF Volt							
Range	0 - 25V						0 - 100V
Resolution	0.1V						
Accuracy	1% of SETTING +0.25V						
Weight	15.2kgs./33.4lbs	19.4kgs/42.7lbs	23.6kgs/51.9lbs	19.4kgs/42.7lbs	23.6kgs/51.9lbs	23.6kgs/51.9lbs	23.6 kgs. / 51.9 lbs.

SL Series : Specifications

75 W–14.4 kW

Common	
Software	LabVIEW Driver can be downloaded at no cost: www.elgar.com/products/SL/SL_Downloads.htm
Regulatory	Certified to UL/CSA 61010 and IEC/EN 61010-1, CE Compliant (LVD and EMC Directives)
Environmental	Operating Temperature: 0˚ to 40˚C Storage Temperature: -10˚ to 65˚C
Cooling	Front, Side, Top Air Inlets, Rear Exhaust, Units may be rackmounted without spacing.
SLH Memory	150 Settings for DC, 5 Settings for AC
Readback	Voltage, Current, Power: 16-bit resolution, VA: Vrms x Arms
Analog Input	SLM: DB9 connector, SLH: BNC connector. DC, Single Input (SLH or SLM), CC Mode: 0-10V = 0 – FS, Bandwidth: 20kHz, Sums Current with Programmed Value
AC (SLH or SLM)	Sync signal on zero crossing
Remote Programming	SLM-1: RS-232C, GPIB (Optional), SLM-4: RS-232C, GPIB, analog, SLH: RS-232C, GPIB, analog
Dynamic Mode (DC Models) (see Figure 4)	Mode: CC, T-high, T-low: 50 µs to 9.999 sec, Slew Rate: See Specification Tables, I high, I low: 0 to Rated Current
Options and Accessories	-1: GPIB, SLM-1 or SLM-4 only -01: 100/200V AC input, SLM-1 only -11: 100/200V AC input and GPIB M12: Front panel bus bar, SLH DC only M23: Front panel bus bar and 100/200V AC Input, SLH DC only
Input Power	
Line:	115V / 230V ± 10%, switch selectable or 100V / 200V ± 10% switch selectable (optional)
Frequency:	50 / 60Hz
Power Consumption	100W Maximum
Protection: AC input fuses	
OVP, OCP, OPP:	~5% above rated maximum
OTP:	~85°C Heat sink temperature
DC Loads:	Reverse Polarity All protection modes turn off LOAD input
Hardware Input Voltage Limit:	60V Rated DC Input: 100V, 250V Rated DC Input: 400V, 500V Rated DC and all AC Input: 900V
SLM Chassis	
Memory	150 memory settings for DC modules, 5 memory settings for AC modules, Memory settings store entire chassis condition
Sequencer (see Figure 2)	
Control	Front panel
Timing	100ms-9.9 secs per step
Maximum Steps per Sequence	16
Number of Sequences	9
Programming	
All Parameters	12-bit resolution
AC Crest Factor (see Figure 3)	Sinewave: √2, 1.5-3.5, Resolution: 0.1 Squarewave: 1.0-3.4, Resolution: 0.1
DC	√2, 2.0-3.5, Resolution: 0.5
Maximum Peak	Current = 2 x Rated Current

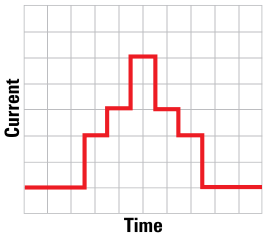


Fig.2 - Sequencer for Modules

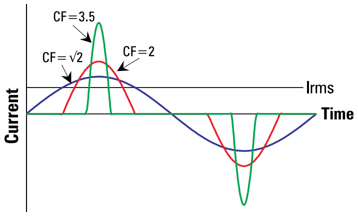


Fig.3 - Crest Factor for AC models

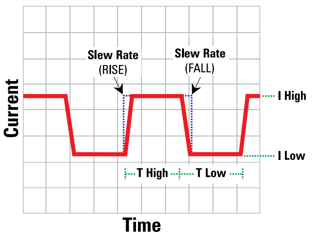
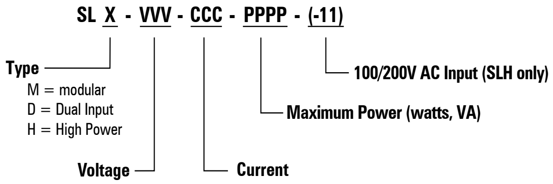


Fig.4 - Dynamic Mode for DC models



SL X - VVV - CCC - PPPP - (-11)

Type
M = modular
D = Dual Input
H = High Power

Voltage

Current

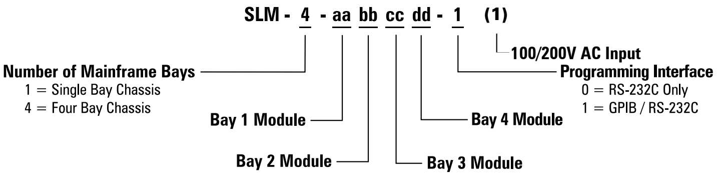
Maximum Power (watts, VA)

100/200V AC Input (SLH only)

SLH Stand Alone DC Loads

Model Number	Description
SLH-60-120-600	60V / 120A / 600W rack mounted, programmable DC load
SLH-60-120-1200	60V / 120A / 1200W rack mounted, programmable DC load
SLH-60-120-1800	60V / 120A / 1800W rack mounted, programmable DC load
SLH-60-240-1200	60V / 240A / 1200W rack mounted, programmable DC load
SLH-60-240-1800	60V / 240A / 1800W rack mounted, programmable DC load
SLH-60-360-1800	60V / 360A / 1800W rack mounted, programmable DC load
SLH-500-60-1800	500V / 60A / 1800W rack mounted, programmable DC load
SLH-500-4-1200	500Vdc/300Vrms / 4A / 1200W rack mounted, programmable DC load
SLH-500-6-1800	500Vdc/300Vrms / 6A / 1800W rack mounted, programmable AC/DC load
SLH-300-12-1200	300Vrms / 12A / 1200W rack mounted, programmable AC/DC load
SLH-300-12-1800	300Vrms / 12A / 1800W rack mounted , programmable AC/DC load
SLH-300-18-1800	300Vrms / 18A / 1800W rack mounted , programmable AC/DC load

All SLH models include rackmount handles with ears.



SLM - 4 - aa bb cc dd - 1 (1)

Number of Mainframe Bays
1 = Single Bay Chassis
4 = Four Bay Chassis

Bay 1 Module

Bay 2 Module

Bay 3 Module

Bay 4 Module

100/200V AC Input

Programming Interface
0 = RS-232C Only
1 = GPIB / RS-232C

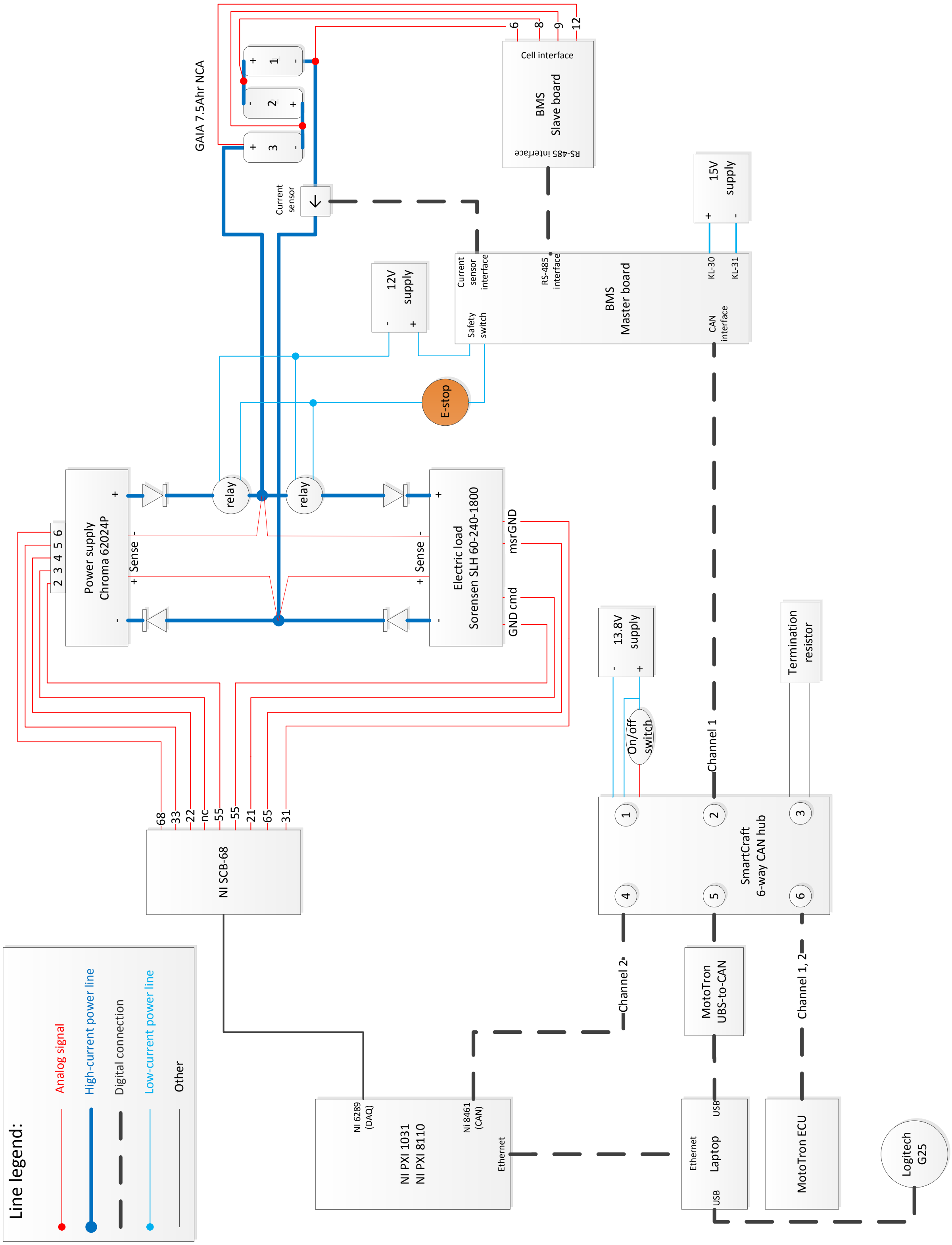
SLM & SLD Modular Loads

Code	Module / Chassis	Description
C	SLM-4	Mainframe Chassis, Four (4) Bay for SLM, SLD modular loads includes GPIB/RS-232C
C	SLM-1	Mainframe Chassis, Single bay for SLM, SLD modular loads
10	SLM-60-30-150	DC Module, 60V / 30A / 150W
11	SLM-60-60-300	DC Module, 60V / 60A / 300W
12	SLM-250-10-300	DC Module, 250V / 10A / 300W
14	SLM-500-10-300	DC Module, 500V / 10A / 300W
15	SLM-60-15-75	DC Module, 60V / 15A / 75W
32	SLD-80-20-102	DC dual input module, 80V / 20A / 100W x 2
30	SLD-60-505-255	DC dual input module, 60V / 50A / 250W, 60V / 5A / 50W
31	SLD-61-505-255	DC dual input module, 60V / 50A / 250W, -60V / 5A / 50W
33	SLD-61-5-752	DC dual input module, 60V / 5A / 75W, -60V / 5A / 75W
34	SLD-62-5-752	DC dual input module, -60V / 5A / 75W x 2
35xx	SLD-60-105-550	DC dual input module, 60V / 100A / 500W, 60V / 5A / 50W
50	SLM-60-20-300	AC/DC Module, 60V / 20A / 300W
51	SLM-150-8-300	AC/DC Module, 150V / 8A / 300W
52	SLM-300-4-300	AC/DC Module, 300V / 4A / 300W
53	SLM-500-1-300	AC/DC Module, 500Vdc/300Vrms / 1A / 300W
BB	SLM-BB	Blank Panel

Appendix G

Hardware Dependdencies

In this section, the details of hardware dependencies in the test setup is presented.


Line legend:
Analog signal
High-current power line
Digital connection
Low-current power line
Other
GAIA 7.5Ahr NCA
1
2
3
+
-
Current sensor
Cell interface
BMS
Slave board
RS-485 interface
6
8
9
12
Current sensor interface
RS-485 interface
BMS
Master board
Safety switch
CAN interface
KL-30
KL-31
15V supply
12V supply
E-stop
relay
relay
Power supply
Chroma 62024P
Sense
2 3 4 5 6
Electric load
Sorensen SLH 60-240-1800
msrGND
GND cmd
NI SCB-68
68
33
22
nc
55
55
21
65
31
13.8V supply
On/off switch
Termination resistor
Channel 1
SmartCraft
6-way CAN hub
4
5
6
Channel 2
MotoTron
UBS-to-CAN
Channel 1, 2
NI PXI 1031
NI PXI 8110
NI 6289
(DAQ)
NI 8461
(CAN)
Ethernet
Ethernet
Laptop
USB
USB
MotoTron ECU
Logitech
G25

Appendix H

LabVIEW Programs

The programs developed for this setup is given in this section. In this section, the details of hardware dependencies in the test setup is presented.

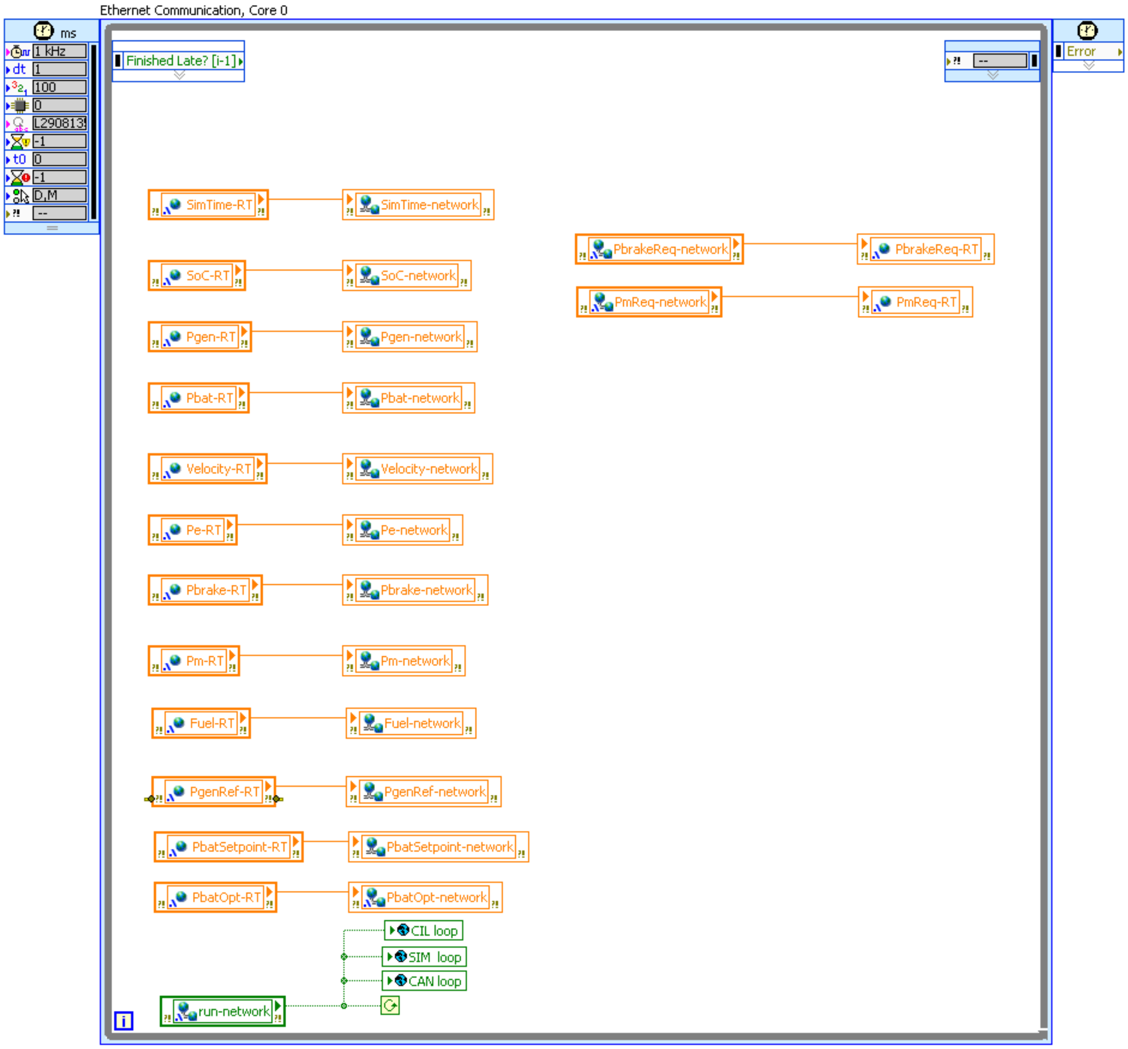


Figure H.1: The Ethernet communication loop, core 1

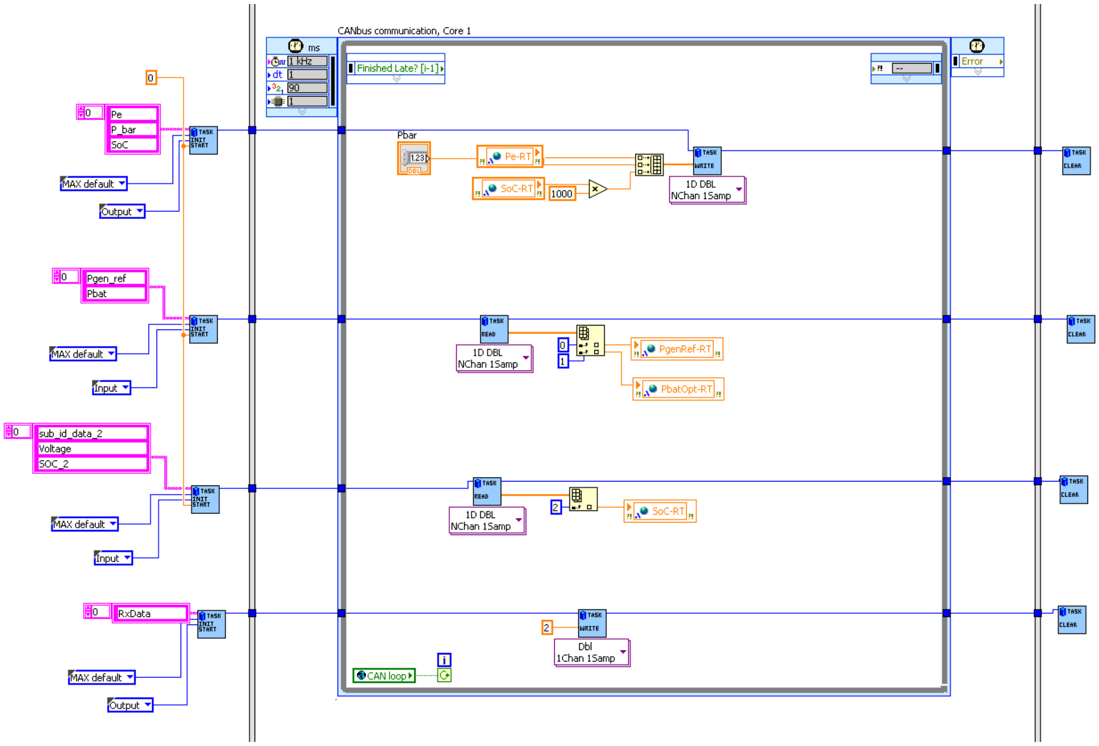


Figure H.2: The CAN communication loop, core 2

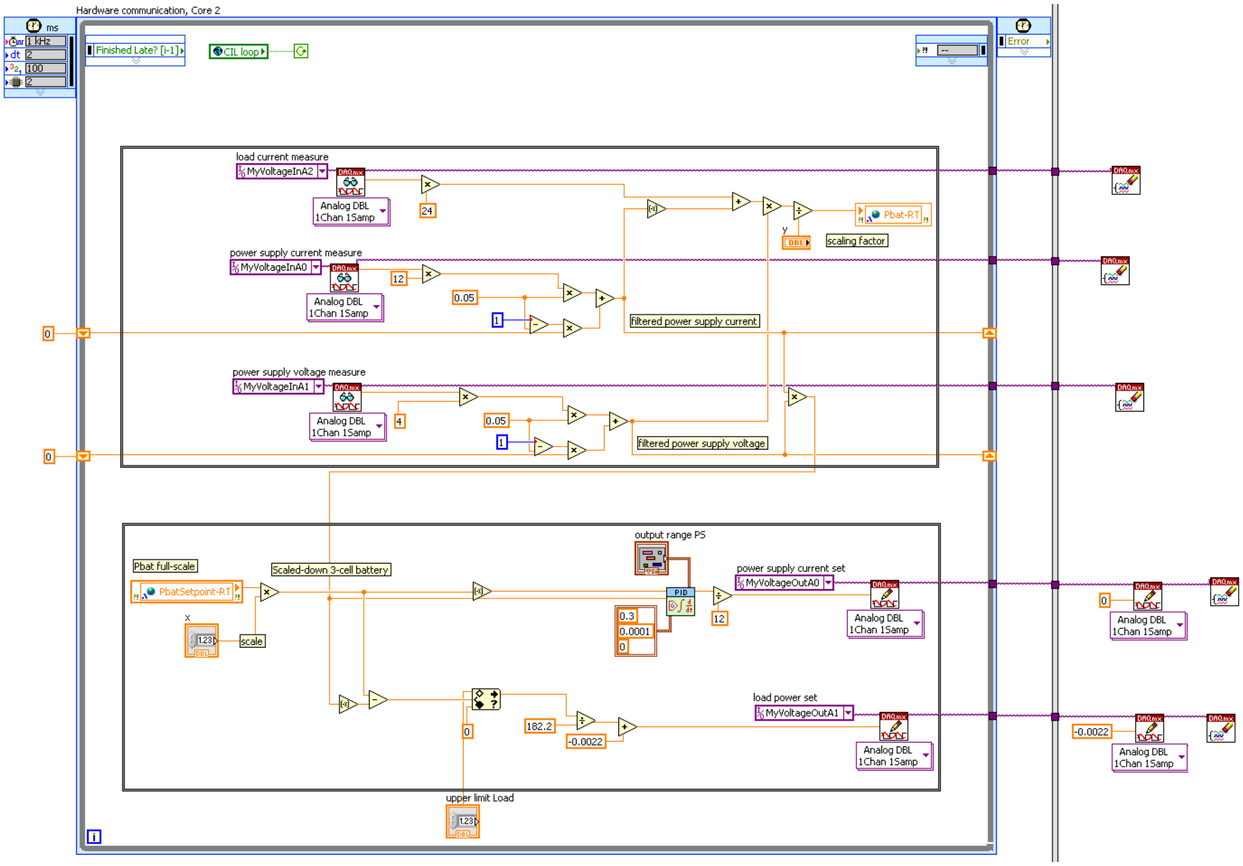


Figure H.3: The battery cycler control loop, core 3

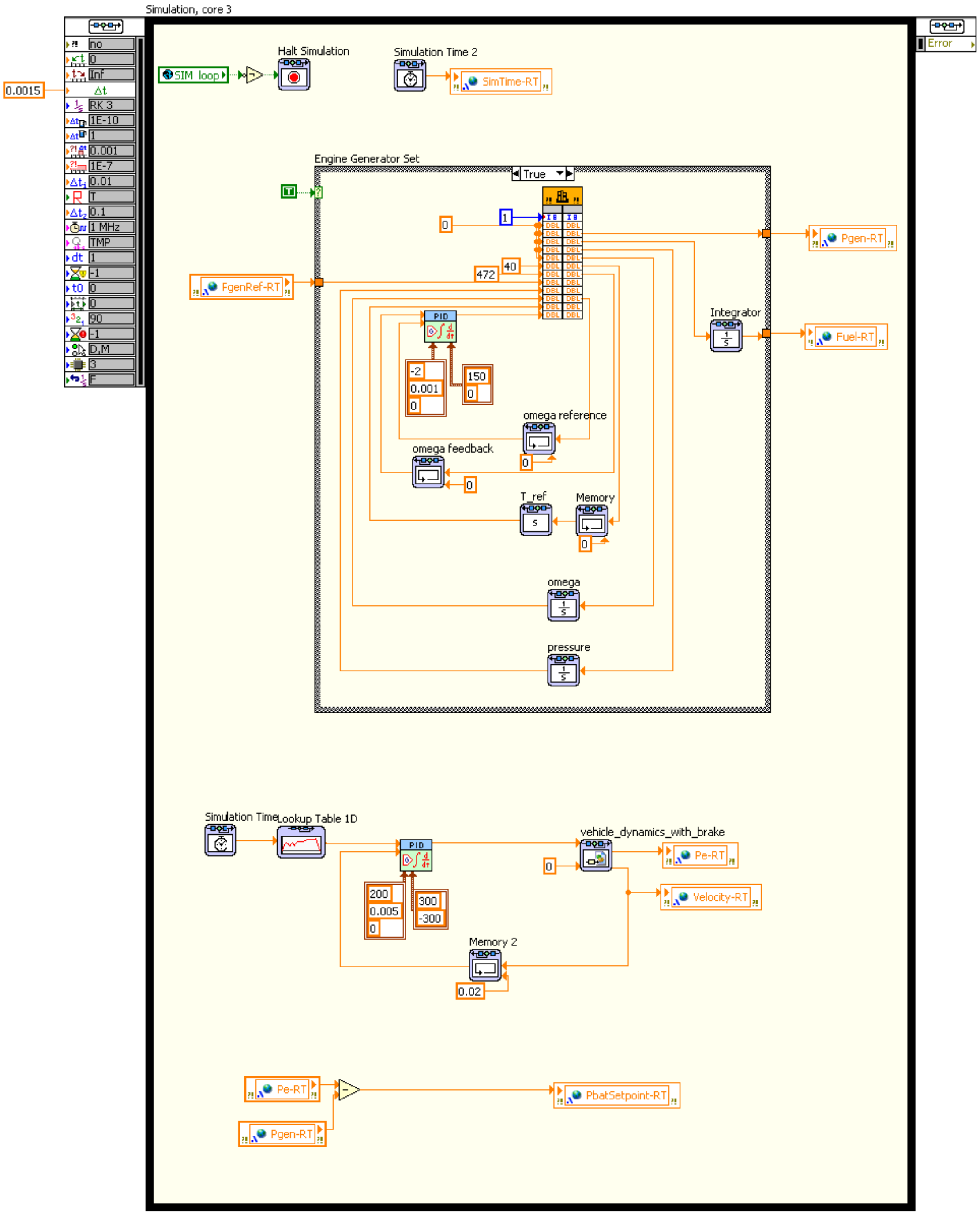


Figure H.4: The high-fidelity model solver, core 4

Appendix I

Battery Managemanet Systems Parameters

```
------- SYSTEM DATAS ------------------------------
FlatModel: 1S (3C) Cells: 3
----------------------------------------------------
PCM-ID  Software Version: 1.3.4 09.07.2008
0x0300= Version DataSp. :     98
0x0301= Number of Slaves:      1
0x0302= Number of Cells :      3
0x0303= Cells per Slaves:      3
0x0304= Sensor  Type    :      1 small sensor
0x0305= Battery Type    :     50 BAT_GAIA_MTA
0x0306= User    Type    :     25 GAIA_MTA
0x0307= Config-Bits     : 0x0001 Datastring on
0x0308= SecRelais Type  :      1 KILOVAC-DIG
0x0309= Current Factor  :   1500
0x030A= CAN Btr0 Btr1   : 0xC149 250 kbps
0x030B= CAN IDs ErrWrn  :  0x000 [->0x001]
0x030C= CAN IDs Wakeup  :  0x010 [->0x011]
0x030D= CAN IDs Data    :  0x100 [->0x101]
0x030E= CAN IDs Diag    :  0x102 [->0x103]
0x030F= CAN IDs PCM     :  0x104 [->0x105]
0x032A= CAN ID OPUSLP10 :  0x640 enable
0x0340= Use segmented BattModel:   0 disable
0x0341= Bat= # of par. Strings :   1
0x0342= Str= # of ser. Moduls  :   1
0x0343= Mod= # of Cells per Slv: 0x0000000A

------- PARAMETER ----------------------------
PCM-ID  Software Version       : 1.3.4 09.07.2008
0x0300: Version DataSpace      :           98
0x0100: Capacity               :     27000000 mAs   7.50 Ah
0x0101: Rtc-Wakeup Short       :           10 min
0x0102: Rtc-Wakeup Long        :          240 min
0x0103: Min KL30               :        24000 mV
0x0104: KL15_on_prz            :           25 %
0x0105: Err-U-Min-Cell         :         2700 mV
0x0106: Err-U-Max-Cell         :         4200 mV
0x0107: Warn-U-Min-Cell        :         2800 mV
0x0108: Warn-U-Max-Cell        :         4100 mV
0x0109: RI-0-Charge            :           10 0.1 mOhm
0x010A: RI-0-Discharge         :           20 0.1 mOhm
0x010B: RI-0-TempFakt          :            0
0x010C: Err-Temp-Min-Charge    :         -250 0.1 Cel
0x010D: Err-Temp-Max-Charge    :          600 0.1 Cel
0x010E: Err-Temp-Min-Discharge :         -300 0.1 Cel
0x010F: Err-Temp-Max-Discharge :          600 0.1 Cel
0x0110: Warn-Temp-Min-Charge   :         -200 0.1 Cel
0x0111: Warn-Temp-Max-Charge   :          500 0.1 Cel
0x0112: Warn-Temp-Min-Discharge:         -200 0.1 Cel
0x0113: Warn-Temp-Max-Discharge:          500 0.1 Cel
0x0114: Err-I-Max-Charge       :       120000 mA  16.00 C
0x0115: Err-I-Max-Discharge    :       150000 mA  20.00 C
0x0116: Warn-I-Max-Charge      :       110000 mA  14.67 C
```

```
0x0117: Warn-I-Max-Discharge   :       140000 mA  18.67 C
0x0118: IPeak-Max-Charge_Im    :       120000 mA  16.00 C
0x0119: IPeak-Max-Discharge_Im :       150000 mA  20.00 C
0x011A: IPeak-Max-Charge       :       120000 mA  16.00 C
0x011B: IPeak-Max-Discharg     :       150000 mA  20.00 C
0x011C: IPeak-40ms_N           :          750 n    30.0 s
0x011D: Warn KiloVAC OverCur-SD:            0 n   disable
0x011E: Err  KiloVAC OverCur-SD:            0 n   disable
0x011F: Err-RSafe-Slv-Min      :           20 Ohm
0x0120: Err-RSafe-Slv-Max      :          200 Ohm
0x0121: IBat-Ignore            :            9 mA
0x0122: Warn-SoC-Underflow     :          200 0.1 %
0x0123: Warn-SoC-Overflow      :          800 0.1 %
0x0124: Cycle-Time TU          :         1000 ms
0x0125: I-Bat0                 :          200 mA
0x0126: t-Bat0-min             :          600 s
0x0127: Dislog-DeltaUc         :           10 mV
0x0128: Dislog-MinUc           :         3400 mV
0x0129: Dislog-IdleCnt         :           10
0x012A: UCErr exp.timer        :           10 s enable
0x012B: UCErr I Charge-Ofl     :          100 mA
0x012C: UCErr I Discharge-Ufl  :          500 mA
0x012D: SD Errorcounter Limit U:            1 n
0x012E: SD Errorcounter Limit I:            1 n
0x012F: SD Errorcounter Limit T:            1 n
0x0130: SD Errorcounter Limit R:            1 n
0x0131: Use dynamic Ri         :            0 disable
0x0132: Use only Hi-Curr-Meas  :            0 no
0x0133: Dislog Activ if IBat > :         -500 mA
0x0134: General Application-ID :     00000000
0x0135: Max UCell-Diff Warning :            0 mV disable
0x0136: Max UCell-Diff Error   :            0 mV disable
0x0137: SD Errorcounter Limit D:            1 n
0x0138: SD Errorcounter Limit S:            1 n
0x0139: Simulate Slave Ntc     :            0 no
0x013A: Simulate Slave RSafe   :            0 no
0x013B: Master RSafe: Polarity :            0 normal
0x013C: SecSwitch   : Polarity :            1 invers
0x013D: SecSwitch -> ShDown KV?:            1 yes
0x013E: Allow CAN ext. CmdSet1?:            0 no
0x013F: CAN WatchDog Mode     ?:            0 no, disabled
0x0140: Cooling Start at       :          300 0.1 Cel
0x0141: Cooling Stop  at       :          250 0.1 Cel
0x0142: Cooling Fan-Port       :            1 Nr
0x0150: Delayed Shutdown Mode  : 00 : None
0x0151: Delayed Shutdown Time  :            0 n  disable
0x0152: Delayed Shdn Presettime:            0 n
0x0153: Usr-Tim-Chg  Limit  I >:            0 mA
0x0154: Usr-Tim-Dchg Limit -I <:            0 mA
0x0155: Fan-Overrides SoC  1..4:  00.00.00.00
0x0180: SoC-OCV Method number  :            0 nr
0x0182: SD Errorcnt Limit CanWD:            1 n
```

```
0x01E0: AutoDiagSend ADS_tab[0]:         0x00 -free-
0x01E1: AutoDiagSend ADS_tab[1]:         0x00 -free-
0x01E2: AutoDiagSend ADS_tab[2]:         0x00 -free-
0x01E3: AutoDiagSend ADS_tab[3]:         0x00 -free-
0x01E4: AutoDiagSend ADS_tab[4]:         0x00 -free-
0x01E5: AutoDiagSend ADS_tab[5]:         0x00 -free-
0x01E6: AutoDiagSend ADS_tab[6]:         0x00 -free-
0x01E7: AutoDiagSend ADS_tab[7]:         0x00 -free-
0x0200: soc_tab[ 0]  0%        :         3377 mV
0x0201: soc_tab[ 1] 10%        :         3450 mV
0x0202: soc_tab[ 2] 20%        :         3523 mV
0x0203: soc_tab[ 3] 30%        :         3596 mV
0x0204: soc_tab[ 4] 40%        :         3669 mV
0x0205: soc_tab[ 5] 50%        :         3742 mV
0x0206: soc_tab[ 6] 60%        :         3815 mV
0x0207: soc_tab[ 7] 70%        :         3888 mV
0x0208: soc_tab[ 8] 80%        :         3961 mV
0x0209: soc_tab[ 9] 90%        :         4034 mV
0x020A: soc_tab[10]100%        :         4107 mV
```