Multidisciplinary Optimization of Hybrid Electric Vehicles: Component Sizing and Power Management Logic

by

Brian Su-Ming Fan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2011

AUTHOR'S DECLARATION

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

A survey of the existing literature indicates that optimization on the power management logic of hybrid electric vehicle is mostly performed after the design of the powertrain architecture or the power source components are finalized. The goal of this research is to utilize Multidisciplinary Design Optimization (MDO) to automate and optimize the vehicle's powertrain component sizes, while simultaneously determining the optimal power management logic in developing the most cost-effective system solution.

A generic, modular, and flexible vehicle model utilizing a backward-looking architecture is created using scalable powertrain components. The objective of the research work is to study the energy efficiency of the vehicle system, where the dynamics of the vehicle is not of concern; a backward-looking architecture could be used to compute the power consumption and the overall efficiency accurately while minimizing the required computing resource. An optimization software platform utilizing multidisciplinary design optimization approach is implemented containing the generic vehicle model and an optimizer of the user's choice. The software model is created in the MATLAB/Simulink environment, where the optimization code and the powertrain component properties are implemented using m-files, and the power consumption calculations of the vehicle system are performed in Simulink. Furthermore, a feature-based optimization technique is developed with the motivation of significantly reducing the simulation run-time. To demonstrate the capabilities of the developed approach and contributions of the research, two optimization case studies are undertaken: (i) series hybrid electric vehicles, and (ii) police vehicle anti-idling system.

As the first case study, a plug-in battery-only series hybrid electric vehicle with similar power components as the Chevrolet Volt is created, where the battery size and the power management logic are simultaneously optimized. The objective function of the optimizer is defined from the financial cost perspective, where the objective is to minimize the initial cost of batteries, gasoline and electricity consumption over a period of five years, and the carbon tax as a penalty function for fuel emissions. The battery-only series hybrid electric vehicle is subsequently extended to include ultracapacitors, and the optimization process is expanded to the rest of the powertrain components and power management logic. A comparison between the optimization algorithms found that only genetic algorithm (GA) was capable of finding the optimal solution during a full simulation, while simulated annealing and pattern search were not able to converge to any solution after a 24-hour period. A comparison between the full genetic algorithm optimization and the feature-based (FB) method with secondary optimization found that although the final cost function of the FB methodology is higher than that of the full GA optimization, the total simulation run-time is approximately ten times less using the FB method. The behaviour of the solutions found via both methods exhibited almost identical characteristics, further confirming the validity of the feature-based methodology. Finally, a benchmarking comparison found that with more accurate manufacturers' component data and additional appropriate performance requirements, the proposed software platform will be capable of predicting a solution that is comparable to the Chevrolet Volt.

The second case study involves optimizing an anti-idling system for police vehicles using the same optimization algorithm and generic vehicle model. The goal of the optimization study is to select an additional battery and determine the power management logic to reduce the engine idling time of a police vehicle. It is found that depending on the SOC threshold, the duration of time over which the engine is activated varies in a non-linear fashion, where local minima and maxima exist. A design study confirmed that by utilizing the anti-idling system, significant cost reduction can be realized when compared to one without the anti-idling system.

A comparison between the various optimization algorithms showed that the feature-based optimization can obtain a relatively accurate solution while reducing simulation time by approximately 90%. This significant reduction in simulation time warrants the feature-based optimization technique a powerful tool for vehicle design. Due to the high cost of the electrical energy storage components, it is currently still more cost-effective to use the fossil fuel as the primary energy source for transportation. However, given the rise of fuel cost and the advancement in the electrical energy storage technology, it is inevitable that the cost of the electrical and chemical energy storage method will reach a balance point. The proposed optimization platform allows the user the capability and flexibility to obtain the optimal vehicle solution with ease at any given time in the future.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my thesis supervisors, Dr. Amir Khajepour in the department of Mechanical and Mechatronics Engineering, and Dr. Mehrdad Kazerani in the department of Electrical and Computer Engineering. This dissertation would not have been possible without their teaching, guidance and encouragement throughout the completion of my study.

I would also like to thank my committee members: Dr. Kaan Erkorkmaz and Dr. John Wen in the department of Mechanical and Mechatronics Engineering, Dr. Claudio Canizares in the department of Electrical and Computer Engineering; and my external examiner, Dr. Giorgio Rizzoni in the department of Mechanical and Aerospace Engineering from the Ohio State University; for their thorough review and valuable suggestions to further improve the quality of my dissertation.

Last but not least, I would like to thank my family, my parents Ellen and K.C., and my sister Sharon. No words can express my utmost appreciation for their unconditional support, inspiration, and motivation over the years of pursuing this degree.

Table of Contents

List of Figures

List of Tables

Nomenclature

$\%_{driving}$	Percent of total cycle time when vehicle is driving [%]
$\dot{m}_f$	Mass flow of fuel [g/s]
A	Frontal area of the vehicle [m^2]
a	Vehicle acceleration [m/s^2]
a_{des}	Desired acceleration of the drive cycle [m/s^2]
a_{fea}	Acceleration vector containing the acceleration range of the drive cycle [m/s^2]
B	IC Engine cylinder bore [m]
$B_{\#}$	Number of battery banks
B_k	Symmetrical and nonsingular matrix of the line search method
$Batt_{cap}$	Battery capacity [Ah]
b_{ch}	Slope of the battery SOC charging power distributing function
b_{dis}	Slope of the battery SOC discharging power distributing function
C	IC Engine fuel consumption [L]
C_D	Drag coefficient of the vehicle
C_{RR}	Rolling resistance constant of the tire
C_{UC}	Ultracapacitor capacitance [F]
$C_{eqv,gas}$	Equivalent gasoline consumption [L]
$C_{f,sec}$	Fuel consumption vector of each of the section of the drive cycle in the feature-based simulation [L]
$C_{f,tot}$	Total fuel consumption of the drive cycle in the feature-based simulation [L]
c_{ch}	Inflection point of the battery SOC charging power distributing function
c_{dis}	Inflection point of the battery SOC discharging power distributing function
c_m	IC Engine mean piston speed [m/s]
D_k	Search direction of the pattern search method
d_{ch}	Slope of the ultracapacitor SOC charging power distributing function
d_{dis}	Slope of the ultracapacitor SOC discharging power distributing function
E_{map}	Energy map generated by the Simulink model in the feature-based simulation [J]
E_{tot}	Total energy consumption of the drive cycle in the feature-based simulation [J]
e	Thermodynamic properties of the engine related to the mean effective pressure
e_{ch}	Inflection point of the ultracapacitor SOC charging power distributing function

e_{dis}	Inflection point of the ultracapacitor SOC discharging power distributing function
F_{RR}	Tire rolling resistance force [N]
F_{drag}	Vehicle drag force [N]
F_{drive}	Vehicle drive force [N]
F_{tot}	Algebraic sum of the vehicle's driving and resisting forces [N]
f_{ch}	Slope of the desired power charging power distributing function
f_{dis}	Slope of the desired power discharging power distributing function
g	Gravitational acceleration [m/s^2]
g_{ch}	Inflection point of the desired power charging power distributing function
g_{dis}	Inflection point of the desired power discharging power distributing function
H_l	Lower heating value of gasoline [J/g]
H_{sec}	Histogram containing the occurrence counts for each section of the drive cycle
h_{dis}	Slope of the rate of change of desired power discharging power distributing function
I_{UC}	Ultracapacitor current [A]
$I_{alt,max}$	Alternator maximum current output [A]
I_{aux}	Current consumption of the auxiliary components [A]
I_{batt}	Battery current [A]
I_{ideal}	Battery ideal current without internal resistance [A]
i_{ch}	Inflection point of the rate of change of the desired power charging power distributing function
i_{dis}	Inflection point of the rate of change of the desired power discharging power distributing function
J	Optimizer objective or fitness function
K	Mass scaling factor of the feature-based optimization
K_{eqf}	Equivalence factor of the Equivalent Consumption Minimization Strategy
k	Battery degradation scaling parameter
k_{SA}	Boltzmann's constant of the simulated annealing method
m	Number of sections that the drive cycle is divided into in the feature-based simulation
m_{veh}	Vehicle mass [kg]
N	IC Engine type: N=4 for a four-stroke engine, N=2 for a two-stroke engine
$P_{UC,des}$	Desired ultracapacitor power [W]
P_{act}	Total available power from the battery and the genset [W]

$P_{batt,act}$	Battery actual terminal power [W]
$P_{batt,chg}$	Battery charging power limit [W]
$P_{batt,des}$	Desired battery power [W]
P_{des}	Desired vehicle power [W]
P_e	IC Engine power [W]
P_{elec}	Electric power consumed or generated by the motor-generator [W]
P_{fuel}	Enthalpy flow of fuel [J/s]
$P_{gen,eff}$	Genset power output determined by the optimizer [W]
$P_{m,req}$	Electrical power required by the motor [W]
P_{map}	Power map generated by the Simulink model in the feature-based simulation [W]
P_{regen}	Motor regenerative braking power [W]
p_k	Search direction of the line search method
p_{loss}	IC Engine losses due to gas exchange and friction [Pa]
p_{loss_f}	IC Engine losses due to gas friction [Pa]
p_{loss_g}	IC Engine losses due to gas exchange [Pa]
p_{me}	IC Engine mean effective pressure [Pa]
p_{mf}	Fuel mean effective pressure [Pa]
Q_{UC}	Ultracapacitor charge [C]
$Q_{UC,nom}$	Ultracapacitor normal charge [C]
R_{ESR}	Ultracapacitor equivalent series resistance [Ω]
R_{int}	Battery internal resistance [Ω]
r_t	Vehicle tire radius [m]
S	IC Engine stroke [m]
SOC_{UC}	Ultracapacitor SOC [%]
SOC_{batt}	Battery SOC [%]
SOC_{final}	Battery final SOC [%]
T	Total time of the drive cycle [s]
T_{SA}	Temperature of the simulated Annealing method
T_{drive}	Powertrain drive torque [Nm]
T_e	IC Engine torque [Nm]
T_m	Electric motor torque [Nm]
t_s	Time step of the drive cycle [s]

U	Multidisciplinary design optimization system output variables
$V_{UC,oc}$	Ultracapacitor open circuit voltage [V]
$V_{batt,oc}$	Battery open circuit voltage [V]
V_{chg}	Battery charging voltage [V]
V_d	IC Engine displacement [m^3]
V_t	Battery terminal voltage [V]
v_{act}	Actual vehicle speed [m/s]
$v_{avg,d}$	Average vehicle speed during a delayed cycle [m/s]
v_{des}	Desired vehicle speed given by the drive cycle [m/s]
v_{fea}	Velocity vector containing the velocity range of the drive cycle [m/s]
v_x	Longitudinal speed of the vehicle [m/s]
X_D	Multidisciplinary design optimization design variables
X_Y	Multidisciplinary design optimization promoted design variables
$x_{tot,d}$	Total distance travelled during a delayed cycle [m]
α	Battery degradation function
α_k	Step length of the line search method
Δt	Additional time required to complete the drive cycle [s]
η_{alt}	Alternator efficiency [%]
η_{batt}	Battery efficiency [%]
η_e	Engine efficiency [%]
η_m	Electric motor efficiency [%]
μ_{ch}	Charging power distributing function of battery [%]
μ_{dis}	Discharging power distributing function of battery [%]
ρ_{air}	Density of air [kg/m^3]
ρ_{gas}	Density of gasoline [g/L]
ω_e	IC Engine speed [rad/s]
ω_m	Electric motor speed [rad/s]

Chapter 1
Introduction

In recent years, the global economy and industrial world have stridden towards alternative green technologies in the face of climate change. Automobiles are currently a major source of air pollution, prompting collaborations among governments, academia, and industrial institutions to search for a solution to reduce vehicle emissions, while reducing the consumption of fossil fuels. Hybrid electric vehicle systems became one of the best working solutions by utilizing the advantages of both internal combustion (IC) engine and electric energy source. By definition, a hybrid vehicle is one that employs two or more power sources to improve the overall efficiency of the system. The advantage of an IC engine is that fuels with high-energy content can be transported with ease, while the disadvantage is that burning of fossil fuels creates emissions that are hazardous to the environment. Alternatively, an electric vehicle uses electric energy from a battery or fuel cell, and converts it into kinetic energy via electric motors. The advantage of an electric vehicle is that zero emissions are produced when electric energy is converted into kinetic energy. However, current electrical energy storage technologies do not present a working solution with reasonable vehicle cost and range. By combining an IC engine with an electric battery-motor system, the problem of energy portability can be solved. In addition to achieving low emissions and reducing fuel consumption, hybrid electric vehicle can recapture the otherwise lost kinetic energy during the braking cycle, further improving the efficiency of the vehicle system. In order to increase the efficiency and accuracy of automotive design, Computer Aided Engineering (CAE) has played an ever increasingly significant role throughout the process of vehicle design. With the increase of computing power, manufacturers are now able to perform design, testing, and optimization of a vehicle through computer simulation, all prior to the actual manufacturing of a vehicle. Given the complexity of automobile design, the greatest challenge for automotive engineers is to research and optimize component designs in their respective field while communicating with other disciplines to determine the optimal vehicle system design. Only in recent years, CAE software products such as topology optimization from Hyperworks for structural optimization and MSC Software MD which combines ADAMS (dynamics) and NASTRAN (finite element analysis) for multi-disciplinary simulations, became available to assist automotive engineers in realizing optimal solutions across various disciplines.

The key contribution of the research is to develop a Multidisciplinary Design Optimization (MDO) methodology for hybrid electric vehicle design. MDO is currently widely utilized in the aerospace industry, where engineers seek a balance between the performances of aerodynamics and structural design. However, in the field of hybrid electric vehicles, researchers are still working on improving efficiencies and performance at the component level, rather than from the system perspective. A survey of the existing literature indicated that optimization on the power management logic is mostly performed after the design of the powertrain architecture or the power source components were finalized. The goal of this research is to utilize MDO approach to automate and optimize the hybrid electric vehicle's powertrain component sizing, while simultaneously determining the optimal power management logic in developing the most effective system solution. The objective function seeks to minimize the cost from a financial perspective rather than only fuel consumption or emissions. Since the target user of the optimized hybrid electric vehicles is the consumer market, it is more realistic to propose a cost conscious solution in balancing the size of the electrical energy storage devices while minimizing the consumption of fossil fuel. On the other hand, if the target user is for the defense industry where financial cost is not the top priority, the objective function can be easily adjusted to maximize the performance of the system. Using the proposed methodology, an automotive engineer will perform concurrent optimization at the beginning of the design cycle based on the vehicle design objective, and subsequently finalize the detailed design of each of the components only after the most optimal solution has been found. Such methodology not only allows the designer to realize the most optimal system, but also greatly improves the efficiency of the design process while reducing developmental cost.

The developed approach utilizes Multidisciplinary Feasible (MDF) method for multidisciplinary design optimization. Various optimization techniques are implemented to search the design space containing scalable power components and the power management logic parameters. The objective is to develop an optimization software platform to perform concurrent vehicle optimization while determining the most suitable and effective optimization process. To demonstrate the effectiveness and the contribution of the research, concurrent optimizations are performed and demonstrated in two case studies: (i) series hybrid electric vehicles, and (ii) police vehicle anti-idling system. Chapter 2 will first provide some definitions for hybrid electric vehicles, as well as a literature survey of some of the existing optimization approaches and MDO methodologies. Chapter 3 will present the generic vehicle model along with its scalable powertrain components and the power management logic of the electrical energy storage system. Chapter 4 will discuss the overall software structure using the MDF method and the various optimization algorithms available. Furthermore, detailed derivation of the proposed feature-based optimization method will be presented. Chapters 5 and 6 will present the design objectives and simulation results of the two case studies, i.e., series hybrid electric vehicle and police vehicle anti-idling system, respectively. Finally, Chapter 7 will make conclusions based on the research undertaken and will highlight the contributions of the thesis.

Chapter 2
Literature Review

As hybrid electric vehicles (HEVs) are gaining widespread attention and popularity in the industry and the research community, various powertrain architectures and power management schemes have been proposed in order to improve the vehicle's fuel economy and to reduce emissions. This chapter provides an overview of the existing hybrid electric vehicle powertrain structures, and reports surveys of the previously proposed power management controller techniques. In addition, powertrain sizing optimizations are presented, along with examples of existing concurrent optimization on the powertrain sizing and power management logic. Finally, due to the multi-disciplinary nature of HEV, the concept of multidisciplinary design optimization along with the proposed methodology and theory is discussed.

2.1 Hybrid Electric Vehicle Configuration

The most successful hybrid configuration currently utilized by various vehicle manufacturers consists of a gasoline or diesel engine, coupled with a motor and a generator linked with a battery system. Although there exist many different hybrid configurations, most can be categorized under two hybrid system classes: (i) Series Hybrid and (ii) Parallel Hybrid.

2.1.1 Series Hybrid

In the series hybrid system, the IC engine drives the generator, where electricity is generated and supplied to the battery. It is also sometimes referred to as an electric vehicle with a range extender in the industry. The electrical energy from the battery is then delivered to the motor, which in turn drives the wheels to propel the vehicle. Figure 2-1 illustrates the system configuration of a series hybrid electric vehicle [1].

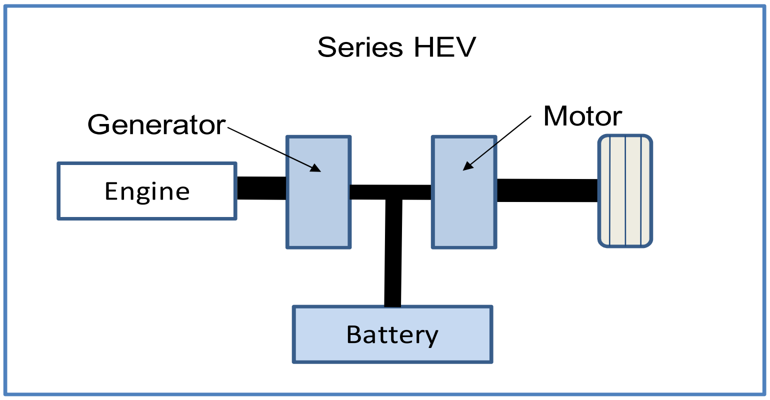


Figure 2-1: Schematic of a Series Hybrid Electric Vehicle [1]

The advantage of the series hybrid architecture is that the engine runs at its best efficiency to generate electrical energy to charge the battery. Since the engine is constantly operating at its optimum efficiency, and the vehicle receives its power solely from the electric motor, this system is most efficient during the stop and go of city driving. In addition, the internal combustion engine and generator of the series hybrid electric vehicle can be replaced by a fuel cell and a DC-DC converter, thus converting it into a pure electric vehicle. The disadvantage of a series hybrid electric vehicle is in that the efficiency of the system is reduced during highway driving cycles. During highway driving, energy losses during the conversion process in addition to the lower torque output of the electric motor at high rotational speeds contribute to the overall lower efficiency of the system [1].

2.1.2 Parallel Hybrid

The parallel hybrid configuration switches between the two power sources, i.e., the internal combustion engine and the electric motor, where the high-efficiency range of each is selected and utilized. Depending on the situation, both power sources can also be used simultaneously to achieve maximum power output and peak performance. Figure 2-2 shows the system configuration of a parallel hybrid electric vehicle [1].

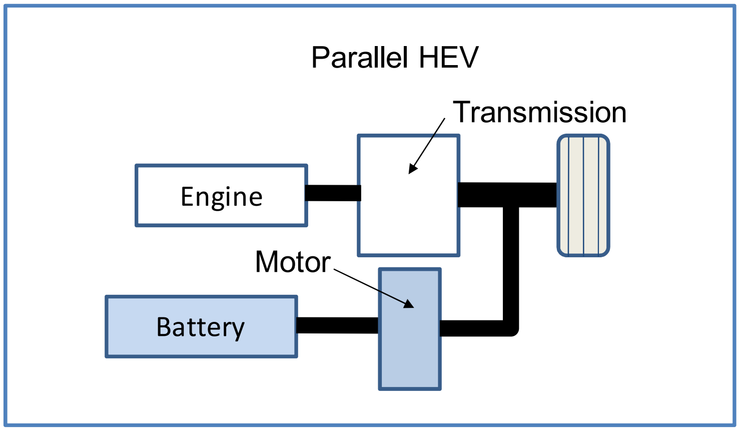


Figure 2-2: Schematic of a Parallel Hybrid Electric Vehicle [1]

The advantage of a parallel hybrid electric vehicle is in that the system has the ability to offer higher efficiency during highway driving conditions. During highway driving, the vehicle speed does not vary significantly and therefore it is more efficient to drive the wheels directly from the IC engine. On the other hand, the electric motor can be used solely during city driving to prevent the IC engine from operating in its low-efficiency range, thus providing higher overall efficiency [1].

2.2 Power Management Control Strategies

As hybrid electric vehicles (HEVs) are gaining more popularity in the market, the efficiencies of the power management system in the hybrid powertrain are receiving increasing attention in the research communities worldwide. Majority of the proposed solutions for the power management control logic can be classified under two types: (i) rule-based approach and (ii) optimization-based approach. Rule-based control strategies consist of deterministic and fuzzy logic rule-based methods, while optimization-based approaches typically utilized global optimization when determining the control strategy [2]. The following sections will provide an overview of the existing power management control strategies in details.

2.2.1 Deterministic Rule-Based Methods

Deterministic rule-based methods are usually based on analysis of power flow in a hybrid drivetrain, efficiency/fuel maps of ICE, and human experiences, generally implemented in the form of lookup tables and by splitting powers between power sources [2]. In the following subsections two types of deterministic rule-based methods will be described.

2.2.1.1 Power Follower Baseline Control Strategy

The baseline control strategy is used in the parallel hybrid configuration, and uses the engine as a primary source of torque, while the electric motor supplements additional power when required. When the battery SOC is low, the system switches to charging mode in order to recharge the battery. The following rules depict the baseline control strategy.

1. Only the electric motor is used below a certain minimum vehicle speed.
2. If the power demand is greater than the maximum engine power at its operating speed, the electric motor is used to provide the additional required power.
3. The batteries are recharged by regenerative braking.
4. The engine shuts off when the power demand falls below a limit at the operating speed to prevent inefficient operation of the engine.
5. If the battery SOC reaches its lower threshold, the engine provides additional power to recharge the battery.

This is a popular strategy for power management in current hybrid systems. For example, the basic control strategy of the Toyota Prius is that the motor provides additional power when required. Additionally, the motor is also exclusively used when the vehicle accelerate from standstill and at low speed. Similarly, the Honda Insight uses the IC engine as the primer power source, with the electric motor assisting the engine during acceleration and when starting from standstill. Even though such an approach is popular and widely implemented, it suffers the drawback that the efficiency of the entire powertrain is not optimized [2,3].

2.2.1.2 Modified Power Follower Control Strategy

In order to improve the baseline control strategy, Johnson *et al.* [3] proposed an adaptive rule-based power management strategy. The main goal of this approach is to optimize both energy usage and emissions by introducing a cost function representing overall fuel consumption and emissions at all candidate operating points. The control strategy uses a time averaged speed to obtain the instantaneous energy use and emission targets. The proposed control strategy can be described as follows.

1. Define the range of candidate operating points represented by the range of acceptable motor torques for the current torque request.
2. For each candidate operating point, calculate the constituent factors for optimization.
    a. Calculate the fuel energy that would be consumed by the engine.
    b. Calculate the effective fuel energy that would be consumed by the electromechanical energy conversion.
    c. Calculate the total energy that would be consumed by the vehicle.
    d. Calculate the emissions that would be produced by the engine.
3. Normalize the constituent factors for each candidate operating point.
4. Apply user weightings to the results from step 3.
5. Apply target performance weightings to the results from step 4.
6. Compute overall impact function, a composite of results from steps 3-5, for all candidate operating points.

The final operating point is the operating point with the minimum impact factor. Although this modified strategy has improved the problems associated with the baseline approach, repeating the above steps for all candidate operating points is not desirable for online implementation [2,3].

2.2.2 Fuzzy Rule-Based Methods

Due to the multi-domain, nonlinear, and time-varying nature of the hybrid electric vehicle's powertrain, many researchers have investigated the implementation of fuzzy logic as a solution. Instead of using deterministic rules, the decision making property of fuzzy logic can be adopted to realize a real-time power-split controller [2]. The past work performed by researchers on applying fuzzy logic to hybrid electric vehicle powertrain can be classified under the following categories.

2.2.2.1 Conventional Fuzzy Strategy

Schouten *el al.*[4] developed a fuzzy logic-based power management control logic that included a compression ignition (CI) engine, an electric motor, and a battery system, and was particularly designed for a parallel hybrid electric vehicle. It was stated that the most efficient operating region of the battery occurred in the high SOC and low-power region for both charge and discharge, meaning that the battery should be frequently charged at low power levels. Therefore, when the power command is below 6kW, only the electric motor is used to drive the vehicle. Between 6 and 50 kW, only the CI engine is used to propel the wheels and charge the battery, if necessary. If the power command is over 50 kW, both the electric motor and the CI engine are used. The proposed fuzzy logic controller determines the optimal generator power and a scaling factor for the electric motor during motor mode. The inputs are the driver power command, the battery SOC, and the electric motor speed. When the SOC is high, the scaling factor equals to one. On the other hand, when the SOC is low, the scaling factor is set to zero to prevent battery damage. Sample rules of the Fuzzy Logic Controller (FLC) are as follows:

1. IF SOC is Low, Pdes is Normal, and ωm is Low, THEN Pgen is 5 kW
2. IF SOC is Low, Pdes is Normal, and ωm is not Low, THEN Pgen is 15 kW

The rules suggest that if the SOC is low, and the requested power is normal, and the electrical motor's rotational speed is close to its optimum efficient region, the battery will be charged at a higher power level than when the electric motor speed is low. Finally, the engine and the motor powers were computed based on the output of the FLC and P_{des}, using simple deterministic rules. The advantage of this approach is that the operating points for the CI engine, electric motor, and the battery can be controlled in their optimal efficiency regions. The drawback, however, is that the resultant vehicle emissions are not taken into account [2,4].

2.2.2.2 Fuzzy Predictive Strategy

An alternate method proposed by researchers to achieve optimal solution is based on minimizing an appropriate cost function over a drive cycle, attainable by knowing the entire trip information beforehand. The problem is to perform real-time control tasks, while accounting for situations in the future along a planned route. In such scenario, Global Positioning System (GPS) can obtain prior knowledge of the vehicle operating environment, i.e., heavy traffic, road grade, etc. The fuzzy logic predictive controller adapts the instantaneous controller parameters to the predictions from future states such as the road grade or speed dictated by traffic conditions. The inputs to the predictive controller are the change in vehicle speed corresponding to the recent speeds, the predicted speed, and the road grade along the predetermined route from the navigational system. The controller then determines the actions to be performed, based on the recent history of the motion of the vehicle, and applies the changes in the near future. The prerequisite for such system is that detailed road grade and traffic information in real time must be known at all time. The approach yields the closest to optimal solution as far as the vehicle operating efficiency is concerned, however, due to the current level of roadway infrastructure, it is still unrealistic to implement such controller for mass production [2,5].

2.2.3 Optimization-Based Methods

The goal of the optimization-based control strategies is to optimize the output power of the power components by minimizing a cost function typically represented by the fuel consumption and/or emissions. Global optimum solution can be found by performing global optimization over a fixed drive cycle, which is non-casual since it finds the minimum fuel consumption using knowledge of future and past power demands. The drawback of such approach is that it cannot be used directly for real-time power management. However, it can be used as a basis for designing rules for online implementation or comparison for evaluating other control strategies.

2.2.3.1 Dynamic Programming

Dynamic programming (DP) is a powerful tool to solve general dynamic optimization problems, due to its ease of handling the constraints and nonlinearity of the problem while obtaining a globally optimal solution. Optimal solution can be found by minimizing the optimization parameters by evaluating the objective function at every time step of the drive cycle. The drawback however, is the complexity and the expensive computational resources required to obtain the solution [6,7,8].

Lin et al. [9] applied the DP technique to solve the optimal power management problem of a hybrid electric truck by minimizing fuel consumption, NOx, and emissions as cost functions over a drive cycle. To reduce the computational burden of the DP, only three state variables, the vehicle speed, transmission gear number, and the battery SOC were included in the state vector x to implement a dynamic model in the form of x(k+1)=f(x(k),u(k)) for the hybrid truck under study. The control variables u(k) contains the desired output torque from the engine/motor and gear shift command to the transmission. The overall dynamic optimization problem can be decomposed into a sequence of simpler minimization problems as follows.

$$\begin{aligned}&\text{Step } N-1:\\ &J^*_{N-1}\big(x(N-1)\big) = \min_{u(N-1)}\big[L\big(x(N-1),u(N-1)\big) + G(x(N))\big]\\ &\text{Step k, for } 0 \leq \text{k} < N-1:\\ &J^*_k\big(x(k)\big) = \min_{u(k)}\big[L\big(x(k),u(k)\big) + J^*_{k+1}(x(k+1))\big]\end{aligned} \tag{2.1}$$

where $J^*_k(\boldsymbol{x(k)})$ is the optimal cost-to-go function or optimal value function at state $\boldsymbol{x(k)}$ starting from time state k. It represents the optimal resulting cost that at stage k, the system starts at state $\boldsymbol{x(k)}$ and follows the optimal control law thereafter until the final stage. The above recursive equation is solved backward to find the optimal control policy [2,9].

Another solution proposed by Perez *et al.* [10] is to utilize dynamic programming to determine the optimal solution of a series HEV. The control objective is to determine the value of the engine and the motor power to minimize the fuel consumption. Since the total required power (P_{req}) is the sum of the engine, or the power from the fuel tank (P_{FT}), and the motor, or the power from the electrical storage system (P_{ESS}), and either P_{FT} or P_{ESS} can be taken as the control action or independent variable. The dynamic programming equations can be expressed in the following recursive algorithm.

$$\begin{aligned} &V_N(i) = a_{it}^N, \qquad i \in S_N \\ &V_k(i) = \min_{j \in S_{k+1}} \left[a_{ij}^k + V_{k+1}(j)\right], \qquad i \in S_k, k = 0, \dots, N-1 \end{aligned} \tag{2.2}$$

where a^N_{it} is the arc-cost from node *i* to stage *N* to a fictitious terminal node t and $V_k(i)$ is the minimum cost from node *i* to stage *k* to the terminal node. This algorithm is known as the backward algorithm. In order to manage the integral constraints, a penalization term is introduced in the objective function. The arc-cost is modified as follows.

$$\begin{aligned} a_{ij}^k = &\frac{1}{2}\left(\frac{P_{FT\ i,k}}{\eta_{FT\ i,k}} + \frac{P_{FT\ j,k+1}}{\eta_{FT\ j,k+1}}\right)\Delta t \\ &+ \alpha \left|\frac{1}{2}\left(\frac{P_{req}(k\Delta t) - P_{FT\ i,k}}{\eta_{ESS\ i,k}} + \frac{P_{req}((k+1)\Delta t) - P_{FT\ j,k+1}}{\eta_{ESS\ j,k+1}}\right)\Delta t\right| \end{aligned} \tag{2.3}$$

The parameter *α* is chosen by trial and error for the demanded cycle, and the resulting consumed energy profile satisfies the constraint for each *t*. The DP algorithm can then be used to determine the split between the two power sources given a known drive cycle [10].

2.2.3.2 Equivalent Consumption Minimization Strategy

Equivalent Consumption Minimization Strategy (ECMS) solves the local optimization problem instantaneously by considering the total energy consumption, while maintaining a constant battery state of charge (SOC). Essentially, ECMS regulates the SOC around a reference point while providing the required power at the wheels and achieving minimum fuel consumption. The concept of equivalent fuel consumption is based on the fact that in a hybrid powertrain the energy consumption from the battery is replenished by running the engine, and it is used in the objective function for the control optimization. The objective function for the ECMS is

$$J(x) = \dot{m}_{ICE}(k) + \dot{m}_{batt,eq}(k) \tag{2.4}$$

where $\dot{m}_{ICE}(k)$ is the fuel consumed by the IC engine. The $\dot{m}_{batt,eq}(k)$ is the equivalent fuel consumed while charging/discharging the battery, k is the discrete time index. The equivalent fuel by the battery is

$$\dot{m}_{batt,eq}(k) = \frac{K_{eqf}\eta_{total}}{H_l}u \tag{2.5}$$

where *u* is the battery power which is a control input, K_{eqf} is the equivalence factor that acts as a weighting factor for the electric energy, η_{total} is the average efficiency of the electric drivetrain including the battery charge-discharge and the electric machine efficiency, and H_l is the lower heating value of the fuel. The equivalence factor is very important and it affects the optimum power sharing between the engine and the motor. Further details of the equivalence factor for different application and derivation can be found in [11,12,13].

2.2.3.3 Particle Swarm

Wu *et al.* [14] proposed a control strategy parameter optimization using particle swarm optimization method for a series plug-in electric hybrid vehicle. The goal of the control logic is to manage the energy consumption of the engine and the electric motor such that when the battery state of charge (SOC) is high, the energy consumption will be primarily from the electric source. Once the SOC drops below a lower limit threshold, the engine will be used as the primary energy source, while maintaining the battery's SOC to prevent damage and cycle life reduction. Table 2-1 describes the parameters of the control strategy to be optimized.

Table 2-1: Energy Management Strategy Parameter for Particle Swarm Optimization [14]

Parameter	Description
LSOC	Lower limit on the battery State of Charge
HSOC	Upper limit on the battery State of Charge
Tch	Torque load on engine to recharge the battery when the engine is on
Tmin	Fraction of maximum engine torque above which the engine must operate if SOC<LSOC
VL	Vehicle Speed below which the vehicle attempts to run all electrically at low SOC
VH	Vehicle Speed below which the vehicle attempts to run all electrically at High SOC

In this work, the fuel economy (FE) is selected as the optimization target, and the objective function is defined as follows.

$$J(x) = \frac{FE_{st}}{\int FE(t)dt} \tag{2.6}$$

The problem can be defined as the solution for a constrained nonlinear programming problem described as

$$\begin{gathered} minJ(x) \\ x \in \Omega \\ s.t. g_i(x) \leq 0 \quad i = 1,2,\dots,n \end{gathered} \tag{2.7}$$

where Ω is the solution space, $g_i(x) \leq 0$ a group of nonlinear constraints, $J(x)$ the objective function and n the number of constraints. To apply particle swarm optimization to the control strategy parameters of the HEV, a fitness function is required to evaluate the performance of each particle. However, since particle swarm optimization is applicable only to unconstrained optimization problem, the constraints are handled by using a penalty function that penalizes the infeasible solutions by adding their fitness values. The fitness function is described as:

$$F(x) = J(x) + h(k)H(x) \tag{2.8}$$

where $h(k)$ is a dynamically modified penalty value, k the algorithm's current iteration number, and $H(x)$ the penalty factor. The optimized parameters were used to perform two drive cycles, and the results were compared to those of ADVISOR (ADvanced VehIcle SimulatOR). It was concluded that the optimal parameters successfully reduced the fuel consumption when compared to the original model [14].

2.2.3.4 Genetic Algorithm

Huang *et al.* [15] conducted power management control strategy optimization on a series hybrid electric vehicle utilizing genetic algorithm, and compared the results against those of Thermostatic and DIRECT (DIvided RECTangles). The optimization problem is defined as:

$$\begin{gathered} minJ(x_i) \\ s.t. g(x_i) \geq 0 \quad i = 1,2,\dots,n \end{gathered} \tag{2.9}$$

where x_i consist of parameters for power control strategy and $g(x_i) \geq 0$ is a group of nonlinear inequality constraints. The optimization objectives are the fuel economy and emission (NOx, CO, and HC) reduction, where each component of the fitness function is weighted by factor w_i, which can be used to take into account the relative importance of each objective. For the SHEV model, there are five possible operation modes: electric power only, fuel power only, power-assist (electric power plus fuel power), recharging, and regenerative braking. The control strategy determines the torque and speed of which the engine should operate at, to generate electric power by the generator. Table 2-2 shows the upper and lower limits of the controls parameters optimization variables and the results of genetic algorithm [15].

Table 2-2: Powertrain Optimization Parameters of Genetic Algorithm [15]

Parameter	Lower Limit	Upper Limit	GA Results
SOC Upper Limit [%]	50	90	90
SOC Lower Limit [%]	10	50	30.47
Min Power Command [kW]	0	25	16.557
Max Power Command [kW]	25	50	34.651
Charge Power [kW]	0	25	12.539
Engine-off Time [s]	10	1000	434

Upon comparing the results, it was found that genetic algorithm performed significantly better than those of Thermostatic control and DIRECT, demonstrating the effectiveness of the utilizing genetic algorithm for the power control strategy optimization [15].

2.2.3.5 Simulated Annealing

In conjunction with the research performed by Huang *et al.* [15], Wang *et al.*[16] conducted the power management control optimization utilizing simulated annealing algorithm. The problem statement and the fitness function remained the same as those introduced in Subsection 2.2.3.4, while utilizing different power component sizes. The results obtained from simulated annealing were compared to those of DIRECT, and indicated that simulated annealing provided significantly better results. Table 2-3 summarizes the upper and lower limits of the optimization variables and the results of simulated annealing [16].

Table 2-3: Powertrain Optimization Parameters of Simulated Annealing [16]

Parameter	Lower Limit	Upper Limit	SA Results
SOC Upper Limit [%]	50	90	56.67
SOC Lower Limit [%]	10	50	34.44
Min Power Command [kW]	0	20.5	17.083
Max Power Command [kW]	20.5	41	26.194
Charge Power [kW]	0	20.5	3.416
Engine-off Time [s]	10	1000	434

2.3 Hybrid Electric Vehicle Optimization

In addition to power management control strategy optimization, additional research had been focusing on the optimization of the powertrain components. Furthermore, concurrent optimization had been conducted in attempt to incorporate the optimizations on the powertrain sizing and the power management logic simultaneously. The following sections describe some of the powertrain sizing optimization on different hybrid configurations. Additionally, examples on existing concurrent optimization on hybrid electric vehicles are discussed.

2.3.1 Powertrain Optimization

2.3.1.1 Series Hybrid Electric Vehicle

A design optimization was undertaken for a series hybrid electric mini-bus designed for the Beijing Olympic gymnasium by Liu *et al.* [17]. The vehicle model was created using advisor, and a real-coded, adaptive based hybrid genetic algorithm combined with a local search method SQP was used to optimize the vehicle. Details of the improvements of the genetic algorithm can be found in [17]. The optimization problem is to minimize the fuel economy of the vehicle, where a city-highway test procedure is used to evaluate the fuel economy calculated by the following equation.

$$MPG_{tot} = \frac{1}{\frac{0.55}{MPG_{city}} + \frac{0.45}{MPG_{hwy}}} \tag{2.10}$$

where MPG_{tot} is the overall fuel economy, MPG_{city} the fuel economy of the city drive cycle, and MPG_{hwy} corresponds to the highway fuel economy. The population size was set as 20, and the optimization program was terminated after 50 iterations. The component sizing results is summarized in Table 2-4.

Table 2-4: Series Hybrid Electric Bus Component Sizing Results [17]

Parameter	Before Optimization	After Optimization
Fuel Converter Max Power (kW)	38	82
Generator Max Power (kW)	30	78
Motor Max Power (kW)	120	82
Battery Number	28	20
Battery Capacity (Ah)	85	42.5

It was found that the optimizer increased the size of the engine and the generator, while reducing the size of the motor and the battery size, leading to the conclusion that the original vehicle may have been oversized. Finally, even though the acceleration and top speed performance of the optimized vehicle was less than the original vehicle, the fuel economy was improved [17].

2.3.1.2 Parallel Hybrid Electric Vehicle

A parallel hybrid electric vehicle powertrain optimization was conducted by Gao and Porandla [18], utilizing PSAT for vehicle modeling. Three optimization algorithms were used to perform the design optimization: DIRECT (DIvided RECTangles), simulated annealing, and genetic algorithm. The objective was to increase the overall fuel economy of the parallel HEV on a composite city and highway driving cycle, as defined in Equation (2.10). Table 2-5 shows the design variables and their corresponding upper and lower bound, while Table 2-6 summaries the results of the optimization algorithms. [18]

Table 2-5: Parallel Hybrid Powertrain Design Variables [18]

Description	Lower Bound	Upper Bound
Fuel Converter Power (kW)	40	100
Motor Controller Power (kW)	10	80
Number Battery Cells	150	350
Minimum SOC Allowed (%)	20	40
Maximum SOC Allowed (%)	60	90
Final Drive Ratio	2	4

Table 2-6: Parallel Hybrid Powertrain Optimization Results [18]

Description	Original	DIRECT	Simulated Annealing	Genetic Algorithm
Fuel Converter Power (kW)	86	83.1	82.4	53.8
Motor Controller Power (kW)	65.9	20.2	21.9	65.4
Number Battery Cells	240	245	311	220
Minimum SOC Allowed (%)	0	25	22	21
Maximum SOC Allowed (%)	100	84	78	83
Final Drive Ratio	3.63	3.9	4	3.49
Fuel Economy (MPG)	35.1	39.64	40.37	36.6

It can be seen that all three optimization algorithms improved the fuel economy when compared to the original configuration, where simulated annealing produced the best solution. However, since the control strategy was not report, further investigation is required to fully understand the behaviour of the vehicle.

2.3.1.3 Fuel Cell Hybrid Electric Vehicle

A fuel cell hybrid electric vehicle powertrain sizing optimization was performed by Hegazy and van Mierlo[19], where they sought to balance the sizing of the powertrain from the cost perspective. The fuel cell hybrid powertrain consisted of the fuel cell pack to provide power during steady state operation while utilizing an ultracapacitor pack for transient and instantaneous peak power demand. The fuel cell and the ultracapacitor systems were connected via a set of DC/DC converters, and the powertrain model was created in MATLAB/Simulink. Two drive cycles were considered for vehicle power calculation: Federal Test Procedure (FTP75) and New European Driving Cycle (NEDC) [19].

The goal of the optimization problem is to minimize the cost of the fuel cell and the ultracapacitors. The objective function *J(x)* is defined as:

$$J(x) = C1 \times Nfcs \times Nfcp + C2 \times Nscs \times Nscp \tag{2.11}$$

where *C1* and *C2* are the unit cost of the fuel cell and ultracapacitor. *Nfcs* and *Nfcp* respectively denote the number of fuel cell in series and parallel, while *Nscs* and *Nscp* are respectively the number of ultracapacitor in series and parallel. Three methods were utilized to achieve the optimal sizing: trial and error, genetic algorithm, and particle swarm optimization. It was found that in both the NEDC and FTP75 drive cycle, both optimization methods produced results better than the trial and error, while the results of particle swarm optimization was slightly better than those obtained by the genetic algorithm. It was found with that the fuel cell hybrid electric vehicle improved the hydrogen consumption when compared to a fuel cell vehicle without the ultracapacitor by 9.22% on the NEDC and 13.29% on the FTP75 cycle. Furthermore, the total cost reduction on the fuel cell and ultracapacitor components were around 13.4% and the NEDC and 12.21% on the FTP75 drive cycle [19].

2.3.2 Concurrent Optimization

2.3.2.1 Parallel Hybrid Electric Vehicle

The challenge of concurrent optimization for powertrain components and the control system parameters is due to the large amount of coupled design parameters, conflicting design objectives, and nonlinear constraints. One effective strategy to solve such a problem was to utilize multi-objective genetic algorithms to find the Pareto-optimal solution proposed by Fang and Qin [20]. The aim of their work was to optimize the parameters of powertrain components and the control system of a parallel hybrid electric vehicle to improve fuel economy and reduce emissions (CO, HC, and NO_x). The design optimization problem is defined as follows.

$$\begin{cases} \min\limits_{X\in\Omega} J(X) = [Fuel(X), CO(X), HC(X), NO_x(X)] \\ \qquad\qquad s.t.\ \ g_j(X) > 0 \ \ j = 1,2..n \end{cases} \tag{2.12}$$

where X is the variable vector which includes the parameters of powertrain components and control system, and Ω the feasible solution space, governed by constraints $g_j, j=1,2,..n$. The vehicle performance constraints imposed on the design problem were taken from those set out by the U.S. Consortium for Automotive Research for the PNGV (The Partnership for a New Generation of Vehicles). The optimization variables are summarized in Table 2-7 [20].

Table 2-7: Optimization Variables for Multi-Objective Genetic Algorithm [20]

Variables	Description
P_{ICE}	Peak power of ICE
P_{EM}	Peak power of electric motor
N_{bat}	Number of the battery cells
Fd	Final reduction ratio
H_{SOC}	Highest desired battery SOC
L_{SOC}	Lowest desired battery SOC
V_L	Vehicle speed threshold for ICE to turn off
F_{off}	The minimum torque fraction of ICE turn-off
T_{chg}	The minimum torque for battery recharge
F_{min}	Torque fraction for battery recharge

Genetic algorithm was utilized to optimize the parameters in Table 2-7 using a population size of 200, maximum number of 2000 generations, crossover probability of 0.9, and a mutation probability of 0.01. Finally, eight sets of Pareto-optimal solution were found by the optimizer, and simulated using ADVISOR with its default parallel hybrid electric vehicle model to obtain the fuel consumption and the emissions. Final results indicated a reduction of fuel consumption and emission, demonstrating the effectiveness of genetic algorithm to perform concurrent optimization on a parallel hybrid powertrain parameters and its control strategy. However, the drawback of such approach was that for a series hybrid powertrain, the optimizer will seek the minimal battery size to achieve the design constraints without activating the IC engine to avoid any fuel consumption. Such approach may not be feasible to strike a balance between the cost of batteries and fuel consumption from a financial cost perspective.

2.3.2.2 Series Hybrid Electric Vehicle

Similar to the concurrent optimization of parallel hybrid electric vehicle, Zhang *et al.* [21] optimized a series hybrid electric vehicle using multi-objective genetic algorithm. The optimization seeks to minimize the fuel consumption and vehicle emission by optimizing the powertrain sizing and the power management logic. The optimization procedure and vehicle simulation was again performed using ADVISOR. The optimization is defined as:

$$\begin{cases} \min_{X \in \Omega} J(X) = 0.7Fuel(X) + 0.1CO(X) + 0.1HC(X) + 0.1NO_x(X) \\ \qquad\qquad g_j(X) \geq 0,\ \ j = 1,2,..,J \\ s.t. \quad h_k(X) = 0,\ \ k = 1,2,..,L \\ \qquad\qquad x_i^{(l)} \leq x_i \leq x_i^{(u)},\ \ i = 1,2,..,n \end{cases} \tag{2.13}$$

where *J(X)* is the multi-objective function, $g_j(X) \geq 0$, $h_k(X) = 0$ is a group of constraints, and the design variables x_i bounded within a lower $x_i^{(l)}$and upper $x_i^{(u)}$ limit. The power management logic utilizes a thermostat control strategy utilizing the generator and the IC engine to generate electrical energy for the traction motor. The control strategy is described as follows [21].

- To maintain charge in the battery, the engine turns on when the state of charge reaches the low limit
- The engine turns off when the SOC reaches the high limit
- The engine operates at the most efficient speed and torque level

The desired drive cycle composed of one highway (HWFET) and one city (UDDS) drive cycles. The optimization algorithm was implemented in ADVISOR with an initial population of 40, and a terminating condition of 80 generations. Simulation was performed on a 3.4GHz Pentium computer, and took about 4 days for the program to complete. The design variables and the optimized results are shown in Table 2-8 [21].

Table 2-8: Design Variables and the Results of SHEV Optimization [21]

Description	Default Value	Lower Bound	Upper Bound	Optimized Results
Engine Power	41kW	25kW	53kW	25.1kW
Motor Power	75kW	38kW	112kW	80.9kW
Battery Capacity	26Ah	13Ah	39Ah	38Ah
Highest SOC	80%	70%	85%	74%
Lower SOC	60%	30%	50%	49%
Max Power Command	30kW	25kW	40kW	33kW
Min Power Command	20kW	5kW	20kW	5.6kW
Fuel Converter off Duration	inf	10s	1000s	519s

Table 2-8 showed the optimizer determined a minimum engine while indicating a battery size almost at its maximum value. This is not surprising since the most fuel efficient configuration is for a vehicle to operate in pure electric mode throughout the drive cycle. The key deciding factor of the powertrain sizing thus becomes determining the upper and the lower limit of the power components. Again, such approach may not be feasible to ensure a balance between the cost of batteries and fuel consumption from a financial cost perspective.

2.4 Multidisciplinary Design Optimization

Large engineering systems are usually complex and contain different components from various disciplines. Optimal design of such complex systems usually requires various engineering teams specializing in different disciplines collaborating to provide a solution. In order to increase the efficiency of the design process, researchers in the past two decades have proposed different optimization methods in attempt to solve various disciplines simultaneously, termed Multidisciplinary Design Optimization (MDO). The proposed multidisciplinary design optimization have been applied to various complex engineering systems, such as spacecraft launch vehicles [22], air launch rockets [23], underwater autonomous vehicles [24], rail car suspension and dynamics control [25,26], and automotive crash and Noise, Vibration, and Harshness (NVH) design [27].

Multidisciplinary design optimization methods are classified into single-level and multilevel methods. Single-level methods have a single optimizer and have a non-hierarchical structure. Multidisciplinary Feasible (MDF) and Individual Discipline Feasible (IDF) are two examples for the MDO approaches classified as single-level methods. On the other hand, multilevel methods contain hierarchical structure, and each level has an optimizer. Collaborative Optimization (CO), Concurrent

Subspace Optimization (CSSO), Bi-Level Integrated System Synthesis (BLISS), and Integrated System of System Synthesis (ISSS) are examples of multi-level MDO methods [28]. Since hybrid electric vehicle powertrain system is considered a single-level system, the following subsections provide a brief overview of the aforementioned single-level MDO techniques.

2.4.1 Multidisciplinary Feasible Method

Multidisciplinary feasible (MDF) method [29], also referred to as All-in-One method, is the most common methodology for solving MDO problems. In such formulation, the vector of design variables X_D is provided by the optimizer to the coupled system of analysis disciplines, and a complete multidisciplinary analysis (MDA) is performed at that value of X_D to obtain the system output variable $U(X_D)$. The system output variable $U(X_D)$ is then subsequently used to evaluate the objective function $J(X_D, U(X_D))$ and the constraints $C(X_D, U(X_D))$. The MDF formulation is as follows.

$$\text{Minimize } J(X_D, U(X_D)) \text{ w.r.t. } X_D$$
$$\text{s.t. } C(X_D, U(X_D))$$

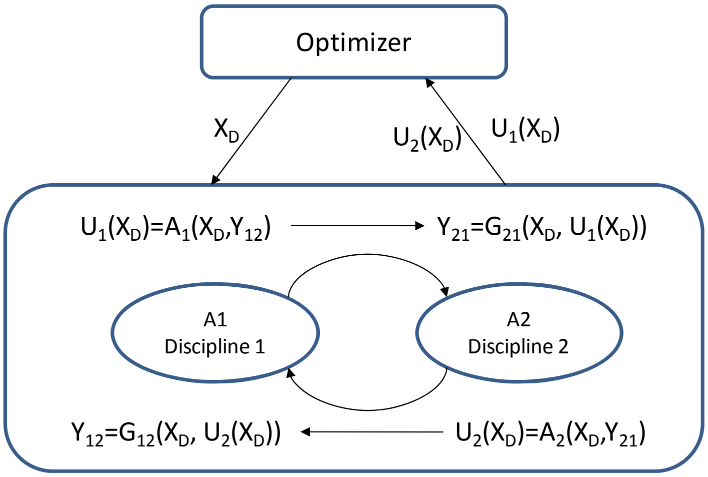


Figure 2-3: Multidisciplinary Feasible (MDF) Formulation

Figure 2-3 illustrates the data flow in an MDF formulation. The system consists of an optimizer that determines the objective function J with respect to the design variable vector X_D, using the system output $U(X_D)$. For each of the optimization loops, the design variable vector X_D is fixed, while analysis A1 and analysis A2 are performed to obtain the system output variable vectors $U_1(X_D)$ and $U_2(X_D)$. The output variable vectors $U_1(X_D)$ and $U_2(X_D)$ are then subsequently returned to the optimizer for evaluating the objective functions $J(X_D, U_1(X_D), U_2(X_D))$ and the constraints $C(X_D, U_1(X_D), U_2(X_D))$. G_{ij} represents the interdisciplinary mapping from the output variable vector U_j of discipline j to a suitable input variable Y_{ij} for use by discipline i. If a gradient approach optimization algorithm is to be used to solve the above problem, then a complete MDA is necessary not just at each iteration, but at every point where derivatives are to be evaluated. Therefore, this can be computationally expensive for realistic applications [30].

2.4.2 Individual Discipline Feasible Method

In order to avoid a complete multidisciplinary analysis every time an objective function, constraint, or sensitivity evaluation is required, Individual Discipline Feasible (IDF) method can be used. The IDF approach maintains the feasibility of the individual disciplines, while allowing the optimizer to obtain the multidisciplinary feasibility and optimality by controlling the interdisciplinary coupling variables. In such approach, the specific analysis variables representing the coupling, or the mappings between analysis disciplines, are "promoted" to become optimization variables. From the point of view of a single analysis discipline solver, these optimization variables are indistinguishable. The general IDF formulation can be written as follows:

$$\begin{gathered}\text{Minimize } F(X_D, U(X)) \text{ w.r.t. } X=(X_D, X_Y)\\ \text{s.t. } C(X_D, U(X)),\ C_{aux} \triangleq X_Y - G(X_D, U(X))=0\\ U(X)=A(X)\end{gathered}$$

X_D is defined as the design variable vector of the optimizer, X_Y denotes the "promoted" optimization design variable vector from the input variable vector Y for an analysis discipline, $U(X)$ is the system output variable vector and $A(X)$ is the analysis mapping from the inputs X_D and X_Y. $F(X_D, U(X))$ and $C(X_D, U(X))$ represent, respectively, the objective function and the constraints of the system. Also, G represents the interdisciplinary mappings and the condition $C_{aux} \triangleq X_Y - G(X_D, U(X))=0$ converts the interdisciplinary mappings into auxiliary optimization constraints. In order to evaluate $U(X)=A(X)$, all the single discipline analysis codes can be executed simultaneously with the available multidisciplinary design variable vector X. Therefore, these computations can be performed independently and concurrently, thus minimizing communication cost. Hence, the IDF method is ideal for applications utilizing a parallel processing computing system [30].

Chapter 3
Hybrid Electric Vehicle Modeling

A generic, modular, and flexible vehicle model is created for the purpose of performing vehicle system optimization. In particular, the powertrain components need to be scalable for the optimizer to determine the optimal sizes of the components. The following sections discuss the vehicle system and the component modeling in detail.

3.1 Generic Vehicle Structure

There are currently two modeling approaches to perform vehicle simulations: (i) forward looking and (ii) backward looking. In forward-looking type of modeling, simulation begins from the driver's point of view, where a power demand from the driver is sent to the powertrain components, and the resulting power available from the powertrain is subsequently fed to the wheels of the vehicle. The advantage of such architecture is that it is more realistic, since it mimics the actual driving of a human in the real world. In addition, hardware-in-the-loop can be easily implemented to enhance the engineering and development of a vehicle system. However, the drawback is that a high fidelity model is required for each of the components, and the cost of computation can be high. On the other hand, backward-looking modeling begins with determining the required vehicle power using a known drive cycle. It then issues power demands for the powertrain components, where the actual power consumed is subsequently calculated while taking into consideration the power components' efficiencies. The benefit of such approach is that it greatly reduces the computational time during simulation, and that simplified quasi-static models can be used. The drawback, however, is in that the dynamics of the components and the vehicle system is not considered, thereby making the approach not as realistic as the forward-looking models.

In order to create a generic vehicle model to be used for optimization, a vehicle model covering all possible hybrid electric vehicle configurations is created. Figures 3-1 and 3-2 illustrate, respectively, a generic vehicle structure in forward- and backward-looking simulation architectures.

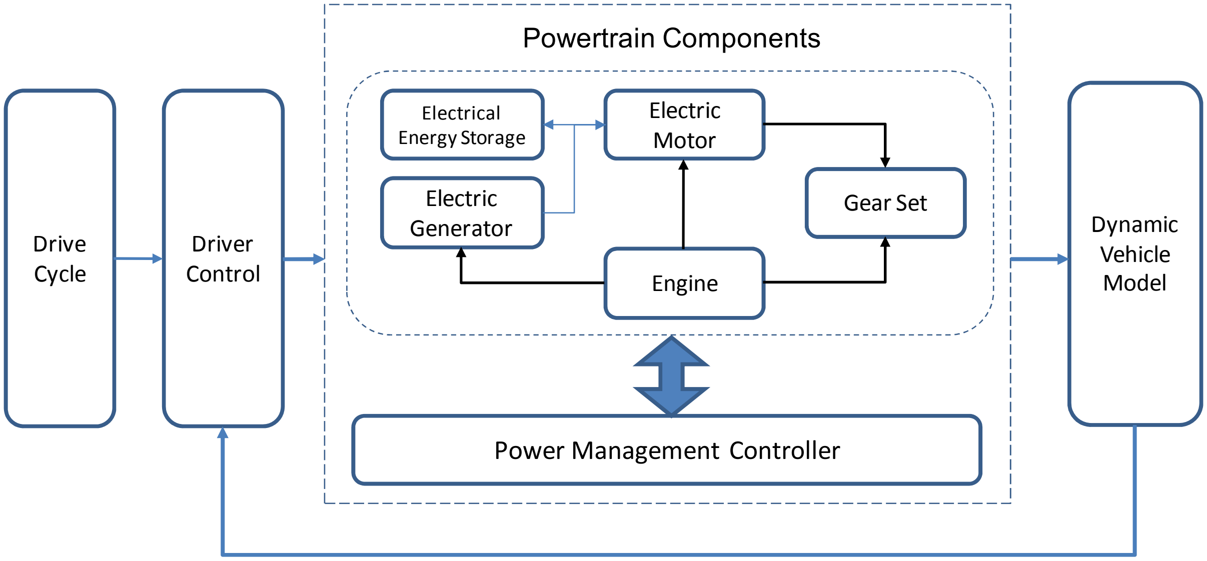


Figure 3-1: A Generic Vehicle Model in a Forward-Looking Simulation Architecture

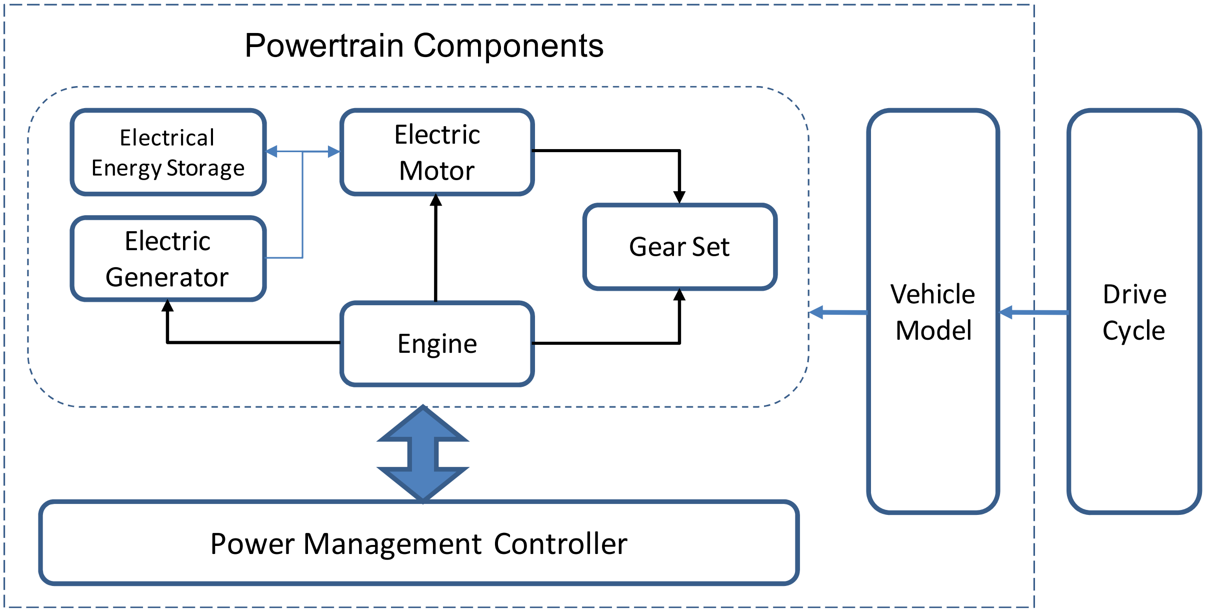


Figure 3-2: A Generic Vehicle Model in a Backward-Looking Simulation Architecture

The generic vehicle structure consists of all possible components of a hybrid electric vehicle powertrain, where any desired hybrid electric vehicle configurations can be created. For example, by removing the connections between the engine, the gear set, and the electric motor, a series hybrid structure can be realized, as shown in Figure 3-3.

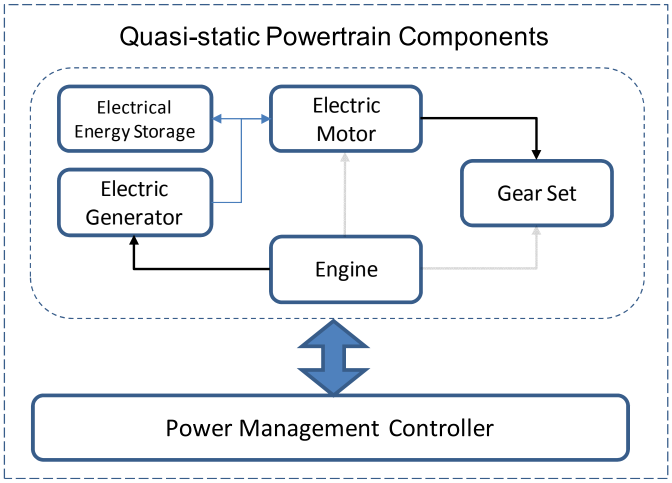

Quasi-static Powertrain Components
Electrical Energy Storage
Electric Motor
Electric Generator
Gear Set
Engine
Power Management Controller

Figure 3-3: Series Hybrid Electric Vehicle Model Created from the Generic Vehicle Structure

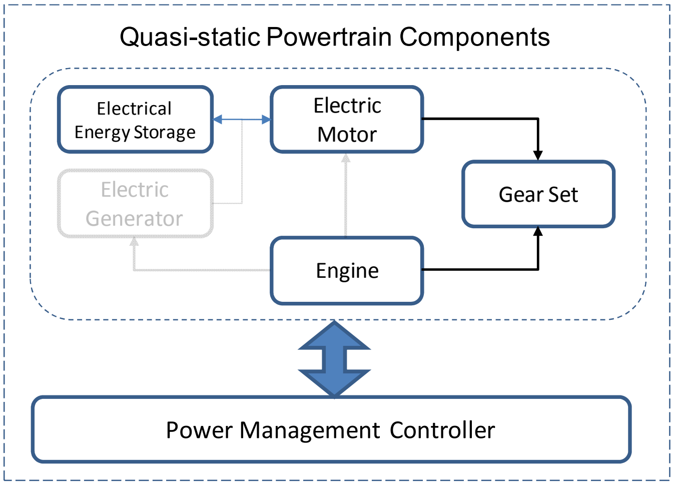

Quasi-static Powertrain Components
Electrical Energy Storage
Electric Motor
Electric Generator
Gear Set
Engine
Power Management Controller

Figure 3-4: Parallel Hybrid Electric Vehicle Model Created from the Generic Vehicle Structure

Similarly, Figure 3-4 shows a parallel hybrid electric vehicle configuration, where the electric generator and the connection between the engine and the electric motor were deactivated. Another type of hybrid electric vehicle currently available in the market is the Honda Integrated Motor Assist (IMA) architecture, where the electric motor is mounted directly between the engine and the transmission. In such a system, the engine is the prime power source to the vehicle, and the electric motor act as an assisting power source, where it supplements the engine power when necessary. Figure 3-5 depicts the Honda IMA configuration using the generic vehicle model.

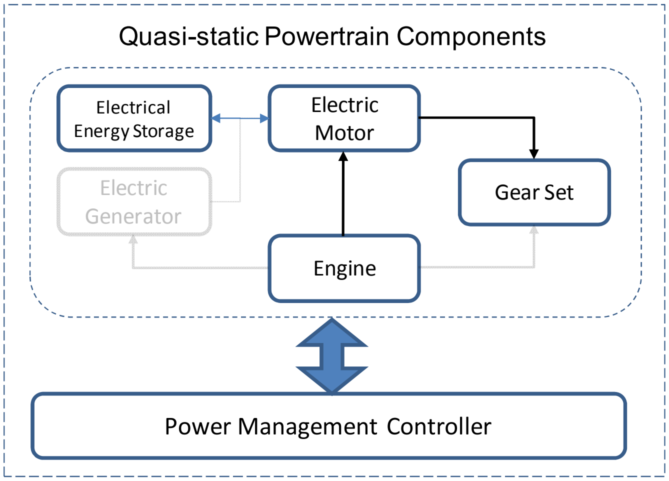

Quasi-static Powertrain Components
Electrical Energy Storage
Electric Motor
Electric Generator
Gear Set
Engine
Power Management Controller

Figure 3-5: Honda IMA Vehicle Model Created from the Generic Vehicle Structure

As shown above, the generic vehicle structure can be used to simulate all possible types of hybrid electric vehicle system as required by the user.

3.1.1 Forward-Looking Vehicle Model

In the forward-looking simulation platform, the simplest vehicle model that can be utilized is a particle vehicle model, where only the drive torque, the resistive aerodynamic and rolling resistance forces are considered. Since only the overall power consumption of the vehicle is of interest, the effects of vehicle dynamics due to the suspension can be safely ignored.

The vehicle model receives the drive torque from the powertrain components and outputs the vehicle speed and wheels. The input and output variables of the vehicle model are illustrated in Figure 3-6 and the free-body diagram of the vehicle is depicted in Figure 3-7.

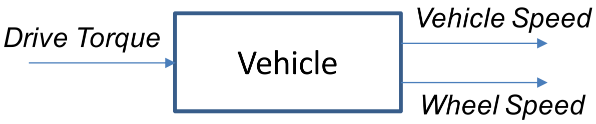


Figure 3-6: Input and Output Variables of the Particle Vehicle Model

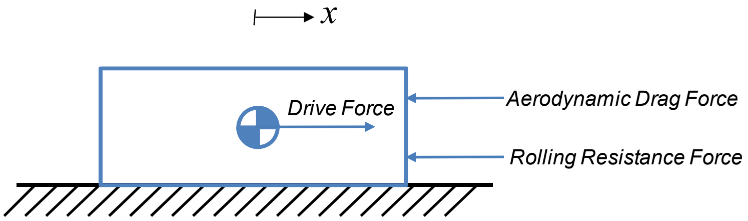


Figure 3-7: Free Body Diagram of the Particle Vehicle Model

The summation of forces acting on the vehicle body with a given mass (m_{veh}) is illustrated by

$$\sum F_x(t) = m_{veh}\ddot{x}(t) = F_{drive}(t) - F_{drag}(t) - F_{RR} \tag{3.1}$$

where $F_{drive}(t)$, $F_{drag}(t)$, and F_{RR} represent the drive force, aerodynamics drag force, and tire rolling resistance force respectively. By integrating the acceleration, the vehicle speed can be obtained. In addition, assuming constant tire radius (r_t), the wheel speed can be calculated. The drive force of the vehicle is obtained using

$$F_{drive}(t) = \frac{T_{drive}(t)}{r_t} \tag{3.2}$$

where $T_{drive}(t)$ is the input drive torque from the powertrain components. The aerodynamics drag force is calculated by

$$F_{drag}(t) = \frac{1}{2}C_D\rho v_x(t)^2 A \tag{3.3}$$

where C_D is the drag coefficient, ρ the air density, $v_x(t)$ the longitudinal speed of the vehicle, and A the frontal area of the vehicle. Finally, the tire rolling resistance force is expressed by

$$F_{RR} = C_{RR}m_{veh}g \tag{3.4}$$

where C_{RR} is the rolling resistance constant of the tire, m_{veh} the vehicle mass, and g gravitational acceleration. Using the above-mentioned formulation, a simple vehicle model for power consumption calculation can be utilized.

3.1.2 Backward-Looking Vehicle Model

The basic components of the backward-looking vehicle model are similar to that of the forward-looking particle vehicle model, where the drag force and rolling resistance components remain the same. However, the backward-looking vehicle model differs from the particle model from the viewpoints of the input and output variables to the model, as depicted in Figure 3-8.

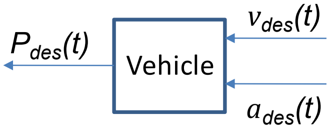


Figure 3-8: Input and Output Variables of the Backward-Looking Vehicle Model

As previously mentioned, in the backward-looking vehicle model the desired drive cycle is known, and is used as the input to the vehicle model in the form of vehicle speed and acceleration. The total desired power ($P_{des}(t)$) to realize the drive cycle and to overcome the aerodynamic drag and tire rolling resistance force is given by

$$P_{des}(t) = \left[F_{drive}(t) + F_{drag}(t) + F_{RR}\right]v_{des}(t) \tag{3.5}$$

where $v_{des}(t)$ is the vehicle speed form the drive cycle, $F_{drive}(t)$ the drive force of the vehicle, $F_{drag}(t)$ the aerodynamics drag force, and F_{RR} the rolling resistance force of the tires. The drive force of the vehicle is given by

$$F_{drive}(t) = m_{veh}a_{des}(t) \tag{3.6}$$

where m_{veh} is the vehicle mass and $a_{des}(t)$ is the acceleration defined by the drive cycle. $F_{drag}(t)$ and F_{RR} are defined by equations (3.3) and (3.4), respectively. For the purpose of optimization, it is desire to use a backward-looking model to reduce the computational efforts; thus, a simple vehicle model is created to perform the aforementioned power consumption calculations using the above-mentioned formulation.

Due to the fact that the backward-looking vehicle model looks at the vehicle performance only from the power consumption point of view, it lacks the information on the dynamics of the vehicle system. Specifically, it lacks the capability of calculating whether the vehicle is capable of completing the drive cycle, which is critical in determining the final configuration of the vehicle model. One method is to look at the additional amount of time that the vehicle needs to complete the drive cycle if the powertrain is not powerful enough to achieve the desired speed and acceleration. Consider the average speed ($v_{avg,d}$) of the vehicle when a delay in completion of drive cycle occurs as described by the following equation.

$$v_{avg,d} = \frac{x_{tot,d}}{t_{tot,d}} = \frac{x_{tot,d}}{T + \Delta t} \tag{3.7}$$

where $x_{tot,d}$ and $t_{tot,d}$ denote the total distance traveled and the total time the vehicle needs to complete the drive cycle. T is the total time of the drive cycle with no delay and Δt is the delay. Rearranging Equation (3.7) yields

$$\Delta t = \frac{x_{tot,d}}{v_{avg,d}} - T \tag{3.8}$$

The average speed ($v_{avg,d}$) can be calculated by averaging the actual speed ($v_{act}(t)$) of the vehicle during simulation time, where v_{act} can be determined using the following equation.

$$\Delta v(t) = (v_{des}(t) - v_{act}(t)) = \frac{(P_{des}(t) - P_{act}(t))}{F_{tot}(t)} \tag{3.9}$$

where $v_{des}(t)$ is the desired speed given by the drive cycle, $P_{des}(t)$ is the desired power from Equation (3.5), $F_{tot}(t)$ is the algebraic sum of the vehicle's driving and resisting forces, and $P_{act}(t)$ is the total available power from the battery and genset. Rearranging Equation (3.9), $v_{act}(t)$ can be calculated as follows.

$$v_{act}(t) = v_{des}(t) - \frac{(P_{des}(t) - P_{act}(t))}{F_{tot}(t)} \tag{3.10}$$

By averaging the time history of $v_{act}(t)$, the average speed can be found. Substituting $v_{avg,d}$ into Equation (3.8), the additional time required to complete the drive cycle can be easily determined. It can also be used to check whether or not the vehicle powertrain is capable of completing the drive cycle during optimization.

3.2 Component Descriptions

The powertrain components are modeled using a scalable quasi-static backwards approach as proposed by Guzzella *et al.* [31,32,33,34], where the actual consumed power of the energy converters (i.e., engine and electric motor) is calculated by the required component torque at its current velocity state, while taking into account the components' efficiencies. Further details will be provided in the subsequent subsections.

3.2.1 Engine

Willans line modeling approach for powertrain components which was first proposed by Rizzoni *et al.* [31], and further developed by Guzzella *et al.* [32,33,34], describes a scalable quasi-static engine model that calculates the engine efficiency of converting fuel energy to output power as a function of the internal combustion engine's property, given by the following function:

$$\eta_e(t) = \frac{T_e(t)\omega_e(t)}{P_{fuel}(t)} \tag{3.11}$$

where $\eta_e(t)$ is the efficiency of the engine, $P_{fuel}(t)$ the enthalpy flow associated with the fuel mass flow, and $T_e(t)$ and $\omega_e(t)$ denote the engine torque and speed, respectively. The fuel mass flow is calculated by:

$$\dot{m}_f(t) = \frac{P_{fuel}(t)}{H_l} \tag{3.12}$$

where H_l is the fuel's lower heating value. The input and output variables to the quasi-static engine model are illustrated in Figure 3-9.

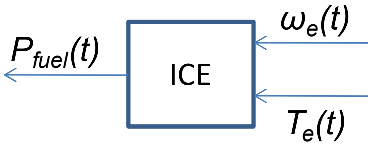


Figure 3-9: Input and Output Variables of the Internal Combustion Engine

The key idea in developing a scalable model of a combustion engine is to use the concept of *mean effective pressure* ($p_{me}(t)$) to describe the engine's ability to produce mechanical work, and to use its *mean piston speed* ($c_m(t)$) to describe its operating speed. [32] When the engine is running in steady-state condition, $p_{me}(t)$ and $c_m(t)$ describes its operating point in the following equations:

$$p_{me}(t) = \frac{N\pi}{V_d} T_e(t) \tag{3.13}$$

$$c_m(t) = \frac{S}{\pi}\omega_e(t) \tag{3.14}$$

where S is the engine's stroke and V_d its displacement. The parameter N depends on the engine type: N=4 for a four-stroke engine and N=2 for a two-stroke engine [32]. Considering the thermodynamic efficiency and the internal losses during the engine cycle, and introducing $p_{mf}(t)$ as the fuel mean effective pressure, the following relationships can be established:

$$\eta_e(t) = \frac{p_{me}(t)}{p_{mf}(t)} \tag{3.15}$$

$$p_{me}(t) = ep_{mf}(t) - p_{loss}(t) \tag{3.16}$$

where *e* represents the thermodynamic properties of the engine related to the indicated mean effective pressure [32]. $p_{loss}(t)$ represents the engine's losses due to gas exchange (p_{loss_g}) and friction ($p_{loss\_f}(t)$) and is illustrated as follows:

$$p_{loss}(t) = p_{loss\_g} + p_{loss\_f}(t) \tag{3.17}$$

Using the *mean piston speed* and experimental results, $p_{loss\_f}(t)$ is defined as:

$$p_{loss\_f}(t) = k_1(k_2 + k_3 S^2 \omega_e(t)^2)\Pi_{max}\sqrt{\frac{k_4}{B}} \tag{3.18}$$

where Π_{max} is the maximum boost pressure and *B* denotes the engine cylinder bore. The *k* parameters are experimentally determined and are listed in Appendix A [32]. By combining Equations (3.16), (3.17), and (3.18), and substituting $p_{me}(t)$ in Equation (3.15), $\eta_e(t)$ can be found. Subsequently, the fuel consumption can be calculated for a given engine output torque and speed using Equations (3.11) and (3.12). Detailed derivations can be found in references [31,32,33,34]. By utilizing of the Willans line model, a scalable engine model can be used for size optimization.

3.2.2 Electric Motor-Generator

The roles of the electric motor-generator besides acting as a tractive motor to provide motor torque during acceleration, is to perform regenerative braking to capture the otherwise lost kinetic energy of the vehicle. The input and output variables of the electric motor-generator are depicted in Figure 3-10.

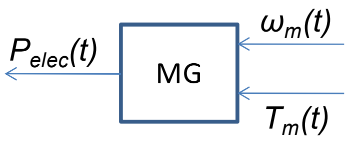


Figure 3-10: Input and Output Variables of the Electric Motor-Generator

$T_m(t)$ and *$\omega_m(t)$* represent, respectively, the output torque and speed of the motor, and *$P_{elec}(t)$* denotes the electrical power consumed or generated by the electric motor-generator. As a sign convention, positive torque and power represent the unit acting in the motor mode, while negative values denote the unit operating in the generator mode.

Guzzella *et al.* [33] at the Swiss Federal Institute of Technology Zurich have developed a Quasi-static (QSS) Toolbox that contained a scalable electric motor-generator using experimentally obtained efficiency map for a generic electric motor-generator. The efficiency map is modeled as a look-up table indexed by motor-generator's torque (*$T_m(t)$*) and speed (*$\omega_m(t)$*), which is provided in Appendix A. During operation in motor or generator mode, the electric power (*$P_{elec}(t)$*) consumed or generated is given by the following equation.

$$P_{elec}(t) = \eta_m(t) T_m(t) \omega_m(t) \tag{3.19}$$

3.2.3 Generator

Using the same modeling approach as the electric motor-generator, the same model can be used solely as a generator, which is coupled to the IC engine to generate electrical power for the battery or the tractive motor. Similar to the electric motor-generator, the size of generator can be scaled appropriately as necessary.

3.2.4 Gear Set

The purpose of the transmission is to perform torque multiplication from the power source to the wheels, while maintaining the operating range of the power source within its maximum torque range. For this reason, the number of gears of the transmission is dependent on the primary power source. For example, the torque of an IC engine typically peaks around the middle of its speed range; therefore, a common transmission for conventional vehicle contains 5 or 6 gears to ensure the engine operates within its maximum torque band as frequently as possible. On the other hand, the maximum torque band of an electric motor is usually in the first half of its speed range; therefore, a transmission with 2 or 3 gears will satisfy the operation requirement. Based on the aforementioned reason, the exact type of gear set will be dependent on the hybrid architecture utilized.

3.2.5 Electrical Energy Storage

The electrical energy storage (EES) system of the vehicle stores the electric energy in batteries and/or ultracapacitors. Depending on the desired configuration, the EES may consist of only the battery, the ultracapacitor, or a combination of both. In addition to providing electric energy to the traction motor, the EES also stores the energy captured during regenerative braking. The following subsections describe the modeling details of the battery and ultracapacitor.

3.2.5.1 Battery

The battery is an electrochemical device that stores electrical energy in the form of chemical energy. Many types of batteries that are used for providing traction energy for electric vehicles; some examples are lead-acid, nickel-metal-hydride, and lithium-ion batteries. For this research, a simple battery model is utilized where the current is calculated while taking into consideration the internal resistance when calculating the charge and discharge power of the battery. Figure 3-11 shows the electrical circuit diagram of the battery model.

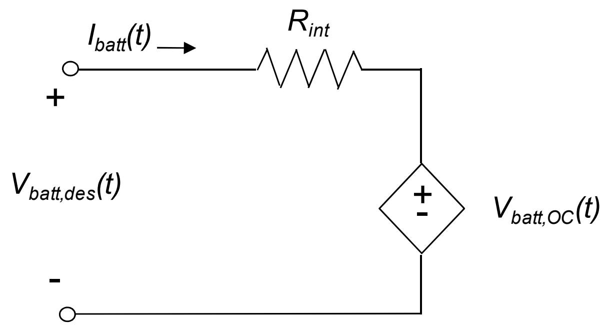


Figure 3-11: Electrical Circuit Diagram of the Battery Model

The desired battery power, $P_{batt,des}(t)$, can be expressed as:

$$P_{batt,des}(t) = I_{batt}^2(t)R_{int} + V_{batt,oc}(t)I_{batt}(t) \tag{3.20}$$

where R_{int} and $V_{batt,oc}(t)$ are the internal resistance and the open circuit voltage of the battery, respectively. As a sign convention, let positive power and current denote charging while negative power and current denote discharging of the battery. Note that Equation (3.20) is valid for both charging and discharging modes even though the directions and signs of battery power and current will be different in these two cases. Solving the quadratic Equation (3.20), the current can be determined by

$$I_{batt}(t) = \frac{-V_{batt,oc}(t) \pm \sqrt{V_{batt,oc}^2(t) + 4P_{batt,des}(t)R_{int}}}{2R_{int}} \tag{3.21}$$

Solving Equation (3.21) yields two values. To determine which one of the two values is the correct solution, one can assume that when the internal resistance is zero, the ideal current can be determined by

$$I_{ideal}(t) = \frac{P_{des}(t)}{V_{oc}(t)} \tag{3.22}$$

where $P_{des}(t)$ and $V_{oc}(t)$ are the desired power and the open circuit voltage of the electric energy storage system, respectively. Upon inspection of the solutions from Equation (3.21), it is found that the larger value of the two answers is higher than the ideal current as determined by Equation (3.22), which is not acceptable. Therefore, the solution from Equation (3.21) with the smaller absolute value will be chosen as the answer. In addition, the current will be limited by the charge and discharge current limits specified by the battery manufacturer. The open circuit voltage ($V_{batt,oc}(t)$) as function of the battery SOC is modeled using a look-up table, which can also be obtained from the manufacturer. Finally, once the current and the open circuit voltage are known the actual charge and discharge power at the battery can be found by:

$$P_{batt,act}(t) = V_{batt,oc}(t) I_{batt}(t) \tag{3.23}$$

In order to update the SOC of the battery, the actual charge and discharge power of the battery $P_{batt,act}(t)$, is integrated to obtain change in energy, and is added to the existing energy level of the battery. The capacity of the battery is determined by taking the nominal capacity of each of the battery cell, as specified by the manufacturer, and multiplying it by the number of cells in the battery system. The size of the battery system can therefore be scaled by changing the number of cells.

3.2.5.2 Ultra-capacitor

An ultracapacitor, also known as a supercapacitor or double-layer capacitor, is an electrochemical capacitor with relatively high energy density. It has a much higher power density than that of a battery; therefore, it is advantageous to utilize the ultracapacitor during hard acceleration and braking period.

Using a modeling approach similar to that used for the battery, the terminal voltage and the current of the ultracapacitor are calculated while taking into account the internal resistance in order to calculate the charge and discharge power of the ultracapacitor. Figure 3-12 illustrates the electrical circuit diagram of the ultracapacitor model.

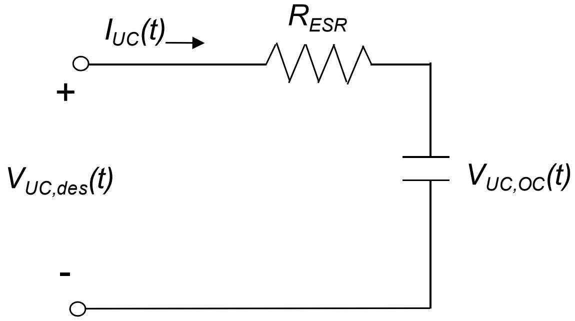


Figure 3-12: Electrical Circuit Diagram of the Ultracapacitor Model

The desired ultracapacitor power, *$P_{UC,des}(t)$* can be expressed as:

$$P_{UC,des}(t) = I_{UC}^2(t)R_{ESR} + V_{UC,oc}(t)I_{UC}(t) \tag{3.24}$$

where *R_{ESR}* and *$V_{UC,oc}(t)$* are the equivalent series resistance and the open circuit voltage of the ultracapacitor, respectively. For an ultracapacitor, the open circuit voltage is simply

$$V_{UC,oc}(t) = \frac{Q_{UC}(t)}{C_{UC}} \tag{3.25}$$

where *$Q_{UC}(t)$* and *C_{UC}* denote the charge and capacitance of the ultracapacitor, respectively. Again, as a sign convention, let positive power and current denote charging while negative power and current denote discharging of the ultracapacitor. Solving the quadratic Equation (3.24), the current can be determined by

$$I_{UC}(t) = \frac{-V_{UC,oc}(t) \pm \sqrt{V_{UC,oc}^2(t) + 4P_{UC,des}(t)R_{ESR}}}{2R_{ESR}} \tag{3.26}$$

Solving Equation (3.26) yields two values. Upon inspection of the solutions from Equation (3.26), it is found that the larger value of the two answers is higher than the ideal current as depicted by Equation (3.22), which is not acceptable. Therefore, the solution from Equation (3.26) with the smaller absolute value will be chosen as the answer. Additionally, the current will be limited by the charge and discharge current limits specified by the ultracapacitor manufacturer. Once the charging or discharging current is calculated, it is then integrated to update the charge status (*$Q_{UC}(t)$*) of the ultracapacitor. Finally, the state of charge (*$SOC_{UC}(t)$*) of the ultracapacitor can be calculated by

$$SOC_{UC}(t) = \frac{Q_{UC}(t)}{Q_{UC,nom}} \tag{3.27}$$

where $Q_{UC,nom}$ is the nominal charge of the ultracapacitor, as specified by the manufacturer.

3.3 Power Management of the Electrical Energy Storage System

Due to the difference in characteristics of the battery and ultracapacitor, it is important to utilize their strength while compensating for their weaknesses. For example, ultracapacitor has a much higher power density, and therefore should be used during hard acceleration and regenerative braking periods. On the other hand, the battery has a larger energy storage capacity than that of the ultracapacitor, and therefore should be used for longer cruising and milder regenerative braking durations. However, the design and controls of a combined battery and ultracapacitor system is complex enough to warrant an entirely different field of study. For the purpose of this research, it is sufficient to operate the battery and the ultracapacitor solely from the charging and discharging power transfer point of view.

3.3.1 Power Distributing Function

A simple approach to balance the operation of the battery and the ultracapacitor is to utilize a function that splits the desired electric power between the two components, defined as follows:

$$\begin{aligned} P_{des}(t) &= P_{batt,des}(t) + P_{UC,des}(t) \\ &= \mu(t) \times P_{des}(t) + \big(1 - \mu(t)\big) \times P_{des}(t) \end{aligned} \tag{3.28}$$

$P_{des}(t)$ denotes the combined desired power, while $P_{batt,des}(t)$ and $P_{UC,des}(t)$ represent the desired power of the battery and the ultracapacitor respectively. $\mu(t)$ is the power distributing function of the battery, where a value of 1 represents battery operation only with no power from the ultracapacitor, and 0 signals using no battery power with all the power dedicated to charge or discharge the ultracapacitor.

When defining the power distributing function ($\mu(t)$), it is necessary to monitor the state of charge (SOC) of the components (battery and ultracapacitor) to avoid discharging request when the component is depleted, or charging request when the component is full. Furthermore, since the advantage of the ultracapacitor is the capability to provide high and sudden power request, the power distributing function shall also consider the power desired ($P_{des}(t)$) and the rate of change of the desired power ($dP_{des}(t)/dt$). Utilizing a membership function that is used in fuzzy logic, a sigmoid function can be used as an activating function based on a monitored variable, where the characteristic curve can be modified with ease. A general sigmoid function takes the following form:

$$f(x) = \frac{1}{(1 + e^{a(x-c)})} \tag{3.29}$$

where *f(x)* is the output function based on the input variable *x*. The parameter *a* defines the slope of the curve at the inflection point, while the parameter *c* defines the inflection point. Figure 3-13 shows a general sigmoid function with an inflection point (*c*) of 5 with varying slope (*a*).

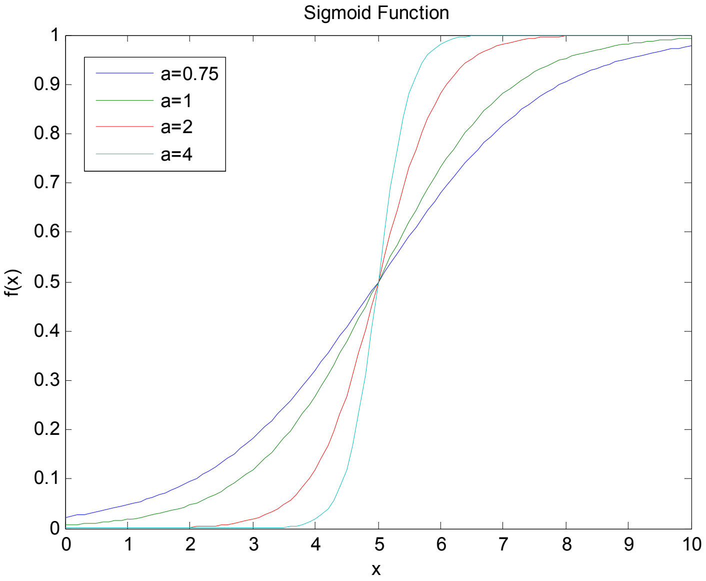


Figure 3-13: A Sigmoid Function with Varying Slopes and Inflection Point of 5

The advantage of utilizing a sigmoid function is that it can easily represent different types of activating function with ease by changing the slope and inflection point. For example, a step function can be created by assigning a large value to the slope of the sigmoid function, thereby creating a continuous function that exhibit the behaviour of a step function. This can be computationally efficient during simulation where the discontinuity of a step function may cause numerical error. On the other hand, by setting the slope to be zero, a constant output function can be created.

As previously mentioned, the power distributing function (*$\mu(t)$*) will be a function of the SOC of the battery (*$SOC_{batt}(t)$*), SOC of the ultracapacitor (*$SOC_{UC}(t)$*), the desired power (*$P_{des}(t)$*), and the rate of change of the desired power (*$dP_{des}(t)/dt$*). Using a sigmoid function for each of the monitored variable, the discharging power distributing function (*$\mu_{dis}(t)$*) is defined as:

$$\mu_{dis}(t) = \frac{1}{\left(1 + \mathrm{e}^{\mathrm{b_{dis}}(\mathrm{SOC_{batt}}(t) - \mathrm{c_{dis}})}\right)} \times \frac{1}{\left(1 + \mathrm{e}^{\mathrm{d_{dis}}(\mathrm{SOC_{UC}}(t) - \mathrm{e_{dis}})}\right)} \times \frac{1}{\left(1 + \mathrm{e}^{\mathrm{f_{dis}}(\mathrm{P_{des}}(t) - \mathrm{g_{dis}})}\right)} \times \frac{1}{\left(1 + \mathrm{e}^{\mathrm{h_{dis}}(\frac{dP_{des}(t)}{dt} - \mathrm{i_{dis}})}\right)} \tag{3.30}$$

with the slopes (b_{dis}, d_{dis}, f_{dis}, and h_{dis}) and inflection points (c_{dis}, e_{dis}, g_{dis}, and i_{dis}) as described in Equation (3.29). It should be noted that the parameter g_{dis} is defined as the discharging power limit of the battery, and therefore its value is dependent on the specification of the component.

The charging power distributing function ($\mu_{ch}(t)$) is similarly defined, with the exception of the desired power, where it is desired that the battery be charged first before the ultracapacitor. Since the power charging limit of the battery is much lower than that of the ultracapacitor, this will ensure maximum possible charge of the battery at any given time. The charging power distributing function ($\mu_{ch}(t)$) is therefore defined as

$$\mu_{ch}(t) = \frac{1}{\left(1 + \mathrm{e}^{\mathrm{b_{ch}}(\mathrm{SOC_{batt}}(t) - \mathrm{c_{ch}})}\right)} \times \frac{1}{\left(1 + \mathrm{e}^{\mathrm{d_{ch}}(\mathrm{SOC_{UC}}(t) - \mathrm{e_{ch}})}\right)} \times \frac{1}{\left(1 + \mathrm{e}^{\mathrm{h_{ch}}(\frac{dP_{des}(t)}{dt} - \mathrm{i_{ch}})}\right)} \times \min\left(\frac{\mathrm{g_{ch}}}{\mathrm{P_{des}}(t)}, 1\right) \tag{3.31}$$

where the slopes (b_{ch}, d_{ch}, and h_{ch}) and inflection points (c_{ch}, e_{ch}, and i_{ch}) are as described in Equation (3.29). It should be noted that the parameter g_{ch} is defined as the charging power limit of the battery, and that the min function ensures that all charging power is first sent to the battery when the charging power ($P_{des}(t)$) is less than the charging power limit of the battery (g_{ch}). Since the battery charge limit is much lower than that of the ultracapacitor, this will ensure the maximum charging of the battery throughout the drive cycle.

3.3.2 Power Distributing Function Illustrations

Due to the multi-dimensional nature of the power distributing function, it is not possible to visually demonstrate the behavior of all the variables simultaneously. Therefore, variables are selected in turn to illustrate visually their relationships with the power distributing function, as presented in the following subsections.

3.3.2.1 Scenario 1: Effects of Battery State of Charge

Figures 3-14 and 3-15 depict the discharge power distributing function (*$\mu_{dis}(t)$*) as a function of the ultracapacitor state of charge (*$SOC_{UC}(t)$*) and the desired power (*$P_{des}(t)$*), while changing the battery SOC (*$SOC_{batt}(t)$*) and holding the rate change of power (*$dP_{des}(t)/dt$*) constant.

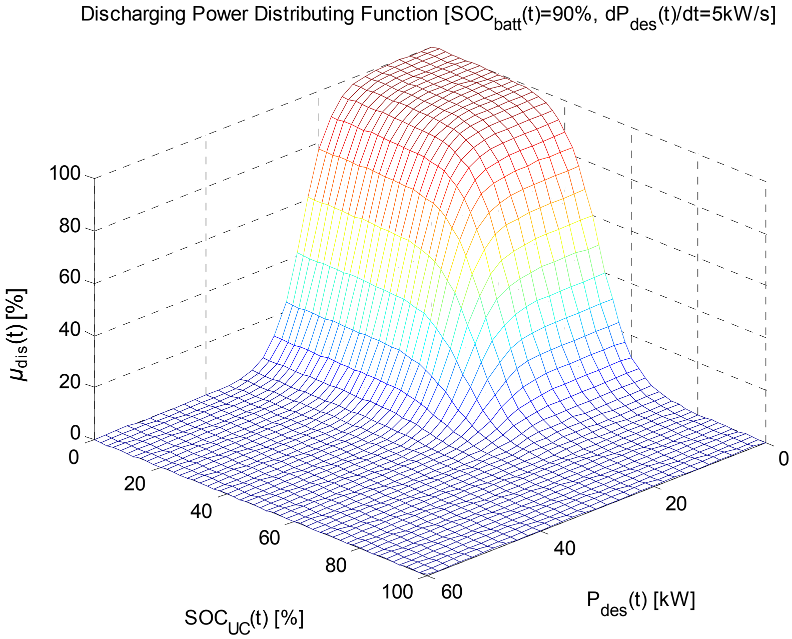


Figure 3-14: Discharging Power Distributing Function [*$SOC_{batt}(t)$*=90%, *$dP_{des}(t)/dt$*=5kW/s]

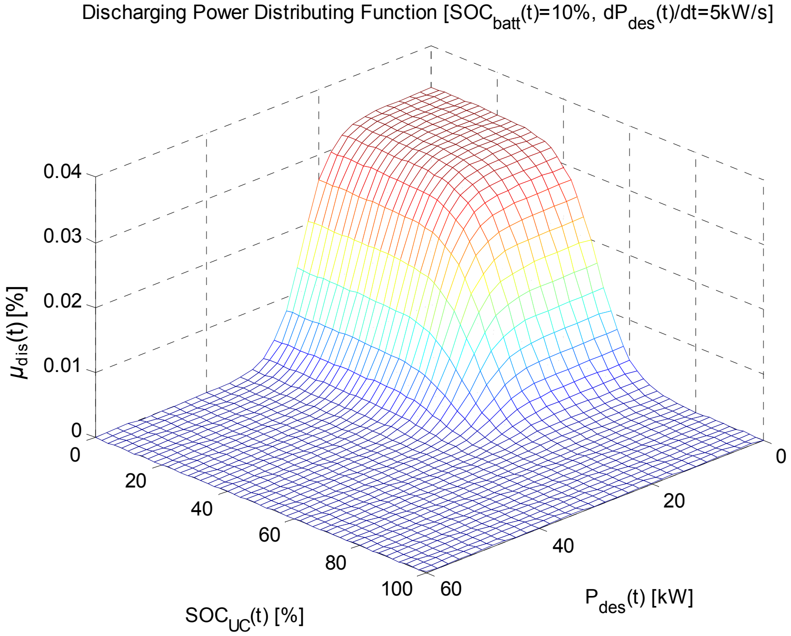


Figure 3-15: Discharging Power Distributing Function [$SOC_{batt}(t)$=10%, $dP_{des}(t)/dt$=5kW/s]

Figure 3-14 indicates a close-to-full battery, while Figure 3-15 represents an almost depleted battery. It can be seen that when the battery if full, depending on the ultracapacitor state of charge and the desired power, it is possible to discharge the battery at 100% share of power. However, when the battery is depleted as shown in Figure 3-15, the discharge power distributing function is close to zero at all times, indicating the discharging power is solely provided by the ultracapacitor. Furthermore, since the inflection point (g_{dis}) of the desired power ($P_{des}(t)$) sigmoid function is the battery discharging power limit, when the desired discharging power is higher than that of the battery limit, power will be primarily discharged from the ultracapacitor regardless of the battery or the ultracapacitor's states of charge. It should be noted that the rate of change of power ($dP_{des}(t)/dt$) is set to a low value in this scenario, and will be studied later in the section. Similarly, the charging power distributing function ($\mu_{ch}(t)$) is shown in Figures 3-16 and 3-17.

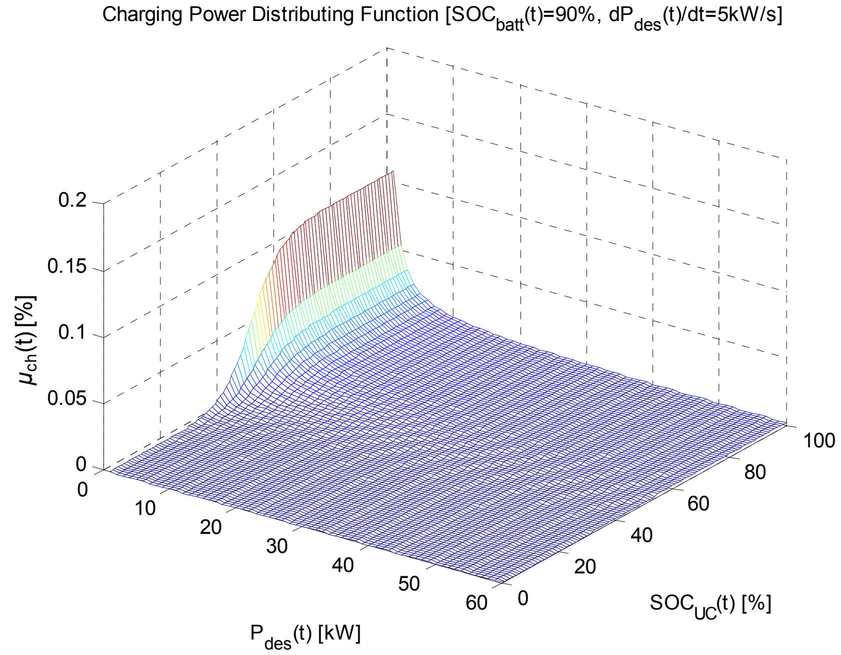


Figure 3-16: Charging Power Distributing Function [$SOC_{batt}(t)$=90%, $dP_{des}(t)/dt$=5kW/s]

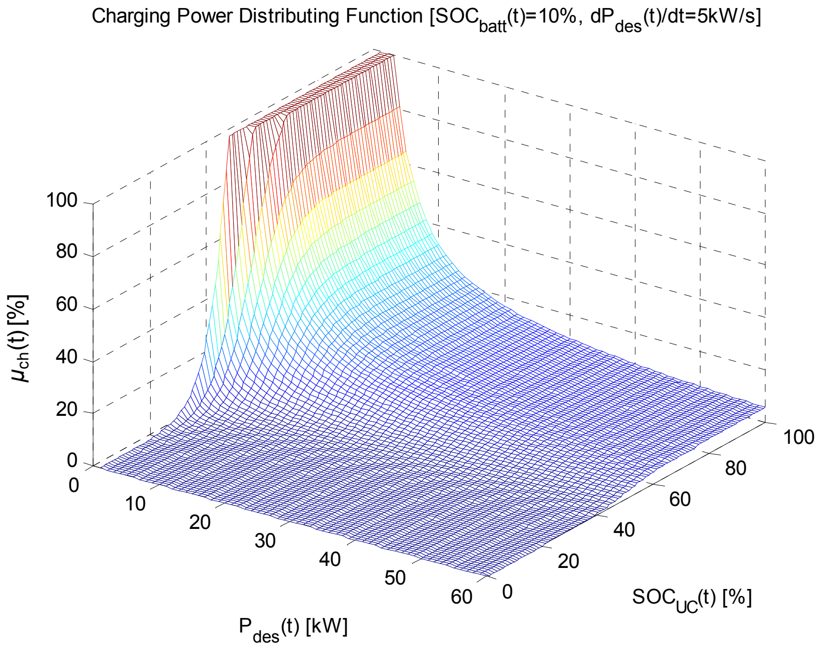


Figure 3-17: Charging Power Distributing Function [*$SOC_{batt}(t)$*=10%, *$dP_{des}(t)/dt$*=5kW/s]

Figure 3-16 depicts a relatively full battery, and it is clear that in this case majority of the charging power is sent to the ultracapacitor. Figure 3-17 shows a depleted battery, and due to the low charging power limit of the battery, it is desired to send all charging power to the battery up to its charging limit. Any additional power will subsequently be used to charge the ultracapacitor. It should be noted that in the above figures the inflection points (e_{dis} and e_{ch}) of the ultracapacitor state of charge ($SOC_{UC}(t)$) are both set to 50% for illustration purposes. They can be manually adjusted or used as optimization variables.

3.3.2.2 Scenario 2: Effects of Ultra-capacitor State of Charge

Similar to the previous scenario, Figures 3-18 and 3-19 illustrate the discharge power distributing function ($\mu_{dis}(t)$) as a function of the battery state of charge ($SOC_{batt}(t)$) and the desired power ($P_{des}(t)$), while changing the ultracapacitor SOC ($SOC_{UC}(t)$) and holding the rate of change of power ($dP_{des}(t)/dt$) constant. Figure 3-18 shows a relatively full ultracapacitor, while Figure 3-19 indicates a depleted ultracapacitor.

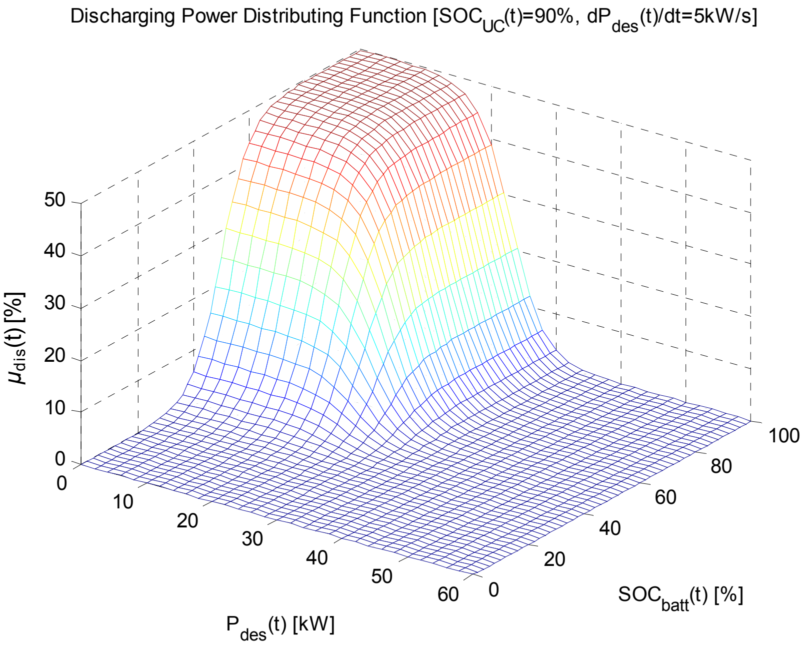


Figure 3-18: Discharging Power Distributing Function [*$SOC_{UC}(t)$*=90%, *$dP_{des}(t)/dt$*=5kW/s]

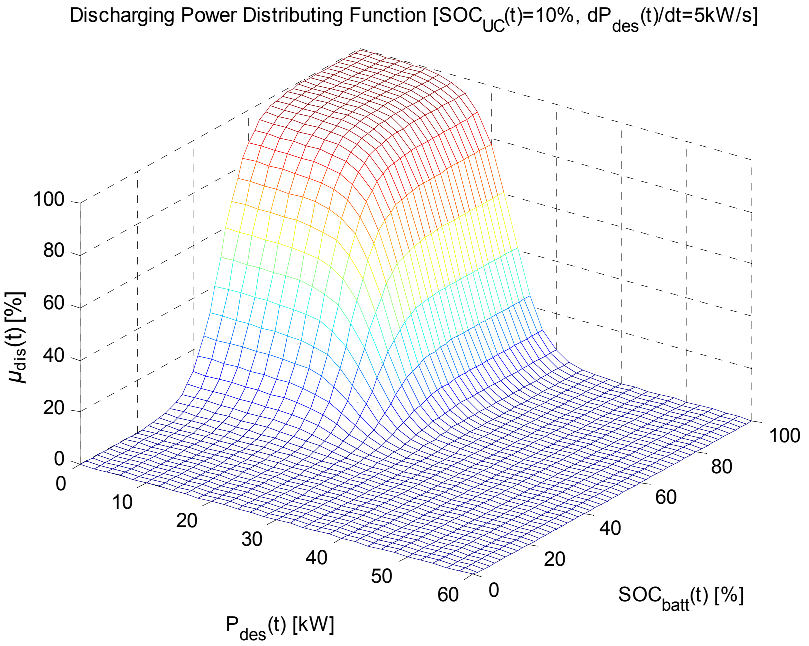


Figure 3-19: Discharging Power Distributing Function [*$SOC_{UC}(t)$*=10%, *$dP_{des}(t)/dt$*=5kW/s]

As expected, it is shown that the discharging power distributing function ($\mu_{dis}(t)$) performs according to the state of charge of the ultracapacitor. It should be noted that the maximum discharging function

($\mu_{dis}(t)$) in Figure 3-18 could be adjusted by changing the inflection point (e_{dis}) of the ultracapacitor's sigmoid function. Similarly, the charging distributing function ($\mu_{ch}(t)$) is shown in Figures 3-20 and 3-21.

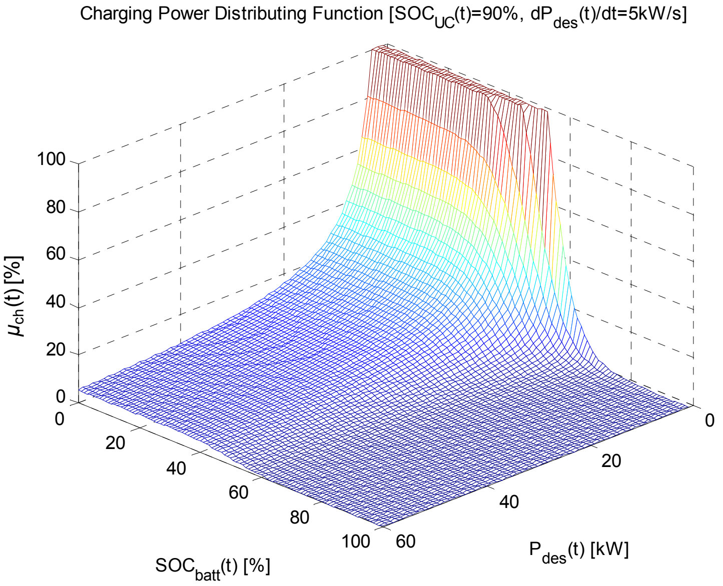


Figure 3-20: Charging Power Distributing Function [*SOC$_{UC}$(t)*=90%, *dP$_{des}$(t)/dt*=5kW/s]

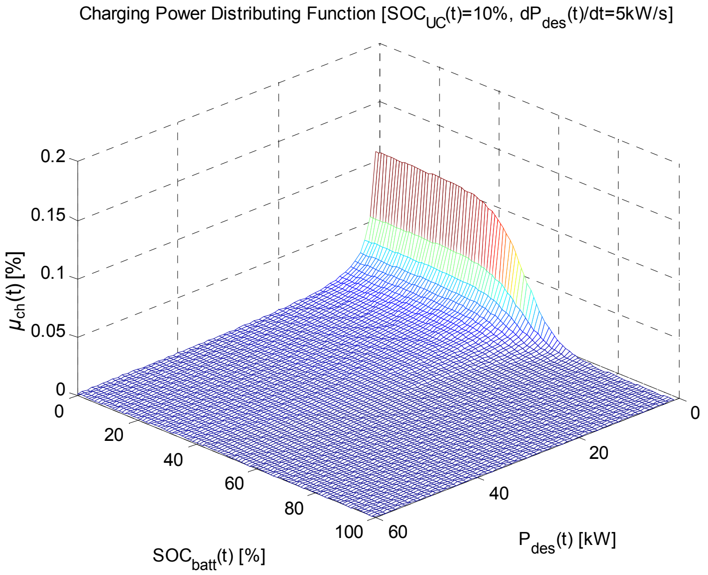


Figure 3-21: Charging Power Distributing Function [*SOC$_{UC}$(t)*=10%, *dP$_{des}$(t)/dt*=5kW/s]

Again, as expected, the charging distributing function (*$\mu_{ch}(t)$*) ensures that the proper portion of the charging power is sent to the ultracapacitor to maintain its charge. Again, it should be noted that in the above figures, the inflection points (*c_{dis}* and *c_{ch}*) of the battery state of charge (*$SOC_{batt}(t)$*) are both set to 50% for illustration purposes, and can be manually adjusted or used as optimization variables.

3.3.2.3 Scenario 3: Effects of Rate of Change of Power

The previous examples examined the effects of the battery's state of charge, the ultracapacitor's state of charge, and the desired power on the power distributing function while setting the rate of change of power constant. In this section, the rate change of power will be varied to illustrate its influence on the power distributing function. Figure 3-22 illustrates the discharge power distributing function (*$\mu_{dis}(t)$*) as a function of the desired power (*$P_{des}(t)$*) and the rate change of power (*$dP_{des}(t)/dt$*) while the battery and the ultracapacitor's states of charge remain at 50%.

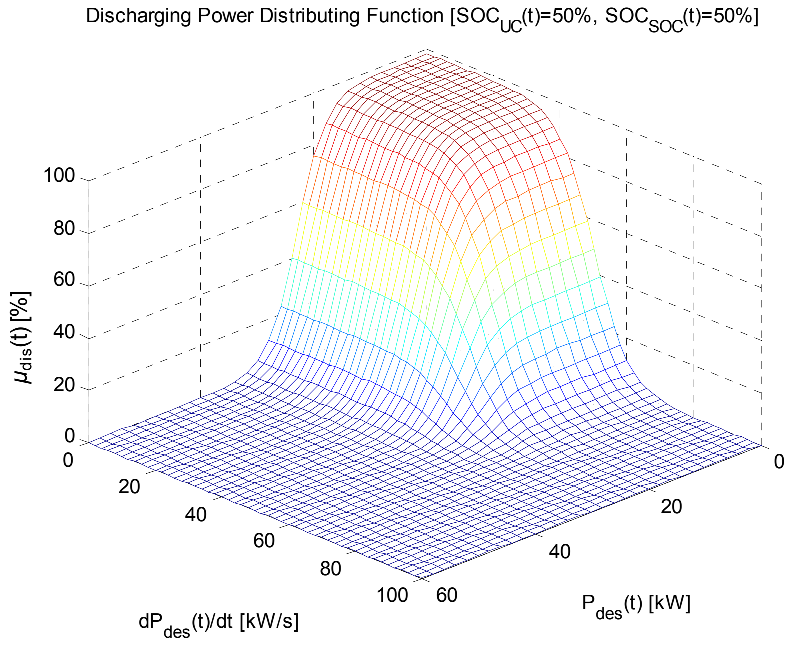


Figure 3-22: Discharging Power Distributing Function [*$SOC_{UC}(t)$*=50%, *$SOC_{batt}(t)$*=50%]

It can be seen from Figure 3-22 that when the desired power and rate of change of power are in the lower region, majority of the power is discharged from the battery. On the other hand, when the desired power or the rate of change of power is high, the power distributing function ensures that the power is discharged from the ultracapacitor. Similarly, the charging distributing function (*$\mu_{ch}(t)$*) is shown in Figure 3-23.

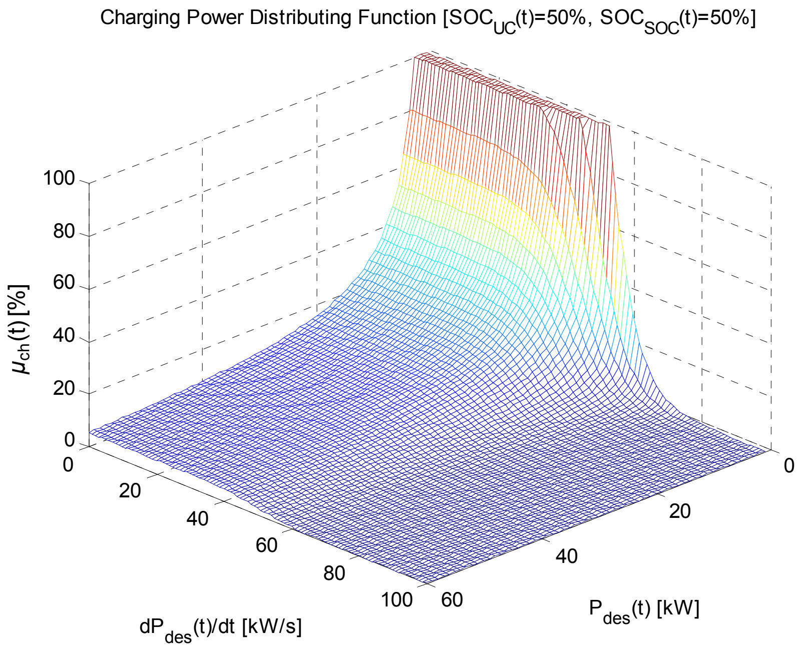


Figure 3-23: Charging Power Distributing Function [*$SOC_{UC}(t)$*=50%, *$SOC_{batt}(t)$*=50%]

Similar to the discharging function, majority of the power is used to charge the battery when the charging power and the rate of change of power are low. Also shown in Figure 3-23, due to the low charging limit of the battery, any additional charging power over the battery's charging limit is sent to the ultracapacitor. Finally, as expected, the ultracapacitor is used almost exclusively when the rate change of charging power is high.

3.4 Summary

For the purpose of optimization, a generic hybrid electric vehicle model consisting of all possible powertrain components was developed. The connections or components can be utilized depending on the desired configuration determined by the user. It was decided that since the objective of the research work was to study the overall efficiencies of the vehicle system, a backward-looking vehicle model would satisfy the power consumption calculation required while minimizing the computing resource. The powertrain components for the generic hybrid electric vehicle model consisted of quasi-static scalable components, where the appropriate size will be determined by the optimizer. A scalable internal combustion (IC) engine was modeled based on the Willans line modeling approach, while an electric motor-generator model consisting of a scalable lookup table was utilized. Due to the characteristic of the electric motor and the IC engine, the number of gears of the transmission will differ depending on which power source is driving the wheel directly. As part of the electrical energy storage (EES) system of the vehicle, an open circuit voltage-based battery model was created, where the battery's characteristic is determined by a lookup table depicting the relationship between the open circuit voltage and battery state of charge, available from the manufacturer. Similar modeling approach was used to create the ultracapacitor model. In order to manage the power flow within the EES, a power distributing function was proposed where the function monitors the battery and the ultracapacitor's SOC, the desired tractive power and the rate of change of tractive power. Based on the monitored values and the function's parameters, the power distributing function determines the power split ratio between the battery and the ultracapacitor within the EES. It was shown that the power distributing function is a simple, yet effective method to manage the power distribution between the battery and the ultracapacitor. The parameters of the sigmoid functions allow the flexibility of manual design or can be automatically determined by an optimizer. In conclusion, the aforementioned generic vehicle model with scalable powertrain components will allow flexibility and modularity during the optimization process, which will be discussed in detail in the subsequent chapters.

Chapter 4
Components and Power Controller Logic Optimization

The key contribution of the research is performing concurrent optimization on a hybrid electric vehicle's powertrain components and power management logic. Each component of the vehicle system, such as engine, motor, and electrical energy storage components along with the power management logic is considered as an individual discipline. The idea is to use Multidisciplinary Design Optimization (MDO) methodology to simultaneously optimize the overall system. It is of importance that global optimization is performed to thoroughly search for the most effective vehicle system design. Since the process of optimization is associated with a large amount of simulation a feature-based optimization approach is proposed to reduce the required simulation time. This chapter discusses the MDO approach, the architecture of the software model including the optimizer, and the feature-based optimization approach.

4.1 Process Overview

Among the MDO methods reviewed in Section 2.4, the Multidisciplinary Feasible (MDF) method is not only the simplest, but also the closest match for the vehicle's powertrain system. In a vehicle's simplest form, consider the vehicle chassis and its engine as two individual disciplines, where the engine speed is dependent by the vehicle's wheel speed, while the wheel torque is mapped from the engine torque. Similar correlations can be drawn between each of the powertrain components and the power management logic of a hybrid electric vehicle system. Since the objective of the vehicle model is to study the energy efficiencies of the system, where the dynamics of the vehicle is not of concern, the backward-looking architecture can compute with ease the vehicle power consumption and the overall efficiency accurately while utilizing minimal computing resource. Combining the MDF method and the modular vehicle layout shown in Figure 3-2, the MDF formulation of the overall system including the optimizer and the vehicle system is as follows.

$$\text{Minimize } J(\boldsymbol{X_D}, \boldsymbol{U}(\boldsymbol{X_D})) \text{ w.r.t. } \boldsymbol{X_D}$$

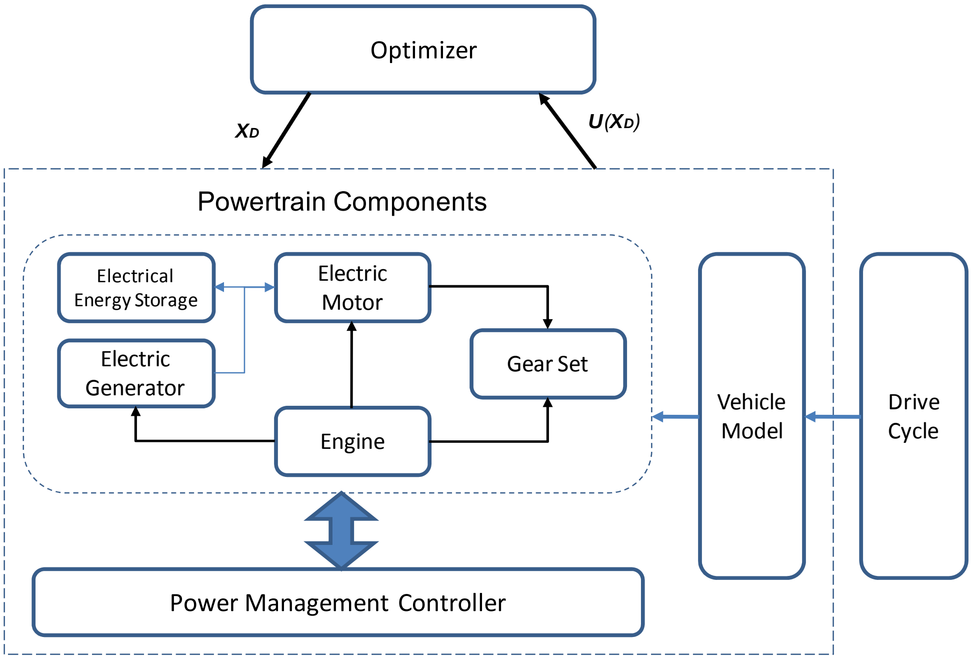


Figure 4-1: Schematics of the Optimizer with the Vehicle System

Depending on the requirements, the vehicle system can be modified to perform various design studies while maintaining the same optimization procedure. During each optimization iteration, the optimizer will choose a set of design candidates $\boldsymbol{X_D}$ to provide to the vehicle system, and a complete vehicle simulation involving various disciplines (powertrain components and power management logic) at the value of $\boldsymbol{X_D}$ is performed to obtain the system output variables $\boldsymbol{U(X_D)}$. The design candidate vector $\boldsymbol{X_D}$ contains optimization variables such as the power management logic and the powertrain component sizing parameters, while the system output variables $\boldsymbol{U(X_D)}$ are the energy consumptions of the vehicle in the form of either fuel and/or electricity consumption. The system output variable vector $\boldsymbol{U(X_D)}$ is then used to evaluate the objective function $J(\boldsymbol{X_D}, \boldsymbol{U(X_D)})$ of the optimization algorithm.

4.2 Optimization Algorithm

Many researchers had contributed to the field of optimization over the years, and numerous algorithms are available to solve different types of minimization problems. The optimization methods can be classified into two categories: classical and global methods. This section describes briefly some of the most commonly used optimization methods.

4.2.1 Classical Optimization

Classical optimization is an established field of study, where most of the algorithms have existed for many years. The benefit of classical methods is its high computational speed; however, the drawback is that they are all local optimization methods by nature. The methods described in this section include both algorithms that require gradient information and derivative-free methods.

4.2.1.1 Line Search Methods

The line search method calculates a search direction p_k from the current point and then decides how far to move along that direction. The iteration is given by

$$x_{k+1} = x_k + \alpha_k p_k \tag{4.1}$$

where the positive scalar α_k is called step length, x_k is the current point, and x_{k+1} the point at the next iteration. The success of a line search method is dependent on the choices of both the direction p_k and the step length α_k. Most line search algorithms require p_k to be a descent direction, for which $p_k^t \nabla J_k < 0$. This property guarantees that the function J can be reduced along this direction. Furthermore, the search direction takes the form

$$p_k = -B_k^{-1} \nabla J_k \tag{4.2}$$

where B_k is a symmetrical and nonsingular matrix. In the steepest descent method, B_k is simply the identity matrix I, while in Newton's method, B_k is the exact Hessian $\nabla^2 J(x_k)$. In quasi-Newton methods, B_k is an approximation to the Hessian that is updated at each iteration by means of a low-rank formula. Detailed definition of various line search methods can be found in [35].

4.2.1.2 Trust Region

Trust region is similar to the line search method in that they both generate a step using a quadratic model of the objective function. However the difference is in that line search methods use it to generate a search direction, while trust region methods define a region around the current iteration within which they *trust* the model to be an adequate representation of the objective function. The algorithm then chooses the step that is the approximate minimizer of the model in this region. If this step is not acceptable, the size of the region is reduced and a new minimizer if found. In practical algorithms, the size of the region is chosen according to the performance of the algorithm during previous iterations. If the model is consistently producing good steps and accurately predicting the behaviour of the objective function along these steps, the size of the trust region may be increased to allow longer and large steps to be taken. On the other hand, if the step failed, then the size of the region is reduced and the process is repeated. Detailed procedures of the trust region algorithm are discussed in [35].

4.2.1.3 Nelder-Mead Simplex Direct Search

Nelder-Mead simplex-reflection method takes its name from the fact that at any stage of the algorithm there are $n + 1$ points of interest in an n-dimensional space, whose convex hull forms a simplex. Given a simplex S with vertices $\{z_1, z_2,\ldots,z_{n+1}\}$, an associated matrix V(S) is defined by taking the n edges along V from one of its vertices (e.g., z_1).

$$V(S) = [z_2 - z_1, z_3 - z_1, \ldots, z_{n+1} - z_1] \tag{4.3}$$

In a single iteration of the Nelder-Mead algorithm, the vertex with the worst function value is removed and replaced with another point with a better value. The new point is obtained by reflecting, expanding, or contracting the simplex along the line joining the worst vertex with the centroid of the remaining vertices. If a better point cannot be found in this manner, the vertex with the best function value is retained, and the simplex is shrunk by moving all other vertices towards the retained vertex. Further information and details of the algorithm can be found in [35].

4.2.1.4 Pattern Search

Pattern search is a constrained derivative-free optimization technique that is similar to line search methods. At each iteration, a certain set of search directions is chosen, and the objective function (J) is evaluated at a given step length along each of these directions. These candidate points form a frame around the current iterate, and if a point with a significantly lower function value is found, it is adopted as the new iterate and the center of the frame is shifted to this new point. At each of the current iterations with a set of design variables x_k, D_k is defined to be the set of possible search directions and γ_k to be the line search parameter. The frame consists of the points $x_k + \gamma_k p_k$, for all $p_k \in D_k$. When one of the points in the frame yields a significant decrease in J, the step is taken and γ_k is increased to expand the frame for the next iteration. If none of the points in the frame has a significantly improved function value with respect to J_k, γ_k is reduced and the process is repeated. The main difference between a pattern search and a line search method is that in pattern search, the direction set p_k does not require information on the derivative information of the objective function, as compared to that of the line search methods. [35] In MATLAB, the direction set is generated based on a set of rational basis vectors [36].

4.2.2 Global Optimization

As opposed to classical optimization techniques, global optimization methods search the entire problem space, thus avoiding the optimizer being "stuck" in local minima. The downside is the additional computational time required to search the global space thoroughly. This section describes two global optimization techniques available: genetic algorithm and simulated annealing.

4.2.2.1 Genetic Algorithm

Genetic algorithm (GA), as its names suggests, is an optimization procedure inspired by the biological process of evolution and the survival-of-the-fittest concept, formally introduced by Holland in the 1970s [37]. It is a global optimization technique and is derivative-free, and thus can be easily applied to both continuous and discontinuous functions. The algorithm is based on the evaluation of the objective function at a set of points within the function's variable space, which is usually first chosen randomly within the search region. Such feature allows the algorithm to be less vulnerable to local optima, and is an excellent method to solve global optimization problems.

The algorithm begins with generating a random population in the range of the optimization variables, and uses the binary encoding procedure to represent each variable as a string of binary digits. The generated variables are termed individuals, which contain a collection of genetic traits or genotypes, and are referred to as chromosomes. The genotypes are represented as strings of binary digits, or genes in GA terminology. The objective functions using individuals are first evaluated and the result is called the fitness of the individuals. A set of genetic operators, such as selection, crossover, and mutation are then applied to the population. Selection is a process in which individuals are copied based on their fitness values, where highly fit individuals will have a higher number of offsprings in the succeeding generation. Crossover is then applied by combining successful individuals by exchanging equivalent lengths of their chromosomes, where the two strings from the reproduced population are mated randomly. Finally, mutation is performed by picking a random chromosome and flipping a gene randomly. In essence, crossover represents searching within the local region, while mutation explores the global space to avoid being trapped in a local optimum. The fitness function of the individuals is then evaluated, and the process is repeated until the best fitness is obtained or the maximum generation is reached. Detailed GA procedure and formulation can be found in [37].

4.2.2.2 Simulated Annealing

Simulated annealing is a stochastic optimization techniques based on random evaluation of the objective function in order to avoid being trapped in a local minimum. The name of the techniques is inspired by the annealling process in metallurgy, where a controlled and slow cooling of heated solid ensures proper solidification of its crystal while reducing their defects. Such process ensures a crystalline state that corresponds to its lowest internal energy.

The optimization problem can be states as:

$$\text{Minimize } J(\boldsymbol{x})$$

$$\text{Subject to } x_m^{(l)} \leq x_k \leq x_m^{(u)}, m = 1,2,\ldots,n$$

where $J(\boldsymbol{x})$ is the objective function, and $\boldsymbol{x}$ is a vector of the optimization variables bounded by its upper ($x_m^{(u)}$) and lower ($x_m^{(l)}$) limits. The algorithm starts with an initial vector $\boldsymbol{x}_1$, and moves randomly along each coordinate direction to generate successively improved point to a global minimum solution. During each iteration, the objective function $J(\mathbf{x}_{k+1})$ is evaluated at a candidate vector $\boldsymbol{x}_{k+1}$ from the current point $\boldsymbol{x}_k$ to determine whether an improvement of the objective function is achieved. If $\Delta J \leq 0$, the new point is accepted and $\boldsymbol{x}_{k+1} = \mathbf{x}$. Otherwise, the new point is accepted with a probability of

$$P(\Delta J) = e^{-\Delta J/k_{SA}T_{SA}} \tag{4.4}$$

where $\Delta J = J(\mathbf{x}_{k+1}) - J(\mathbf{x}_k)$, k_{SA} is a scaling factor called Boltzmann's constant, and T_{SA} is the temperature. The algorithm begins with a "high" temperature ($T_{SA,0}$), and a sequence of design vectors is generated until the equilibrium is reached, where the average value of J stabilizes as k_{SA} increases. Once the thermal equilibrium is reached, the temperature T_{SA} is reduced and a new sequence of moves is made again until thermal equilibrium is reached once again. This process is repeated until a sufficiently low temperature is reached, at which stage the global minimum is found.

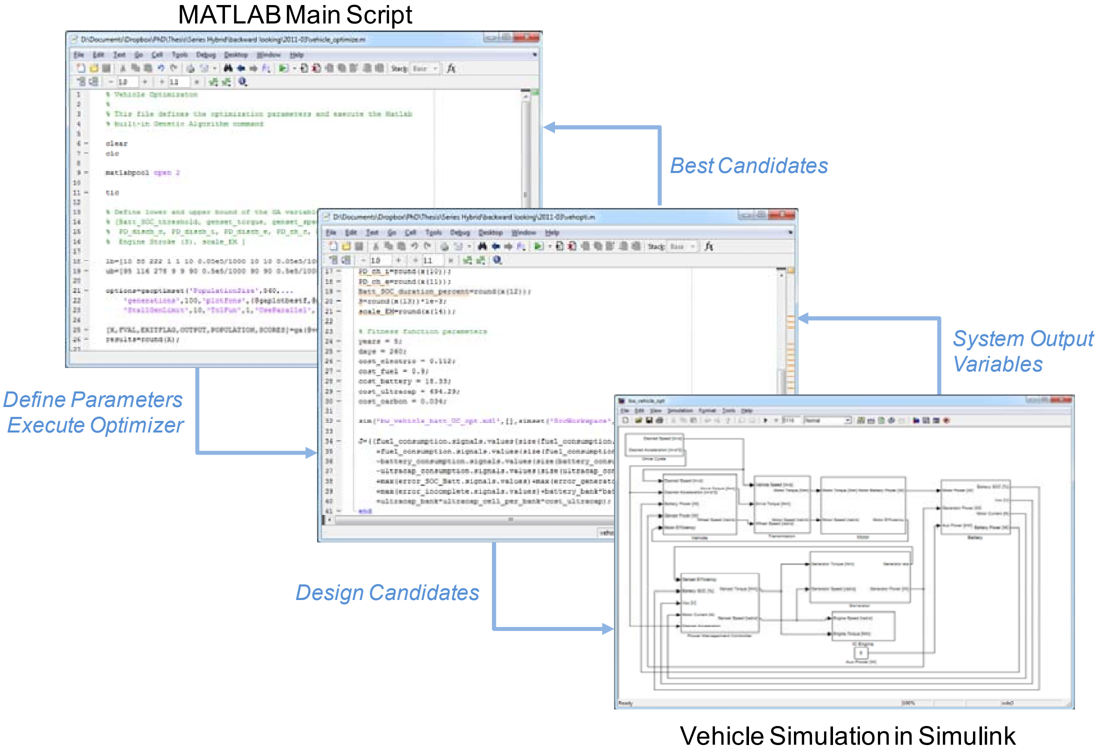
It should be noted that the initial guesses of the optimization variables does not affect the quality of the final solution, except that computational time may increase with worse starting point. Additionally, due to the discrete nature of the function and constraint evaluations, the convergence or transition characteristics are not affected by the continuity or differentiability of the functions. Therefore, the simulated annealing optimization algorithm is well suited to solve discrete problems with the ability to find the global optimal solution [38].

4.2.3 Software Structure

The modeling and simulation work of this research was implemented in the MATLAB/Simulink environment, where the vehicle component modules, described in Chapter 3, were created in Simulink, and the components' properties were created using m-files. The advantage of such approach is that the parameters of the components can be changed with ease and a library of various types of components can be constructed. It is therefore desired to perform the optimization procedure within the MATLAB /Simulink environment.

There are two methods of performing optimization in the MATLAB /Simulink environment: (i) Optimization Toolbox (optimtool) and (ii) Simulink Response Optimization. The Optimization Toolbox is a graphical user interface toolbox that is based on MATLAB commands, while the Simulink Response Optimization is a library of block diagrams in the Simulink environment. After an extensive investigation, it was found that using the MATLAB command provides the greatest flexibility over the control of each optimization algorithm while integrates most seamlessly with the vehicle software model.

To execute an optimization procedure, a main MATLAB script is used to first define the upper ($\boldsymbol{X_{D,u}}$) and lower ($\boldsymbol{X_{D,l}}$) bounds of the design variables, and also the optimization parameters of each of the algorithms. The main script then executes the optimization algorithm's MATLAB command that minimizes a custom function, which takes the form of a user defined m-file that initializes the vehicle and its components' properties, and subsequently performs the vehicle model simulation in Simulink. Once the vehicle simulation is complete, the output variables ($\boldsymbol{U(X_D)}$) are used to evaluate the objective function ($J(\boldsymbol{X_D}, \boldsymbol{U(X_D)})$) and passed back to the optimizer. The vehicle simulation in Simulink and the objective function evaluation are repeated until the terminating condition of the optimizer is satisfied. Once the optimization is completed, the main script executes the vehicle model simulation in Simulink using the best design candidate as determined from the optimization to obtain the final results. Using the above-mentioned procedure, the optimized system configuration along with its results can be obtained with one click of a button. Figure 4-2 illustrates the optimization procedure in the MATLAB/Simulink environment.


MATLAB Main Script
D:\Documents\Dropbox\PhD\Thesis\Series Hybrid\backward looking\2011-03\vehicle_optimize.m
Best Candidates
D:\Documents\Dropbox\PhD\Thesis\Series Hybrid\backward looking\2011-03\vehopti.m
System Output
Variables
Define Parameters
Execute Optimizer
Design Candidates
Vehicle Simulation in Simulink

```
% Vehicle Optimization
%
% This file defines the optimization parameters and execute the Matlab
% built-in Genetic Algorithm command

clear
clc

matlabpool open 2

tic

% Define lower and upper bound of the GA variabl
% [Batt_SOC_threshold, genset_torque, genset_spe
%  FD_disch_n, FD_disch_i, FD_disch_e, FD_ch_n,
%  Engine Stroke (3), scale_EM ]

lb=[10 50 222 1 1 10 0.05e5/1000 10 10 0.05e5/10
ub=[95 116 278 9 9 90 0.5e5/1000 90 90 0.5e5/100

options=gaoptimset('PopulationSize',240,...
    'generations',100,'plotfcns',{@gaplotbestf,@
    'StallGenLimit',10,'TolFun',1,'UseParallel',

[X,FVAL,EXITFLAG,OUTPUT,POPULATION,SCORES]=ga(@v
results=round(X);
```

```
FD_ch_i=round(x(10));
FD_ch_e=round(x(11));
Batt_SOC_duration_percent=round(x(12));
S=round(x(13))*1e-3;
scale_EM=round(x(14));

% Fitness function parameters
years = 5;
days = 260;
cost_electric = 0.112;
cost_fuel = 0.9;
cost_battery = 18.33;
cost_ultracap = 494.29;
cost_carbon = 0.034;

sim('hu_vehicle_batt_SC_opt.mdl',[],simset('SrcWorkspace',

J=((fuel_consumption.signals.values(size(fuel_consumption.
    +fuel_consumption.signals.values(size(fuel_consumptio
    -battery_consumption.signals.values(size(battery_consu
    -ultracap_consumption.signals.values(size(ultracap_con
    +max(error_SOC_Batt.signals.values)+max(error_generato
    +max(error_incomplete.signals.values)+battery_bank*bat
    +ultracap_bank*ultracap_cell_per_bank*cost_ultracap);
end
```

Figure 4-2: Optimization Procedure in MATLAB/Simulink

4.2.4 Optimization Comparison

MATLAB contains a library of optimization algorithms that includes the aforementioned classical and global optimization methods [39]. Due to the functionality available within the software package, it is desired to utilize the MATLAB optimization command to perform optimization of the vehicle system. Due to the different nature of the optimization algorithms, not all will satisfy the requirement of performing the concurrent optimization of the vehicle system. Since the evaluation of the objective function will be performed by the Simulink vehicle model, it is not possible to obtain derivative information, namely the Gradient and the Hessian, for use in derivative-based optimization algorithm. Of those optimization algorithms available in MATLAB, several methods are derivative free and are therefore able to perform optimization using the Simulink vehicle model. Table 4-1 shows some of the derivative-free optimization algorithm available in MATLAB.

Table 4-1: Derivative-Free Optimization Algorithms in MATLAB [39]

Method	MATLAB Command	Type
Nelder-Mead	fminsearch	Local, unconstrained
Pattern Search	patternsearch	Local, constrained
Simulated Annealing	simulannealbnd	Global, constrained
Genetic Algorithm	ga	Global, constrained

In addition to that the algorithm needs to be derivative-free, it is desired to have the capability of searching the global space to avoid being trapped in the local minimum. Table 4-1 shows that both simulated annealing and genetic algorithm are global optimization by nature, and will be implemented to obtain the global optimal solution. Furthermore, classical methods such as the Nelder-Mead and pattern search algorithm will also be used to perform vehicle system optimization for comparison purposes. The optimization results of the aforementioned algorithms will be compared and presented in the subsequent Chapters.

4.3 Feature-Based Optimization

As described in the previous section, when performing vehicle energy consumption calculation, the entire drive cycle is used during simulation. During optimization, thousands of simulations are performed where the vehicle is to complete the entire drive cycle. With the large number of optimization variables of the vehicle powertrain and power management logic, running thousands of simulations for each optimization case could results in hours if not days of simulation time.

To reduce simulation run-time, feature-based optimization is developed by utilizing a statistical approach to extract velocity and acceleration information of a drive cycle, where a 3D histogram of the drive cycle is first generated to determine the range of the velocity and acceleration values. The extracted velocity and acceleration combination is then applied to the Simulink vehicle model to determine the energy consumption, thereby generating an energy map corresponding to the velocity and acceleration range of the histogram. The energy map is then used to perform the energy calculation of the drive cycle. During the optimization process, only the drive cycle energy calculation is repeated when changing the optimization variables, thus eliminating the need to perform Simulink vehicle simulation and reducing the simulation run-time significantly. Additionally, there exists some research work aiming to construct a comprehensive drive cycle utilizing statistical real world driving data collected over a period of time. The collected data is then used to generate a driving cycle that can be used to perform a full vehicle simulation [40]. However, by utilizing the feature-based approach, real world statistical driving data can be used directly to perform vehicle simulation without any additional effort, thereby eliminating the need to generate a specific driving cycle, further improving the process efficiency of the vehicle's design cycle. The following sections describe the process in details.

4.3.1 Drive Cycle Decomposition

A typical drive cycle consists of a desired vehicle velocity time history as defined by the user. Additionally, there are standard drive cycles available and are used worldwide. In this research, standard city and highway drive cycles developed by the US Environmental Protection Agency (EPA) are used. Further information is provided in Appendix B. Typical city and highway drive cycles are shown in Figures 4-3 and 4-4.

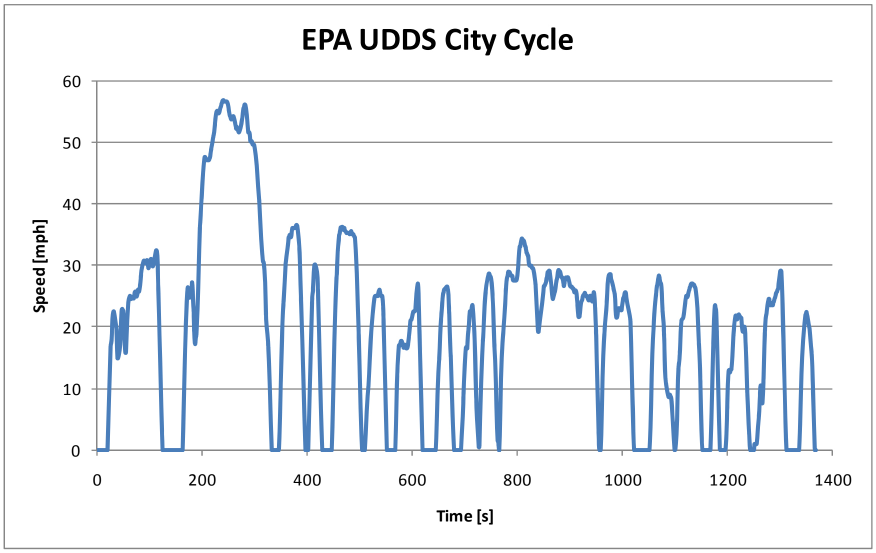


Figure 4-3: Standard EPA UDDS City Cycle

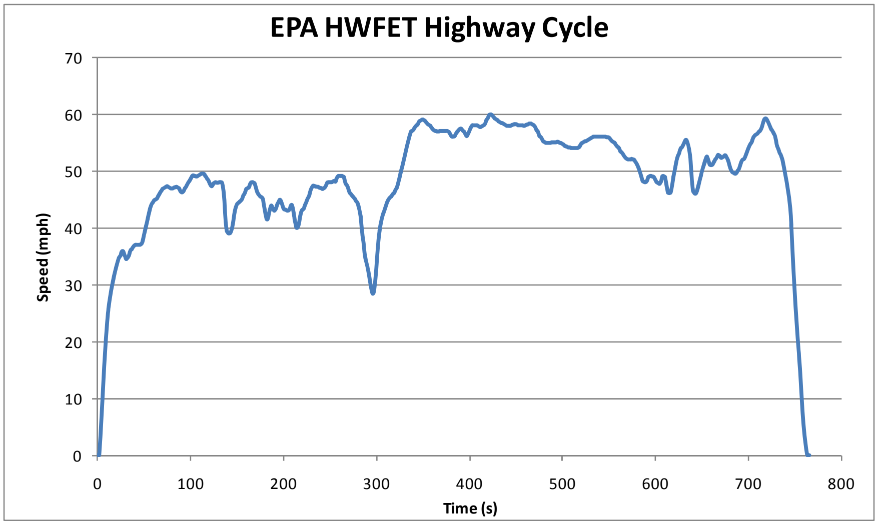


Figure 4-4: Standard EPA HWFET Highway Cycle

The first step in extracting the features of the drive cycle is to analyze the velocity and acceleration time history data using the 3D histogram (hist3) command in MATLAB, where it will automatically determine the range of the velocity and the acceleration of the drive cycle and generate an occurrence count based on a user defined interval. As an example, Figures 4-5 and 4-6, respectively, show a 3D histogram of the city and highway drive cycles, where the velocity and acceleration range are divided into 15 intervals.

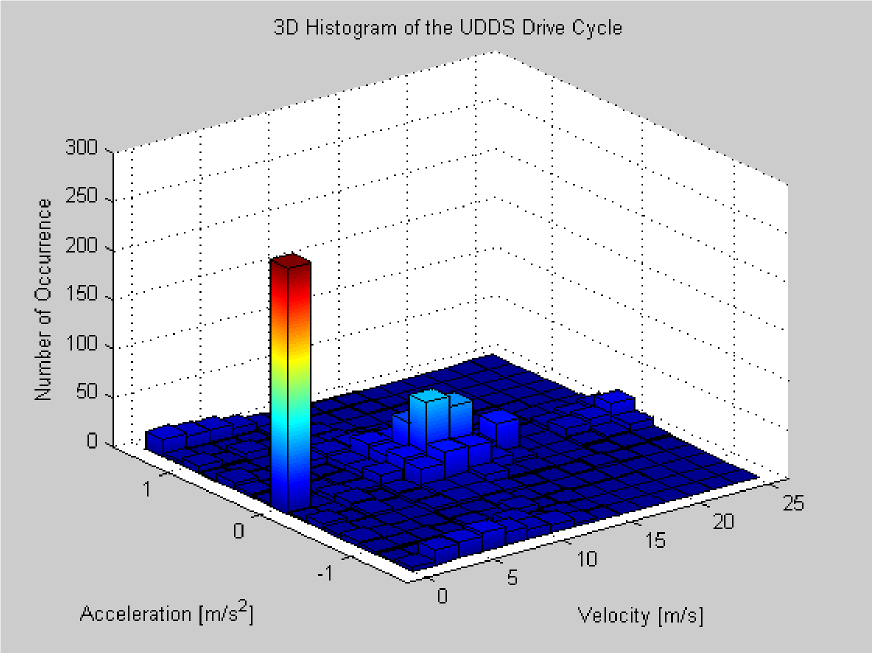


Figure 4-5: 3D Histogram of the UDDS City Cycle

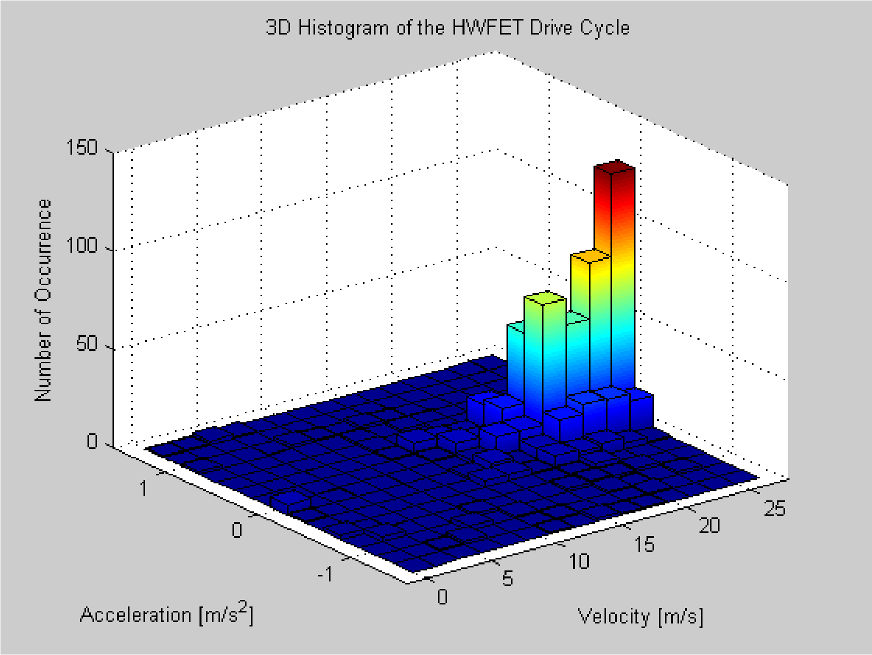


Figure 4-6: 3D Histogram of the HWFET Highway Cycle

As shown in the above figures, the histogram command automatically determined the velocity and acceleration intervals based on the range of the drive cycle and the user desired interval count. Each combination of the velocity and acceleration is called a bin, and each bin contains the number of occurrence of a specific combination of velocity and acceleration over the drive cycle. Naturally, Figure 4-5 indicated the most occurrences at low speed for the city cycle, while Figure 4-6 showed more occurrences at higher speeds for the highway cycle. The actual values of the velocity and accelerations intervals are stored in the vectors $\boldsymbol{v}_{fea}$ and $\boldsymbol{a}_{fea}$, respectively, where the sizes of both vectors are determined by the size of the user-defined bin size. Additionally, the time interval of each data point from the drive cycle time history data is defined as the time step of the drive cycle, and is stored in the variable t_s. The variables are subsequently used to generate the energy map as described in the next section.

4.3.2 Energy Map Generation

Once the velocity and the acceleration intervals were determined using the histogram command, simulations are performed using the vehicle model in Simulink to determine the energy consumption of each bin to generate an energy map, defined by an $i \times j$ vector $\boldsymbol{E_{map}}$. For each i and j location of the energy map, the corresponding velocity ($v_{fea,i}$) and acceleration ($a_{fea,j}$) are used as the initial velocity and acceleration of the Simulink vehicle model, which is simulated for a duration equal to the drive cycle time step (t_s). The energy value is taken as the consumed or generated energy of the electrical energy storage system, thereby taking into account the efficiencies of the power components. As an example, Figures 4-7 and 4-8 depict the energy maps of the city and highway cycles using the same velocity and acceleration intervals as determined from the previous section.

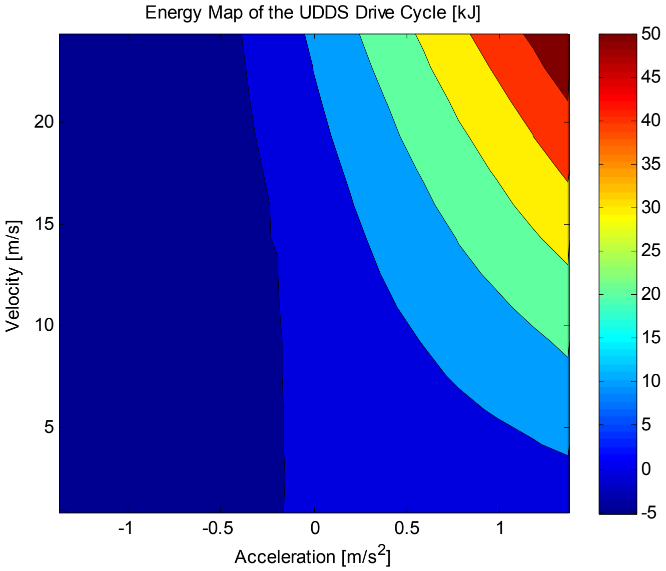


Figure 4-7: Energy Map of the UDDS City Cycle

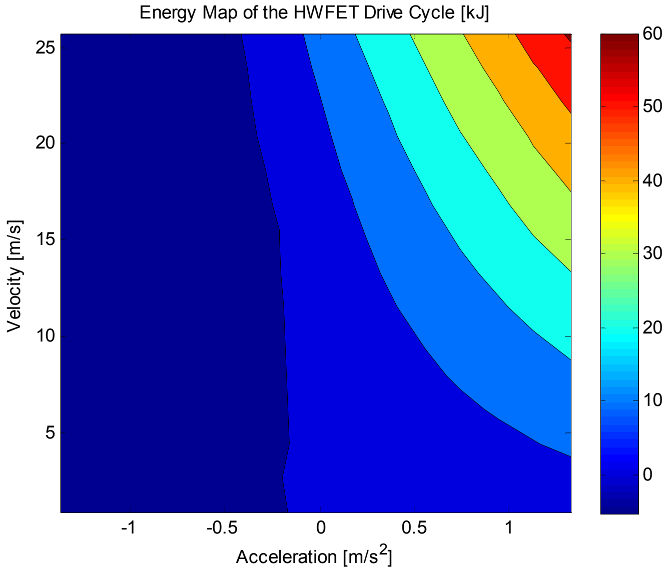


Figure 4-8: Energy Map of the HWFET Highway Cycle

It should be noted that negative acceleration denotes braking, and subsequently negative energy indicates a regenerated energy from regenerative braking. Once the energy map of the drive cycle is created, it is used to calculate the vehicle's energy consumption due to the drive cycle.

4.3.3 Drive Cycle Energy Calculation

The feature-based simulation approach determines only the total amount of energy that is required to complete the drive cycle while taking into account the friction, aerodynamics drag, and power component efficiency losses. This energy consumption could be electrical (battery or ultracapacitor) or chemical (IC engine), and in the case of a hybrid electric vehicle a combination of both. Furthermore, it does not consider the power capability and the capacity of the electrical energy storage components. In order to compensate for the lack of time dependence information, it is necessary to keep track of the state of charge (SOC) of the electrical energy storage system throughout the drive cycle. This is achieved by dividing the drive cycle to *m* segments and performing energy calculation for each section. As an example, the HWFET highway drive cycle shown in Figure 4-4 is divided into 50-second intervals (*m*=16), as illustrated in Figure 4-9.

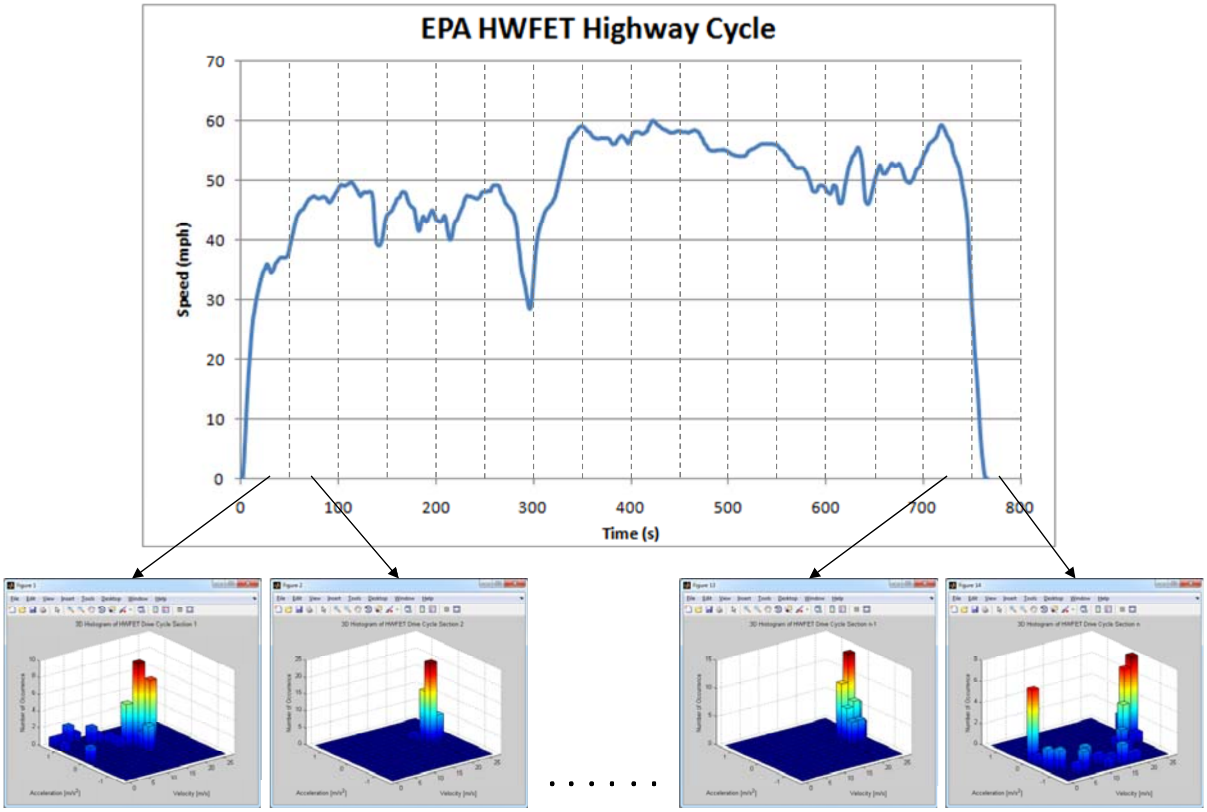


Figure 4-9: HWFET Highway Drive Cycle in Multiple Sections

To calculate the energy consumption of the drive cycle while taking into account the state of charge (SOC) of the electrical energy storage components, the energy consumption of each section is calculated sequentially starting with the first section at initial time. During energy calculation at each section m, a histogram ($\boldsymbol{H}_{sec}$) indexed by the previously defined velocity ($\boldsymbol{v}_{fea}$) and acceleration ($\boldsymbol{a}_{fea}$) range is generated for that particular section as shown in Figure 4-9. Each section's histogram is then multiplied to the energy map to determine the total energy consumption of that section. The state of charge of the electrical energy storage components is then updated to reflect the energy consumption for that particular section, and is subsequently used as the initial state of charge for the next section. Depending on the power management's algorithm, the genset may be activated during each section (m) to provide additional energy to the system, and thus the fuel consumption ($C_{f,sec,m}$) is subsequently calculated using the IC engine formulation as described in Subsection 3.2.1. The energy calculation is repeated for each of the section of the drive cycle, and the total energy consumption (E_{tot}) of the drive cycle is defined by

$$E_{tot} = \sum_{1}^{m}\sum_{1}^{i}\sum_{1}^{j} E_{map,ij} \times H_{sec,m} \tag{4.5}$$

while the total fuel consumption ($C_{f,tot}$) of the drive cycle is simply

$$C_{f,tot} = \sum_{1}^{m} C_{f,sec,m} \tag{4.6}$$

Using the previously described formulation the total energy and fuel consumption of the drive cycle can be determined while taking into account of the power capacity of the electrical energy storage devices. The next subsection will provide discussion of using the feature-based simulation for concurrent optimization of the vehicle system.

4.3.4 Optimization

The motivation behind feature-based optimization is to reduce the simulation run-time by avoiding the repetition of full Simulink vehicle model simulations during optimization. During a typical simulation using conventional method, a full simulation of the Simulink vehicle model is performed whenever an optimization variable is changed. This results in thousands of Simulink vehicle simulation which requires hours of simulation time. By utilizing the feature-based simulation as outlined previously, Simulink vehicle simulation occurs only when generating the energy map, while optimization utilizes only the drive cycle energy calculation as described in Subsection 4.3.3 to determine its objective function. Since performing energy calculation in MATLAB utilizes only a fraction of the time compared to what is required in Simulink, significant amount of time can be reduced. Figure 4-10 illustrates the flow diagram of the feature-based optimization method.

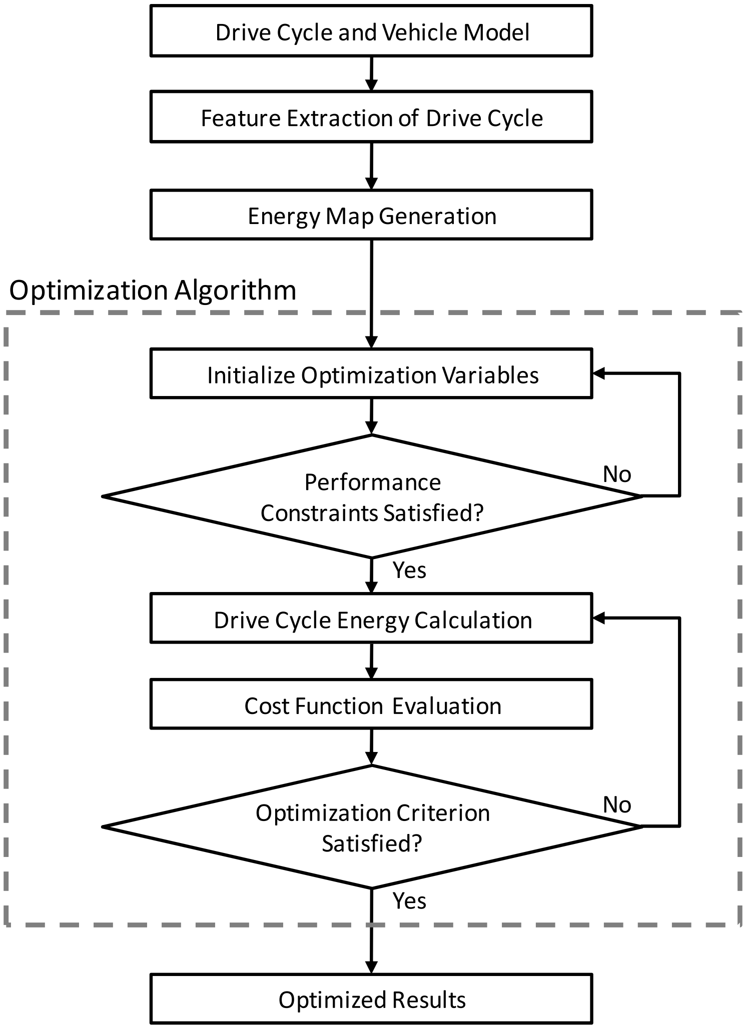

Drive Cycle and Vehicle Model
Feature Extraction of Drive Cycle
Energy Map Generation
Optimization Algorithm
Initialize Optimization Variables
Performance Constraints Satisfied?
No
Yes
Drive Cycle Energy Calculation
Cost Function Evaluation
Optimization Criterion Satisfied?
No
Yes
Optimized Results

Figure 4-10: Feature-Based Optimization Flow Diagram

During optimization, the design candidates such as the power management logic and powertrain component sizing variables are changed, thereby changing the energy consumption values of the drive cycle. To ensure that the vehicle completes the drive cycle with adequate energy and power when the optimization variables are changed by the optimizer, it is necessary to apply the performance constraints prior to calculating the energy consumption of the drive cycle, where the maximum electrical energy storage power needs to exceed the maximum required power of the drive cycle while taking into account the traction motor efficiency. Additionally, since the sizing of the power components are changed during the optimization, thereby affecting the mass of each new configuration, it is necessary to apply a scaling factor to the energy map to accurately capture the effects of component sizing. The new energy map ($E_{map,ite}$) at each of the iterations due to a different mass can be defined as follows

$$E_{map,ite} = K \times E_{map} \tag{4.7}$$

where K is the scaling factor due to the changing mass at each iteration and E_{map} is generated by the Simulink vehicle model as described in Subsection 4.3.2 using the initial configuration. By relating energy map to the desired vehicle power and the drive cycle time step, Equation (4.7) can be rearranged as

$$K = \frac{E_{map,ite}}{E_{map}} = \frac{P_{map,ite} \times \eta_{pt} \times t_s}{P_{map} \times \eta_{pt} \times t_s} \tag{4.8}$$

where $P_{map,ite}$ is the power map of each iteration during the optimization, and P_{map} is the power map generated using the initial configuration. Additionally η_{pt} is the efficiency of the powertrain components and t_s is the time step of the drive cycle. It should be noted that both $P_{map,ite}$ and P_{map} are $i \times j$ vectors corresponding to the velocity (v_{fea}) and acceleration (a_{fea}) range as defined in Subsection 4.3.2. For each i^{th} and j^{th} component, $P_{map,ite,ij}$ can be determined using the derivations discussed in Subsection 3.1.2 as

$$P_{map,ite,ij} = m_{veh,ite} a_{fea,j} + \frac{1}{2} C_D \rho_{air} v_{fea,i}^2 A + C_{RR} m_{veh,ite} g \tag{4.9}$$

where $m_{veh,ite}$ is the vehicle mass at each iteration, and respectively $v_{fea,i}$ and $a_{fea,j}$ are the corresponding velocity and acceleration at the i^{th} and j^{th} component. The vehicle drag and tire rolling resistance parameters were as described in Subsection 3.1.1. Similarly, $P_{map,ij}$ can be expressed as

$$P_{map,ij} = m_{veh} a_{fea,j} + \frac{1}{2} C_D \rho_{air} v_{fea,i}^2 A + C_{RR} m_{veh} g \tag{4.10}$$

where m_{veh} is the vehicle mass of the initial configuration. Substituting Equations (4.9) and (4.10) into Equation (4.8) while cancelling the common terms in the numerator and denominator, the scaling map K can be calculated based on the configuration at each of the iterations during optimization. Finally, applying K to the energy calculation of the drive cycle as described by Equation (4.5), the total energy consumption ($E_{tot,ite}$) at each of the iteration can be expressed as

$$E_{tot,ite} = \sum_{1}^{m} \sum_{1}^{i} \sum_{1}^{j} E_{map,ij} \times H_{sec,m} \times K \tag{4.11}$$

Using the above derivation, the energy consumption can be accurately calculated while reflecting the change in vehicle mass during each iteration. Finally, the energy consumption is used to determine the cost function of the objective function to determine the optimal solution.

4.4 Summary

The key contribution of the research is to perform concurrent optimization on a hybrid electric vehicle's powertrain components and power management logic. Multidisciplinary Design Optimization methodology is utilized to perform concurrent optimization of the vehicle system. The vehicle simulation and optimization is implemented in the MATLAB/Simulink environment, where the vehicle component properties are defined using m-files, while the vehicle model is created in Simulink. Vehicle system optimization is achieved by using the built-in library of MATLAB's optimization algorithms, and is executed using the MATLAB command. Due to the large number of simulations required during a typical optimization process, a feature-based optimization is developed with the objective of reducing simulation time. Feature-based optimization utilized statistical approach to extract drive cycle information, where it is used by the Simulink vehicle model to create an energy map. The energy map is subsequently used to perform vehicle energy calculation during the optimization process. Since Simulink simulations are avoided during optimization, simulation run-time can be reduced considerably. The comparison between various optimization algorithms, along with the performance and results of the feature-based optimization will be discussed in the next Chapter.

Chapter 5

Case Study 1: Series Hybrid Electric Vehicle

A series hybrid electric vehicle (SHEV) is created in MATLAB/Simulink to demonstrate the effectiveness of the proposed concurrent optimization approach, where the component sizing and power management logic of the vehicle are to be optimized simultaneously. The advantage of a series hybrid electric vehicle configuration is that the power management logic is relatively simple, where the genset (engine and generator) will only operate in the most efficient region to recharge the battery and supplement traction power. This chapter will discuss the series hybrid electric vehicle model in detail and provide simulation and optimization results.

5.1 Background and Objective

The design challenge of a series hybrid electric vehicle is in balancing the size of the battery against the engine's fuel consumption. By increasing the battery size, the electric range of the vehicle is increased, thereby reducing the fuel consumption of the engine required to recharge the battery. However, due to the high cost of batteries, having a large battery pack to increase the electrical capacity may not be financially effective. Therefore, the design objective is to determine a system configuration where fuel consumption is minimized while utilizing a reasonably-sized battery pack. Of all hybrid electric vehicles launched recently, General Motors' Chevrolet Volt has perhaps received the most media attention. It is designed as an electric vehicle with a range extender, where an internal combustion (IC) engine is coupled with a generator to provide additional electric energy when the battery is depleted. Additionally, the battery can also be recharged by plugging into a standard household electric receptacle. Furthermore, the Volt is capable of operating in electric mode for 64km (based on standard city cycle) before operating the engine to recharge the 16kWh lithium-ion battery pack [41]. Upon inspection, the system architecture of an electric vehicle with a range extender is identical to that of a series hybrid electric vehicle, and therefore it is used interchangeably in this thesis.

This Chapter contains a two-part case study. In the first part, similar powertrain component (engine, generator, traction motor, etc.) sizes as those of the Chevrolet Volt are used, except for the battery whose size will be determined by the optimizer. In the second part of the case study, the electrical energy storage (EES) system of the series hybrid electric vehicle will be modified to include an ultracapacitor, while the optimization of the vehicle system will be extended to include the power distributing function (PDF) of the EES system and the rest of the powertrain components. In both cases, the same drive cycle and similar objective functions will be applied.

The stated achievable electric range of the Volt is based on 64km of standard city cycle, which may not accurately reflect the actual driving behavior of most commuters. It is therefore decided that the drive cycle for the design optimization will include both city and highway drive cycles for a total of 64km. Furthermore, it is assumed that the EES system is fully charged at the beginning of each commuting day, and that at the end of the commute the user returns home and fully recharges the EES using the household electrical outlet. The goal of the optimizer is minimizing the financial amount that the consumer will be paying, consisting of the initial cost of the EES system (battery and/or ultracapacitor pack), and the combined cost of purchasing gasoline and household electricity to recharge the battery over a period of five years. Due to the lack of information, the maintenance cost of the EES system and the power components is not being considered. Based on the life cycle of lithium-ion batteries [42] and ultracapacitors [43], it is assumed that the EES system pack will last for a period of five years without significant degradation in performance, and therefore a five-year period is used in the objective function of the optimizer. Finally, carbon tax is added to the total cost as a social impact penalty function to take into account the effect of harmful emissions resulting from burning fossil fuels.

5.2 Software Model

The MATLAB/Simulink vehicle model utilizes a backward-looking architecture as described in Chapter 3. The vehicle model consists of a series electric hybrid powertrain architecture, where the electric motor is the sole tractive power source, and the internal combustion (IC) engine is connected to a generator to generate electric power to either charge the battery or to supplement the tractive motor. In addition, the electric motor also acts as a generator during regenerative braking to recapture the otherwise lost vehicle kinetic energy. During a typical simulation, the electric motor delivers the required tractive power to achieve the drive cycle using electrical power primarily from the battery. Once the battery state of charge (SOC) falls below a preset value, the IC engine will drive the generator to recharge the Electric Energy Storage (EES) devices, and in the case where the electric motor requires more electric power than the battery can supply, the genset will also supply the additional required electric power. Figure 5-1 illustrates the overall schematic of the series hybrid electric vehicle model with the optimizer.

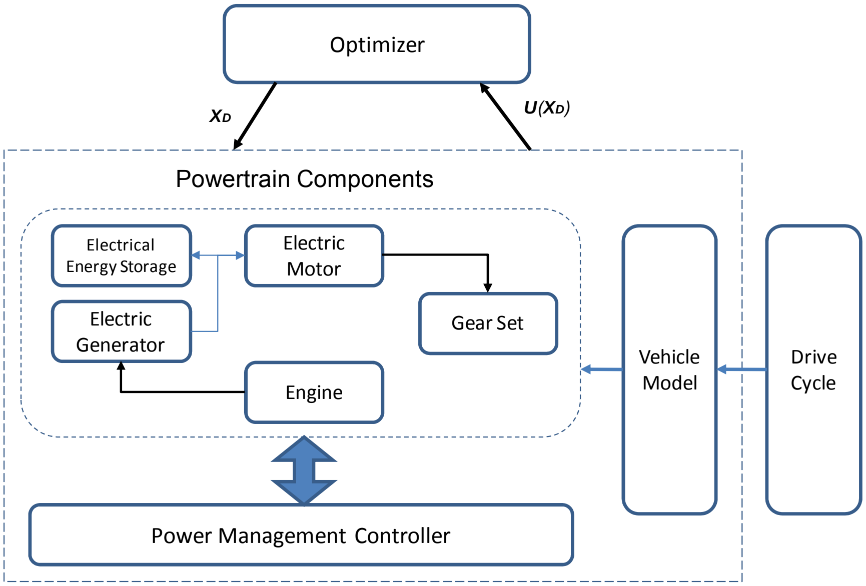


Figure 5-1: Series Hybrid Electric Vehicle Overall Schematic with Optimizer

In Figure 5-1, $\boldsymbol{X_D}$ represents a vector containing the design variables, while $\boldsymbol{U(X_D)}$ is the output variable vector from the vehicle model for the optimizer to evaluate the objective function. The modeling details of each of the vehicle components are discussed in detail in the subsequent subsections.

5.2.1 Drive Cycle

The purpose of the drive cycle for the vehicle model optimization is to simulate the driving speed profile of a typical commuting day. The drive cycle is designed to represent a combined city and highway drive cycle with a total distance of approximately 64km derived from the new test methodology developed by the US Environmental Protection Agency (EPA), introduced in Appendix B. Due to the fact that the vehicle model does not take into consideration the thermal effects of the powertrain components or the presence of air conditioning, it is not possible to simulate the SC03 and cold FTP 75 cycles. It was therefore decided that the drive cycle consists of the US06 cycle in addition to FTP 75 and HWFET, to include high-speed driving and aggressive acceleration and deceleration. Figure 5-2 depicts the speed profile of the combined drive cycle.

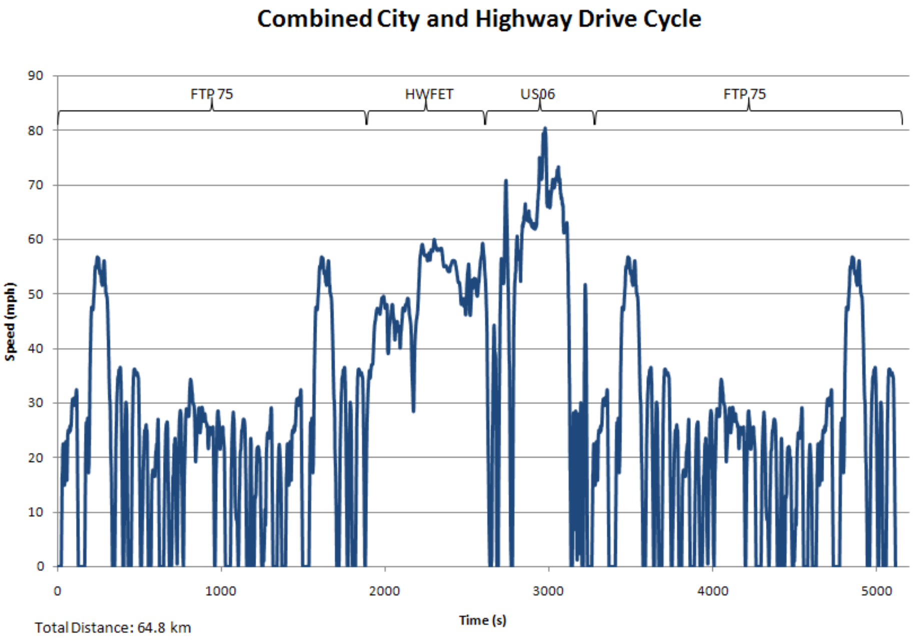


Figure 5-2: Combined City and Highway Drive Cycle

5.2.2 Vehicle Model

The vehicle model utilizes the backward-looking method as described in Subsection 3.1.2, where the desired power is calculated using the speed and acceleration from the drive cycle, while taking the aerodynamic drag and tire rolling resistance into consideration. It should be noted that the base chassis (sprung) mass does not contain the final total mass of the electrical energy storage (EES) devices. Since the size of the EES system is an optimization parameter, the mass will vary according to the optimization results. The final vehicle mass for desired power calculations is therefore the sum of chassis, tire, and EES system masses. The vehicle model parameters are given in Appendix C.

5.2.3 Powertrain Components

5.2.3.1 Transmission

As previously mentioned in Subsection 3.2.4, due to the difference between the locations of the maximum torque regions of an IC engine and the electric motor, the transmission will differ if an IC engine or an electric motor is the primary power source to the wheel. In the case of a series hybrid electric vehicle, where the electric motor is the sole power source, a transmission with a small number of gears will suffice. In this case study, a transmission with 3 gear ratios is selected, and the corresponding operating vehicle speed range is given in Appendix C.

5.2.3.2 Motor

The tractive motor in a series hybrid electric vehicle is the sole power source to the wheels, and also acts as a generator during regenerative braking to capture the otherwise lost vehicle kinetic energy. The motor is created with the same size as in the Chevrolet Volt, i.e., 112kW (150hp) electric motor with a maximum torque of 370Nm [44]. However, since there is no technical information available on the actual motor, the scalable electric motor described in Subsection 3.2.2 was modified to achieve output power and torque similar to that of the Volt. Using the scalable lookup table of the electric motor from the QSS toolbox [33] and applying a scale factor of 7 to the torque index, an electric motor with maximum torque of 375Nm and peak power of 112kW was created.

5.2.3.3 Genset

The genset includes an engine and a generator, which is used to recharge the battery and to supply electrical power to the motor when necessary. In such a configuration, the sole purpose of the IC engine is to drive the generator to produce electricity, and therefore can be operated in its most efficient region. Similar to the electric motor, the generator is modeled using the scalable look-up table described in Subsection 3.2.3 with a scaling factor of 3 to the torque index, yielding a maximum torque of 161Nm and a peak power of 48kW. An engine similar to the GM's family 0 1.4L engine is created using the Willans line modeling approach as illustrated in Subsection 3.2.1, where the parameters of the engine cylinders are defined. Again, due to the lack of technical information on the genset components, the genset model can only be created to have specifications as close as possible to those of the Volt's [44].

Due to the difference in nature between the engine and the electric generator, the high efficiency regions of the two units lie in different locations. One solution is to use a gear set such that both units can be operated in their respective peak efficiency regions. Figure 5-3 illustrates the maximum torque and peak efficiency of the engine and the generator without the added gear set.

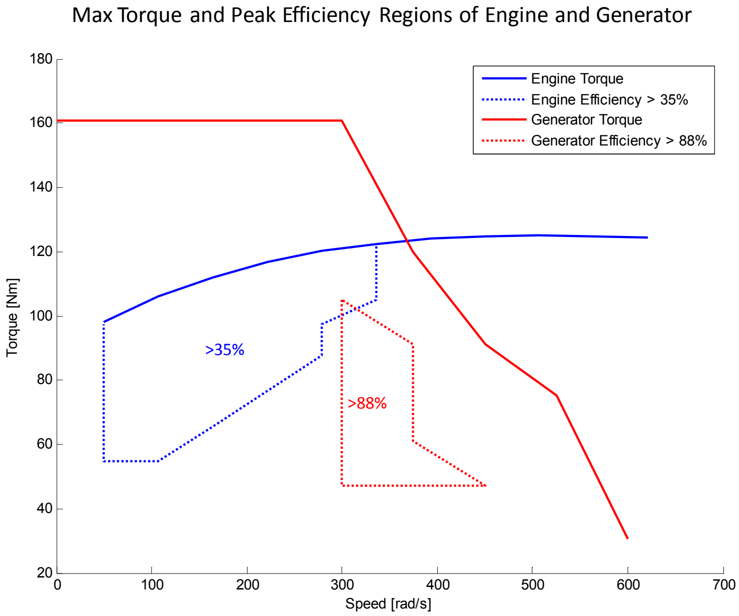


Figure 5-3: Maximum Torque and Peak Efficiency Regions of the Engine and the Generator

It can be seen from Figure 5-3 that there is minimal overlapping between the peak efficiency regions of the two components. By adding a gear set to increase the engine speed while decreasing the engine torque, it is possible to overlay the peak efficiency of both units. Figure 5-4 depicts the torque and efficiency curves of the engine and generator after a gear set with the ratio of 1.35 is added, as well as the combined peak efficiency region of the genset for optimization.

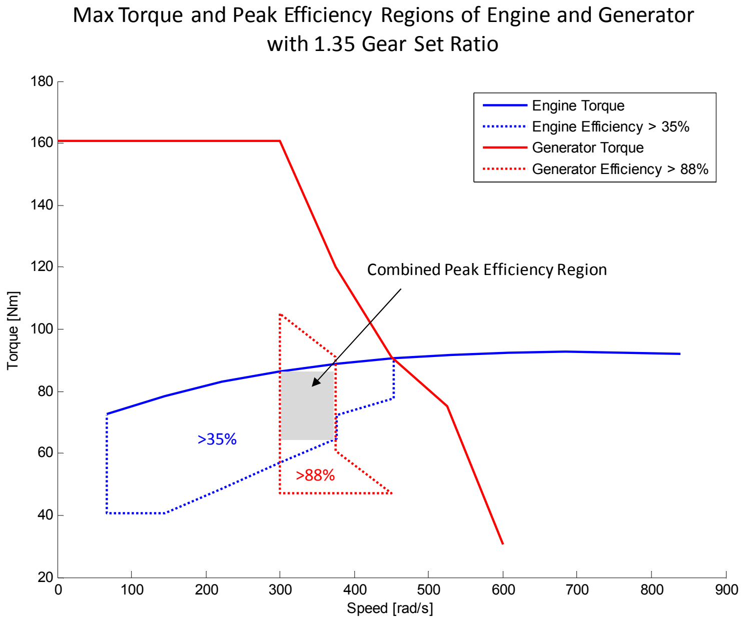


Figure 5-4: Maximum Torque and Peak Efficiency Regions of the Genset with Gear Set Ratio

5.2.3.4 Battery

The battery component used in the series hybrid electric vehicle model is based on the A123System's lithium-ion batteries where each cell has a capacity of 2.3Ah and a nominal voltage of 3.3V [45]. Due to the large capacity of the battery that the vehicle requires, a parallel-series combination of battery cells is used to satisfy the required capacity of the vehicle, while maintaining a reasonable terminal voltage for the system. It was assumed that each battery bank will contain 100 battery cells in series, producing a nominal voltage of 330V. When a larger battery capacity is necessary, additional battery banks will be added in parallel to increase the overall capacity while maintaining the nominal voltage; thus, the number of battery banks will be used as an optimization variable to size the battery. In addition to affecting the overall capacity of the battery, the number of battery banks also changes the charging and discharging power limits of the battery system.

5.2.3.5 Ultracapacitor

The ultracapacitor utilized in the series hybrid electric vehicle model is the Maxwell Boostcap 16 Volt Modules with a rated capacitance of 110F [43]. The rational of selecting the 16 volt module is that its maximum instantaneous power (80kW) is similar to and slightly higher than the approximate maximum power requirement (76kW) of the drive cycle. When a larger capacitance is required, additional ultracapacitors will be added in parallel to increase the overall capacitance while maintaining the nominal voltage; thus, the number of ultracapacitors in parallel will be used as an optimization variable to size the ultracapacitor pack. Since the vehicle modeling approach is based on a backward-looking model, where only power consumption and generation are of interest, the voltage difference between the battery and the ultracapacitor is not being considered. In reality, a DC-DC converter is required to interface the two systems with different voltage levels. However, since the dynamic behaviour and the power electronics of the electrical components are not within the scope of this research, the powertrain model only considers the power transfer between the battery-ultracapacitor combination and the electric motor and generator. Finally, it is assumed that the efficiency of the DC-DC converter is close to ideal; hence, no power losses were modeled for the DC/DC conversion.

It should be noted that currently the cost of an ultracapacitor is considerably higher than that of the battery. However, due to its high power output characteristic, its inclusion is desired to improve the performance during acceleration and efficiency during regenerative braking. With the continuing decrease in the cost of ultracapacitors, there is no doubt that they will play an increasingly important role in the design of electric vehicles' electric energy storage systems in the future.

5.2.4 Power Management Logic

The role of the power management logic is to activate the genset to charge the battery when required. Additionally, the power management logic controls the operating point of the engine, specifically the engine speed and torque, and the total output power of the genset. When the battery state of charge (SOC) falls below a preset threshold value, the power management logic activates the genset to charge the battery until the SOC reaches a predetermined percentage increase before being turned off. This is to avoid having the engine turn on and off frequently when the battery SOC remains around the threshold value, and also reflects real world operation of a genset. Ideally, it is desired that the genset operates in the most efficient region as described in the previous section. However, in the case where the vehicle is performing regenerative braking, with the addition of the power output of the genset at its peak efficiency, the total charging power may be higher than the charging limit of the battery. To ensure that the battery is not overcharged, it is desired that the genset's output power be regulated accordingly. The following rules describe the genset output power (P_{gen}) during various modes of vehicle operation.

$$P_{gen} = \begin{cases} a_{des} \geq 0, & min\left(P_{gen,eff}\;,\; P_{m,req} + P_{batt,chg}\right) \\ a_{des} < 0, & min\left(P_{gen,eff}\;,\; P_{batt,chg} - P_{regen}\right) \end{cases} \tag{5.1}$$

where a_{des} = desired acceleration of the vehicle

$P_{gen,eff}$ = genset power output determined by the optimizer

$P_{m,req}$ = electric power required by the motor

P_{regen} = electric power generated by the motor during regenerative braking

$P_{batt,chg}$ = battery charging power limit

Equation (5.1) depicts that when the vehicle is cruising or accelerating, the genset output power is the sum of the battery's charging power limit and the required discharge power to the traction motor, up to the genset output power determined by the optimizer. Similarly, during deceleration, the genset power output is the difference between the battery's charging power limit and the regenerated electric power from the motor, up to the optimizer-determined output power. This ensures that the genset operates in the most efficient point while not over charging the battery.

5.3 Battery-Only Series Hybrid Electric Vehicle

The battery-only hybrid electric vehicle's architecture is identical to that of the Chevrolet Volt, where the electrical energy storage (EES) system consists of only the battery pack. In addition, the IC engine, generator, and traction motor are in the same way as the corresponding components in the Volt. The optimization effort will focus on determining the battery size and the power management logic. The battery-only series hybrid electric vehicle's Simulink model is given in Appendix D.

5.3.1 Optimization Problem

The optimization procedure employs the optimizer to determine the battery size and the power management logic to obtain the minimal objective function. The optimization problem is

$$\text{Minimize } J(\boldsymbol{X_D}, \boldsymbol{U}(\boldsymbol{X_D})) \text{ w.r.t. } \boldsymbol{X_D}$$
$$\text{Subject to } c(\boldsymbol{X_D})$$

where the optimization parameters ($\boldsymbol{X_D}$) are the SOC threshold, engine torque, and engine speed for the power management logic, and the number of battery banks to determine the size of the battery system. $\boldsymbol{U}(\boldsymbol{X_D})$ are the fuel and the electricity consumption of the vehicle, while $c(\boldsymbol{X_D})$ are the equality and inequality constraints of the problem. The Simulink vehicle model as shown in Appendix D defines the number of equations and the constraints of the optimization problem, which is a mixed integer nonlinear problem in nature. Finally, the size of the optimization problem reported in Table 5-1 is the number of variables at each time step multiplied by the number of time steps that the drive cycle is divided into and is calculated as

$$OptimizationSize = \frac{Variable_{number} \times T}{t_s} \tag{5.2}$$

where T is the total time of the drive cycle, and t_s is time step used during simulation.

Table 5-1: Summary of the Battery-Only Series Hybrid Electric Vehicle Optimization Problem

Drive Cycle Duration [s]	5,114
Time Step [s]	0.5
Number of Variables	15
Size of Optimization Problem	153,420
Number of Constraints ($c(\boldsymbol{X_D})$)	8

Additionally, Table 5-2 depicts the numerical range of the optimization parameters ($\boldsymbol{X_D}$), where the engine torque and speed ranges are determined from the most efficient region of the genset as described in Subsection 5.2.3.3.

Table 5-2: Battery-Only Series Hybrid Electric Vehicle Optimization Variables

Variables	Lower Bound	Upper Bound
SOC Threshold [%]	10	95
Engine Torque [Nm]	88	116
Engine Speed [rad/s]	222	278
Number of Battery Banks	1	9

The objective function (J) is the total financial amount consisting of the initial cost of the battery, the cost of gasoline and household electricity consumption over a period of five years, and the equivalent carbon tax cost due to fossil fuel consumption, as given by the following equation.

$$J = \left[\sum_{t=0}^{T} Fuel_{Consumed} \times (Fuel_{Cost} + Carbon_{\ Tax}) + \sum_{t=0}^{T} Electricity_{Consumed} \times Electricity_{Cost}\right] \times Days \times Years + Battery_{Cell} \times Battery_{Cost} + Error\_Delayed \tag{5.3}$$

where T is the total time of the drive cycle. The costs of fuel and electricity were determined based on the typical gasoline and household hydro cost in Ontario in 2008. The cost of battery includes the battery cell itself and packaging cost determined from the A123 systems' website [45]. The number of days is based on a 260-day work year as reported from statistic Canada [46], while the carbon tax is the average expert consensus indicated in the climate change 2007 report by the Intergovernmental Panel on Climate Change (IPCC) [47]. It should be noted that due to the wide variety of electricity generation methods (i.e., coal, hydro, wind farm, nuclear) [48], it is difficult to affix a precise carbon tax cost to the generation of electricity, and this item is therefore not included in the objective function evaluation. Finally, an error function is included in the objective function evaluation for the case when the vehicle cannot complete the drive cycle without a time delay; therefore, the optimizer will only consider the vehicle settings where powertrain is capable of delivering the required power of the drive cycle. Table 5-3 summarizes the values of the cost function parameters used in the optimization.

Table 5-3: Parameters for the Battery SHEV Objective Function Evaluation

Cost Function Parameters	Values
Cost of Gasoline [$/L]	0.9
Cost of Electricity [$/kW·h]	0.12
Carbon Tax [$/L of Gasoline]	0.034
Cost of A123 Li-Ion Battery [$/cell]	18.33
Number of Working Days per Year	260
Number of Years	5

5.3.2 Optimization Results

Optimization on the battery-only series hybrid electric vehicle was performed using the procedure described in Subsection 4.2.3 along with the algorithms mentioned in Subsection 4.2.4. The algorithms' MATLAB commands were used for the optimization procedure, and the algorithm specific parameters were left at their default values. The m-files containing the MATLAB built-in optimization commands are included in Appendix E. Additionally, feature-based optimization was also performed on the battery-only series hybrid electric vehicle, and its MATLAB code is given in Appendix F. The optimization results of various algorithms and the feature-based optimization are summarized in Table 5-4.

Table 5-4: Battery SHEV Optimization Results of Various Algorithms

	Full Optimization				**Feature-Based using GA**
Algorithm	**Genetic Algorithm**	**Simulated Annealing**	**Pattern Search**	**Nelder-Mead**	
SOC Threshold [%]	12	15	12	7	11
Engine Torque [Nm]	116	116	114	170	116
Engine Speed [rad/s]	225	222	222	164	222
Number of Battery Banks	3	3	3	3	3
Gasoline Consumption [L/day]	2.277	2.279	2.280	2.079	2.273
Electricity Consumption [kWh/day]	1.961	1.885	1.965	2.071	1.980
Battery Cost [$]	5,499	5,499	5,499	5,499	5,499
Fuel Cost [$]	2,664	2,666	2,667	2,433	2,660
Electricity Cost [$]	285	274	286	302	288
Carbon Tax [$]	101	101	101	92	100
Total Cost [$]	**8,549**	**8,540**	**8,553**	**8,326**	**8,547**
Simulation run-time [s]	*6,147*	*13,165*	*701*	*575*	*467*

Table 5-4 shows that out of the four MATLAB optimization algorithms, the solution found by the Nelder-Mead method is out of bound, and therefore is deemed meaningless. Genetic algorithm, simulated annealing, and pattern search all found solutions that yielded approximately the same cost function, where simulated annealing resulted in the lowest cost and the total cost found by pattern search was the largest. However, it took simulated annealing the longest time to reach the solution, while pattern search utilized the least amount of time. This is not surprising since pattern search is a not a global optimization method in nature, and therefore require less time to reach the solution. Furthermore, it should be noted that pattern search method was able to reach the solution regardless of the starting point. Finally, the result found by the feature-based optimization was in between that of genetic algorithm and simulated annealing, demonstrating the accuracy of the feature-based optimization approach. Additionally, it took feature-based optimization significantly less simulation time to reach the solution, further confirming the validity of the proposed methodology.

5.3.3 Design Study

A design study was conducted on various vehicle settings, by varying the size of the battery and the SOC threshold to investigate their effects on gasoline and electricity consumptions, and validate that the optimized parameters indeed yield the optimal results. A series of simulations were conducted with the battery size varying from 1 to 22 battery banks, while the entire range of the SOC threshold (0-100%) was swept at the resolution of 5%. Throughout the simulation, the engine was assumed to be operating at 222rad/s and 116Nm. Figure 5-5 shows the gasoline consumption versus the number of battery banks and SOC threshold.

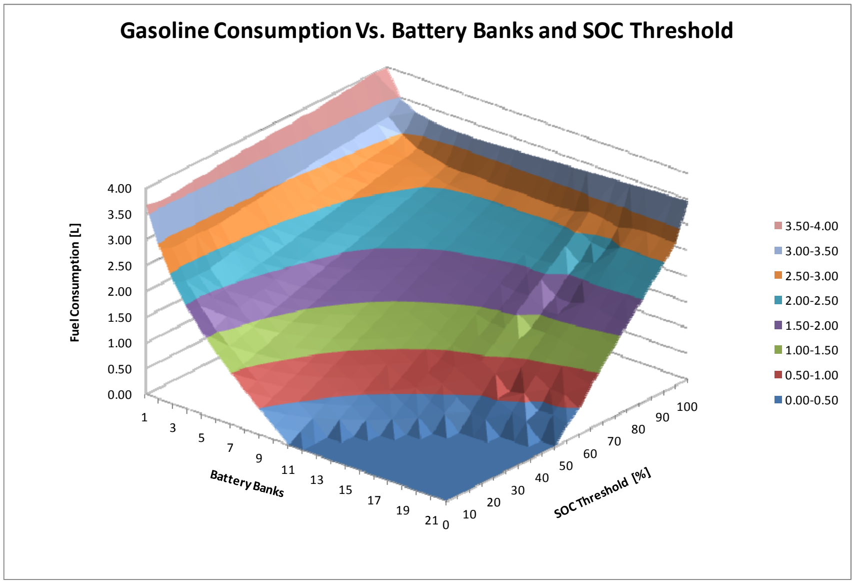


Figure 5-5: Gasoline Consumption versus Number of Battery Banks and SOC Threshold

It can be seen from Figure 5-5 that, as expected, the gasoline consumption of the engine increases with a higher SOC threshold setting. Furthermore, lowering the capacity of the battery will result in an increase in the gasoline consumption due to the genset supplying more power to the traction motor to complete the drive cycle. Figure 5-5 also indicates the region of vehicle parameter settings where pure electric mode can achieve the combined drive cycle without consuming any gasoline. The electricity consumption over varying battery banks and SOC threshold is illustrated in Figure 5-6.

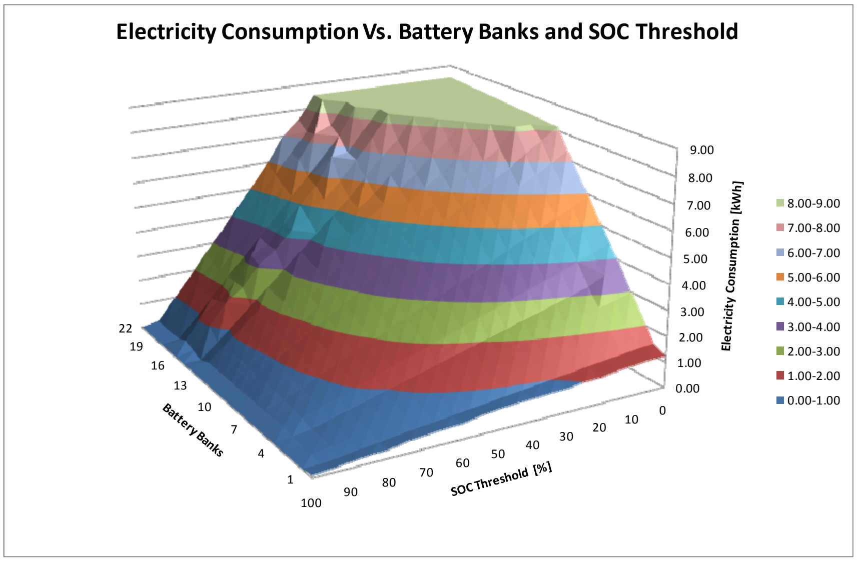
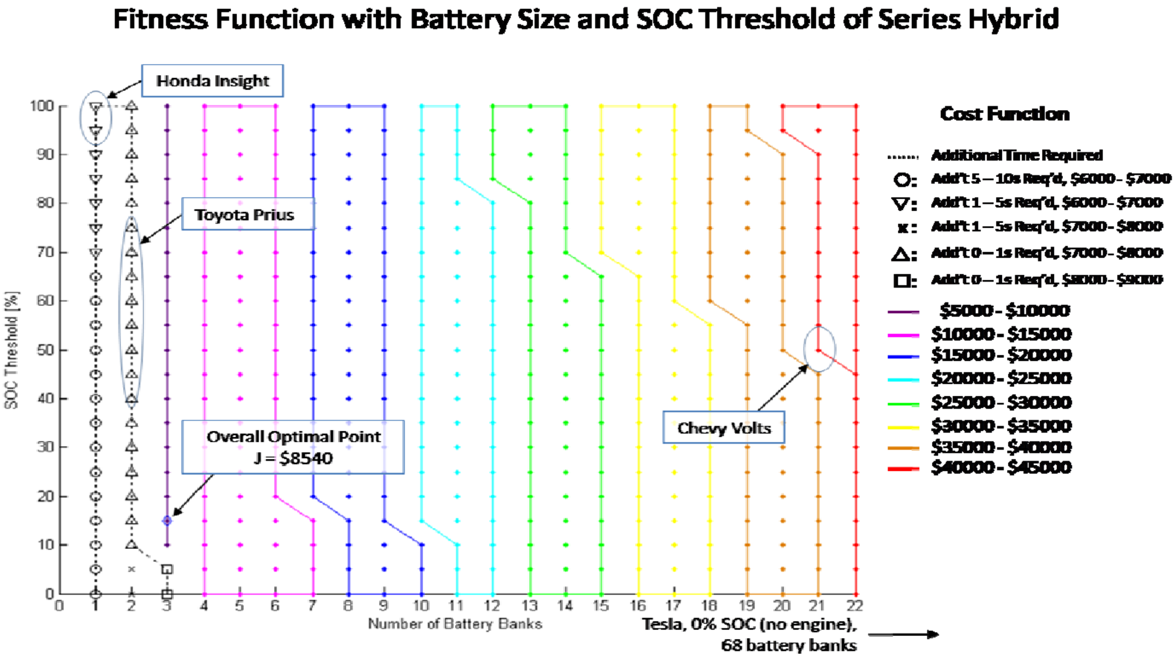


Figure 5-6: Electricity Consumption with Varying Battery Banks and SOC Threshold

As expected, with less gasoline consumed, the more electricity is required for the vehicle to complete the cycle. In addition to the energy consumptions, the effects of varying the battery banks and SOC threshold on the final cost function are shown in Figure 5-7.



Note: Prius and Insight are parallel hybrids with more powerful engines (98 hp)

Figure 5-7: Battery-Only SHEV Cost Function with Battery Banks and SOC Threshold

It can be seen from Figure 5-7 that the most prominent factor affecting the final cost function was the number of battery banks, which is in consensus with the industry that the cost of battery remains the main obstacle in popularizing hybrid electric vehicles on the consumer market. The optimized vehicle configuration as determined by the simulated annealing along with its cost function value was shown in a blue diamond in Figure 5-7. It was also found that the cost function monotonically decreases towards the optimal point; hence, the solutions found by the local optimization will be fairly close to that of the global optimization techniques, as indicated in Table 5-6. In addition to indicating the cost function for each combination of the battery size and SOC threshold setting, the parameters for which the powertrain cannot complete the drive cycle without a time delay are shown in Figure 5-7. It was found that the smaller the battery pack, the larger the amount of additional time required to complete the cycle. Since the battery banks are arranged in parallel, the maximum discharge power decreases as the number of battery banks is reduced, thereby increasing the amount of time required to complete the drive cycle. Finally, based on the published battery size and a guesstimate of the engine operating range, the configurations of the Honda Insight, Toyota Prius, Chevrolet Volt, and Tesla Roadster are also indicated in Figure 5-7. It is worth noting that the location of each of the aforementioned vehicles on the cost function map reflects the order of the list price on the market, where the Honda Insight is marketed as the most affordable hybrid electric vehicle available, and the Tesla Roadster is branded as an ultra premium roadster.

5.4 Combined Battery and Ultracapacitor Series Hybrid Electric Vehicle

To further demonstrate the concurrent optimization approach, the battery-only series hybrid electric vehicle model in the previous section is modified to include the ultracapacitor in the electrical energy storage (EES) system. In order to distribute the charging and discharging power between the batteries and the ultracapacitors, the power distributing function (PDF) described in Section 3.3 is implemented as part of the power management controller logic. Furthermore, the sizing of the powertrain has been extended to include the rest of the power components. The following subsections present the optimization parameters and results of the combined vehicle model. The Simulink models of the ultracapacitor and the power distributing function are given in Appendix G.

5.4.1 Optimization Problem

The optimization problem for the combined vehicle model is formulated as:

$$\text{Minimize } J(\boldsymbol{X_D}, \boldsymbol{U}(\boldsymbol{X_D})) \text{ w.r.t. } \boldsymbol{X_D}$$
$$\text{Subject to } c(\boldsymbol{X_D})$$

where the optimization parameters ($\boldsymbol{X_D}$) are the sizes of the powertrain components such the EES system including battery and ultracapacitor, the IC engine, and the traction motor, along with the power management logic of the genset and the power distributing function (PDF) of the EES system. $\boldsymbol{U}(\boldsymbol{X_D})$ are again the fuel and the electricity consumption of the vehicle, while $c(\boldsymbol{X_D})$ are the equality and inequality constraints of the problem. The Simulink vehicle model as shown in Appendix G defines the number of equations and the constraints of the optimization problem, which is a mixed integer nonlinear problem in nature. Table 5-5 summarizes the size of the optimization problem as determined by Equation (5.2), while Table 5-6 describes the upper and lower bounds of the optimization parameters used in the combined optimization.

Table 5-5: Summary of the Combined Series Hybrid Electric Vehicle Optimization Problem

Drive Cycle Duration [s]	5,114
Time Step [s]	0.5
Number of Variables	19
Size of Optimization Problem	194,332
Number of Constraints ($c(\boldsymbol{X_D})$)	14

Table 5-6: Optimization Variables of the Combined Series Hybrid Electric Vehicle

Variables	Lower Bound	Upper Bound
SOC Threshold [%]	10	95
Genset Active Duration [%]	5	25
Engine Torque [Nm]	88	116
Engine Speed [rad/s]	222	278
Number of Battery Banks	1	9
Number of Ultra-capacitors	1	9
Engine Stroke [mm]	50	120
Traction Motor Scale	5	14
Power Distributing Function		
Discharge $SOC_{batt}(t)$ (c_{dis}) [%]	1	150
Discharge $SOC_{UC}(t)$ (e_{dis}) [%]	1	150
Discharge $dP_{des}(t)/dt$ (i_{dis}) [kW/s]	1	150
Charge $SOC_{batt}(t)$ (c_{ch}) [%]	1	150
Charge $SOC_{UC}(t)$ (e_{ch}) [%]	1	150
Charge $dP_{des}(t)/dt$ (i_{ch}) [kW/s]	1	150

In addition to determining the SOC threshold to turn on the genset, the power management logic includes the duration of time that the genset remains activated. The engine sizing is achieved by changing the engine bore, thereby changing the engine displacement. For the purpose of a series hybrid electric vehicle, it is decided that the engine will remain a four-cylinder engine, since a six- or eight-cylinder engine will be unnecessarily large to be used in a genset. The range of the engine bore is determined by a quick survey of the existing 4-cylinder engines, where the lower limit corresponds to a large motorcycle engine, and the upper limit is a typical 2.0L car engine. The scaling range of the traction motor corresponds to a minimum power of 80kW and a maximum of 224kW electric motor, where the minimum size is to satisfy the power requirement of the drive cycle and the maximum size is similar to that of the Tesla roadster [49]. Finally, for the power distributing function (PDF), it is decided to optimize only the sigmoid function's inflection point of each of the monitored variables, since the inflection point affects the behaviour of the PDF much more significantly than the slope. It should be noted that the inflection points of the desired power ($P_{des}(t)$) described in Subsection 3.3.1 corresponds to the charging and discharging limits of the battery; therefore, the parameters g_{dis} in Equation (3.30) and g_{ch} in Equation (3.31) are not part of the optimization variables.

Similar to the battery-only series hybrid electric vehicle, the objective function (J) includes the total financial amount of the initial cost of the EES system (battery and the ultracapacitor), the cost of gasoline and household electricity consumptions over a period of five years, and the equivalent carbon tax cost due to fossil fuel consumption. Furthermore, cost functions are included to take into account the effects of the IC engine and traction motor sizing, and the objective function (*J*) is given as follows.

$$J = \left[\sum_{t=0}^{T} Fuel_{Consumed} \times (Fuel_{Cost} + Carbon\ _{Tax}) + \sum_{t=0}^{T} Electricity_{Consumed} \times Electricity_{Cost}\right] \times Days \times Years + Battery_{Cell} \times Battery_{Cost} + UC_{Unit} \times UC_{Cost} + ICE_{cost} + Motor_{cost} + Error\_Delayed \tag{5.4}$$

The costs of fuel and electricity were again determined based on the typical gasoline and household hydro cost in Ontario in 2008, and the same objective function parameters described in Subsection 5.3.1 were used. The cost of the ultracapacitor is obtained through a retailer [50] which offers a pre-packaged unit. On the other hand, the costs of the IC engine and traction motor were not readily available from the manufacturers; thus, an interpolation function is utilized to capture the effects of their sizing. The following equation describes the interpolation function of the IC engine.

$$ICE_{cost} = ICE_{base} + (S - S_{lb}) \times Cost_{inc} \tag{5.5}$$

where ICE_{base} is the base cost of the IC engine at the minimum engine size corresponding to the lower bound of the engine stroke (S_{lb}), and *S* is the optimization variable determined by the optimizer during each iteration. $Cost_{inc}$ is a constant value that interpolates the cost increase of the unit corresponding to the unit size. Similarly, the interpolation function of the traction motor is given as follows.

$$Motor_{cost} = Motor_{base} + \left(EM_{scale} - EM_{scale,lb}\right) \times Cost_{inc} \tag{5.6}$$

Again, $Motor_{base}$ is the base cost of the traction motor at the minimum motor size corresponding to the lower bound of the traction motor's scaling factor ($EM_{scale,lb}$), while EM_{scale} is the optimization variable determined by the optimizer during each iteration. Due to the lack of manufacturer's cost information for the electric motor, the same $Cost_{inc}$ used for the IC engine is used for the traction motor. The cost values of ICE_{base}, $Motor_{base}$, and $Cost_{inc}$ are assumed for this case study, and can be updated if information from the manufacturers is available. Finally, an error function is included in the objective function evaluation for the case when the vehicle cannot complete the drive cycle without a time delay; thus, the optimizer will only consider the vehicle settings where the powertrain is capable of delivering the required power of the drive cycle. Table 5-7 summarizes the values of the objective function parameters used in the optimization.

Table 5-7: Parameters for the Combined SHEV Objective Function Evaluation

Cost Function Parameters	Values
Cost of Gasoline [$/L]	0.9
Cost of Electricity [$/kW·h]	0.12
Carbon Tax [$/L of Gasoline]	0.034
Cost of A123 Li-Ion Battery [$/cell]	18.33
Cost of Maxwell Ultra-capacitor [$/unit]	694.29
Cost of Base Engine [$]	2,000
Cost of Base Traction Motor [$]	5,000
Cost Increase [$/unit_increment]	28.57
Number of Working Days per Year	260
Number of Years	5

5.4.2 Optimization Results

Optimization on the combined series hybrid electric vehicle was again performed using the procedure described in Subsection 4.2.3 with the algorithms mentioned in Subsection 4.2.4, along with the feature-based optimization. Based on the results shown in the previous Section, Nelder-Mead algorithm was not able to find any meaningful solution, and was therefore not included to perform optimization for the combined vehicle model. Table 5-8 summarizes the optimization results of various algorithms and the feature-based optimization.

Table 5-8: Combined SHEV Optimization Results of Various Algorithms

	Full Optimization			**Feature-Based using GA**
Algorithm	**Genetic Algorithm**	**Simulated Annealing**	**Pattern Search**	
SOC Threshold [%]	41	85	60	44
Genset Active Duration [%]	5	23	19	5
Engine Torque [Nm]	114	110	88	115
Engine Speed [rad/s]	224	226	243	251
Number of Battery Banks	2	2	2	2
Number of Ultra-capacitor	1	1	1	2
Engine Stroke [mm]	51	51	59	53
Traction Motor Scale	6	6	5	5
Power Distributing Function				
Discharge $SOC_{batt}(t)$ (c_{dis}) [%]	9.68	103.5	106.68	92.51
Discharge $SOC_{UC}(t)$ (e_{dis}) [%]	107.68	7.5	87.02	47.15
Discharge $dP_{des}(t)/dt$ (i_{dis}) [kW/s]	116.19	149.5	132.44	n/a
Charge $SOC_{batt}(t)$ (c_{ch}) [%]	130.83	107.5	121.37	60.23
Charge $SOC_{UC}(t)$ (e_{ch}) [%]	108.57	75.5	100.00	23.55
Charge $dP_{des}(t)/dt$ (i_{ch}) [kW/s]	144.09	147.5	113.39	n/a
Incomplete Time Delay [s]	**0**	**23.61**	**16.01**	**14.18**
Simulation run-time [s]	*25,576*	*99,684*[1]	*122,875*[1]	*621*

Table 5-8 describes the values of the optimization variables found by the optimizer, the time delay to complete the drive cycle, and the simulation run-time required by the optimizer. A time delay value greater than zero indicated that the solution found by the optimizer cannot complete the drive cycle, since appropriate PDF parameters are required to ensure proper power management of the EES components. Table 5-8 shows that only genetic algorithm was able to find a solution without any time delay. Furthermore, both the simulated annealing and the pattern search optimizations were terminated manually, since both processes took over 24 hours and were still not able to converge to a solution. This further demonstrates that out of the available optimization algorithms in the MATLAB library, genetic algorithm is the most suitable method to perform concurrent optimization of the series hybrid electric vehicle. Finally, due to the nature of the feature-based simulation which lacks the time history information of the drive cycle, it is not possible to obtain the rate change of desired power ($dP_{des}(t)/dt$); therefore, the PDF used during the feature-based optimization does not contain the sigmoid function for $dP_{des}(t)/dt$ and its parameters i_{dis} and i_{ch}. Hence, the lack of time information causes the inaccuracy of the PDF parameters, thereby contributing to the time delay. However, upon

[1] Simulation manually terminated

inspection of the results for feature-based and genetic algorithm, it was found that both algorithms found similar solutions for the vehicle powertrain sizing and genset activation threshold. To further improve the solution of the feature-based method, a secondary optimization using the initially found powertrain parameters was performed to optimize only the power distributing function parameters. Based on the optimization comparison made in Subsection 5.3.2, where simulated annealing did not offer significant improvement over genetic algorithm, it was decided to only use genetic algorithm and pattern search as the optimizers to perform the secondary simulation. Table 5-9 describes the results of the secondary simulation.

Table 5-9: PDF Parameters of the Secondary Optimization

Algorithm	**Genetic Algorithm**	**Pattern Search[2]**
Discharge $SOC_{batt}(t)$ (c_{dis}) [%]	8.4	19.54
Discharge $SOC_{UC}(t)$ (e_{dis}) [%]	140.23	116.19
Discharge $dP_{des}(t)/dt$ (i_{dis}) [kW/s]	107.54	143.15
Charge $SOC_{batt}(t)$ (c_{ch}) [%]	45.58	92.23
Charge $SOC_{UC}(t)$ (e_{ch}) [%]	46.85	23.23
Charge $dP_{des}(t)/dt$ (i_{ch}) [kW/s]	122.37	80.09
Incomplete Time Delay [s]	**0**	**0**
Simulation run-time [s]	*1,967*	*1,666*

Table 5-9 showed that both genetic algorithm and pattern search were able to obtain a solution where the power distributing function was able to properly manage the EES to complete the drive cycle. However, it should be noted that the pattern search method is highly sensitive to its initial conditions, where solutions can only be reached if the initial PDF values first found by feature-based from Table 5-9 are used as the initial values in the secondary optimization. Any other initial values such as random numbers or midpoint of the upper and lower bound will result in an incomplete solution. On the other hand, genetic algorithm did not require an initial condition and was able to find a solution at the first attempt. The simulation results of the solution found by the full genetic algorithm optimization and feature-based with secondary optimization are summarized in Table 5-10.

[2] Solution highly sensitive to the initial condition

Table 5-10: Simulation Results of the Optimized Solution

	GA Full Optimization	**Feature-Based with Secondary GA**	**Feature-Based with Secondary PS**
Gasoline Consumption [L/day]	2.80	3.05	3.05
Electricity Consumption [kWh/day]	0.77	0.72	0.73
Battery Cost [$]	3,666	3,666	3,666
Ultra-capacitor Cost [$]	694.29	1,388.58	1,388.58
Engine Cost [$]	2,028.57	2,085.71	2085.71
Motor Cost [$]	5,028.57	5,000	5,000
Fuel Cost [$]	3,271.95	3,568.72	3,566.04
Electricity Cost [$]	112.25	105.2	105.73
Carbon Tax [$]	123.61	134.82	134.72
Total Cost [$]	**14,925**	**15,949**	**15,947**
Total Simulation run-time [s]	*25,576*	*2,588*	*2,287*

Table 5-10 showed that the results of genetic algorithm's full optimization produced the lowest cost function, while the cost found by feature-based method with secondary optimizations is higher due to the extra ultracapacitor unit. However, the most significant advantage of the feature-based method with secondary optimization is that the combined simulation time was ten times less than that of genetic algorithm full optimization, reducing the required simulation run-time from over 7 hours down to less than 45 minutes. Even though the feature-based optimization approach may not provide the optimal solution, the significant reduction in simulation time warrants the proposed method a powerful tool for vehicle design.

5.4.3 Design Study

The solutions of the full genetic algorithm optimization and feature-based method with secondary optimization are further investigated to examine the behaviour of the electrical energy storage (EES) components and the power distributing function (PDF) during the drive cycle. The simulation results of the genetic algorithm full optimization are first presented, followed by those of the feature-based method with a secondary GA optimization. Figure 5-8 shows the state of charge of the EES components corresponding to the drive cycle from the solution of full genetic algorithm optimization.

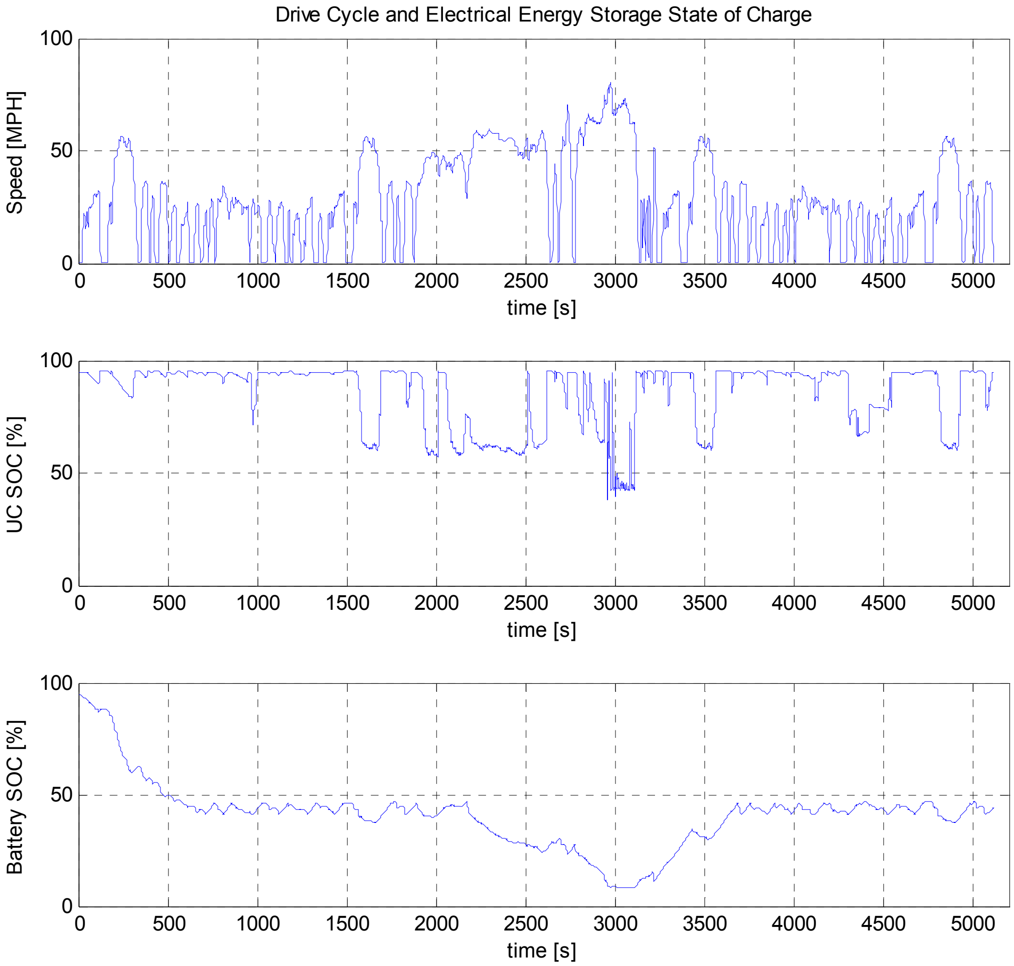


Figure 5-8: Drive Cycle and EES SOC Behaviour of the Full GA Solution

Figure 5-8 showed that the genset was activated to maintain charge while providing traction power once the battery SOC reached the threshold value. Furthermore, during higher acceleration periods, the ultracapacitor provided additional power to achieve the acceleration demand. During the deceleration period, the ultracapacitor was recharged to its upper threshold (95%), indicating the recovering of kinetic energy during regenerative braking. Additionally, it can be seen that during the 2,000 to 3,000 second period, the battery was steadily discharged to provide power in addition to the genset to achieve the higher vehicle speeds. This again confirms that given the current high price of the battery and ultracapacitor, it is still more cost effective to utilize fossil fuel as a primary source of traction energy while minimizing the size of the EES system. To further observe the behaviour of the EES, the period between 2,500 and 3,500 seconds is illustrated in Figure 5-9.

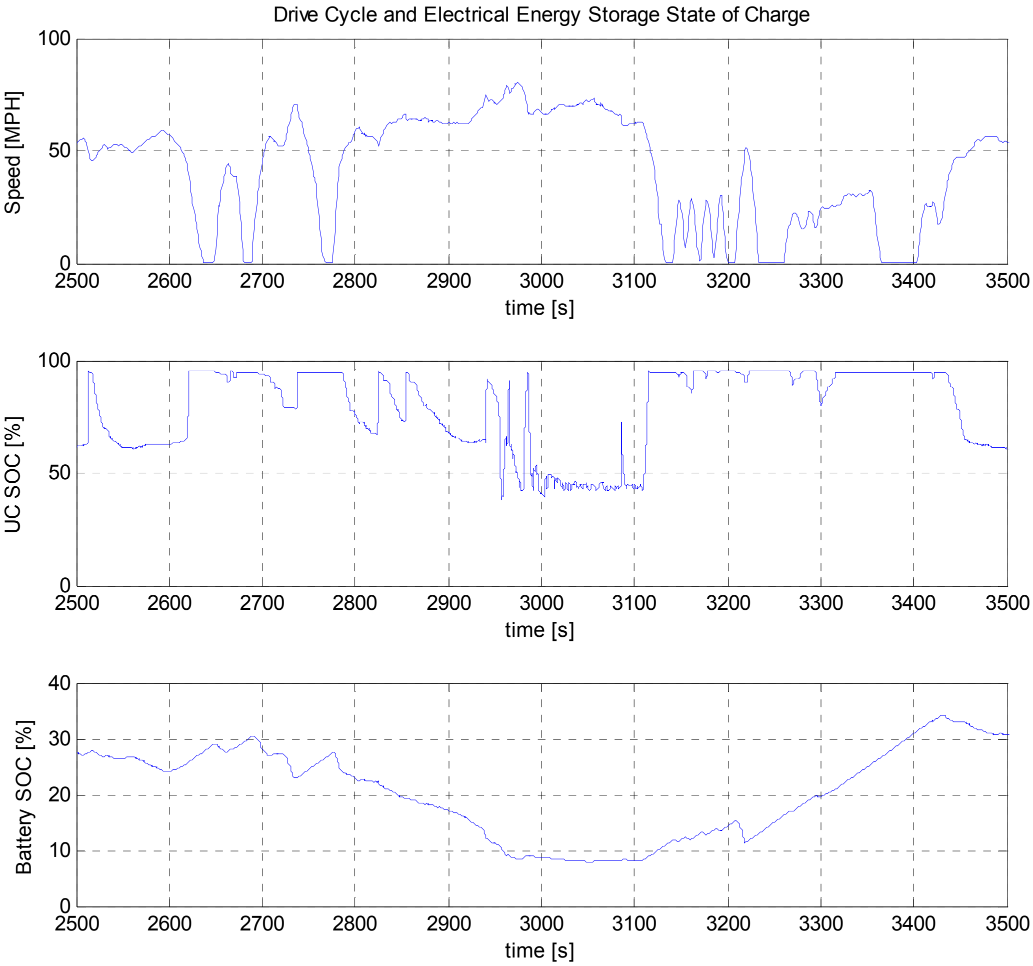


Figure 5-9: Drive Cycle and EES SOC Behaviour of the Full GA Solution [2500-3500s]

Figure 5-9 illustrated that the system behaved correctly during the acceleration and the braking of the vehicle. Upon further inspection, it can be seen that there is a slight lag to the recharging of the ultracapacitor when regenerative braking begins. This is due to the fact that the charging PDF described in Equation (3.31) ensures the charging power is first used to charge the battery, and any additional power beyond the battery's charging limit is subsequently used to charge the ultracapacitor. Similar observation can be made in several instances where the ultracapacitor is only discharged once the required power reaches over the discharging limit of the battery. Finally, the power distributing function and its dependent variables are illustrated in Figure 5-10.

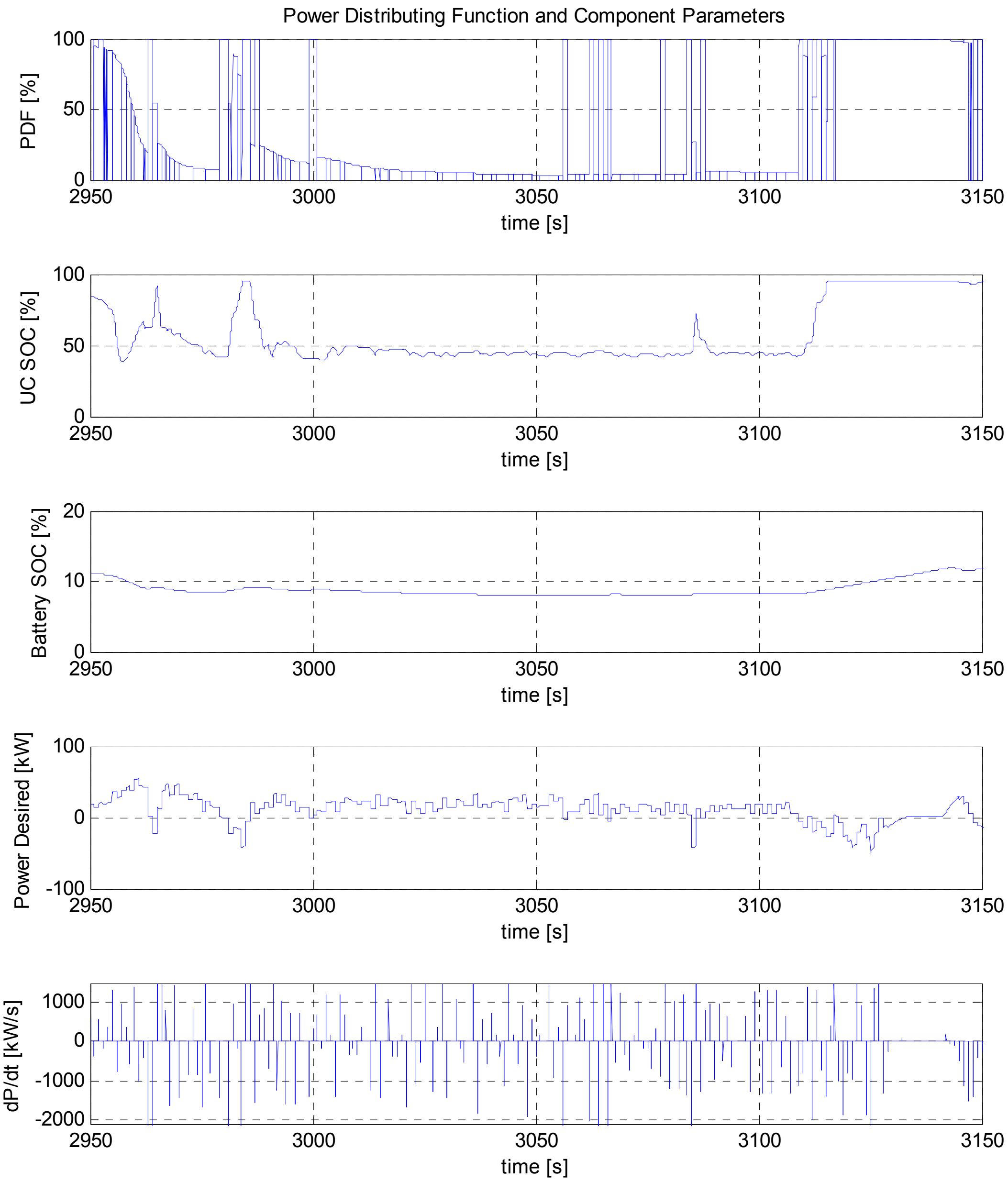


Figure 5-10: Power Distributing Function of Full GA Solution

It is observed from Figure 5-10 that the PDF corresponded to the state of charge of the EES components accurately. When the ultracapacitor is full, the PDF is indicating 100% operation from the battery. During the period when the ultracapacitor's SOC is around 50% and when the battery's SOC is low, the PDF is operating primarily from the ultracapacitor. Additionally, the desired power and the rate change of power were also affecting the PDF, where higher $P_{des}(t)$ and $dP_{des}(t)/dt$

indicated operation primarily from the ultracapacitor. These observations demonstrated the effectiveness of the PDF as the power management of the EES. Finally, it should be noted that the drive cycles obtained from the EPA have a relatively coarse time step compared to that used to perform the Simulink simulations, thus causing the jaggedness of the desired power's signal.

Similarly, the state of charge of the EES components corresponding to the drive cycle from the solution of feature-based method with secondary GA optimization is illustrated in Figure 5-11.

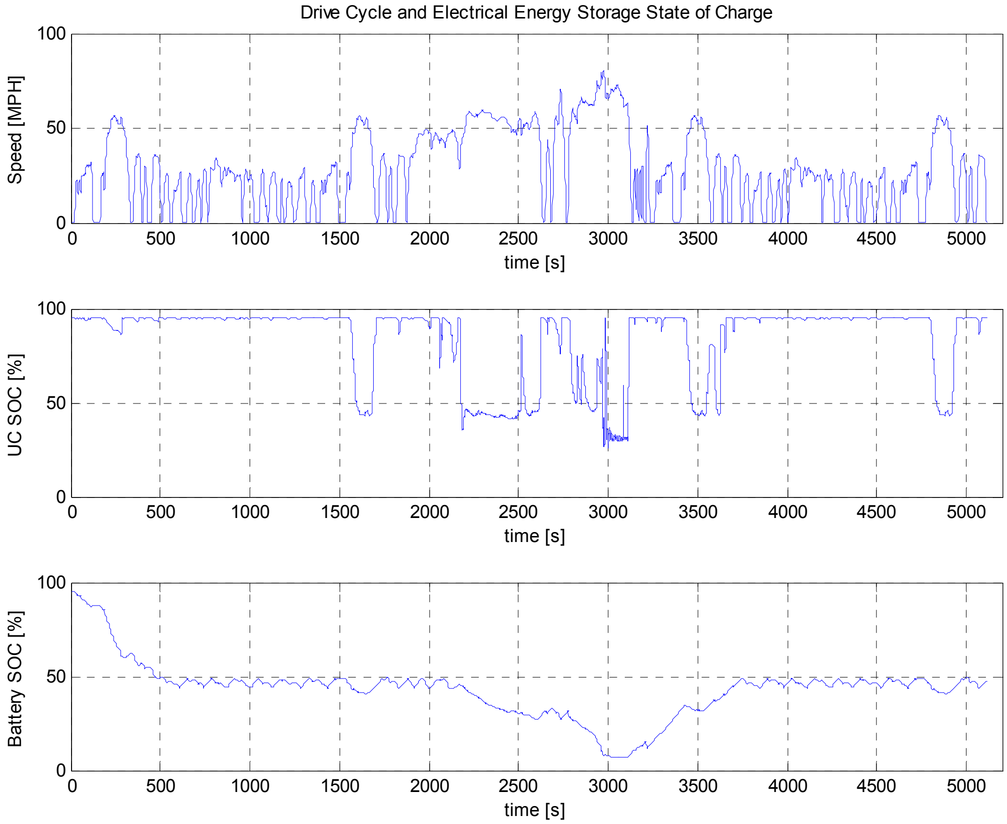


Figure 5-11: Drive Cycle and EES SOC Behaviour of the FB with Secondary GA Solution

Comparing Figure 5-11 and Figure 5-8, one can see that the solution of feature-based optimization with GA secondary optimization exhibited almost identical behaviour as the solution of the full GA optimization, demonstrating the same characteristic of utilizing the genset during the majority of the drive cycle. The time period between 2,500 and 3,500 seconds of Figure 5-11 is illustrated in Figure 5-12.

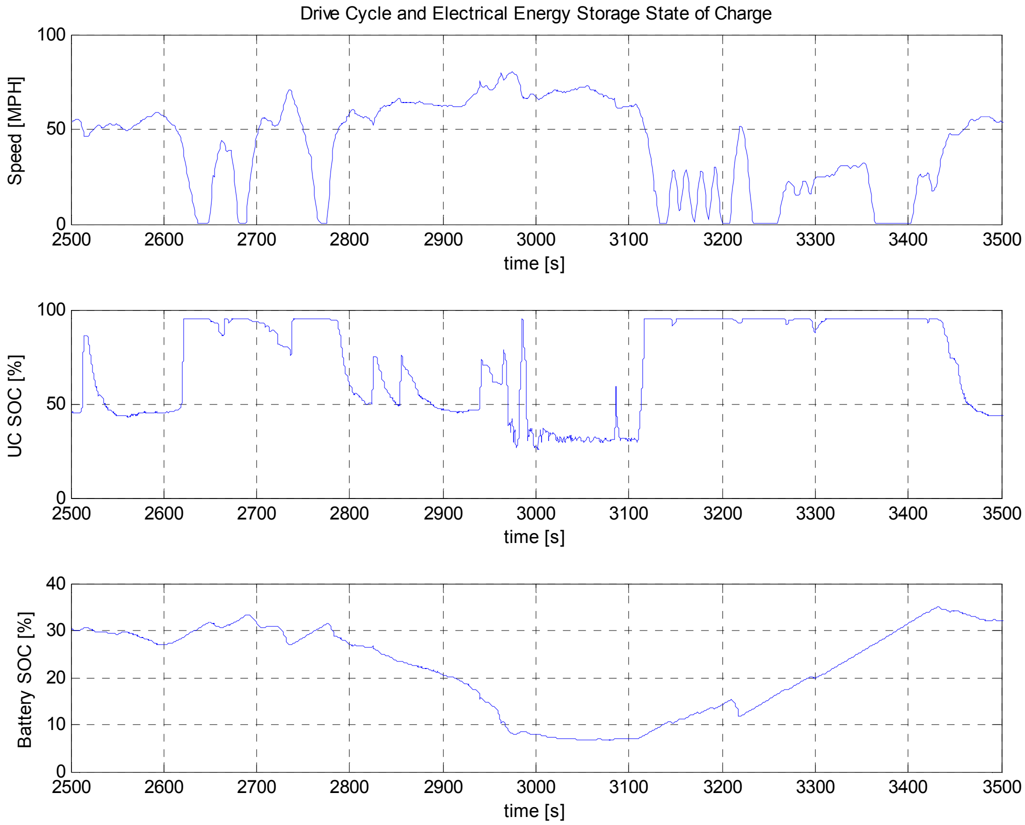


Figure 5-12: Drive Cycle and EES SOC Behaviour of the FB with Secondary GA Solution [2500-3500s]

As expected, Figure 5-12 again showed very similar behaviour as those illustrated by the full genetic algorithm optimization. Finally, the power distributing function and its dependent variables are illustrated in Figure 5-13.

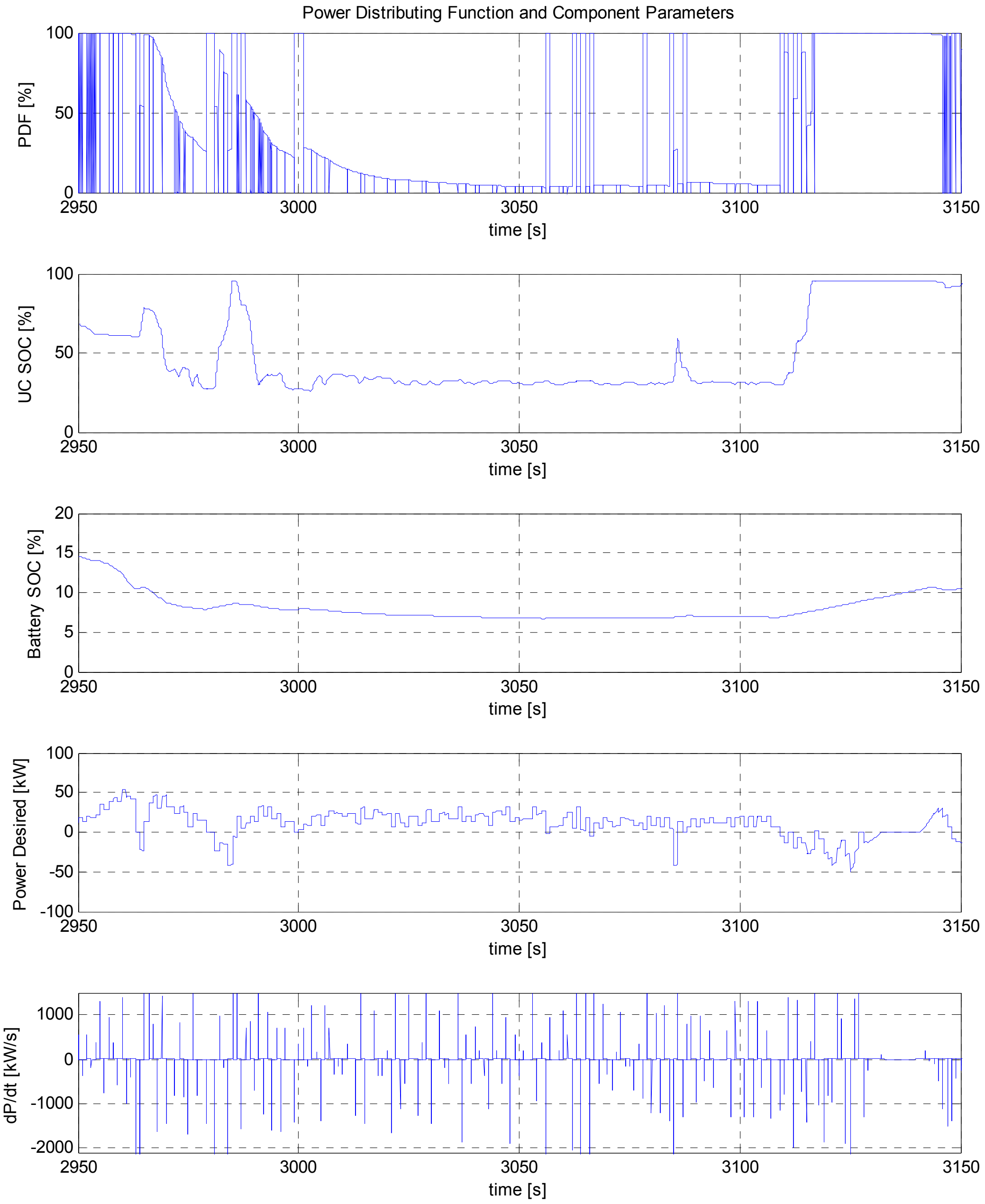


Figure 5-13: Power Distributing Function of FB Method with Secondary GA Optimization

Again, Figure 5-13 showed that the PDF determined by the feature-based method with secondary GA optimization illustrated the same characteristic as the solution found by the full GA optimization.

This further demonstrates that appropriate PDF parameters were found to properly manage the EES components using the feature-based method with secondary optimization.

Finally, to show the location of the optimal solution found by the full genetic algorithm optimization, a design study was performed where the cost function was evaluated by varying the battery size and SOC threshold. Additionally, it can be seen from Table 5-7 that the single most expensive function parameter for the objective function evaluation is the cost of the ultracapacitor; hence, for the design study, the number of ultracapacitors is set to one unit. Throughout the design study, the engine is assumed to be operating at 224rad/s and 114Nm, and the power distributing function remained unchanged. Figure 5-14 illustrates the cost function map indexed by battery size and SOC threshold.

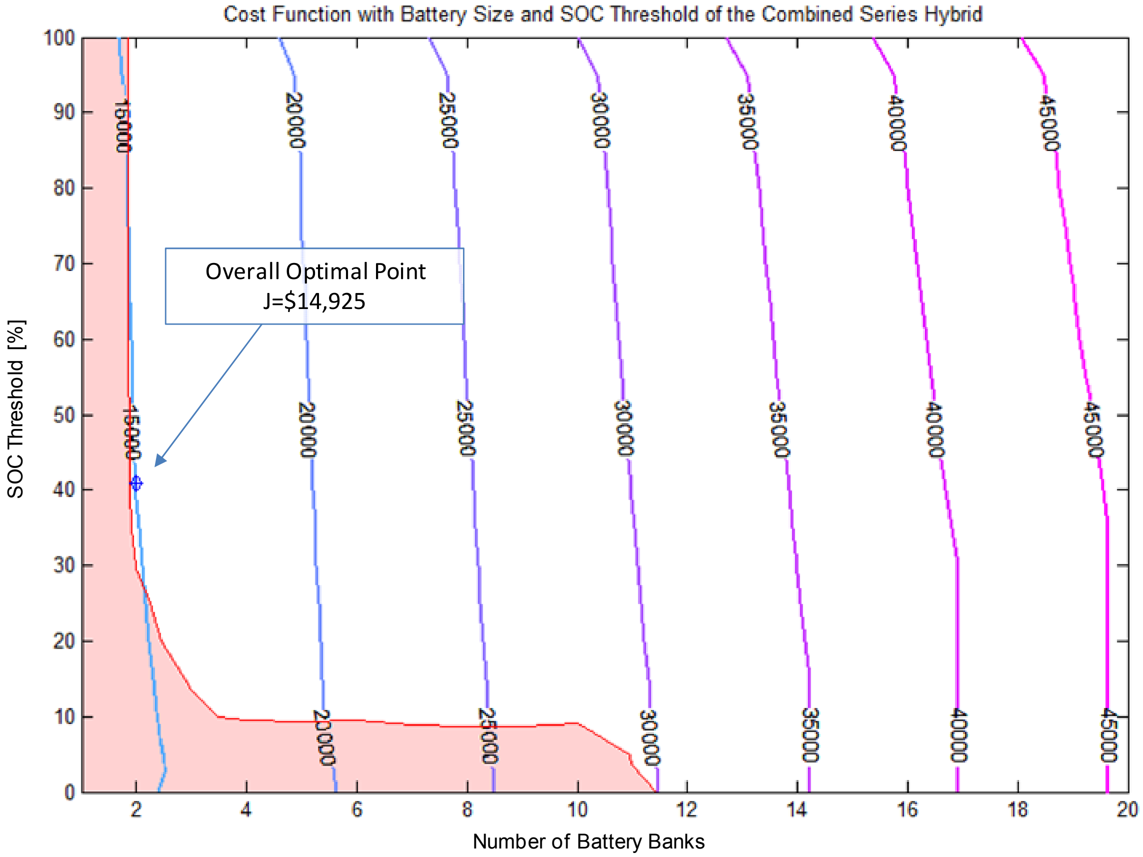


Figure 5-14: Combined SHEV Cost Function with Battery Banks and SOC Threshold

It can be seen from Figure 5-14 that the most prominent factor affecting the final cost function was still the number of battery banks. The optimized vehicle configuration as determined by the full genetic algorithm optimization along with its cost function value was shown in Figure 5-14. It was also observed that the cost function monotonically decreases towards the overall optimal point. However, unlike the case in the battery-only SHEV, local optimization technique was not able to converge to any results, while only the full genetic algorithm optimization and the feature-based method with secondary optimization were able to obtain solutions. Finally, the area where the vehicle was not able to complete a drive cycle without a time delay is shown as a shaded area in Figure 5-14. It should be noted that the PDF was optimized for a SOC threshold of 41%, and remained the same when performing the design study. Therefore the lower left region where the SOC threshold is less than 10% and up to 11 battery banks were used, the PDF was not able to manage the EES system power properly to complete the drive cycle. Once the battery pack contained up to 12 battery banks, the vehicle can operate in pure electric mode, therefore no time delay will occur when completing the drive cycle.

In conclusion, the combined SHEV optimization demonstrated the capability of performing optimization on the vehicle powertrain sizing and the power management logic simultaneously. Furthermore, it was observed that when performing full simulation, genetic algorithm was the most suitable method to solve a large optimization problem due to its capability of converging to an optimal solution. Finally, it was shown that although the feature-based optimization was not able to obtain a complete solution by itself, when combined with a secondary optimization the proposed methodology can obtain a reasonable solution while reducing simulation time by approximately 90%. This significant reduction in simulation time demonstrates that the feature-based methodology is a powerful and capable tool for vehicle design and optimization.

5.5 Benchmarking against the Chevrolet Volt

The optimization of battery-only and combined battery and ultracapacitor series hybrid electric has demonstrated the effectiveness of the concurrent optimization methodology, along with the computational efficiency of the feature-based optimization approach. To validate the accuracy of the proposed method, it is desired to benchmark the developed software platform against an existing vehicle, namely the Chevrolet Volt. In order to satisfy the performance constraints of the Chevrolet Volt, two more requirements are added to the series hybrid electric vehicle model: 1) minimum pure-electric range, and 2) battery degradation function.

One performance requirement for the Chevrolet Volt as a range extender is to have a pure-electric range of 40 miles (64km), before the genset is activated to provide additional electric energy to the powertrain. To implement such constraint during optimization, it is desired to have the minimum battery pack to contain adequate energy to traverse the aforementioned distance before utilizing the genset to sustain the charge of the battery. Therefore, the lower bound of the battery size is defined to ensure that the optimized vehicle model has a minimum pure-electric range of the aforementioned distance.

Secondly, in order to capture the effect of battery depth of discharge on the battery life, a battery degradation function (α) is applied to the cost of the battery, and is defined as follows.

$$\alpha = e^{\left(DOD/_{k}\right)} \tag{5.7}$$

where DOD is the depth of discharge threshold of the battery, which is the difference between 100% and the SOC threshold as determined by the optimizer. k is the scaling parameter to define the battery degradation function over the period of operation. The battery degradation function represents a weighting factor to capture the degradation of the battery over the period of operation. For example, if the depth of discharge is 0%, which indicates no battery usage, α will equals to 1 as calculated by Equation (5.7), thereby having no effect on the cost function of the battery. On the other hand, by adjusting the scaling parameter (k), the desired amount of increase on the cost function of the battery can be obtained. During each iteration of the optimization process, the depth of discharge threshold as determined by the optimizer is used to calculate the battery degradation function and applied to the cost of battery to evaluate the overall objective function. Therefore, the deeper the discharge of the battery, the higher the cost function of the battery will be, thereby capturing the degradation of the battery over the life cycle. Again due to the lack of manufacturers' information, a scaling parameter is assumed and can be adjusted based on the type and the specification of the actual battery pack.

5.5.1 Optimization Problem

The optimization problem is again defined as:

Minimize $J(\boldsymbol{X_D}, \boldsymbol{U}(\boldsymbol{X_D}))$ w.r.t. $\boldsymbol{X_D}$
Subject to $c(\boldsymbol{X_D})$

where the optimization parameters ($\boldsymbol{X_D}$) are the SOC threshold, genset activation duration, engine torque, and engine speed for the power management logic, and the sizes of the battery banks, IC engine, and electric traction motor. $\boldsymbol{U}(\boldsymbol{X_D})$ are the fuel and the electricity consumption of the vehicle, while $c(\boldsymbol{X_D})$ are the equality and inequality constraints of the problem. The Simulink vehicle model as shown in Appendix D defines the number of equations and the constraints of the optimization problem, and is a mixed integer nonlinear problem in nature. Finally, it has been stated that the Chevrolet Volt has an electric range of 40 miles (65km) and will reach a SOC of 30% before the genset is activated. A quick calculation determined that a minimum of 17 battery banks was required to achieve such performance constraints, and therefore, was used as the lower bound for the battery size during optimization. Tables 5-11 and 5-12 depict the size of the optimization problem and the numerical range of the optimization parameters, respectively.

Table 5-11: Summary of the Battery-Only Series Hybrid Electric Vehicle Optimization Problem

Drive Cycle Duration [s]	10,228
Time Step [s]	0.5
Number of Variables	15
Size of Optimization Problem	306,840
Number of Constraints ($c(\boldsymbol{X_D})$)	8

Table 5-12: Optimization Variables for Benchmarking Battery-only SHEV

Variables	Lower Bound	Upper Bound
SOC Threshold [%]	10	95
Genset Active Duration [%]	5	25
Engine Torque [Nm]	88	116
Engine Speed [rad/s]	222	278
Number of Battery Banks	17	28
Engine Stroke [mm]	50	120
Traction Motor Scale	5	14

The objective function (*J*) is again the total financial amount consisting of the initial cost of the battery, the cost effect of the IC engine and traction motor sizing, the cost of gasoline and household electricity consumption over a period of five years, and the equivalent carbon tax cost due to fossil fuel consumption, as given by the following equation.

$$J = \left[\sum_{t=0}^{T} Fuel_{Consumed} \times (Fuel_{Cost} + Carbon_{\ Tax}) + \sum_{t=0}^{T} Electricity_{Consumed} \times Electricity_{Cost} \right] \times Days \times Years + \alpha \times Battery_{Cell} \times Battery_{Cost} + ICE_{cost} + Motor_{cost} + Error\_Delayed \tag{5.8}$$

where T is the total time of the drive cycle. The costs of fuel and electricity were again determined based on the typical gasoline and household hydro cost in Ontario in 2008, and the same objective function parameters described in Subsection 5.3.1 were used. Furthermore, the cost effect of sizing the IC engine and the traction motor were included as described in Subsection 5.4.1. Table 5-13 summarizes the values of the cost function parameters used in the optimization.

Table 5-13: Parameters for the Benchmarking SHEV Cost Function Evaluation

Cost Function Parameters	Values
Cost of Gasoline [$/L]	0.9
Cost of Electricity [$/kW·h]	0.12
Carbon Tax [$/L of Gasoline]	0.034
Cost of A123 Li-Ion Battery [$/cell]	18.33
Battery Degradation Function (α)	145
Cost of Base Engine [$]	2,000
Cost of Base Traction Motor [$]	5,000
Cost Increase [$/unit_increment]	28.57
Number of Working Days per Year	260
Number of Years	5

5.5.2 Optimization Results

Optimization on the benchmarking battery-only series hybrid electric vehicle was performed using the procedure described in Subsection 4.2.3 utilizing genetic algorithm. It should be noted that due to the lack of manufacturers' data, the degradation function (α) is assumed to be one during the comparison. Table 5-14compares the optimization results against the specification of the Chevrolet Volt.

Table 5-14: Optimization Results of the Benchmarking SHEV and the Chevrolet Volt

Algorithm	**Full Optimization using GA**	**Feature-Based Using GA**	**Chevrolet Volt Specifications**
SOC Threshold [%]	10	10	30
Genset Active Duration [%]	19	5	5
Engine Torque [Nm]	116	88	116
Engine Speed [rad/s]	222	222	222
Number of Battery Banks	17	17	21
Engine Stroke [mm]	50	50	80.6
Traction Motor Scale	5	5	7
Gasoline Consumption [L/day]	1.87	2.04	2.40
Electricity Consumption [kWh/day]	10.73	10.49	9.63
Battery Cost [$]	31,161	31,161	38,493
Engine Cost [$]	2,000	2,000	2,874
Motor Cost [$]	5,000	5,000	5,057
Fuel Cost [$]	2,187	2,386	2,803
Electricity Cost [$]	1,562	1,527	1,403
Carbon Tax [$]	83	90	106
Total Cost [$]	**41,993**	**42,164**	**50,736**
Simulation run-time [s]	*3,889*	*589*	*n/a*

Table 5-14 compares the values of the optimization variables and results found by the optimizer, using both full optimization and feature-based approaches, against the specifications of the Chevrolet Volt. Due to the lack of information on the power management logic of the Chevrolet Volt, the genset activation duration is assumed to be the minimum value, while the IC engine's most efficient operating point are used. It was found that the optimizer in both cases of the full optimization and the feature-based approaches determined the minimum SOC threshold and the smallest allowable powertrain sizes; hence, the total cost is less than the Chevrolet Volt. This is expected since the battery degradation and the possibly additional performance constraints were not included. Additionally, the optimization case study only considered satisfying the predefined drive cycle and did not contain performance requirements such as top speed and maximum acceleration. However, with accurate manufacturers' component data and by implementing the appropriate performance constraints, the developed concurrent optimization software platform will be able to accurately predict a solution similar to that of the Volt.

Chapter 6
Case Study 2: Anti-Idling System

In this study the optimization of an anti-idling system for police vehicles is considered. The developed generic vehicle model was modified to represent the power generation and consumption of a conventional vehicle, specifically an idling engine running an alternator to power the auxiliary consumption and recharge the battery. This chapter will discuss the modeling details and present the simulation and optimization results of the anti-idling system based on a 2009 Chevrolet Impala police vehicle.

6.1 Background and Objective

Emergency service vehicles, such as police vehicles, fire trucks, and ambulances generally have higher auxiliary power consumption than conventional vehicles due to the additional electrical equipments, such as the roof top light bar, take down lights, communication equipments, and laptop computers. In addition, these vehicles usually have an unconventional drive cycle where majority of the operation time are spent idling. During the idling period, the engine is usually left running in order to power the aforementioned electrical equipment. In recent years, anti-idling has been receiving a lot of attention in the automotive industry as a method to further improve the fuel economy while reducing harmful emissions. It has been shown that the implementation of anti-idling on a standard urban drive cycle can improve the fuel efficiency by as much as 8% [51,52], and manufacturers are gradually introducing the anti-idling feature to their existing vehicle line-up [53,54].

Using the concept of anti-idling, the objective of the case study is to reduce the engine idling time, by installing an additional (secondary) battery to power the on board electrical equipments during the idling period while minimizing design changes to the original vehicle. The Original Equipment Manufacturer (OEM) battery will not be supplying electrical power to the auxiliary equipments, since it is necessary for the OEM battery to remain fully charged for the vehicle operator to start the engine whenever necessary. The current approach in the anti-idling system is to turn on the engine to recharge the battery at the factory default engine idling speed when the battery state of charge falls below a predetermined threshold value. The goal of the optimization is to determine the battery's state of charge (SOC) threshold below which the engine shall be activated to charge the battery, leading to minimum overall cost. Additionally, the effect of the engine speed during the recharging of the battery will be investigated. Finally, in addition to the specific lead-acid batteries suggested by the industry partner, optimization will also be performed using lithium-ion batteries for the comparison of results with those of the lead-acid batteries.

6.2 System Model

Similar to the series hybrid electric vehicle model described in the previous chapter, the anti-idling system model was created in MATLAB/Simulink utilizing a backward-looking architecture. The battery is discharged as determined by the load cycle, and an IC engine model calculates the fuel consumption required to drive the alternator to charge the battery. During a typical simulation, the battery delivers the required power to run the auxiliary components, and once the battery state of charge (SOC) falls below a preset value, the IC engine will be turned on to drive the alternator to recharge the battery. Figure 6-1 depicts the overall schematic of the anti-idling system with the optimizer.

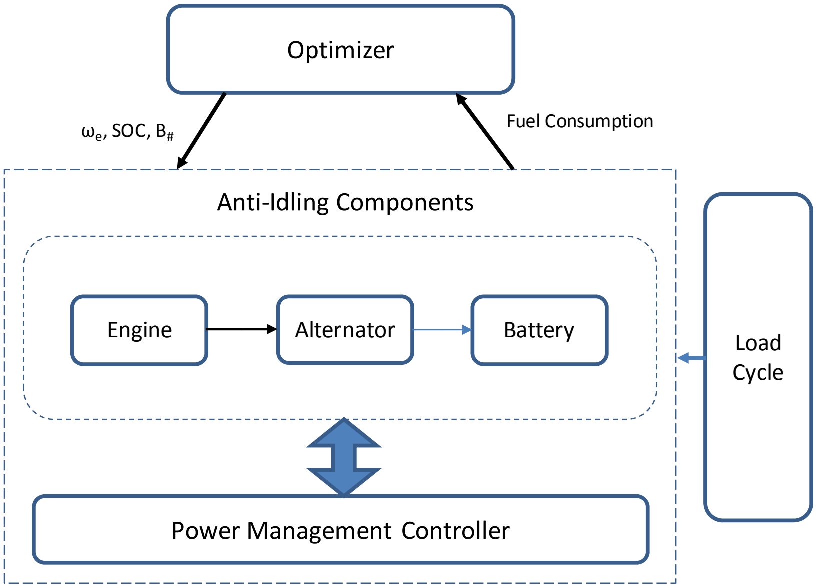


Figure 6-1: Anti-Idling System Overall Schematic with Optimizer

In Figure 6-1, ω_e represents the engine speed, *SOC* is the battery's state of charge (SOC) threshold below which the engine is activated, and $B_\#$ is the number of battery cells in the battery pack. The details of each of the vehicle components are presented in the subsequent sections.

6.2.1 Load Cycle

The load of the anti-idling system includes the consumptions of all electrical components of the vehicle. In reality, the electrical equipments may not be running at a constant power; therefore, the load cycle time history for the anti-idling system Simulink model was created using a random number generator producing an average load power given in Table 6-1.

Table 6-1: Anti-Idling System Power Load Summary

Vehicle System	1.8 A
Light Bar with LED, Take Down Lights, Flashing Front and Rear Lights	30.5 A
Night Time Head and Tail Lights	10.9 A
Laptop Computer	8 A
Total Load	51.2
System Voltage	12V
Total Power	614.4W

Furthermore, the idling pattern of the police vehicles is not known at this time; therefore, for the simulation purposes, it is assumed that the vehicle idles one hour at a time before driving off. Figure 6-2 shows the electrical power load cycle time history for the Simulink model during one-hour vehicle idle time.

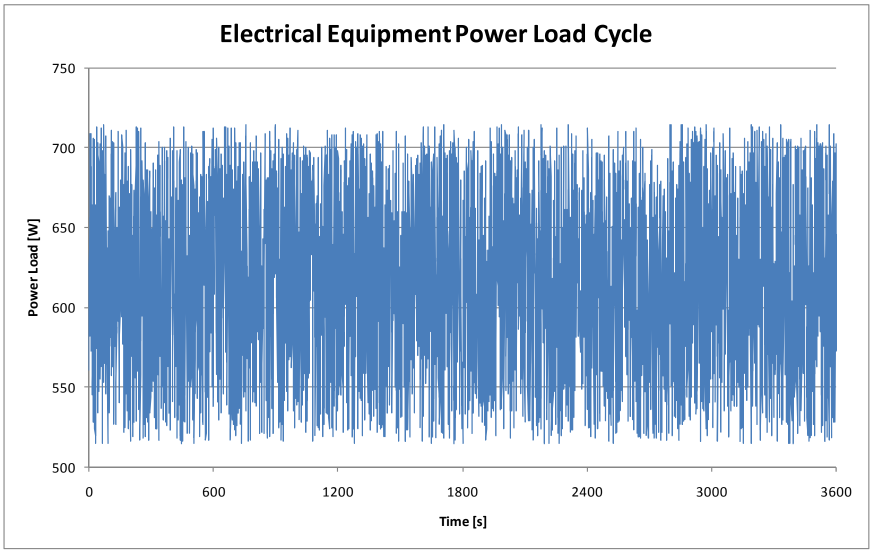


Figure 6-2: Electrical Equipment Power Load Cycle Time History

6.2.2 System Components

6.2.2.1 Battery

The battery model represents the additional (secondary) battery to power the electrical components while the engine is turned off. Since the goal is to prevent using the OEM battery, it is not being considered during the system modeling and optimization; hence, only the secondary battery is modeled. Three battery types were considered for the optimization: two Discover dry cell batteries with different capacities, and the A123 Lithium-Ion battery used in the series hybrid electric vehicle optimization. The specifications of the Discover batteries (EV12-180X and EV12-140X) and the A123 Lithium-Ion battery were obtained from their respective websites [42,45]. It should be noted the Discover dry cell batteries are prepackaged to match the vehicle's electrical system (12V) along with a sufficiently large capacity; therefore, the battery size is not a design variable during optimization. On the other hand, similar to the case of the series hybrid electric vehicle battery system, a parallel-series configuration is necessary when using the A123 Li-Ion batteries, where 4 cells are connected in series to match the electrical system's voltage. The optimization procedure will therefore determine the number of battery banks required when using the A123 Li-Ion batteries. The same Simulink battery model used for the series hybrid electric vehicle is implemented for the anti-idling system.

6.2.2.2 Alternator

A simple alternator model was used for the anti-idling system, where a look-up table was used to model the output current of the alternator indexed by the engine speed. The alternator is modeled after Denso SC2 as indicated by the GM Impala 9C1/9C3 police package specification [55]. The information is given in Appendix I. In addition to the factory default idling speed, it is desired to study the effect of increasing the alternator output power during the optimization. Therefore, the range of engine speed for optimization starts at 650RPM and is increased up to the engine speed corresponding to the alternator's maximum output current (1800RPM). Finally, since the alternator efficiency was not available, it was assumed to be 90%.

6.2.2.3 Engine

The engine model using the Willans line modeling approach was created using the engine parameters of the 3.9L V6 (LGD) engine as specified by the Impala police package [55]. The engine module calculates fuel consumption based on the engine idling speed and the desired power output of the alternator, while taking into consideration the alternator's efficiency.

6.2.3 Power Management Logic

The role of the power management logic is to activate the engine to charge the battery when required. In addition, the power management logic controls the idling speed of the engine and the output torque based on the alternator output power. When the battery state of charge (SOC) falls below a preset threshold value, the power management logic activates the engine until the battery is fully charged before being turned off.

As previously mentioned, the police vehicles idles 80% of the time; hence, the vehicle is only in motion for 20% of the time. It is only this time that the alternator is outputting maximum current to run the auxiliary components and to charge the battery. However a quick calculation shows that when the vehicle is in motion, the difference between the maximum current output and the auxiliary system's consumption is 98.8A, which may not be enough to fully charge the batteries in the short period of time while the vehicle is in motion. It is therefore necessary to ensure that the batteries are charged up to a certain level at the end of the drive cycle in order to fully charge the battery when the vehicle is in motion. The final battery SOC required at the end of the idling cycle is described by Equation (6.1).

$$SOC_{final} = \frac{Batt_{cap} - \left(I_{alt,max} - I_{aux}\right) \times T/3600 \times \%_{driving}}{Batt_{cap}} \tag{6.1}$$

where SOC_{final} = required final battery SOC

$Batt_{cap}$ = battery capacity [Ah]

$I_{alt,max}$ = alternator maximum current output [A]

I_{aux} = current consumption of the auxiliary components [A]

T = total cycle time including idling and driving [s]

$\%_{driving}$ = percent of total cycle time when vehicle is driving

6.2.4 Optimization Problem

The goal of the optimization procedure is to determine the power management logic that will result in the minimal objective function. In addition, the optimization will determine the battery size in the case of the A123 batteries. The optimization problem is formulated as

$$\text{Minimize } J(\boldsymbol{X_D}, \boldsymbol{U}(\boldsymbol{X_D})) \text{ w.r.t. } \boldsymbol{X_D}$$
$$\text{Subject to } c(\boldsymbol{X_D})$$

where the optimization parameters ($\boldsymbol{X_D}$) are the SOC threshold and engine speed for all three batteries, as well as the number of battery banks for the A123 battery system. $\boldsymbol{U}(\boldsymbol{X_D})$ is the fuel consumption of the engine, while $c(\boldsymbol{X_D})$ are the equality and inequality constraints of the problem. The Simulink vehicle model as shown in Appendix H defines the number of equations and the constraints of the optimization problem, which is a mixed integer nonlinear problem in nature. Table 6-2 summarizes the size of the optimization problem.

Table 6-2: Summary of the Anti-Idling Optimization Problem

Drive Cycle Duration [s]	3,600
Time Step [s]	0.5
Number of Variables	8
Size of Optimization Problem	57,600
Number of Constraints ($c(\boldsymbol{X_D})$)	5

Additionally, it should be noted that the optimization range of the A123's battery bank is determined as the equivalent capacity as compared to the Discover EV12-180X battery. Table 6-3 depicts the numerical range of the optimization variables.

Table 6-3: Anti-Idling Optimization Variables

Variables	Range
SOC Threshold	1-95
Engine Speed [rad/s, RPM]	68-189, 650-1800
Number of Battery Banks (A123 only)	1-20

The objective function (*J*) is the financial amount of the total cost of the batteries, the operating cost over a period of five years, and the equivalent carbon tax due to fuel consumption, as given by:

$$J = Fuel_{Consumed} \times (Fuel_{Cost} + Carbon\ _{Tax}) \times Cycles_{per\ day} \times Days \times Years \\ + Battery_{number} \times Battery_{Cost} + Error\_SOC\_final \tag{6.2}$$

The cost of the Discover dry cell batteries were obtained from the Ontario distributer, while the cost of A123 was determined from the manufacture's website [45]. The rest of the cost function parameters were determined from the same source introduced in Subsection 5.3.1. Due to the lack of idling pattern information at the current stage, it was assumed that a police vehicle idles approximately 6 times a day. Finally, an error function is included in the objective function evaluation for the case where the battery SOC does not meet the required level at the end of the idling cycle as defined in Equation (6.1). Table 6-4 summarizes the values of the cost function parameters used in the optimization.

Table 6-4: Parameters for the Anti-Idling Objective Function Evaluation

Cost Function Parameters	Values
Cost of Gasoline [$/L]	0.9
Carbon Tax [$/L of Gasoline]	0.034
Discover EV12-180X [$/unit]	550.51
Discover EV12-140X [$/unit]	458.37
Cost of A123 Li-Ion Battery [$/cell]	18.33
Number of Idling Cycles per Day	6
Number of Working Days per Year	260
Number of Years	5

Finally, due to the high number of discharge cycles over the five-year period, it is necessary to consider the degradation of the battery after reaching its useful life cycle. In addition, since the life cycle of the lead acid battery is dependent on its depth of discharge, its useful life cycle will be determined by the minimal battery SOC that is reached during the idling cycle. The objective function described by Equation (6.2) includes the number of replacement batteries needed throughout the five year period. The number of replacement batteries needed is determined using the following relation.

$$\begin{aligned}&\#\ of\ Replacement\ Battery\\ &\quad = \frac{Idling\ Cycles_{per\ day} \times Days \times Years \times Cycle_{per\ idle\ period}}{Battery\ Cycle\ Life}\end{aligned} \tag{6.3}$$

When the number of A123 Li-Ion battery banks is one, it was observed that the battery system was fully discharged twice per idling period due to its relative small capacity. Hence it is important to include the number of cycles per idle period in such a scenario. The cycle lives of the batteries are given in Appendix I.

6.3 Simulation Results

Optimization was first conducted on the anti-idling system using genetic algorithm for the three different battery types. In addition, optimizations were performed with the algorithms outlined in Subsection 4.2.3 to compare the differences among different optimization algorithms. Finally, design studies were conducted by varying the design parameters. The following sections report the simulation results and discuss the effectiveness of the optimization.

6.3.1 Optimization Results

Genetic algorithm optimization was first performed on the anti idling system using the procedure described in Subsection 4.2.3 using parameters introduced in Subsection 6.2.4. The values of genetic operators used in the optimization process are summarized in Table 6-5.

Table 6-5: Values of Genetic Algorithm Operators

Population	80
Maximum Generation	50
Crossover Probability	0.9
Mutation Probability	0.05

The fitness function of each population was evaluated using the cost function parameters summarized in Table 6-4. Due to the fact that the battery may not be fully charged at the end of the idling cycle, it is necessary to account for the additional gasoline required to fully recharge the battery. Since the drive cycle of the vehicle in motion is not known, the concept of equivalent gasoline consumption based on the additional energy required to fully charge the battery is introduced. The equivalent gasoline ($C_{eqv,gas}$) is calculated by:

$$C_{eqv,gas} = \frac{\left(100\% - SOC_{final}\right) \times Batt_{cap} \times V_{chg} \times 3600}{\rho_{gas} \times H_l \times \eta_{alt} \times \eta_e} \tag{6.4}$$

where SOC_{final} = battery final SOC

$Batt_{cap}$ = battery capacity [Ah]

V_{chg} = battery charging voltage [V]

ρ_{gas} = volume density of gasoline [g/L]

H_l = lower heating value of gasoline [kJ/g]

η_{alt} = alternator efficiency

η_e = engine efficiency

The equivalent fuel consumption is then the sum of the actual gasoline consumed during idling and the equivalent gasoline consumption to fully charge the battery when the vehicle is in motion, where the engine efficiency is assumed to be 30%. The best individuals, the required final SOC, the number of battery replacement, and the cost function of each of the battery types and the system without the anti-idling are summarized in Table 6-6.

Table 6-6: Optimized Vehicle Configuration and Final Cost Function

	Discover EV12-140X	**Discover EV12-180X**	**A123 System ANR26650**	**Without Anti-Idling**[3]
SOC Threshold [%]	73	80	30	n/a
Engine Speed [rad/s]	70	68	68	68
Battery Banks (A123 only)	n/a	n/a	2	n/a
Final SOC Required	82.4	86.4	10	n/a
# of Replacement Batteries	3	3	8	0
Equiv. Fuel Consumption [L/day]	0.3996	0.4069	0.8119	0.7995
Battery Cost [$]	1,375	1,652	1,173	0
Fuel Cost [$]	2,805	2,857	5,700	5,612
Carbon Tax [$]	106	108	215	212
Total Cost [$]	**4,286**	**4,616**	**7,088**	**5,825**

It can be seen from the above results that the Discover EV12-140X battery induced the lowest cost over the operating period, where it resulted in a slightly less fuel consumption in addition to a lower

[3] Engine operating at the idle torque (10Nm).

cost per battery unit when compared to the larger-capacity battery (EV12-180X) of the same type. On the other hand, the deciding factor of the A123 battery was again the cost of the battery, and thus, by reducing the battery size the gasoline consumption at idle speed increased significantly when compared to the Discover lead acid batteries. It can be concluded that the Discover batteries were the obvious choice for the anti-idling system, while utilizing the EV12-140X battery will cost slightly less than the larger capacity battery (EV12-180X) over the operating period.

The same optimization procedure was performed with the EV12-140X battery while utilizing different optimization algorithms. Table 6-7 compares the results of the various optimization methods.

Table 6-7: Optimization Results of Various Algorithms

Algorithm	**Genetic Algorithm**	**Simulated Annealing**	**Pattern Search**	**Nelder-Mead**
SOC Threshold [%]	73	72	87	59
Engine Speed [rad/s]	70	76	76	94
# of Replacement Batteries	3	3	3	7
Equivalent Fuel Consumption [L/day]	0.3996	0.3981	0.4168	0.3148
Battery Cost [$]	1,375	1,375	1,375	3,209
Fuel Cost [$]	2,805	2,795	2,926	2,210
Carbon Tax [$]	106	106	111	83
Total Cost [$]	**4,286**	**4,276**	**4,412**	**5,502**
Simulation run-time [s]	1,252	1,744	116	76

Table 6-7 shows that the simulated annealing (SA) algorithm found the lowest total cost while requiring the longest simulation run-time. The result of genetic algorithm (GA) was slightly higher than that of SA while using approximately 28% less simulation time. Pattern search utilized significantly less simulation time; however, it did not find the minimal solution. Finally, it was found that the solution of the unconstrained Nelder-Mead method was lower than the required final SOC, and therefore was not capable of reaching any meaningful solution. The comparison between the optimization algorithms showed once again that genetic algorithm is the most practical method to solve the optimization problem.

6.3.2 Design Study

A design study was conducted using the Discover EV12-140X battery, where a series of simulations were performed with various engine speeds and SOC thresholds. The engine speed ranged from

68.07 rad/s (650RPM) to 188.5 rad/s (1800RPM), while the SOC threshold varied between 71% and 98%. It was found that when the SOC threshold is below 70%, the battery was not able to return to the desired final SOC, and therefore is not included in the design study. Figure 6-3 illustrates the cost function of the EV12-140X battery with different engine speeds and SOC thresholds.

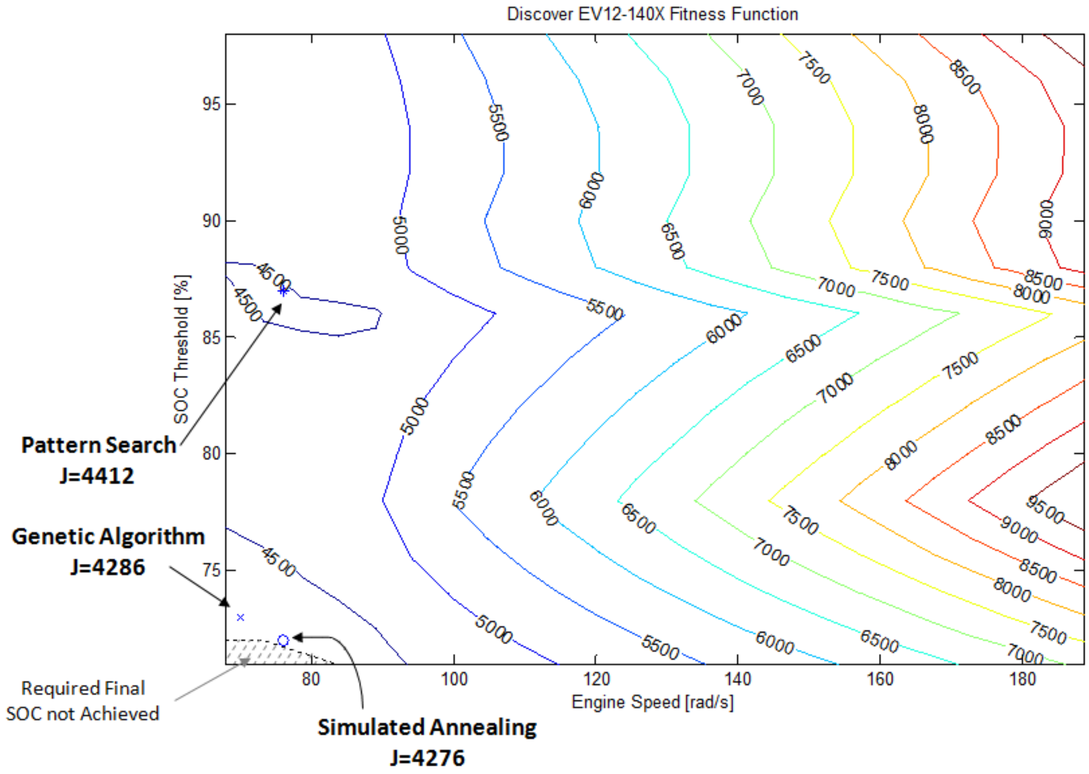


Figure 6-3: Anti-Idling System Cost Function of EV12-140X Battery

It is noticed in Figure 6-3 that the battery was not able to return to the required final SOC level in the lower left corner region. Additionally, there is a local minimum occurring between 85% and 90% SOC threshold, which is where the pattern search algorithm found its solution. Based on the locations of the solutions found by the various optimization algorithms, Figure 6-3 confirmed that classical optimization methods are more susceptible of being trapped in a local minimum, while global optimization algorithm are capable of finding a global solution. To better understand the system's behavior, the equivalent fuel consumption is plotted in Figure 6-4 versus the SOC threshold when the engine is operating at the speed of 78rad/s.

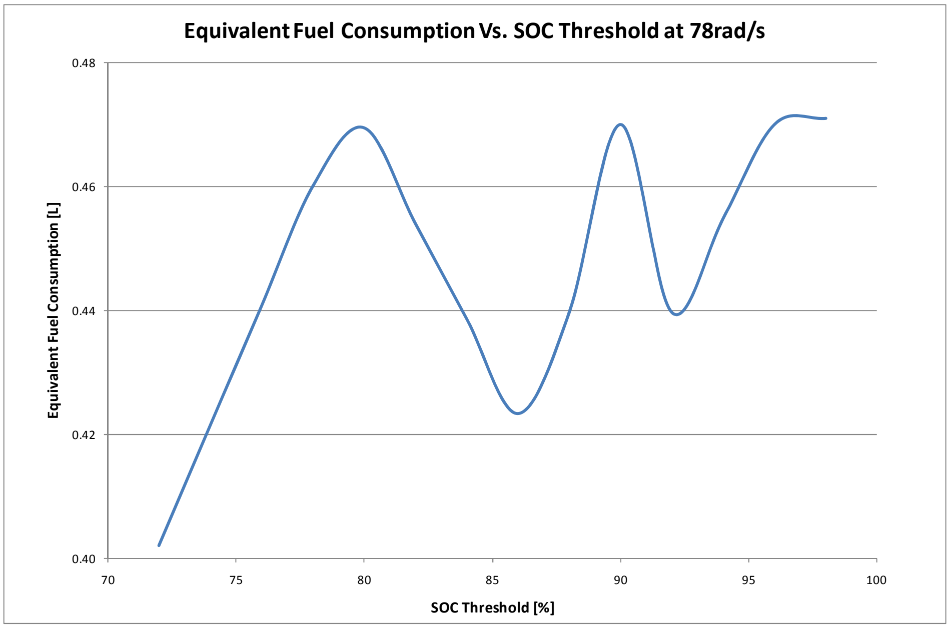


Figure 6-4: Equivalent Fuel Consumption of Anti-Idling System at 78rad/s Engine Speed

Upon further inspection of the model, it was noticed that the local minima and maxima correspond to the period during which the engine is activated. For example, when the SOC threshold is at 86%, the battery is first discharged to 86%, then recharged by the engine to full capacity, and finally discharged back down to exactly 86% at the end of the idling cycle. On the other hand, when the SOC threshold is set at 80%, the battery was fully charged back to 100% at the end of the idling cycle. Even though equivalent gasoline is added to fully recharge the battery at the end of the drive cycle, the engine efficiency is much lower at idling speed than that during normal operation. Therefore, when the SOC threshold is varied, the period of time during which the engine is operating at idle speed is different, thus contributing to the difference in the final equivalent fuel consumption. The design study confirmed that the global optimization algorithms were able to search for the global optimal point while avoiding being trapped in the local minima, and at the same time disregarding the regions where the required final SOC level was not met. Finally, it was found that for the anti-idling system, it was more efficient from the financial cost perspective to operate at a lower engine speed. By utilizing the anti-idling system using the Discover EV12-140X battery, significant cost reduction can be realized when compared to one without the anti-idling system.

Chapter 7
Conclusions

7.1 Summary

The key contribution of the research was the development of a Multidisciplinary Design Optimization (MDO) approach to perform concurrent optimization of a hybrid electric vehicle, where the system's component sizing and power management logic were optimized simultaneously based on the design objectives of the system. Furthermore, a feature-based optimization technique was developed to reduce simulation run-time of the optimization process. To demonstrate the capability of the proposed methodology, two optimization case studies were undertaken: (i) battery-only and combined battery-ultracapacitor series hybrid electric vehicle and (ii) police vehicle anti-idling system. For both case studies, results of various optimization algorithms were compared and discussed.

A plug-in battery-only series hybrid electric vehicle (SHEV) with power components similar to those of Chevrolet Volt was created as part of the first case study using the backward-looking architecture. To demonstrate the effectiveness of the proposed concurrent optimization, the battery size and power management logic were simultaneously optimized. The parameters for the power management logic included the battery's state of charge (SOC) threshold, the SOC below which engine should be turned on to recharge the battery, and the engine speed and torque, for which the genset should be operating at the most efficient point. The objective function of the optimizer was defined from the financial cost perspective, where the objective was to minimize the sum of the initial cost of batteries, fuel and electricity consumption over a period of five years, and the carbon tax as a penalty function for harmful emissions. A drive cycle consisting of standard urban and highway speed profiles with a total distance of 65km was used. The minimum battery size, while still large enough to successfully complete the drive cycle, was found by the optimizer. A comparison between different optimization algorithms found that feature-based technique required the smallest amount of simulation run-time, while simulated annealing required the largest. Furthermore, since the objective function map monotonically decreased towards the optimal solution, local optimization method such as pattern search was able to find the solution in a relatively short time. Finally, the optimized configuration was compared with the Honda Insight, Toyota Prius, Chevrolet Volt, and Tesla Roaster on a cost function map. It was observed that the optimized configuration indeed resulted in the most cost effective battery-only SHEV. In addition, the location of the aforementioned hybrid electric vehicles on the cost function map accurately reflected the order of the vehicles' respective list prices.

In the second part of the case study, ultracapacitors were added to the electrical energy storage (EES) system of the SHEV. The optimization parameters were expanded to size the EES components, the IC engine, and the electric traction motor. Additionally, the power management logic included the duration of time that the genset remained activated and the power distributing function (PDF) that managed the combined operation of the batteries and the ultracapacitors within the EES system. The drive cycle of the vehicle model remained the same as the battery-only SHEV, while the objective function included the cost of the ultracapacitor and the interpolated cost due to the sizing of the IC engine and the electric traction motor. A comparison of the optimization algorithms found that only genetic algorithm was capable of finding the optimal solution during a full optimization, while simulated annealing and pattern search were not able to converge to any solution after a 24-hour period. Due to the nature of the feature-based (FB) optimization technique, it was not capable of solving the time sensitive parameters such as the power distributing function. A secondary optimization was conducted using the powertrain configuration initially determined by FB, and the solutions of the PDF parameters were obtained. A comparison between the full genetic algorithm optimization and the feature-based method with secondary optimization found that although the final cost function of the FB methodology was higher than that of the full GA optimization, the total simulation run-time was significantly decreased using the FB method. The required simulation run-time was reduced to less than 45 minutes using the feature-based technique from over 7 hours as required by the full genetic algorithm optimization. Finally, time history behaviour of the EES system and the PDF in the solutions found by both methods exhibited almost identical characteristics, further confirming the validity of the feature-based method with secondary optimization. Finally, a benchmarking comparison against Chevrolet Volt found that with more accurate manufacturers' component data and by implementing the appropriate performance constraints, the developed concurrent optimization software platform will be able to accurately predict a solution similar to that of the Volt.

The second case study involved optimizing an anti-idling system for the police vehicles using the same optimization algorithm and generic vehicle model. It was estimated that in average for 80% of the time a police vehicle will remain parked while idling with the engine on, in order to power the onboard electrical equipment. The goal was to select and size an additional battery, and determine the power management logic indicating when the engine should be operated to recharge the battery. Two Discover lead-acid batteries with different capacities and a lithium-ion battery were used during the optimization study. It was found that depending on the SOC threshold, the period of time during which the engine was activated varied in a non-linear fashion, where local minima and maxima existed. A comparison of different optimization algorithms demonstrated that only global optimization methods such as genetic algorithm and simulated annealing were capable of reaching at the optimal solution, while the pattern search algorithm was stuck in a local minimum.

The optimizers used in the simulation platform were derivative-free methods that were available within the MATLAB library. Two global and two local optimization methods were chosen to demonstrate the characteristic of the algorithms and to determine the most suitable method of choice for conducting the concurrent vehicle optimization. Additionally, the results of the developed feature-based optimization were compared to those of the existing optimization algorithms. It was found that the Nelder-Mead method could not reach any meaningful solution in all cases, and was therefore disregarded throughout the discussion. Table 7-1 summarizes the characteristic and the performances of various optimization algorithms used in this research.

Table 7-1: Performances of Various Optimization Algorithms used in the Case Studies

	Genetic Algorithm	**Simulated Annealing**	**Pattern Search**	**Feature-Based**
	Global	Global	Local	Custom
Battery SHEV Optimization Variables = 4	**$8,549**	**$8,540**	**$8,553**	**$8,547**
	6,147s	*13,165s*	*701s*	*467s*
Combined SHEV Optimization Variables = 14	**$14,925**	**incomplete**	**incomplete**	**$15,949**
	25,576s	*> 24hrs*	*> 24hrs*	*2,588s*
Anti-Idling System Optimization Variables = 2	**$4,286**	**$4,276**	**$4,412**	**n/a**
	1,252s	*1,744s*	*116s*	*n/a*

Table 7-1 showed that in the case when simulated annealing was able to complete the optimization, it found the lowest cost function while utilizing the largest amount of time. On the other hand, even though the solutions of genetic algorithm were slightly higher than that of simulated annealing in the case of the battery-only SHEV, it was able to converge to a solution for the combined SHEV while simulated annealing could not. Pattern search was able to reach a solution fairly quickly; it was, however, susceptible to be stuck in a local minimum. Similar to the simulated annealing, it was not capable of reaching a solution for the combined SHEV. Additionally, the developed feature-based optimization demonstrated its capability of reaching a reasonable solution while utilizing significantly less simulation run-time. It should be noted that since the drive cycle of the anti-idling system was assumed to be of random nature, feature-based optimization was not conducted for the anti-idling case study. The comparison between the various optimization algorithms found that the genetic algorithm could reach an optimal solution regardless of the complexity of the system. Furthermore, it was shown that the feature-based optimization could obtain a relatively accurate solution while reducing simulation time by approximately 90%. This significant reduction in simulation time warrants the feature-based optimization technique a powerful tool for vehicle design and optimization.

It was observed that the due to the high cost of the electrical energy storage components, it was currently still more cost effective to use the fossil fuel as the primary method of energy transportation for automobiles. However, given the rise of fuel cost and the advancement in the EES technology, it is inevitable that the cost of the electrical and chemical energy storage methods will reach a balance point. The proposed optimization platform provides the user with the capability and flexibility to obtain the optimal vehicle solution with ease at any given time in the future.

7.2 Thesis Contributions

- A complete multidisciplinary vehicle optimization software platform was developed to perform concurrent optimization on powertrain sizing and power management logic. A modular and flexible backward-looking vehicle model was created to perform vehicle energy consumption calculations. Series hybrid electric vehicles and an anti-idling system for police vehicles were created and used as case studies to conduct concurrent optimization using the developed software platform, and demonstrate the usefulness of the developed software solution.
- A power distributing function was created to manage the charging and discharging power between the battery and the ultracapacitor. The power distributing function demonstrated the capability of managing the power flow between the electric energy storage devices and the power components, and also the flexibility to fine tune the behaviour of the EES system using the optimizer.

- A survey and comparison of different optimization algorithms was conducted and implemented in the case studies. The advantages and the shortcoming of the different algorithms were discussed, and it was found that the genetic algorithm was not only capable of finding the global optimal solution in a reasonable time frame, but was also able to converge to a solution in all cases.

- During a full vehicle optimization, simulations in Simulink were performed hundreds and thousands of times to evaluate the objective function, resulting in hours of simulation time. A feature-based optimization methodology was developed with the motivation of reducing the total simulation time. It was shown that the feature-based technique was capable of reducing the total simulation time by up to 90% while maintaining reasonable accuracy of the final solution.

7.3 Future Work

- Currently, to determine the power consumptions of IC engines and electric motor-generators, only efficiency calculations are performed. It is desired that further work be performed on increasing the fidelity of these energy conversion models to improve the accuracy of the fuel and electricity consumptions. Similarly, additional improvements on the battery and ultracapacitor models will provide an improved accuracy when determining the power transfer efficiencies. A more realistic battery model can be realized by considering the internal resistance of the battery as a function of the battery's state of charge.

- The developed optimization platform had only conducted optimization on a series hybrid electric vehicle configuration. To fully utilize the capability of the proposed concurrent optimization approach, it is desirable to expand the case studies to parallel hybrid electric vehicles and battery electric vehicles.

- Due to the lack of OEM information on the powertrain components, assumptions were made on the specification and costs of the IC engine and electric motor-generator. With the wealth of information on power component specifications, manufacturers can utilize the developed software platform with accurate component parameters for their vehicle design and optimization, achieving the full potential of cost and time savings during the vehicle design cycle.

Bibliography

[1] B. S Fan, "Modeling and Siumulation of a Hybrid Electric Vehicle Using MATLAB/Simulink and ADAMS," University of Waterloo, Waterloo, ON, MASc Thesis 2007.

[2] F. R. Salmasi, "Control Strategies for Hybrid Electric Vehicles: Evolution, Classification, Comparison, and Future Trends," *IEEE Transaction on Vehicular Techonology*, vol. 56, no. 5, pp. 2393-2404, September 2007.

[3] V. H. Johnson, K. B. Wipke, and D. J. Rausen, "HEV Control Strategy for Real-Time Optimization of Fuel Economy and Emissions," in *2000 Future Car Congres*, Arlington, Virginia, SAE 2000-01-1543.

[4] N. J. Shouten, M. Salman, and N. Kheir, "Fuzzy Logic Control for Parallel Hybrid Vehicles," *IEEE Transaction on Control Systems Technology*, vol. 10, no. 3, pp. 460-468, May 2002.

[5] S. Ichikawa et al., "Novel Energy Management System for Hybrid Electric Vehicles Utilizing Car Navigation over a Commuting Route," in *2004 IEEE Intelligent Vehicles Symposiium*, Parma, Italy, pp. 161-166.

[6] O. Sundstrom, L. Guzzella, and P. Soltic, "Torque-Assist Hybrid Electric Powertrain Sizing: From Optimal Control Towards a Sizing Law," *IEEE Transaction on Control Systems Technology*, vol. 18, no. 4, pp. 837-849, July 2010.

[7] L. Serrao, S. Onori, and G. Rizzoni, "A Comparitive Analysis of Energy Management Strategies for Hybrid Electric Vehicles," *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, May 2011.

[8] D. Bianchi et al., "A Rule-Based Strategy for A Series/Parallel Hybrid Electric Vehicle: An Approach Based on Dynamic Programming," in *ASME 2010 Dynamic Systems and Control Conference*, Cambridge, MA, 2010.

[9] C. C. Lin, H. Pen, J. W. Grizzle, and J. Kang, "Power Management Strategy for a Parallel Hybrid Electric Truck," *IEEE Transaction on Control System Technology*, vol. 11, no. 6, pp. 839-849, 2003.

[10] L. V. Perez, G. R. Bossio, D. Moitre, and G. O. Garcia, "Optimization of Power Management in an Hybrid Electric Vehicle using Dynamic Programming," *Mathematics and Computers in Simulation*, vol. 73, pp. 244-254, 2003.

[11] P. Tulpule, V. Marano, and G. Rizzoni, "Energy Management for Plug-in Hybrid Electric Vehicles Using Equivalent Consumption Minimisation Strategy," *Int. J. Electric and Hybrid Vehicles*, vol. 2, no. 4, pp. 329-349, 2010.

[12] P. Pisu and R. Giorgio, "A Comparative Study of Supervisory Control Strategies for Hybrid Electric Vehicles," *IEEE Transactions on Control System Technology*, vol. 15, no. 3, pp. 506-518, May 2007.

[13] T. Donateo, L. Serrao, and G. Rizzoni, "A Two-Step Optimisation Method for the Preliminary Design of a Hybrid Electric Vehicle," *Int. J. Electric and Hybrid Vehicles*, vol. 1, no. 2, pp. 142-164, 2008.

[14] X. Wu, B. Cao, J. Wen, and Y. Bian, "Particle Swarm Optimization for Plug-in Hybrid Electric Vehicle Control Strategy Parameter," in *IEEE Vehicle Power and Propulsion Conference (VPPC)*, Harbin, China, 2008.

[15] B. Huang, X. Shi, and Y. Xu, "Parameter Optimization of Power Control Strategy for Series Hyrbrid Electric Vehicle," in *IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, 2006, pp. 1989-1994.

[16] Z. Wang, B. Huang, Y. Xu, and W. Li, "Optimization of Series Hybrid Electric Vehicle Operational Parameters By Simulated Annealing Algorithm," in *IEEE International Conference on Control Automation*, Guangzhou, China, 2007, pp. 1536-1541.

[17] X. Liu, Y. Wu, and J. Duan, "Optimal Sizing of a Series Hybrid Electric Vehicle Using a Hybrid Genetic Algorithm," in *Proceedings of the IEEE International Conference on Automation and Logistics*, Jinan, China, 2007, pp. 1125-1129.

[18] W. Gao and S. K. Porandla, "Design Optimization of a Parallel Hybrid Electric Powertrain," in *Vehicle Power and Propulsion Conference*, 2005, pp. 530-535.

[19] O. Hegazy and J. Van Mierlo, "Particle Swarm Optimization for Optimal Powertrain Component Sizing and Design of Fuel Cell Hybrid Electric Vehicle," in *12th International Conference on Optimization of Electrical and Electronic Equipment*, Brasov, Romania, 2010, pp. 601-609.

[20] Li-Cun Fang and Shi-Yin Qin, "Concurrent Optimization for Parameters of Powertrain and Control System of Hybrid Electric Vehicle Based on Multi-Objective Genetic Algorithm," in *SICE-ICASE International Joint Conference 2006*, Busan, Korea, 2006, pp. 2424-2429.

[21] B. Zhang, Z. Chen, C. Mi, and Y.L. Murphey, "Multi-objective Parameter Optimization of a

Series Hybrid Electric Vehicle Using Evolutionary Algorithms," in *Vehicle Power and Propulsion Conference*, Dearborn, MI, 2009, pp. 961-925.

[22] R. D. Braun, R. W. Powell, R. A. Lepsch, D. O. Stanley, and I. M. Kroo, "Comparison of Two Multidisciplinary Optimization Strategies for Launch-Vehicle Design," *Journal of Spacecraft and Rockets*, vol. 32, no. 3, pp. 404-410, 1995.

[23] J. W. Lee, Y. C. Choi, and Y. H. Byun, "Optimal Supersonic Air-Launching Rocket Design Using Multidisciplinary System Optimization Approach," in *Fuzzy Systems and Knowledge Discovery*.: Springer Berlin / Heidelberg, 2005, pp. 1108-1112.

[24] P. Wang, B. Song, Y. Wang, and L. Zhang, "Application of Concurrent Subspace Design to Shape Design of Autonomous Underwaterloo Vehicle," in *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distribted Computing*, vol. 3, 2007, pp. 1068-1071.

[25] Y. He and J. McPhee, "Multidisciplinary Optimization of Multibody Systems with Application to the Design of Rail Vehicles," *Journal of Multibody System Dynamics*, vol. 14, no. 2, pp. 111-135, 2005.

[26] Y. He and J. McPhee, "Multidisciplinary Design Optimization of Mechatronics Vehicles with Active Suspensions," *Journal of Sound and Vibration*, vol. 283, no. 1-2, pp. 217-241, 2005.

[27] S. Kodiyalam, R. J. Yan, L. Gu, and C.-H. Tho, "Multidisciplinary Design Optimization of a Vehicle System in a Scalable, High Performance Computing Environment," *Structural and Multidisciplinary Optimization*, vol. 26, no. 3-4, pp. 256-263, February 2004.

[28] S. I. Yi, J. K. Shin, and G. J. Park, "Comparison of MDO Methods with Mathematical Examples," *Journal of Structural and Multidisciplinary Optimization*, vol. 35, no. 5, pp. 391-402, 2008.

[29] E. J. Cramer, J. E. Dennis, P. D. Frank, R. M. Lewis, and G. R. Shubin, "Problem Formulation for Multidisciplinary Optimization," *SIAM Journal on Optimization*, vol. 4, no. 4, pp. 754-776, 1994.

[30] S Kodiyalam and J Sobieszczanski-Sobieski, "Multidisciplinary Design Optimization - some formal methods, framework requirements, and applications to vehicle design," *International Journal of Vehicle Design*, vol. 25, no. Special Issue 25, pp. 3-22, 2001.

[31] G. Rizzoni, L. Guzzella, and B. M. Baumann, "Unified Modeling of Hybrid Electric Vehicle

Drivetrains," *IEEE/ASME Transactions on Mechatronics*, vol. 4, no. 3, pp. 246-257, September 1999.

[32] L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*, 2nd ed. Berlin Heidelberg: Springer-Verlag, 2007.

[33] L. Guzzella and A. Amstutz. (2006, July) ETH-IMRT-QSS. [Online]. http://www.imrt.ethz.ch/research/qss

[34] L. Guzzella and C. H. Onder, *Introduction to Modeling and Control of Internal Combustion Engine Systems*. Berlin: Springer-Verlag, 2004.

[35] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, 2nd ed. New York, USA: Springer Science+Business Media, LLC. , 2006.

[36] Global Optimization Toolbox - Pattern Search Options Demo. [Online]. http://www.mathworks.com/products/global-optimization/demos.html?file=/products/demos/shipping/globaloptim/psoptionsdemo.html

[37] Fakhreddine O. Karray and Clarence De Silva, *Soft Computing and Intelligent Systems Design*, 1st ed. Essex, England: Pearson Education Limited, 2004.

[38] Singiresu S. Rao, *Engineerin Optimization, Theory and Practice*, 3rd ed. USA: John Wiley & Sons, 1996.

[39] Mathworks. (2010, July) Optimization Toolbox User's Guide. [Online]. www.mathworks.com/access/helpdesk/help/pdf_doc/optim/optim_tb.pdf

[40] Q. Gong, S. Midlam-Mohler, V. Marano, and G. Rizzoni, "An Iterative Markov Chain Approach for Generating Vehicle Driving Cycles," in *SAE Paper*, 2011-01-0880.

[41] General Motors. (2011) Chevrolet Volt. [Online]. http://www.chevrolet.com/volt/features-specs/

[42] Discover Energy. By Application. [Online]. http://discover-energy.com/productsearch?filter0=9&filter1=**ALL**&filter2=**ALL**&filter3=

[43] Maxwell Technologies. (2011) Boostcap Ultracapacitor - 16 Volt Large Modules - General Purpose Modules. [Online]. http://www.maxwell.com/products/ultracapacitors/product.aspx?PID=16V-LARGE-MODULES

[44] General Motors. Chevrolet Volt. [Online]. http://media.gm.com/volt/index.html

[45] (2007) A123Systems - Products. [Online]. http://a123systems.com/products

[46] (2007, February 13) The Daily. Study: Time with the Family. [Online]. http://www.statcan.gc.ca/daily-quotidien/070213/dq070213b-eng.htm

[47] "Climate Change 2007: Synthesis Report," Intergovernmental Panel on Climate Change, 2007.

[48] Energy Information Administration. Voluntary Reporting of Greenhouse Gases Program - Electricity Factors. [Online]. http://www.eia.doe.gov/oiaf/1605/coefficients.html

[49] Tesla Motors. (2011) Roadster Features and Specifications. [Online]. http://www.teslamotors.com/roadster/specs

[50] Mouser Electronics. (2011) MOD0110 P016 B01 Maxwell Technologies Supercapacitors. [Online]. http://ca.mouser.com/ProductDetail/Maxwell-Technologies/BMOD0110-P016-B01/?qs=JV7lzlMm3yIKJuYqYR7UEw%3d%3d

[51] K. Matsuura, K. Korematsu, and J. Tanaka, "Fuel Consumption Improvement of Vehicles by Idling Stop," in *SAE Paper*, pp. No. 2004-01-1896.

[52] M. Hong, J. Zhang, J. Li, and M. Ouyang, "Potential Fuel Consumption Improvement Analysis for Integrated Started Generator System Base on the New European Drive-cycle," in *SAE Paper*, pp. No. 2008-01-1570.

[53] H. Takahara et al., "Continuously Variable Transimission Control System for Toyota Intelligent Idling Stop System," in *SAE Paper*, pp. No. 2004-01-1635.

[54] G. Tamei et al., "Development of the Hybrid System for the Saturn VUE Hybrid," in *SAE Paper*, pp. No. 2006-01-1502.

[55] GM Fleet. Police Vehicles. [Online]. http://www.gmfleet.com/government/products/police.jsp

[56] "Fuel Economy Labeling of Motor Vehicles: Revisions To Improve Calculation of Fuel Economy Estimates; Final Rule," Environmental Protection Agency, Federal Register Vol. 71, No. 248, December 27, 2006.

[57] C. C. Lin, H. Pen, and J. W. Grizzle, "A Stochastic Control Strategy for Hybrid Electric Vehicles," in *2004 American Control COonference*, Boston, MA, 2004, pp. 4710-4715.

Appendix A

Experimental Parameters of the IC Engine and Motor-Generator

Table A-1 gives the Willans Line engine model *k* parameters used to describe the Otto and Diesel cycles.

Table A-1: Parameters of the Mean Effective Pressure Losses due to Friction [34]

Parameter	Otto Cycle	Diesel Cycle
k1	1.44E5 [Pa]	1.44E5 [Pa]
k2	0.46	0.50
k3	9.1E-4 $[s^2/m^2]$	1.1E-3 $[s^2/m^2]$
k4	0.075 [m]	0.075 [m]

Figure A-1 illustrates the experimentally determined efficiency map of the scalable motor-generator provided in the QSS toolbox. Note that the upper quadrant depicts the efficiency during motor mode, while the lower quadrant shows the generator efficiency. For ease of calculation in the backward modeling approach, the numerical value of the motor efficiency is higher than one when using Equation (3.19)

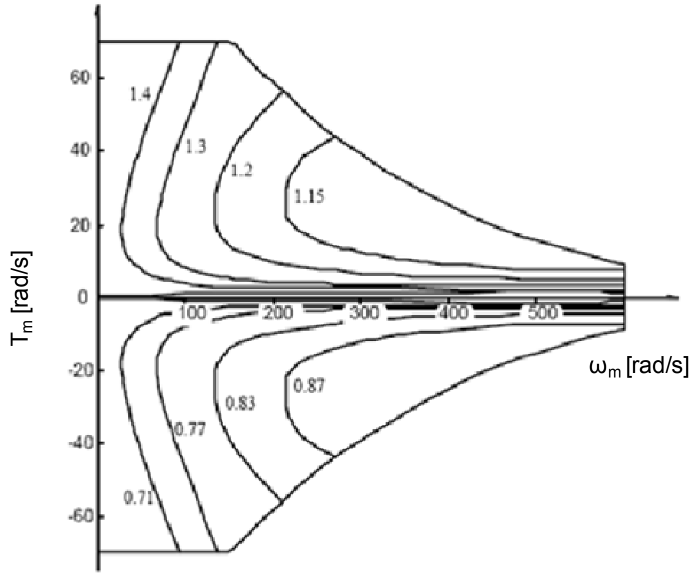


Figure A-1: Efficiency Map of the QSS Toolbox Electric Motor-Generator [33]

Appendix B

New EPA Fuel Economy Test Method

Prior to the 2008 model year, fuel economy estimate performed by the US Environmental Protection Agency (EPA) was based on two drive cycles: (i) Federal Test Procedure (FTP) 75, derived from Urban Dynamometer Driving Schedule (UDDS), for city driving, and (ii) Highway Fuel Economy Test (HWFET) for highway fuel consumption estimates. However, such methodology lacked several important factors that affected fuel economy in the real world, namely high speed, aggressive accelerations and decelerations, the use of air conditioning, and cold temperature operations. Therefore, starting with the 2008 model year, a new 5-cycle method was adopted by the US EPA in order to provide a more realistic fuel economy estimate. The test results from each of the test cycles are substituted into a series of equations to determine the city, highway, and combined fuel consumption estimates. Table B-1 summarizes the characteristics of the 5-cycle methodology. [56]

Table B-1: Fuel Economy 5-Cycle Testing Method

Test	Description	Average Speed (mph)	Max Speed (mph)	Max Acceleration (mph/sec)	Ambient Condition
FTP 75	Urban Stop-and-go driving from 1970’s	21	58	3.3	75°F
HWFET	Rural Driving	48	60	3.3	75°F
US06	High Speed and aggressive driving	48	80	8.5	75°F
SC03	Air conditioner operation	22	55	5.1	95°F & 40% relative humidity
Cold FTP 75	Cold temperature operation	21	58	3.3	20°F

Appendix C

Series Hybrid Electric Vehicle Model Parameters

Table C-1: Vehicle Model Parameters

Parameters	Values
Chassis (sprung) Mass [kg][4]	1200
Tire (unsprung) Mass [kg]	25
Tire Radius [m]	0.31
Air Density ρ [kg/m^3]	1.2
Frontal Area A [m^2]	2.26
Drag Coefficient C_d	0.32

Table C-2: Series Hybrid Transmission Gear Ratio and Operating Vehicle Speed

Vehicle Speed [km/h]	Gear Ratio
0 – 50	5
50-120	3.5
>120	1

[4] Not including battery mass

Appendix D

Battery-Only Series Hybrid Electric Vehicle Simulink Model

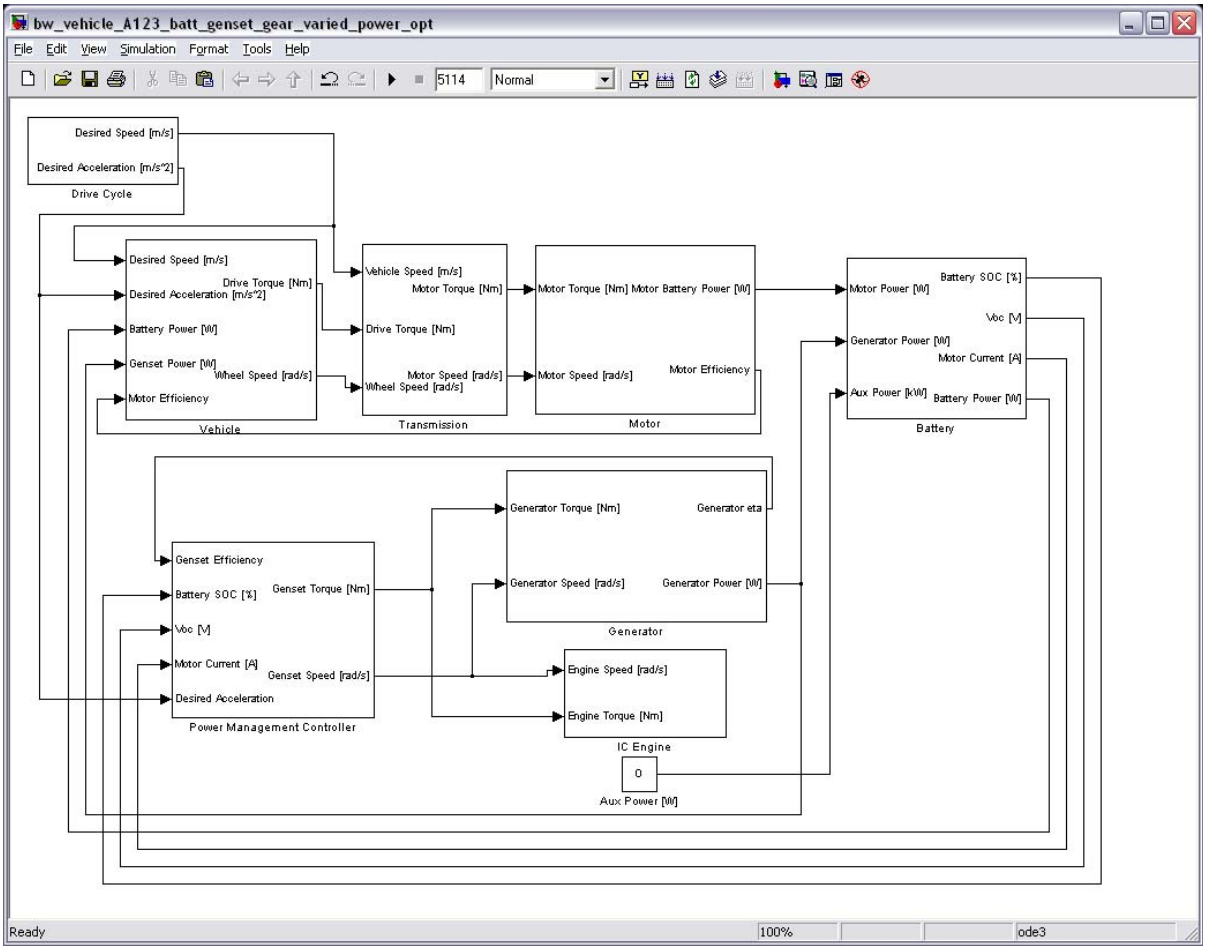


Figure D-1: Battery-Only Series Hybrid Electric Vehicle Model in MATLAB/Simulink

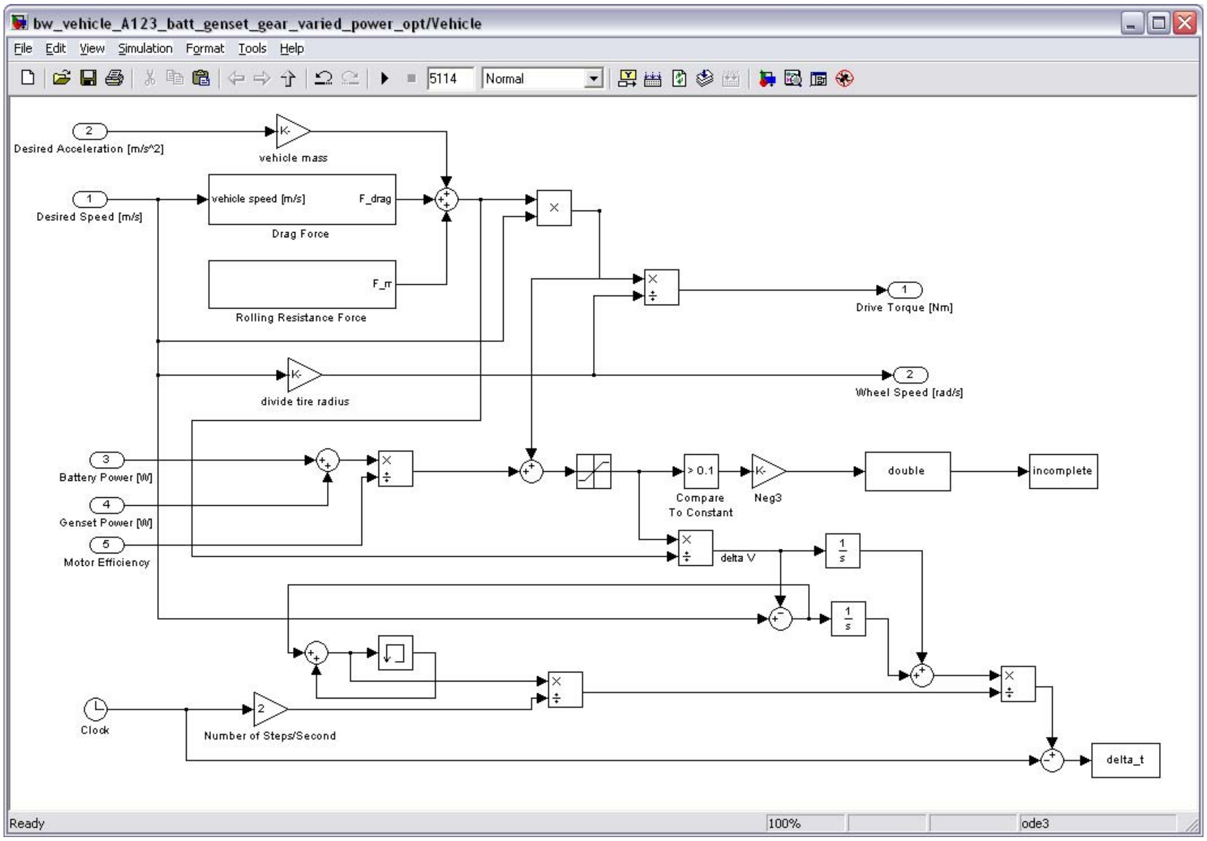


Figure D-2: Battery-Only SHEV Backward-Looking MATLAB/Simulink Vehicle Model

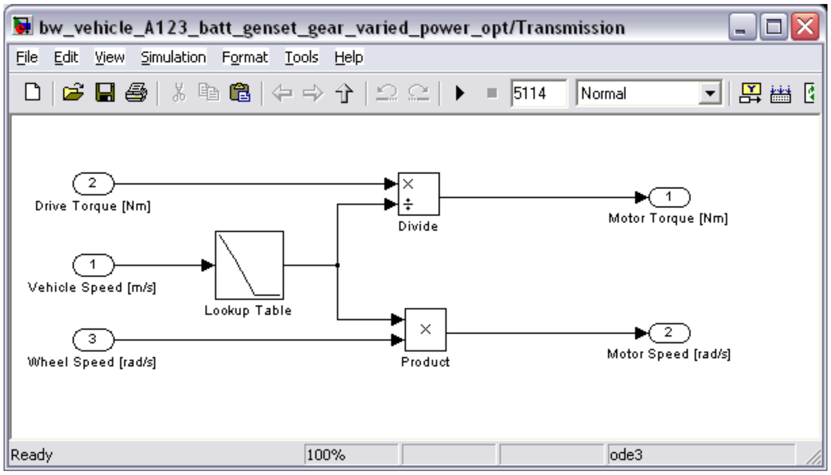


Figure D-3: Battery-Only SHEV MATLAB/Simulink Transmission Model

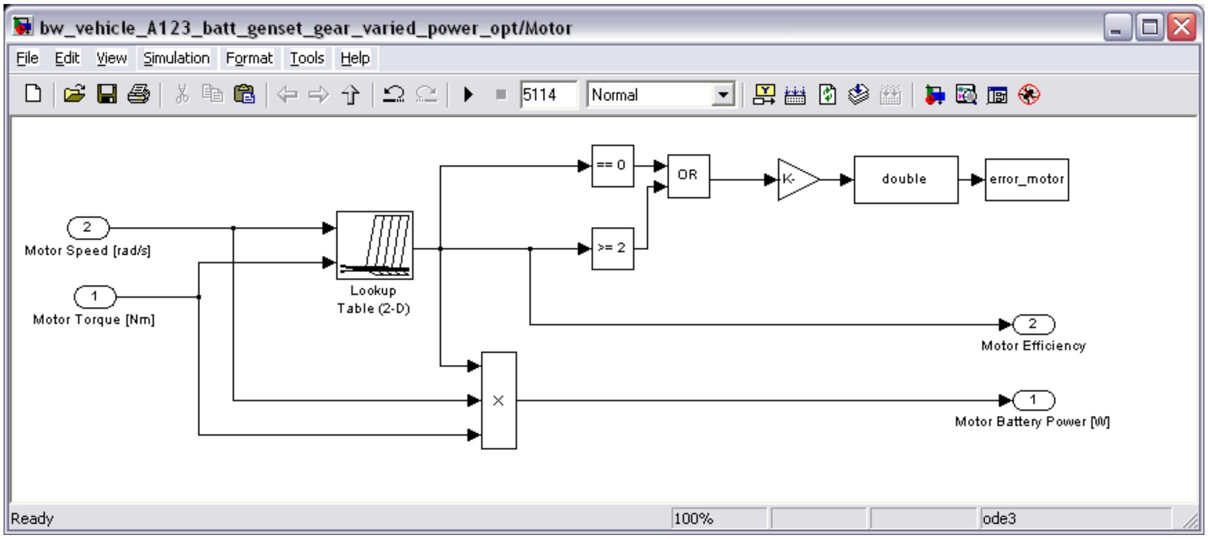


Figure D-4: Battery-Only SHEV MATLAB/Simulink Motor Model

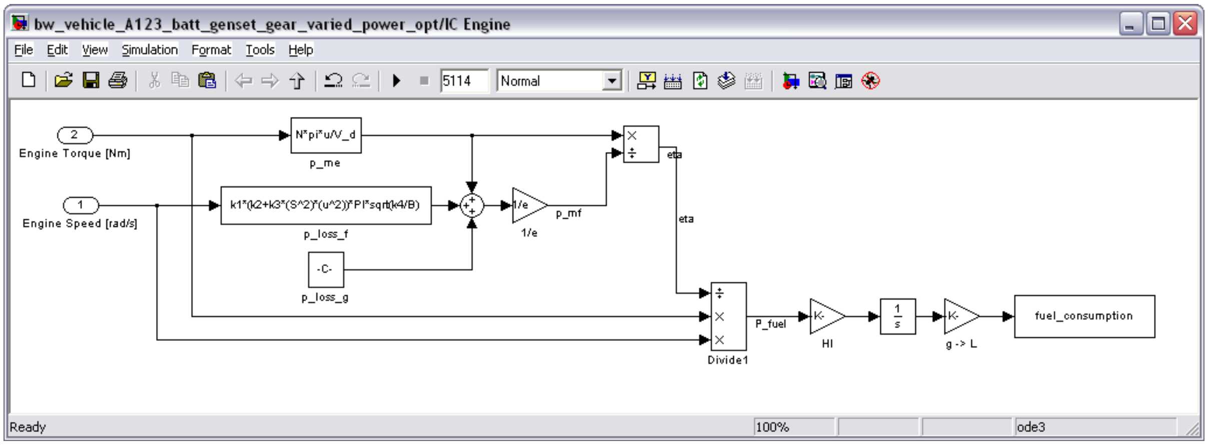


Figure D-5: Battery-Only SHEV MATLAB/Simulink IC Engine Model

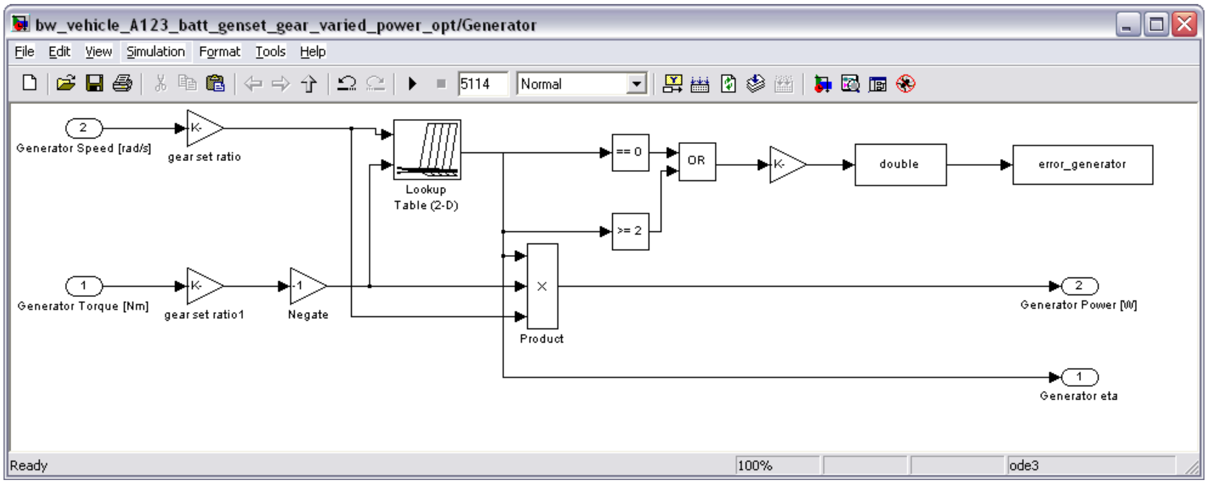


Figure D-6: Battery-Only SHEV MATLAB/Simulink Generator Model

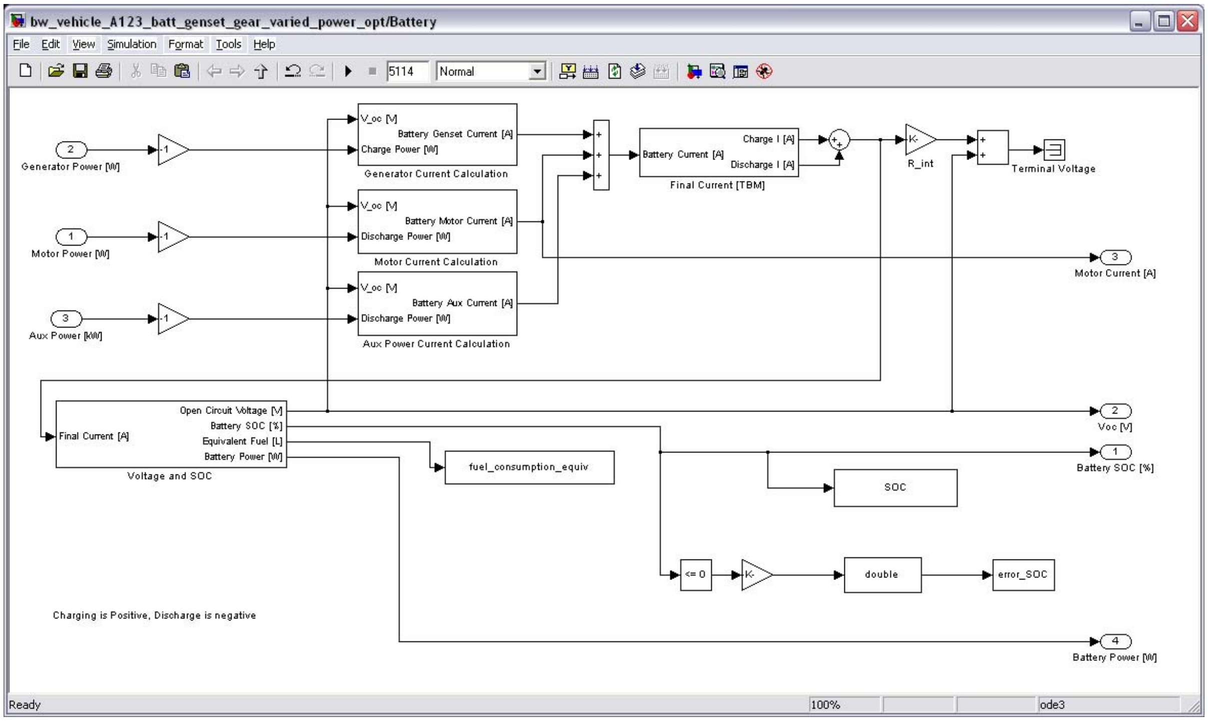


Figure D-7: Battery-Only SHEV MATLAB/Simulink Battery Model

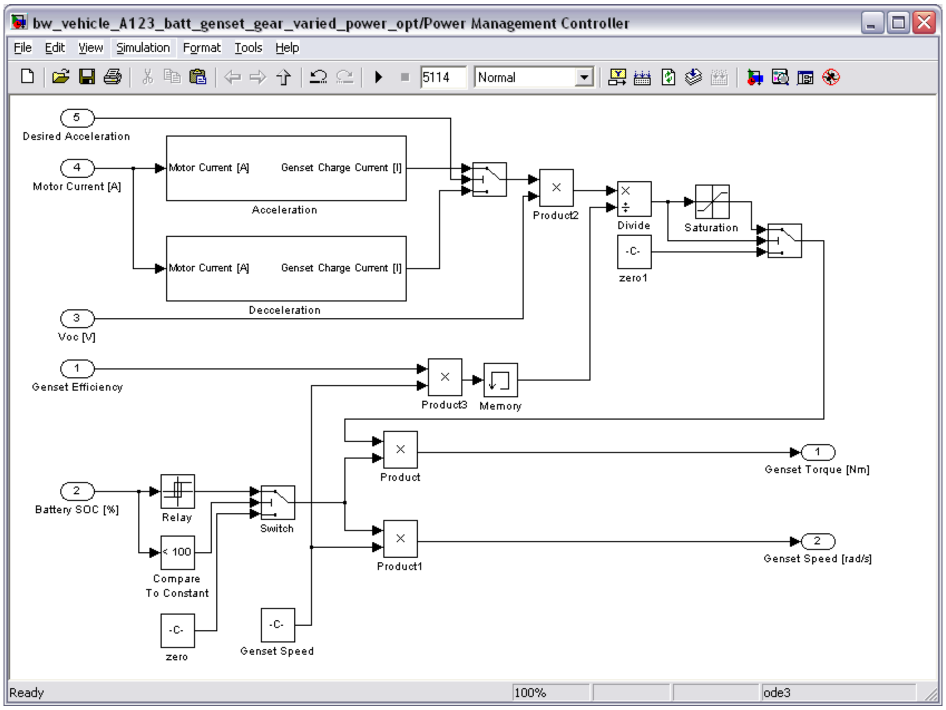


Figure D-8: Battery-Only SHEV MATLAB/Simulink Power Management Logic Model

Appendix E

MATLAB Optimization m-file

vehicle_optimize.m

```
% Vehicle Optimization
%
% This file defines the optimization parameters and execute the MATLAB
% built-in genetic algorithm command

clear
clc

matlabpool open 3

tic

% Define lower and upper bound of the GA variables
% [Batt_SOC_threshold, Batt_SOC_duration_percent, genset_torque, genset_speed,
%  battery_bank, ultracap_bank, PD_disch_c, PD_disch_i, PD_disch_e, PD_ch_c,
%  PD_ch_i, PD_ch_e, Engine Stroke (S), scale_EM ]

lb=[10  5  88 222 1 1   1   1   1   1   1   1  50  5];
ub=[95 25 116 278 9 9 150 150 150 150 150 150 120 14];

options=gaoptimset('PopulationSize',560,...
    'plotfcns',{@gaplotbestf,@gaplotbestindiv},...
    'UseParallel','always','StallGenLimit',1);

[X,FVAL,EXITFLAG,OUTPUT,POPULATION,SCORES]=ga(@vehopti,14,[],[],[],[],lb,ub,[],options);

results(1,1)=round(X(1));
results(2,1)=round(X(2));
results(3,1)=round(X(3));
results(4,1)=round(X(4));
results(5,1)=round(X(5));
results(6,1)=round(X(6));
results(7,1)=X(7);
results(8,1)=X(8);
results(9,1)=X(9);
results(10,1)=X(10);
results(11,1)=X(11);
results(12,1)=X(12);
results(13,1)=round(X(13));
results(14,1)=round(X(14));

toc

sim('bw_vehicle_batt_UC_plot.mdl')

matlabpool close
```

vehopti.m

```
function [J]=vehopti(x)

vehicle_opt;

Batt_SOC_threshold=round(x(1));
Batt_SOC_duration_percent=round(x(2));
genset_torque=round(x(3));
genset_speed=round(x(4));
battery_bank=round(x(5));
ultracap_bank=round(x(6));
PD_disch_c=x(7);
PD_disch_i=x(8);
PD_disch_e=x(9);
PD_ch_c=x(10);
PD_ch_i=x(11);
PD_ch_e=x(12);
S=round(x(13))*1e-3;
scale_EM=round(x(14));

% Fitness function parameters
years = 5;
days = 260;
cost_electric = 0.112;
cost_fuel = 0.9;
cost_battery = 18.33;
cost_ultracap = 694.29; % 2851.2;
cost_carbon = 0.034;
cost_engine_base = 2000;
cost_motor_base = 5000;
cost_increase = 2000/70;

cost_engine_final = cost_engine_base + (S*1e3 - 50)*cost_increase;
cost_motor_final = cost_motor_base + (scale_EM - 5)*cost_increase;

sim('bw_vehicle_batt_UC_opt.mdl',[],simset('SrcWorkspace','current','DstWorkspace','current')
);

J=((fuel_consumption.signals.values(size(fuel_consumption.signals.values,1))*cost_fuel...
    +fuel_consumption.signals.values(size(fuel_consumption.signals.values,1))*cost_carbon...
    -battery_consumption.signals.values(size(battery_consumption.signals.values,1))...
    *cost_electric...
    -ultracap_consumption.signals.values(size(ultracap_consumption.signals.values,1))...
    *cost_electric)*days*years+max(error_SOC_Batt.signals.values)...
    +max(error_generator.signals.values)+max(error_motor.signals.values)...
    +max(error_incomplete.signals.values)+battery_bank*battery_cell_per_bank*cost_battery...
    +ultracap_bank*ultracap_cell_per_bank*cost_ultracap)+cost_engine_final+cost_motor_final;
end
```

Appendix F

Feature-Based Optimization m-file

Feature_extraction.m

```
% Nov 27, 2010
%
% Brian Fan
%
% This m-file performs the feature based extraction using the specified
% drive cycle. The script first reads the drive cycle data points and
% calculates the acceleration between each points. Based on the specified
% resolution, it takes the average acceleration of all point from the
% current point to the next.
%
% The range of the velocity and the acceleration was first determined by
% the 3D histogram of the entire data range. The electrical energy
% consumption map was then determined by running simulation of each of bin
% indexed by the velocity and acceleration range of the histogram.
%
% The drive cycle was divided into sections, where a histogram of each of
% the sections was determined, and dot multiply by the energy map to
% calculate the energy consumption of each section. The battery SOC was
% subsequently calculated based on the energy consumption of each of the
% sections, and if SOC falls bellow the desired threshold, the genset is
% activated throughout the next section, where the SOC is recalculated. The
% process is repeated until the end of the drive cycle.
%

clc
clear all

tic

%Drive Cycle
% load HWFET.mat;  % 765 Sec
% load UDDS.mat;  % 1369 Sec
% load US06.mat;  % 600 Sec
% load NYCC.mat;  % 598 Sec
load combined_65k.mat;  % 5114 Sec
% load UDDS64k.mat;  % 7330 Sec
% load ramp20mph.mat;  % 150 Sec
% load ramp40mph.mat;  % 193 Sec
% load ramp60mph.mat;  % 225 Sec
% load constant40mph.mat;  % 99 Sec

% Create Velocity and Acceleration dataset for histogram
feature.res = 1; % resolution
feature.sec = 101; % No. of sections to divide the drive cycle
feature.binsize = 30;

drive_cycle(:,2)=drive_cycle(:,2)*1.6/3.6; % converting MPH to m/s

%%
% Generate m by 1 vector of velocity and acceleration values
if size(drive_cycle,1)/feature.res > floor(size(drive_cycle,1)/feature.res)
    feature.m=ceil(size(drive_cycle,1)/feature.res);
else
    feature.m=size(drive_cycle,1)/feature.res;
end

for i=1:feature.m
```

```
    feature.v_ext(i,1)=drive_cycle(i+(i-1)*(feature.res-1),2);
end

for i=1:feature.m-1
    feature.a_ext(i,1)=0;
    for j=1:feature.res
        feature.a(j)=drive_cycle(i+(i-1)*(feature.res-1)+j,2)-drive_cycle(i+(i-
1)*(feature.res-1)+j-1,2);
        feature.a_ext(i,1)=feature.a_ext(i,1)+feature.a(j);
    end
    feature.a_ext(i,1)=feature.a_ext(i,1)/feature.res;
end

feature.a_ext(feature.m,1)=0;
for j=1:size(drive_cycle,1)-(feature.m-1)*feature.res-1
    feature.a(j)=drive_cycle(feature.m+(feature.m-1)*(feature.res-1)+j,2)-
drive_cycle(feature.m+(feature.m-1)*(feature.res-1)+j-1,2);
    feature.a_ext(feature.m,1)=feature.a_ext(feature.m,1)+feature.a(j);
end
feature.a_ext(feature.m,1)=feature.a_ext(feature.m,1)/feature.res;

feature.ext_init(:,1)=feature.v_ext;
feature.ext_init(:,2)=feature.a_ext;

[feature.Z,feature.C]=hist3(feature.ext_init,[feature.binsize feature.binsize]);
%%

% Define velocity and acceleration range from overall histogram
feature.v_range.start = min(feature.C{1,1});
feature.v_range.end = max(feature.C{1,1});
feature.a_range.start = min(feature.C{1,2});
feature.a_range.end = max(feature.C{1,2});
feature.v_range = feature.v_range.start:((feature.v_range.end-
feature.v_range.start)/(feature.binsize-1)):feature.v_range.end; % velocity range of the
histogram
feature.a_range = feature.a_range.start:(feature.a_range.end-
feature.a_range.start)/(feature.binsize-1):feature.a_range.end; % acceleration range of the
histogram

if size(drive_cycle,1)/feature.sec/feature.res >
floor(size(drive_cycle,1)/feature.sec/feature.res)
    feature.n=ceil(size(drive_cycle,1)/feature.sec/feature.res);
else
    feature.n=size(drive_cycle,1)/feature.sec/feature.res;
end

% Generate n by sec table of velocity
for j=1:feature.sec-1
    for i=1:feature.n
        feature.v_ext_sec(i,j)=drive_cycle(feature.n*(j-1)+i+(i-1)*(feature.res-1),2);
    end
end

for i=1:size(drive_cycle,1)-feature.n*(feature.sec-1);
    feature.v_ext_sec(i,feature.sec)=drive_cycle(feature.n*(feature.sec-1)+i,2);
end

% Generate n by sec table of accceleration values
for j=1:feature.sec-1
    for i=1:feature.n
        feature.a_ext_sec(i,j)=feature.a_ext(feature.n*(j-1)+i+(i-1)*(feature.res-1));
    end
end

for i=1:size(drive_cycle,1)-feature.n*(feature.sec-1)
    feature.a_ext_sec(i,feature.sec)=feature.a_ext(feature.n*(feature.sec-1)+i);
end
```

```
%%
% Generate Energy map of the histogram range (v_range & a_range)
vehicle_feature

for i=1:size(feature.v_range,2)
    for j=1:size(feature.a_range,2)
        feature.v_init=feature.v_range(i);
        feature.a_avg=feature.a_range(j);
        sim('bw_vehicle_2011_04.mdl',feature.res);
        feature.energy_map(i,j)=(-
battery_consumption.signals.values(size(battery_consumption.signals.values,1))-...
ultracap_consumption.signals.values(size(ultracap_consumption.signals.values,1)))*3600;
    end
end
%%

toc
% pause

matlabpool open 3

tic

feature_variables;

% GA Optimization starts here
[X,FVAL,EXITFLAG,OUTPUT,POPULATION,SCORES,feature]=vehopti_fea_UC(feature);

results(1,1)=round(X(1));
results(2,1)=round(X(2));
results(3,1)=round(X(3));
results(4,1)=round(X(4));
results(5,1)=round(X(5));
results(6,1)=round(X(6));
results(7,1)=X(7);
results(8,1)=X(8);
results(9,1)=X(9);
results(10,1)=X(10);
results(11,1)=round(X(11));
results(12,1)=round(X(12));

toc

matlabpool close
```

vehopti_fea_UC.m

```
function [X,FVAL,EXITFLAG,OUTPUT,POPULATION,SCORES, feature]=vehopti_fea_UC(feature)

% Define lower and upper bound of the GA variables
% [Batt_SOC_threshold, genset_torque, genset_speed, battery_bank, ultracap_bank,
%  PD_disch_c, PD_disch_g, PD_disch_i, PD_ch_c, PD_ch_g, PD_ch_i, Batt_SOC_duration_percent,
%  Engine Stroke (S), scale_EM ]

lb=[10  5  88 222 1 1   1   1   1   1  50  5];
ub=[95 25 116 278 9 9 150 150 150 150 120 14];

options=gaoptimset('PopulationSize',560,'StallGenLimit',10,...
    'plotfcns',{@gaplotbestf,@gaplotbestindiv},'TolFun',1,'UseParallel','always');

% Figure out passing variables
[X,FVAL,EXITFLAG,OUTPUT,POPULATION,SCORES]=ga(@vehopti_feat,12,[],[],[],[],lb,ub,[],options);
results=round(X);

   function [J]=vehopti_feat(x)

% Note: not optimizaing dP/dt parameter (charge/discharge i), since dP/dt
% not used in feature based PDF

feature.Batt_SOC_threshold=round(x(1));
feature.Batt_SOC_duration_percent=round(x(2));
feature.genset_torque=round(x(3));
feature.genset_speed=round(x(4));
feature.battery_bank=round(x(5));
feature.ultracap_bank=round(x(6));
feature.PD_disch_c=x(7);
feature.PD_disch_e=x(8);
feature.PD_ch_c=x(9);
feature.PD_ch_e=x(10);
feature.S=round(x(11))*1e-3;
feature.scale_EM=round(x(12));

% Battery Energy [kJ]
feature.batt_capacity_init=feature.battery_capacity*feature.battery_nominal_voltage...
*feature.battery_cell_per_bank*feature.battery_bank/1000*feature.battery_init_SOC/100*3600;

feature.batt_capacity=feature.battery_capacity*feature.battery_nominal_voltage...
    *feature.battery_cell_per_bank*feature.battery_bank/1000*3600;

% Ultra-capacitor Energy [kJ]
feature.UC_Q=(feature.ultracap_capacitance/feature.ultracap_cell_per_bank...
*feature.ultracap_bank)*(feature.ultracap_nominal_voltage*feature.ultracap_cell_per_bank);

feature.UC_Q_init=feature.UC_Q*feature.ultracap_init_SOC/100;

feature.UC_C =
feature.ultracap_capacitance/feature.ultracap_cell_per_bank*feature.ultracap_bank;

feature.UC_capacity_init=(feature.UC_Q_init)^2/feature.UC_C/2/1000;
feature.UC_capacity=(feature.UC_Q)^2/feature.UC_C/2/1000;

% Calculate energy consumption per section
feature.genset_flag=0;

i=1;
repeat=0;
feature.energy_section=0;

% Power Available
feature.mass_new=feature.M+feature.battery_cell_mass*feature.battery_cell_per_bank*...
    feature.battery_bank+feature.ultracap_cell_mass*feature.ultracap_cell_per_bank*...
    feature.ultracap_bank+4*feature.m_T;

feature.power_batt=feature.battery_discharge_peak*feature.battery_bank*...
    feature.battery_nominal_voltage*feature.battery_cell_per_bank;

feature.power_UC=feature.ultracap_discharge_peak*feature.ultracap_bank*...
    feature.ultracap_nominal_voltage*feature.ultracap_cell_per_bank;

% Max Power of Section
feature.power_desired=(feature.mass_new.*feature.a_ext + (feature.Cd*feature.rho.*...
   (feature.v_ext).^2*feature.A)/2 + feature.Crr*feature.mass_new*9.81).*feature.v_ext;

% Calculate Performance Constraints
if max(feature.power_desired)/feature.motor_traction_eta>feature.power_batt+feature.power_UC
    feature.error_incomplete(i) = 100000;
else
    feature.error_incomplete(i) = 0;
end


% Calculate energy map increase based on new mass
for n=1:size(feature.v_range,2)
    for m=1:size(feature.a_range,2)
        feature.energy_map_scale(n,m)=(feature.mass_new*feature.a_range(m)+...
            (feature.Cd*feature.rho.*(feature.v_range(n)).^2*feature.A)/2 +...
            feature.Crr*feature.mass_new*9.81)/(feature.mass_base*feature.a_range(m)+...
            (feature.Cd*feature.rho.*(feature.v_range(n)).^2*feature.A)/2 +...
            feature.Crr*feature.mass_base*9.81);
    end
end

% Energy calculation of sections
while i<=feature.sec
    feature.ext(:,1)=feature.v_ext_sec(:,i);
    feature.ext(:,2)=feature.a_ext_sec(:,i);
    feature.Z=hist3(feature.ext,{feature.v_range feature.a_range});
    feature.energy(i)=sum(sum(feature.Z.*feature.energy_map));
    feature.energy_section=feature.Z+feature.energy_section;

    % Power Distributing Function
    if i==1
        % Use SOC init
        feature.PDF_ch(i)=(1/((1+exp(feature.PD_ch_b*(feature.battery_init_SOC-
            feature.PD_ch_c)))...
            *(1+exp(feature.PD_ch_d*(feature.ultracap_init_SOC-feature.PD_ch_e)))))...
            *(feature.PD_ch_g/(feature.energy(i)/feature.n));

        feature.PDF_dis(i)=1/((1+exp(feature.PD_disch_b.*(feature.battery_init_SOC-
            feature.PD_disch_c)))...
            *(1+exp(feature.PD_disch_f*((feature.energy(i)/feature.n)-
            feature.PD_disch_g)))...
            *(1+exp(feature.PD_disch_d*(feature.ultracap_init_SOC-feature.PD_disch_e))));
    else
        % Use SOC(i-1)
        feature.PDF_ch(i)=(1/((1+exp(feature.PD_ch_b*(feature.batt_SOC(i-1)-
            feature.PD_ch_c)))...
            *(1+exp(feature.PD_ch_d*(feature.UC_SOC(i-1)-feature.PD_ch_e)))))...
            *(feature.PD_ch_g/feature.energy(i)/feature.n);

        feature.PDF_dis(i)=1/((1+exp(feature.PD_disch_b.*(feature.batt_SOC(i-1)-
            feature.PD_disch_c)))...
            *(1+exp(feature.PD_disch_f*((feature.energy(i)/feature.n)-
            feature.PD_disch_g)))...
            *(1+exp(feature.PD_disch_d*(feature.UC_SOC(i-1)-feature.PD_disch_e))));
    end

    % Genset
```

```
if feature.genset_flag==1;
    % Calculate genset power
    feature.eta_g=interp2(feature.T_EM_col_genset, feature.w_EM_row, feature.eta_EM_map,
        -feature.genset_torque/feature.genset_gear,
        feature.genset_speed*feature.genset_gear);

    feature.genset(i)=feature.genset_torque*feature.genset_speed*feature.eta_g/1000*(featu
    re.n-1);

    % Calculate fuel consumption
    feature.p_me=feature.N*pi*feature.genset_torque/feature.V_d;

    feature.p_loss_f=feature.k1*(feature.k2+feature.k3*(feature.S^2)*(feature.genset_speed
    ^2))*feature.PI*sqrt(feature.k4/feature.B);

    feature.eta_e=feature.p_me/((feature.p_loss_f+feature.p_me+feature.p_loss_g)/feature.e
    );
    feature.P_fuel=feature.genset_torque*feature.genset_speed/feature.eta_e;
    feature.mf_dot=feature.P_fuel/(feature.gas_energy_density*1000); %[g/s]
    feature.mf=feature.mf_dot/feature.gas_vol_density; % [L]
    feature.fuel(i)=feature.mf*(feature.n-1);
else
    feature.genset(i)=0;
    feature.fuel(i)=0;
end

% Assign charging or discharging PDF based on energy of the section
if feature.energy(i)>0
    feature.PDF(i)=feature.PDF_dis(i);
else
    feature.PDF(i)=feature.PDF_ch(i);
end

feature.batt_SOC(i)=min(100,(feature.batt_capacity_init-(sum(feature.energy))...
    *feature.PDF(i)+(sum(feature.genset))*feature.PDF_ch(i))/feature.batt_capacity*100);

feature.UC_SOC(i)=min(100,(feature.UC_capacity_init-(sum(feature.energy))*(1-
     feature.PDF(i))+(sum(feature.genset))*(1-
     feature.PDF_ch(i)))/feature.UC_capacity*100);

% Calculate EES SOC error function
if feature.batt_SOC(i)<=0
    feature.error_SOC_Batt = 1000000;
else
    feature.error_SOC_Batt = 0;
end

if feature.UC_SOC(i)<=0
    feature.error_SOC_UC = 1000000;
else
    feature.error_SOC_UC = 0;
end

if (feature.batt_SOC(i) <= feature.Batt_SOC_threshold)&&(repeat==0) %check if SOC <
        threshold, and whether section repeated
    feature.genset_flag=1;
    i=i;
    repeat=1;
elseif (feature.batt_SOC(i) <= feature.Batt_SOC_threshold)&&(repeat==1) %check if SOC <
        threshold, after section repeated
    feature.genset_flag=1;
    i=i+1;
    repeat=0;
else
    feature.genset_flag=0;
    i=i+1;
    repeat=0;
    end
end



feature.results(:,1)=feature.energy';
feature.results(:,2)=feature.genset';
feature.results(:,3)=feature.fuel';
feature.results(:,4)=feature.batt_SOC';

feature.results_sum(1,1)=sum(feature.energy); % [kJ]
feature.results_sum(2,1)=sum(feature.genset); % [kJ]
feature.results_sum(3,1)=sum(feature.fuel);

% fuel_total=sum(feature.fuel)
% energy_total=sum(feature.energy)

% Fitness function parameters
years = 5;
days = 260;
cost_electric = 0.112;
cost_fuel = 0.9;
cost_battery = 18.33;
cost_ultracap = 694.29; %2851.2;
cost_carbon = 0.034;
cost_engine_base = 2000;
cost_motor_base = 5000;
cost_increase = 2000/70;

feature.cost_engine_final = cost_engine_base + (feature.S*1e3 - 50)*cost_increase;
feature.cost_motor_final = cost_motor_base + (feature.scale_EM - 5)*cost_increase;

J=((sum(feature.fuel)*cost_fuel+sum(feature.fuel)*cost_carbon...
    +(sum(feature.energy)-sum(feature.genset))/3600*cost_electric)*days*years...
    +feature.error_SOC_Batt+feature.error_SOC_UC+max(feature.error_incomplete)...
    +feature.battery_bank*feature.battery_cell_per_bank*cost_battery...
    +feature.ultracap_bank*feature.ultracap_cell_per_bank*cost_ultracap)...
    +feature.cost_engine_final+feature.cost_motor_final;

   end

end
```

Appendix G

Ultracapacitor and Power Distributing Function Simulink Model

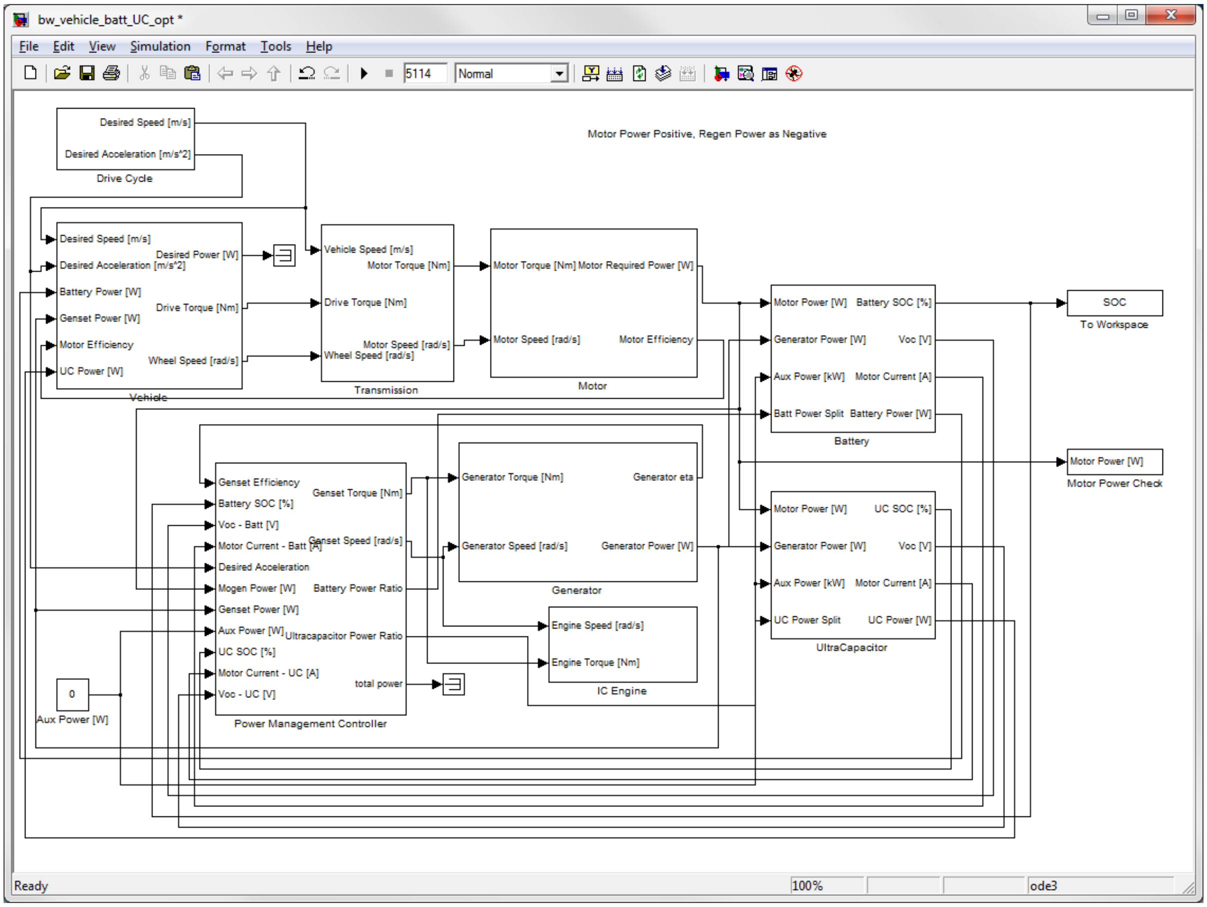
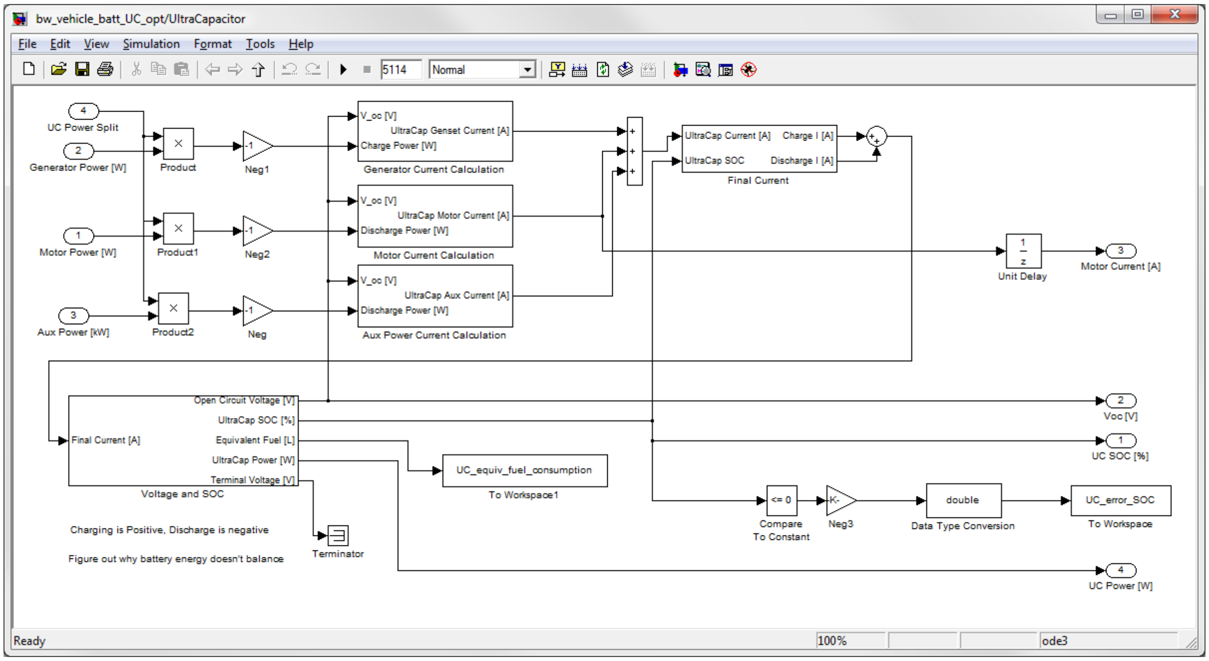


Figure G-1: Combined SHEV MATLAB/Simulink Vehicle Model



Charging is Positive, Discharge is negative

Figure out why battery energy doesn't balance

Figure G-2: Combined SHEV MATLAB/Simulink Ultracapacitor Model

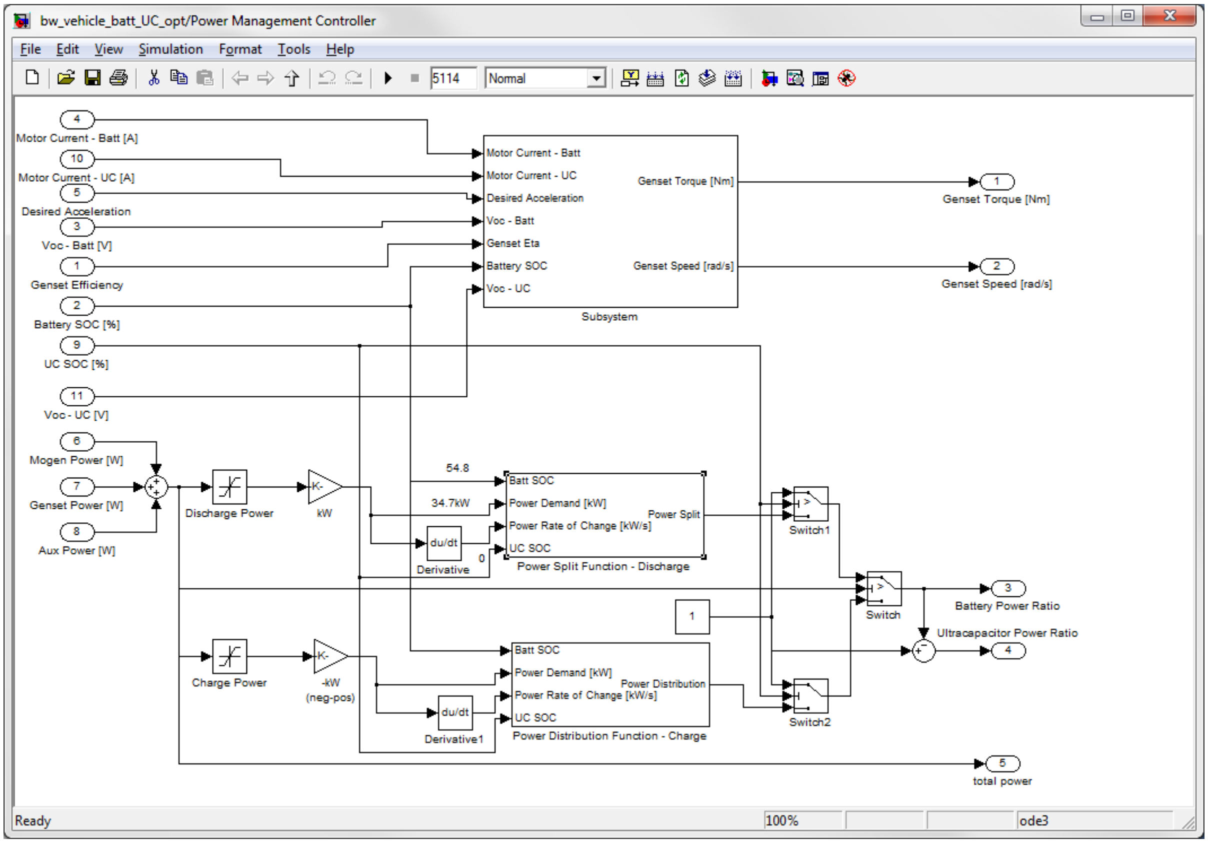
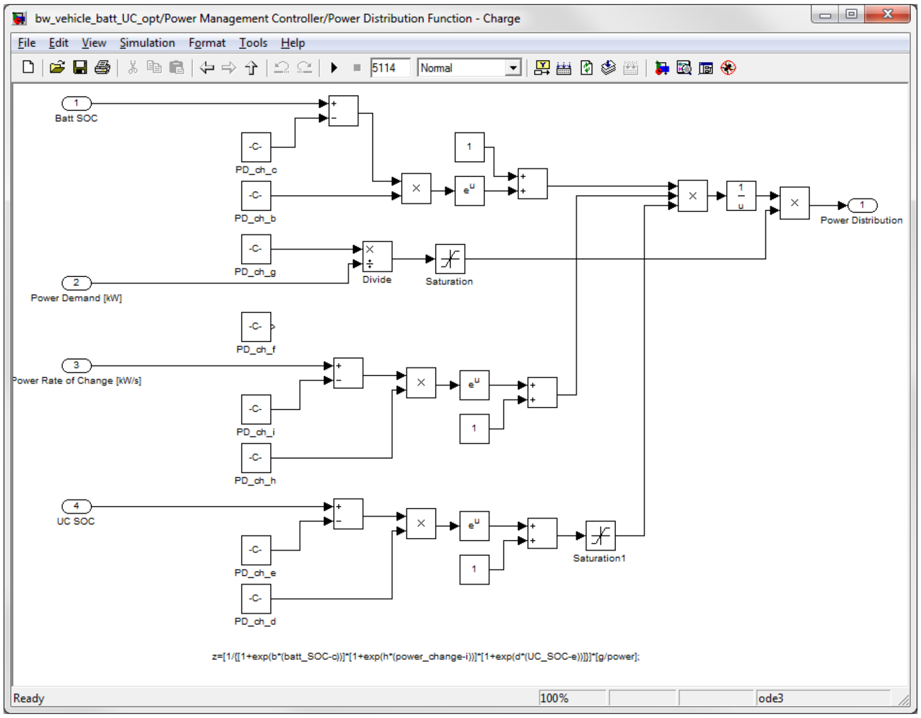
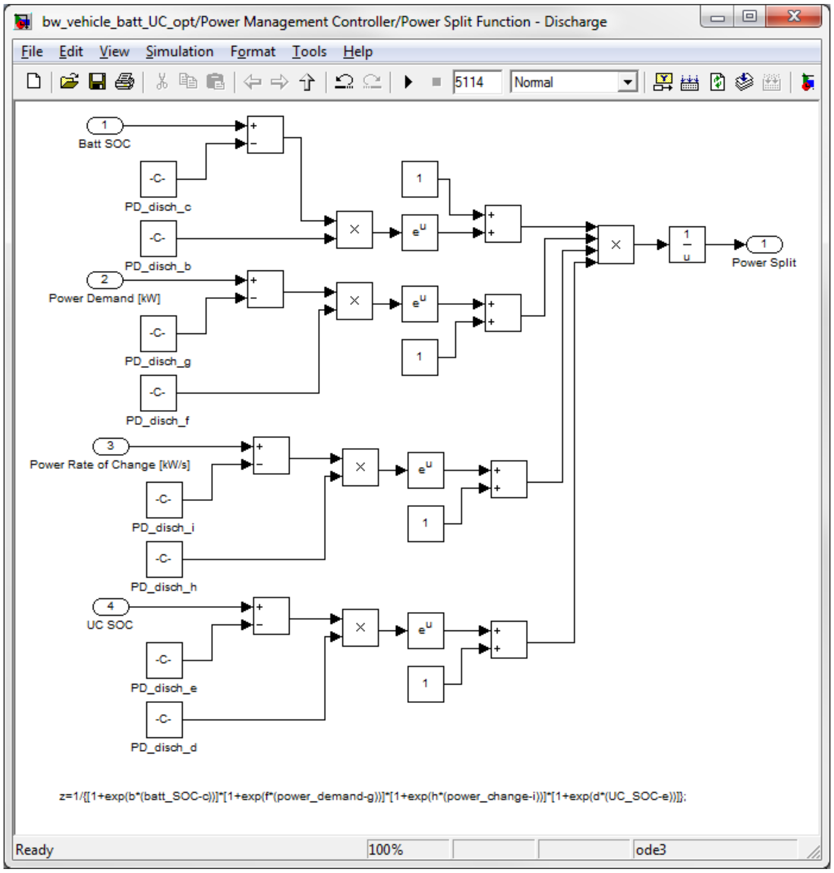


Figure G-3: Combined SHEV MATLAB/Simulink Power Management Controller Model



```
z=[1/{[1+exp(b*(batt_SOC-c))]*[1+exp(h*(power_change-i))]*[1+exp(d*(UC_SOC-e))]}]*[g/power];
```

Figure G-4: Combined SHEV MATLAB/Simulink Charge Power Distributing Function



```
z=1/{[1+exp(b*(batt_SOC-c))]*[1+exp(f*(power_demand-g))]*[1+exp(h*(power_change-i))]*[1+exp(d*(UC_SOC-e))]};
```

Figure G-5: Combined SHEV MATLAB/Simulink Discharge Power Distributing Function

Appendix H
Anti-Idling System Simulink Model

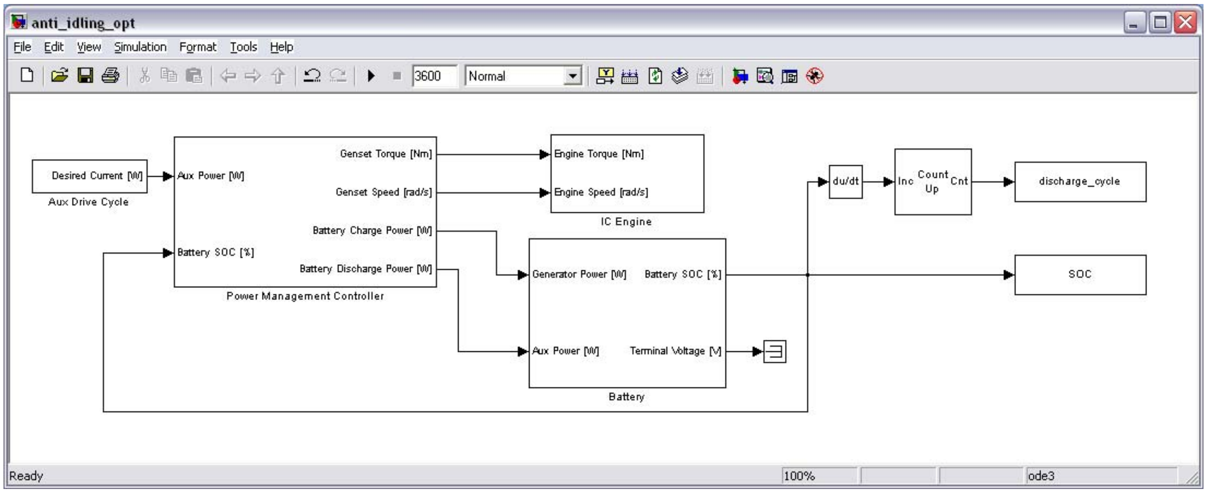


Figure H-1: Anti-Idling System Model in MATLAB/Simulink

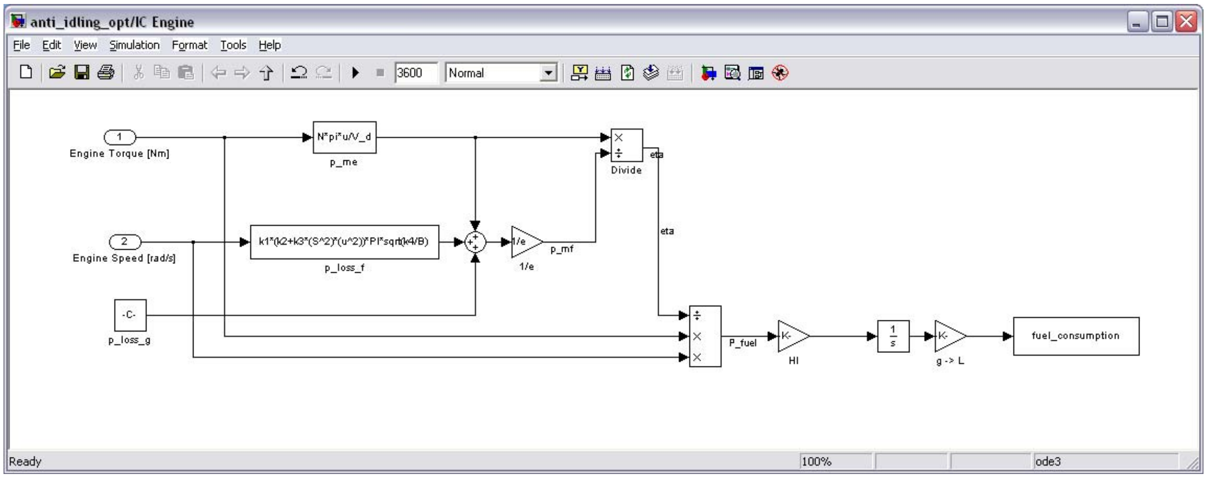


Figure H-2: Anti-Idling MATLAB/Simulink IC Engine Model

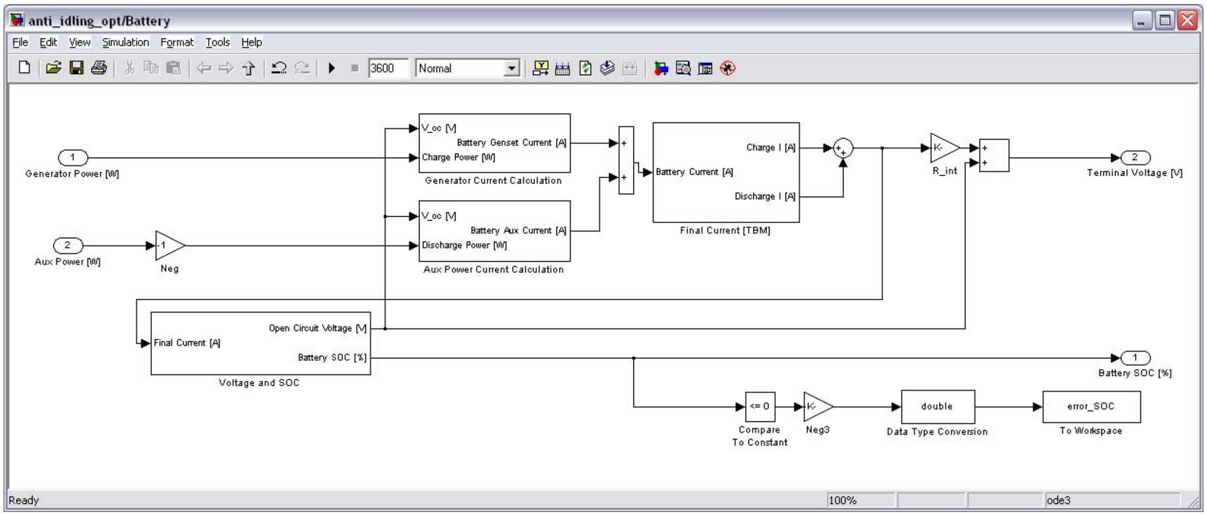


Figure H-3: Anti-Idling MATLAB/Simulink Battery Model

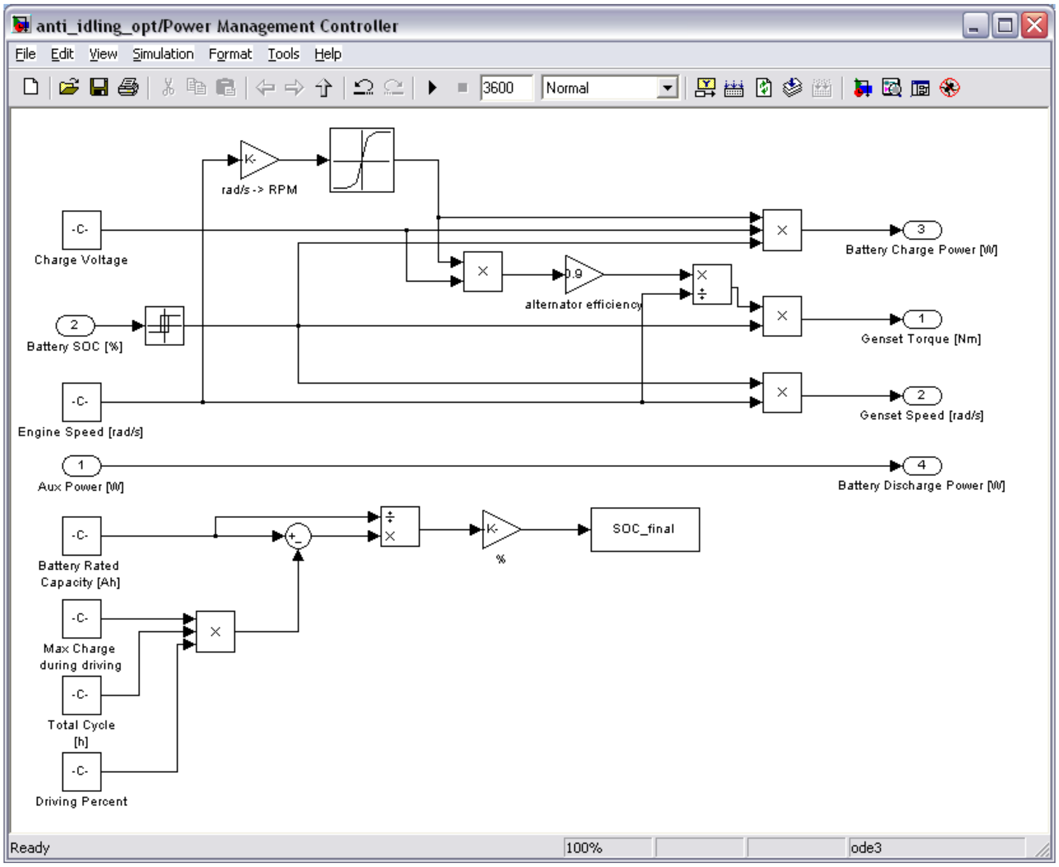


Figure H-4: Anti-Idling MATLAB/Simulink Power Management Logic Model

Appendix I

Anti-Idling System Model Parameters

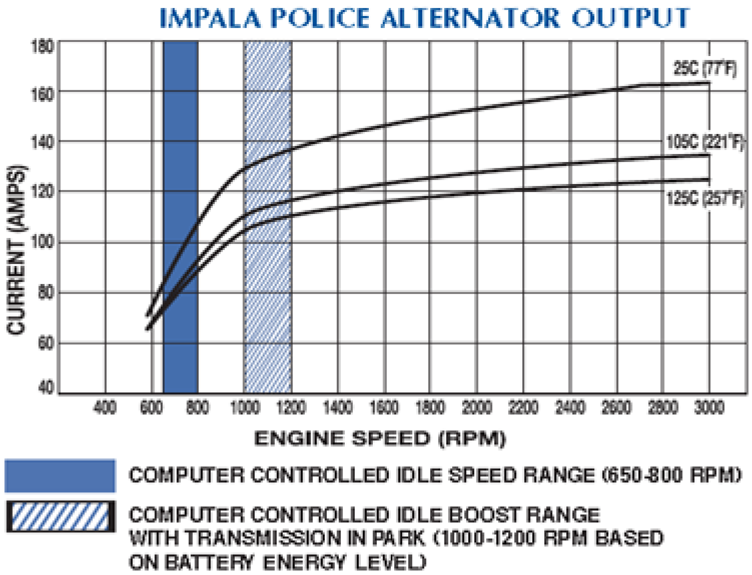


Figure I-1: Anti-Idling Alternator Current Output

Table I-1: Cycle Life of the Batteries

Discover (Lead-Acid)		**A123 (Lithium-Ion)**	
Depth of Discharge	Cycle Life	Depth of Discharge	Cycle Life
>80%	300	100%	1000
>50%	700		
>30%	1200		
<30%	3000		