Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Emrah Tolga Yildiz

Entitled
Nonlinear Constrained Component Optimization of a Plug-in Hybrid Electric Vehicle

For the degree of Master of Science in Mechanical Engineering

Is approved by the final examining committee:

Sohel Anwar
Chair

Yaobin Chen

Afshin Izadian

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Sohel Anwar

Approved by: Jie Chen 12/7/2010
Head of the Graduate Program Date

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

Nonlinear Constrained Component Optimization of a Plug-in Hybrid Electric Vehicle

For the degree of Master of Science in Mechanical Engineering

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Emrah Tolga Yildiz

Printed Name and Signature of Candidate

11/22/2010

Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

NONLINEAR CONSTRAINED COMPONENT OPTIMIZATION OF A PLUG-IN HYBRID ELECTRIC VEHICLE

A Thesis
Submitted to the Faculty
of
Purdue University
by
Emrah Tolga Yildiz

In Partial Fulfillment of the
Requirements for the Degree
of
Master of Science in Mechanical Engineering

December 2010
Purdue University
Indianapolis, Indiana

To my family

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge my professors, Sohel Anwar, Yaobin Chen and Afshin Izadian for their guidance, assistance and supervision during the entire course of this research.

Very special thanks to my thesis advisor Dr. Anwar, for sharing his significant research experience and supervision with me generously throughout the completion of this work.

I would also like to thank my research fellows at the Mechatronics Laboratory, Mr. Harpreetsingh Banvait and Mr. Umut Tugsal for their tremendous help and support. I also thank my department assistant to the chair Mrs. Ginger Lauderback and graduate engineering program coordinator Mrs. Valerie Lim Diemer.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ABSTRACT

Yildiz, Emrah Tolga. M.S.M.E., Purdue University, December 2010. Nonlinear Constrained Component Optimization of a Plug-in Hybrid Electric Vehicle. Major Professor: Sohel Anwar.

Today transportation is one of the rapidly evolving technologies in the world. With the stringent mandatory emission regulations and high fuel prices, researchers and manufacturers are ever increasingly pushed to the frontiers of research in pursuit of alternative propulsion systems. Electrically propelled vehicles are one of the most promising solutions among all the other alternatives, as far as; reliability, availability, feasibility and safety issues are concerned. However, the shortcomings of a fully electric vehicle in fulfilling all performance requirements make the electrification of the conventional engine powered vehicles in the form of a plug-in hybrid electric vehicle (PHEV) the most feasible propulsion systems. The optimal combination of the properly sized components such as internal combustion engine, electric motor, energy storage unit are crucial for the vehicle to meet the performance requirements, improve fuel efficiency, reduce emissions, and cost effectiveness.

In this thesis an application of Particle Swarm Optimization (PSO) approach to optimally size the vehicle powertrain components (e.g. engine power, electric motor power, and battery energy capacity) while meeting all the critical performance requirements, such as acceleration, grade and maximum speed is studied. Compared to conventional optimization methods, PSO handles the nonlinear constrained optimization problems more efficiently and precisely.

The PHEV powertrain configuration with the determined sizes of the components has been used in a new vehicle model in PSAT (Powertrain System Analysis Toolkit) platform. The simulation results show that with the optimized component sizes of the PHEV vehicle (via PSO), the performance and the fuel efficiency of the vehicle are significantly improved.

The optimal solution of the component sizes found in this research increased the performance and the fuel efficiency of the vehicle. Furthermore, after reaching the desired values of the component sizes that meet all the performance requirements, the overall emission of hazardous pollutants from the PHEV powertrain is included in the optimization problem in order to obtain updated PHEV component sizes that would also meet additional design specifications and requirements.

1 INTRODUCTION

A hybrid electric vehicle is a type of hybrid vehicle, which utilizes the combination of a conventional internal combustion engine propulsion system and electric propulsion system. The existence of electric propulsion system is intended to enhance the fuel economy, reduce pollutant emissions and/or improve the performance.

The idea of hybrid electric vehicle (HEV) belongs to Prof. Ferdinand Porsche in 1899. The next 30 years manufacturers made various concepts. This technology was not a center of interest after the early development period for a long time. However, in 1990s, researchers and manufacturers started intensely leaning on improving the HEV technology. Its potential of being highly fuel-efficient and significantly low levels of emissions made this technology one of the brightest research subjects of the era.

Plug-in hybrid electric vehicle (PHEV) is a modified version of an HEV in which the vehicle has a relatively larger energy storage system (ESS) when compared with HEV that can be charged by external sources as well as the internal sources such as regenerative braking, generator, etc. In order to minimize the usage of gasoline engine and to utilize more of the energy stored in the energy storage system (ESS), the energy from the utility grid is used to recharge the ESS with plug-in charging capability. Thus, Plug-in Hybrid Electric Vehicle (PHEV) holds the promise to further improve the energy efficiency and reduce environmental cost of a vehicle. GM manufactured the first PHEV in 1969 that was using lead acid batteries as ESS [1]. However, the last decade was the bright era of these vehicles. Due to its significant improvement in fuel consumption, manufacturers are increasingly interested in improving PHEVs.

1.1 Problem Statement

As mentioned earlier the powertrain components are crucial for the vehicle performance, fuel efficiency and emissions. Configuring the right sizes of powertrain components to meet designated goals or to improve the performance is a complex and a significant problem to resolve. In order to determine the sizes of these components, such as; engine, electrical motor and energy storage system, various methods have been used so far and some significant achievements were made.

Increased power of mathematical computations and related software resulted in various advanced techniques and solutions methods for optimization problems. Particle Swarm Optimization (PSO) is one of such methodologies for solving nonlinear constrained optimization problems such as the problem under consideration.

The increase of energy storage capacity of ESS consequently increases the cost and mass of the vehicle; on the other hand, the increase of the engine size is somewhat redundant if the available configuration is sufficient to provide the required range, while maintaining the performance requirement set by the current commercial standards. To address these issues, which involve multiple nonlinear boundary conditions, Particle Swarm Optimization (PSO) is applied to determine the optimal sizing of components (e.g. engine power, electric motor power, battery energy capacity) for a PHEV vehicle in this research. The cost function and boundaries are determined by the dynamic equation representation of the performance requirements and design constraints. Maximum possible sizing values of three most significant components, engine, electric motor and the battery, are used in determining the major nonlinear constraints of the optimization problem.

Therefore, a search for an optimal configuration of PHEV powertrain components is the main essence of this work. The increased importance of this specific type of combined propulsion systems, also, made this area a promising research subject. Thus, a

PSO algorithm is developed to find the optimal sizes of the powertrain components to achieve the desired performance objectives.

1.2 Literature Survey

There has been lots of research done in this field for both HEVs and PHEVs. Since the fuel consumption, emission levels and performance requirements depend on the powertrain components and configuration of these vehicles, component sizing was one of the branches that researchers gravitated around.

Assanis did a component optimization on series and parallel HEVs separately by integrating vehicle and engine simulations. He used a modified feed forward model for the engine simulation to link it with the vehicle simulation. He found the optimal sizes of components by using a gradient free algorithm to minimize the fuel consumption while meeting the performance requirements [2].

Fellini does an optimization of component sizes for hybrid diesel-electric vehicles. He derived the mathematical model of the vehicle and the powertrain components instead of using vehicle models. This allowed him to use different optimization tools for his sizing. Making a comparison of different algorithms the optimum component sizes were deliberately derived in this research [3].

Galdi used a genetic-based methodology to size major components of an HEV. In this research, reduction of pollutant emissions was included in the objective as well as the fuel economy improvement. Also, critical energy flow management parameters were integrated into the cost function to better enhance the optimal component sizes. The research aimed to minimize an objective function which takes into account not only technical specifications but also environmental, social, and economic aspects [4]. Montezari did another component sizing research with genetic algorithm. In his research he used parallel HEV drivetrain and computed his simulations by three different drive cycles to compare them in order to reach and test the optimum sizes of the drivetrain components [5].

Zhengli did a research on powertrain design by optimal sizing of a series HEV using an adaptive based hybrid genetic algorithm [6]. Similarly, Liu, using a hybrid genetic algorithm searched for the optimal sizes of components for a series HEV [7].

Hasanzadeh introduced an HEV simulation tool with an HEXA optimal sizing method combining optimization algorithm. Then, he developed a real-coded, adaptive based hybrid genetic algorithm and applied to the optimal sizing of a series hybrid electric vehicle. He used ADVISOR2002 as the vehicle simulator [8].

Markel, in his research, he used equivalent fuel consumption method, offline for computing a compromise solution to generate optimum power distribution between the hybrid components for a given driving cycle. Then he used ADVISOR for simulations to find the optimal component sizes by genetic algorithm [9].

Wu did an optimization on components sizes for Parallel HEVs using particle swarm optimization technique [10]. Another Parallel HEV powertrain component sizing optimization research was done by Gao [11]. He used global optimization algorithms, DIRECT (Divided RECTangles), simulated annealing, and genetic algorithm and compared the results of those three.

1.3 About This Thesis

A large number of optimization methodologies and different drivetrain configurations of PHEVs are explored in this research. It was observed that there are different paths that can be taken to solve this component optimization problem of which PSO methodology was found to be most attractive.

Chapter 2, Modeling, the mathematical modeling of a PHEV is explained in terms of each major component, such as; engine, electric motor, energy storage system, transmission, differential as well as ground contact and the driver as they are used in the PSAT software for PSO optimization in search for optimal component sizes.

Chapter 3, Problem Formulation, explains the development of the optimization algorithm and the cooperation of the simulation model along with this optimization tool.

Chapter 4, Simulation, describes the construction of the simulation model, the tools that are used to build the vehicle model and the Matlab script that helps the two powerful computation algorithms, optimization method and simulation model, work together in harmony. Finally, the results of the optimum component sizes, the comparison of the baseline model with the optimized model and the improvements in the PHEV model are presented in the secondary section of this chapter.

Chapter 5, Conclusion and Recommendations, a conclusion is drawn that states achievements that are made with this research and gives some recommendations to further enhance the research results for possible better achievements.

Finally, in Appendices, the script that is written in MATLAB and the objective function values are posted.

2 MODELING

There are three main drivetrain configurations of Plug-in Hybrid Electric Vehicles, namely: series, parallel and series-parallel (or powersplit).

Series Hybrid: Only the electrical motor is mechanically connected to the powertrain. The power provided by the engine is first converted to electrical energy through the generator and then converted to mechanical energy through the electrical motor. The engine is typically smaller in a series drivetrain because it only has to meet average driving power demands; the battery pack is generally more powerful than that in parallel hybrids in order to provide remaining peak driving power needs. This larger battery and motor, along with the generator, add to the cost, making series hybrids more expensive than parallel hybrids.

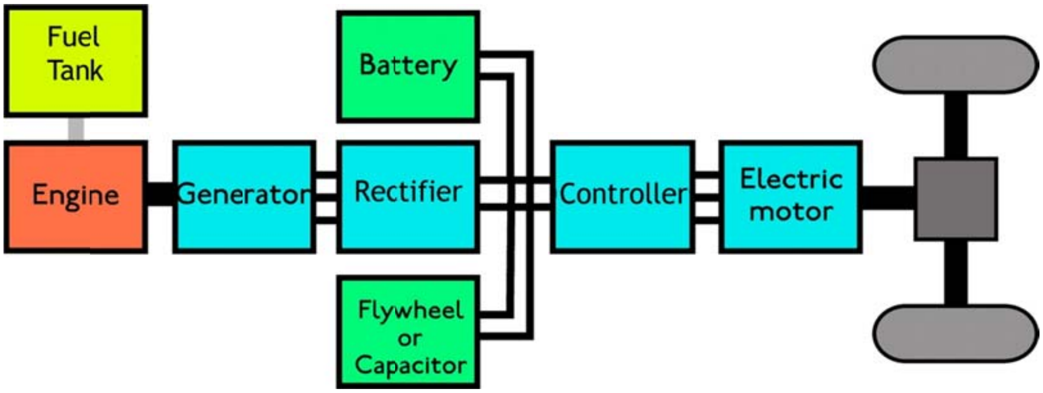

Figure 2.1 Series hybrid drivetrain schematic

Parallel Hybrid: In this configuration the engine and the electrical motor are both connected to the powertrain through torque coupling, speed coupling or torque and speed coupling, eliminating the inefficiency of converting mechanical power to electricity and back which makes these hybrids relatively more efficient on the highway. Yet, the same direct connection between the engine and the wheels that increases highway efficiency compared to a series hybrid does reduce, but not eliminate, the city driving efficiency benefits (i.e. the engine operates inefficiently in stop-and-go driving because it is forced to meet the associated widely varying power demands).

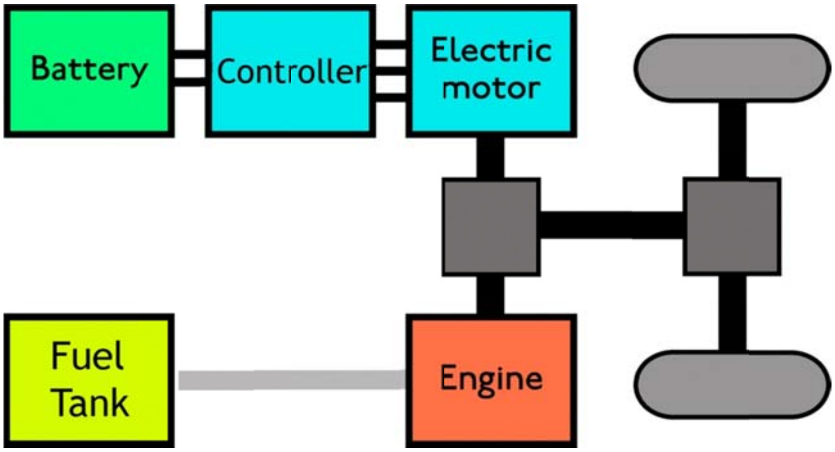

Figure 2.2 Parallel hybrid drivetrain schematic

Series-Parallel Hybrid (Powersplit hybrid): This drivetrain merges the advantages and complications of the parallel and series drivetrains. By combining the two designs, the engine can both drive the wheels directly (as in the parallel drivetrain) and be effectively disconnected from the wheels so that only the electric motor powers the wheels (as in the series drivetrain).

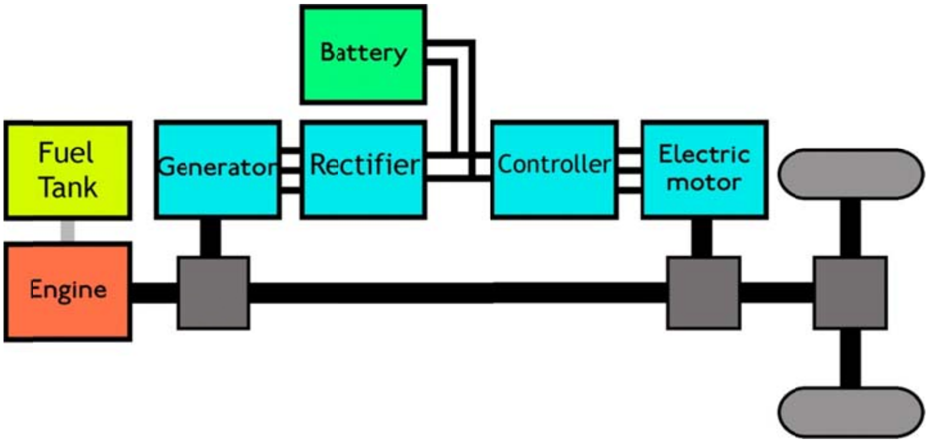

Figure 2.3 Powersplit hybrid drivetrain schematic

Since the powersplit hybrid configuration has most of the benefits and the least amount of drawbacks, it is used as a baseline model. The power split PHEV model configuration is shown in Figure 2.1. In this model, planetary gear set is used whose sun gear is connected to the motor/generator and the carrier gear is connected to the engine. The output of this planetary gear set is connected to the motor through a torque coupler, which gives its output to final drive and wheels.

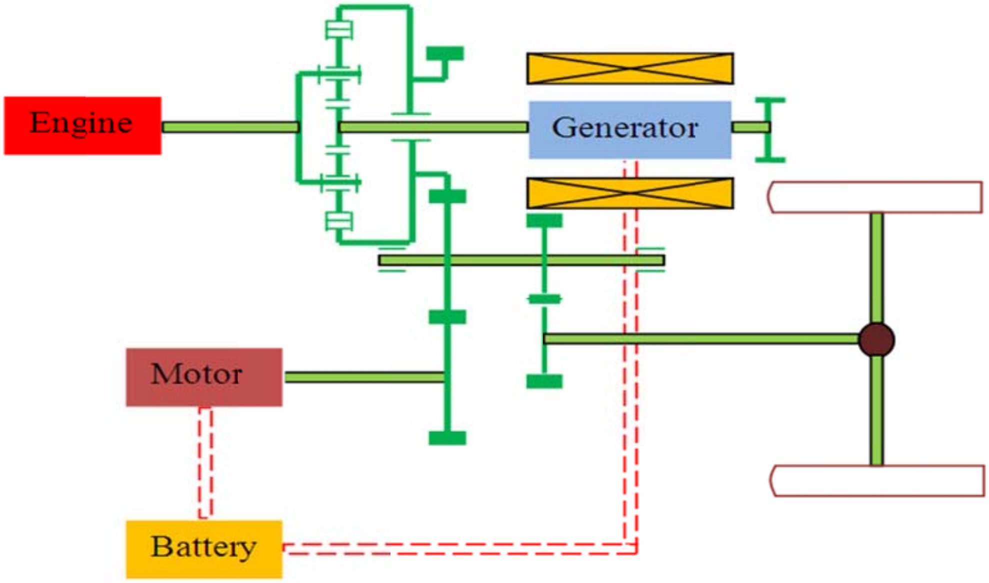

Figure 2.4 PHEV drivetrain schematic

2.1 Driver Model

Driver component designed in the PSAT simulation model is used to imitate the actions of a real driver to follow a specific pattern of an already determined drive cycle. In this thesis Urban Dynamometer Driving Schedule (UDDS) drive cycle of the Environmental Protection Agency, US (EPA) is used. The model is constructed as a PI controller where the proportional gain K_p and the integral gain K_i are found experimentally in the software. Equation 2.1 represents the driver mathematical model in which it is assumed to be an automatic transmission usage. The outputs of the driver model are the demanded torque and the demanded speed that are shown as follows:

$$\tau_{dmd} = K_p e + K_i \int e \, dt \tag{2.1}$$

$$v = v_{DC} \tag{2.2}$$

Here the speed error is:

$$Err = v - v_{dmd} \tag{2.3}$$

Furthermore, in order to overcome the stability and oscillating issues some time delay is added to the torque command generated by the driver model response.

2.2 Vehicle Model

The mathematical model of the vehicle is constructed considering the three major phenomenons: grade, aerodynamic drag and rolling resistance.

Grade force: First of all, the grade force F_g that the vehicle has to overcome has a significant impact on the dynamic model of the vehicle. This grade force is calculated using Newton’s second law of motion, Equation 2.4. Depending on various variables, such as; vehicle mass and grade angle, the grade force can change, which has a large impact on the force required to drive the vehicle, and can result in changing the accuracy of the dynamic model.

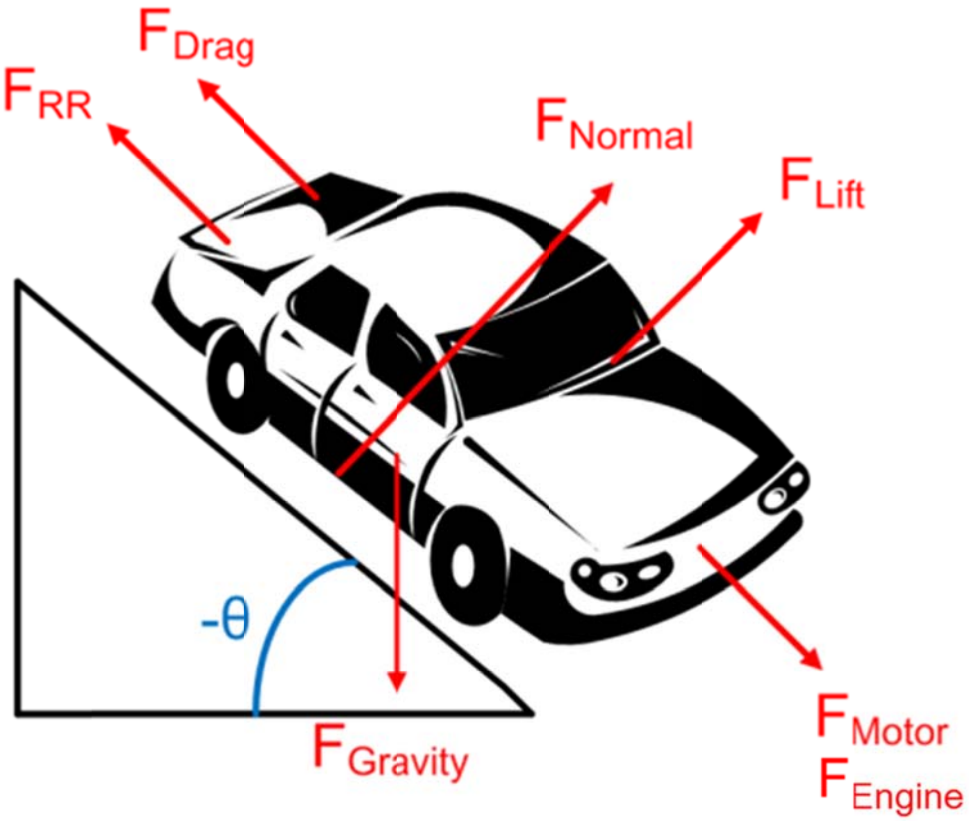

Figure 2.5 Forces on the vehicle

$$F_g = g\, m_v \sin(\tan^{-1}(\aleph)) \tag{2.4}$$

where g is the gravitational acceleration, m_v is the vehicle mass and $\aleph$ is the road grade.

Aerodynamic drag: Second major phenomenon is the aerodynamic drag. As the vehicle moves it is resisted by the air surrounding the vehicle creating a resistive force called aerodynamic drag. In order to calculate this drag force, the lateral forces are neglected and the losses were estimated using Equation 2.5.

$$F_{drag} = {}^{1}\!/_{2}\, \rho C_{air} A v^2 \tag{2.5}$$

where σ is the air density, $\mathcal{A}$ is the estimated frontal area of the vehicle and $\mathcal{C}_{air}$ is the coefficient of aerodynamic air resistance. These constants are determined from experimental results.

Rolling resistance: The last one is the rolling resistance, which relatively has a small impact compared to the other two. This resistive force is created because of the deformation of the tires at the point of the contact with the ground during rolling motion. This is calculated by Equation 2.6 [15].

$$F_{RR} = gm_v(K_1 + K_2 v)\cos\left(\tan^{-1}(\aleph)\right) \tag{2.6}$$

Here K_1 and K_2 are the coefficients of the rolling resistance that are found experimentally and v is the vehicle velocity at a given instant time.

In PSAT model the rolling resistance and the aerodynamic drag force are approximated as a second order polynomial by using above two equations and empirical data as shown in Equation 2.7.

$$F_{ARR} = \min(A_0, \frac{A_0}{0.05})v + A_1 v + A_2 v^2 \tag{2.7}$$

where F_{ARR} is the combined resistive force (Rolling Resistance and Aerodynamic Drag) and the constants A_0, A_1, A_2 coefficients are the combined coefficients that is found experimentally. The first term in this equation decreases at lower speeds therefore it expresses the rolling resistance, the second term expresses the higher order coefficients of rolling resistance and some other parasitic losses whereas the third term represents the aerodynamic drag.

Finally, the required force to drive the vehicle at the demanded speed is calculated by Equation 2.8. F_{dmd} is the demanded force.

$$F_{REQ} = F_{ARR} + F_g + F_{dmd} \tag{2.8}$$

2.2.1 Engine

The engine used in this vehicle model is a generic spark ignition (SI) engine model, using gasoline as fuel to produce mechanical energy. Using the drive cycle parameter values the required torque and speed, as a function of time, is computed. These demand torque and demand speed values are controlled in the vehicle control unit model and afterwards feed to the engine control unit model. This specific engine control module model controlled the engine to operate in the desired torque and speed ranges. In PSAT, the engine is not modeled as a very detailed, high-end dynamic model. The inertial losses and the thermal losses are considered for the control purposes and the parasitic loads on the engine, both mechanical and electrical, are assumed to be constant. The following equations, Equation 2.9 and Equation 2.10, are used to calculate the torque and speed values that are available.

$$\tau_{total\_e} = \tau_e + I_e \frac{d\omega_e}{dt} + L \tag{2.9}$$

$$\omega_{REQ\_TOTAL} = \omega_{REQ} \tag{2.10}$$

Here, τ_e is the engine torque, I_e is the inertia of the engine, L is the constant value that is assumed to represent the mechanical and electrical parasitic losses. ω_{REQ} is the required angular velocity.

The fuel consumption is determined through a 2D look up table based on engine speed and engine torque. The emission levels of hydrocarbon (HC), carbon monoxide (CO) nitrogen oxide (NO_x) and particulate matter (PM), as well as the oxygen content in the exhaust gas, are also estimated through 2D lookup tables as a function of engine speed and engine torque. The 2D look up tables that are used in estimating these important parameters are all defined empirically in the PSAT software.

2.2.2 Electric Motor

Electric motor modeled in a way that the model itself includes the motor’s torque speed-dependent capability and the motor losses due to its inertia. Power losses in terms of efficiency are specifically determined by empirical look up tables in the PSAT software. The motor model constructed by using two dynamic equations shown below:

$$\tau_{total\_m} = \tau_m + I_m \frac{d\omega_m}{dt} \tag{2.11}$$

$$\omega_m = \omega_{REQ} \tag{2.12}$$

where τ_m is the motor torque, I_m is the motor inertia and ω_m is the angular velocity of the motor.

Additionally, maximum torque that the motor can provide is also determined and limited by a look up table so that the motor current does not exceed the maximum allowable rate. Regarding the thermal model of the motor, only the heat index is calculated to determine the maximum torque constraint. Detailed information about the look up tables used for the model, refer to PSAT documentation [15].

2.2.3 Energy Storage System

Since all the power demand from the motor is provided by the energy storage system this component plays a significant role in the drivetrain. The energy storage system, simply the battery is modeled as an open circuit voltage model, equivalent circuit consisting of open circuit voltage, which is in series with battery internal resistance.

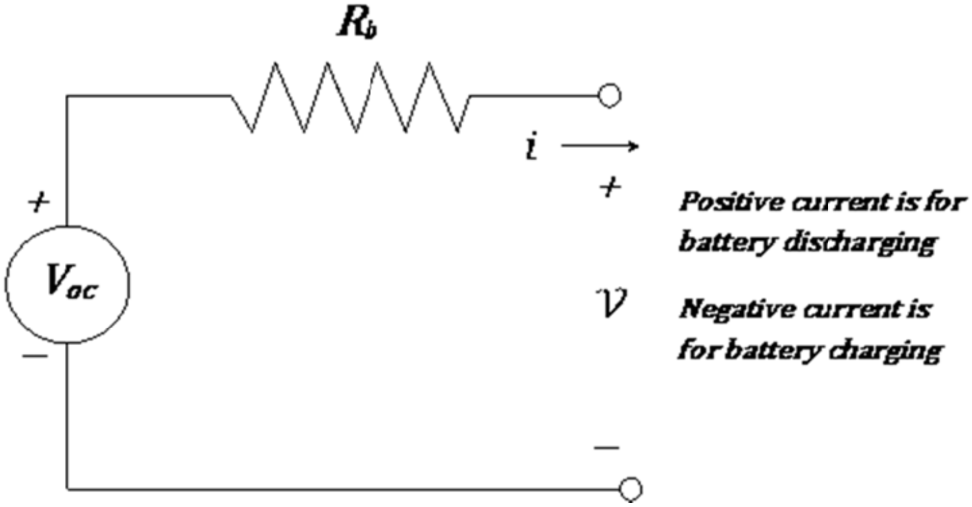
R_b
i
+
+
V_{oc}
−
V
−
Positive current is for battery discharging
Negative current is for battery charging

Figure 2.6 Equivalent circuit diagram for the energy storage system

The quadratic Equation 2.14 is used along with Kirschoff's voltage law in battery equivalent open circuit diagram to obtain the bus current.

$$V = \frac{P_b}{I} = V_{OC} - R_b I \tag{2.13}$$

$$\Rightarrow R_b I^2 - V_{OC}\, I + P_b = 0 \tag{2.14}$$

From Equation 2.36, Equation 2.15 is derived:

$$I = \frac{V_{OC} - \sqrt{V_{OC}^2 - 4P_b}}{2R_b} \tag{2.15}$$

where V_{OC} and R_b are found from 2D look up tables determined experimentally.

To calculate the bus voltage, again, Kirschoff's voltage law is used,

$$V = V_{OC} - R_b I \tag{2.16}$$

The maximum allowable power is determined using look up tables that are constructed through manufacturer's battery specifications.

The battery pack consists of number of battery modules connected in series; these modules are constructed by certain number of cells that are connected in a certain series and/or parallel pattern. Number of battery modules, number of cells and the pattern they are combined are determined by performance requirements of the energy storage system via various optimization techniques. Columbic inefficiency is used to model the power losses- I^2R losses.

The State of Charge (SOC) of the battery is calculated by integrating the current on the time interval. The SOC value corresponding to the optimum set of operating point would then be recorded as previous SOC value for the next time interval. Below is the equation that is used to calculate SOC for each time interval.

$$\gamma_k = \frac{1}{\mathcal{C}_{max}} \int_{t=k-1}^{t=k} i\mathrm{d}t + \gamma_{k-1} \tag{2.17}$$

where γ is SOC, $\mathcal{C}_{max}$ is maximum ampere-hour capacity of battery, k is discrete time.

2.2.4 Continuous Variable Transmission

The Powersplit drivetrain configuration used in this project requires Continuous Variable Transmission (CVT) because of its internal dynamic structure. As mentioned in the earlier sections of this chapter, planetary gear set is used for speed coupling in which the planetary gear is torque coupled with the motor to transfer the power generated to the drivetrain. The sun gear in the gear set is connected to the generator, which converts mechanical energy to electrical energy, also, is defined as 'Motor 2'. Moreover, the engine is connected to the carrier gear and the motor is connected to the ring gear of the gear set.

Equation 2.18 shows the motor torque in the PSAT model.

$$\tau_m = \tau_r - \frac{(\alpha_1 \tau_g + \alpha_2 \tau_e)}{\alpha_3} \qquad (2.18)$$

where τ_m is motor torque, τ_g is the generator torque, τ_e is the engine torque and τ_r is the ring torque. The coefficients α_1, α_2, α_3 are experimentally found by the PSAT tool to simplify the planetary gear set ratios and parasitic losses [15].

2.3 Ground Contact Model

2.3.1 Wheels and Axle

In PSAT wheels and axle are modeled as a single component in which a pair of wheels are combined with an axle and attached to the vehicle. Losses due to the slip are neglected by PSAT tool; instead the tool computes the angular wheel speed from actual vehicle speed assuming that they are equal. For simplification, the braking torque and inertia are added corresponding to all the wheels. The model kinematics equation is shown below:

$$\tau_a = (F_B - F_{REQ}) r_w + L_a + I_w \frac{d\omega_w}{dt} \qquad (2.19)$$

where τ_a is the torque acting on the axle, F_B is the equivalent break torque of the wheels, F_{REQ} is the required force, r_w is the radius of the wheel, L_a is the losses, I_w is the wheel inertia and ω_w is the wheel angular velocity.

2.3.2 Differential

The connection between the transmission and the wheel axle is established with this component. Its role is to distribute transmission power over the wheels through the axle in order to eliminate the slip during a turn. Due to this component, the slip losses are reduced in significant amounts. That is the reason the PSAT tool neglected the slip losses for the wheel and axle model. However, losses due to inertia and the component itself are modeled in the differential mathematical model. Equation 2.20 and Equation 2.21 which are shown below explains the dynamics of the differential torque and the differential angular velocity as:

$$\tau_d = \frac{\tau_a}{\Re_d} + I_d \frac{d\omega_d}{dt} + L_d \tag{2.20}$$

$$\omega_d = \Re_d \omega_w \tag{2.21}$$

Here τ_d is the differential torque, τ_a is the axle torque, $\Re_d$ is the differential gear ratio, I_d is the inertia of the component itself, ω_d is the differential angular velocity and L_d is the loss due to the differential.

3 PROBLEM FORMULATION

The problem arises from the search for a better fuel economy whilst meeting the performance requirements. The search for a better configuration of drivetrain components in terms of fuel economy and better performance is an open-ended research subject. Continuous improvement in this field is a significant technical achievement that should be taken care of.

Since the vehicle itself is dynamically, highly, nonlinear and most of the drivetrain components in a PHEV, directly or indirectly, has an effect on each other, optimization process of the major powertrain components have to be done via proper methodology to represent effects of each component modification on the others. In order to explain this effect briefly, for example, if the engine on the vehicle is decided to be more powerful, changing the existing engine with a 20% more powerful engine will increase the total mass of the vehicle because the bigger engine's block mass will be higher than the replaced one. Therefore, each and every component's effect has to be considered.

The objective of using an optimization tool is, as briefly explained above, to express the effects of all the components on the others and on themselves mathematically in the optimization methodology structure [11].

After constructing the methodology mathematically in detail, a Matlab script can be used to collaborate with PSAT simulation software with the PSO script to search the optimum points in the solution space. The solution space is constructed through dynamic equations related to the performance requirements, such as maximum acceleration, maximum cruise speed, and grade, which is explained briefly later in this chapter.

In the previous studies, researchers have focused on mostly HEVs and various methodologies. This research is the first attempts of its kind to apply PSO methodology to find the optimal sizes of the powertrain components of a PHEV. This research shows a different path of utilizing and combining a well-known optimization tool with a relatively newer technology type of hybrid vehicles.

3.1 Optimization Methodology

This section of problem formulation chapter is explained in two different subsections. The first one is advantages of PSO tool and why it is chosen for this project and the second subsection describes briefly the PSO optimization tool.

3.1.1 Advantages of PSO

Using gradient-based algorithms the optimization problem could be solved [3]. However, since these algorithms depend on the gradients to find the optimum solution, they do not always give the global maximum or minimum as the solution. Therefore, derivative free algorithms such as Genetic Algorithm (GA), DIRECT, Dynamic Programming, Simulated Annealing, Particle Swarm Optimization, etc. can be used. Since they aren’t gradient-based, they provide global solution to the optimization problem.

Most evolutionary techniques mentioned above shares some common procedure, such as: random generation of initial population, reckoning of a fitness value for each subject, reproduction of the population. However, PSO does not have genetic operators like mutation and crossover [14]. Particles update themselves with the internal velocity. Also, each particle has a memory, which is significant for the algorithm. The information sharing mechanism of PSO tool is significantly different than the others. Instead of having massive amount of variables to tell the system about the previous iteration, PSO has just one variable to rule that functionality to the algorithm. It is simply a one-way information sharing mechanism. Another major advantage of PSO tool is that the method only searches for the best solution and at the end all the particles converge to the best solution quickly in most cases [23].

Strengths and advantages of the PSO tool mentioned in above paragraphs over the other gradient based algorithms, made this tool the most convenient optimization algorithm to work with a highly nonlinear and component dependent system like PHEV.

3.1.2 Particle Swarm Optimization

Particle Swarm optimization was developed by Kennedy and Eberhart in 1995 [14]. The algorithm is based on the social behavioral model of the society, similar to the social behaviors of bird flocking or fish schooling; in other words, it is based on the stochastic optimization technique. The difference of this method from the other evolutionary computation techniques, like Genetic Algorithm (GA), is that it does not use evolution operators, such as mutation, crossover and etc. The system is initialized with a population of particles with their own position and velocity values in n-dimensional space. Each particle in the solution space is a possible optimum solution. The particles fly through the solution space by following current optimum particles using the equations defined by the PSO algorithm as shown below:

$$V(k+1) = wV(k) + c_1 r_1\big(pBest(k) - x(k)\big) + c_2 r_2\big(gBest(k) - x(k)\big) \tag{3.1}$$

$$x(k+1) = x(k) + V(k+1) \tag{3.2}$$

For the next iteration the velocity of each particle is calculated by Equation 3.1 and Equation 3.2 is the position of the particle for the next iteration. Here c_1 is the cognition-learning rate, c_2 is social learning rate of particle and w is the inertial weight, which enhances the performance of PSO in various applications [11]. r_1 and r_2 are random numbers between 0 and 1. $pBest$ is the particles' own best position and $gBest$ is the global best position determined by comparing the $pBest$ of all particles. The particles will be updated using these equations iteratively until the optimal solution is obtained by convergence of all the particles. Since the method requires very few parameters, this is a significant advantage over the other methodologies in terms of computation time and relatively less amount of variable determination.

This particular PSO technique was developed for unconstrained optimization problems. However, researchers have developed various versions of PSO algorithm, which can also be valid for constrained optimization problems. Gregorio proposed a PSO approach with variation in velocity computation formula, turbulence operator and different mechanism to handle the constraints [24]. Another approach, the penalty function approach, can be used for solving constrained optimization problems, was shown by Parsopuulos [15]. An additional penalty function is added to the fitness function to replace the constraints, in other words, expressing the constraints as a penalty function to determine the solution space boundaries and to limit the particles from flying out the boundaries.

A different approach by Hu and Eberhart was proposed as well. They suggested a method with some modifications in the PSO algorithm in [24]. The two modifications to the original PSO algorithm are: all the particles have to be reinitialized in the feasible space and also only the feasible points are assigned for the $gBest$ and $pBest$ variables. Therefore, the PSO algorithm always starts and gets values in the constrained region. Thus, the motion of the particles is always in the feasible solution space.

Since the components optimized in this research are restricted by physical limitations and the availability and/or feasibility of some specific components forced the PSO tool to be used with constraints. Hu and Eberhart's modified PSO algorithm is used to express the constraints. The boundaries of the components that are optimized are determined by mathematical equations that are found by expressing the dynamics of the performance requirements. The way in which the methodology is constructed will be examined in detail in the following sections of this chapter.

Figure 3.1 shows the flow chart of the constrained optimization PSO methodology that explains the strategy and the logic behind the modified technique briefly.

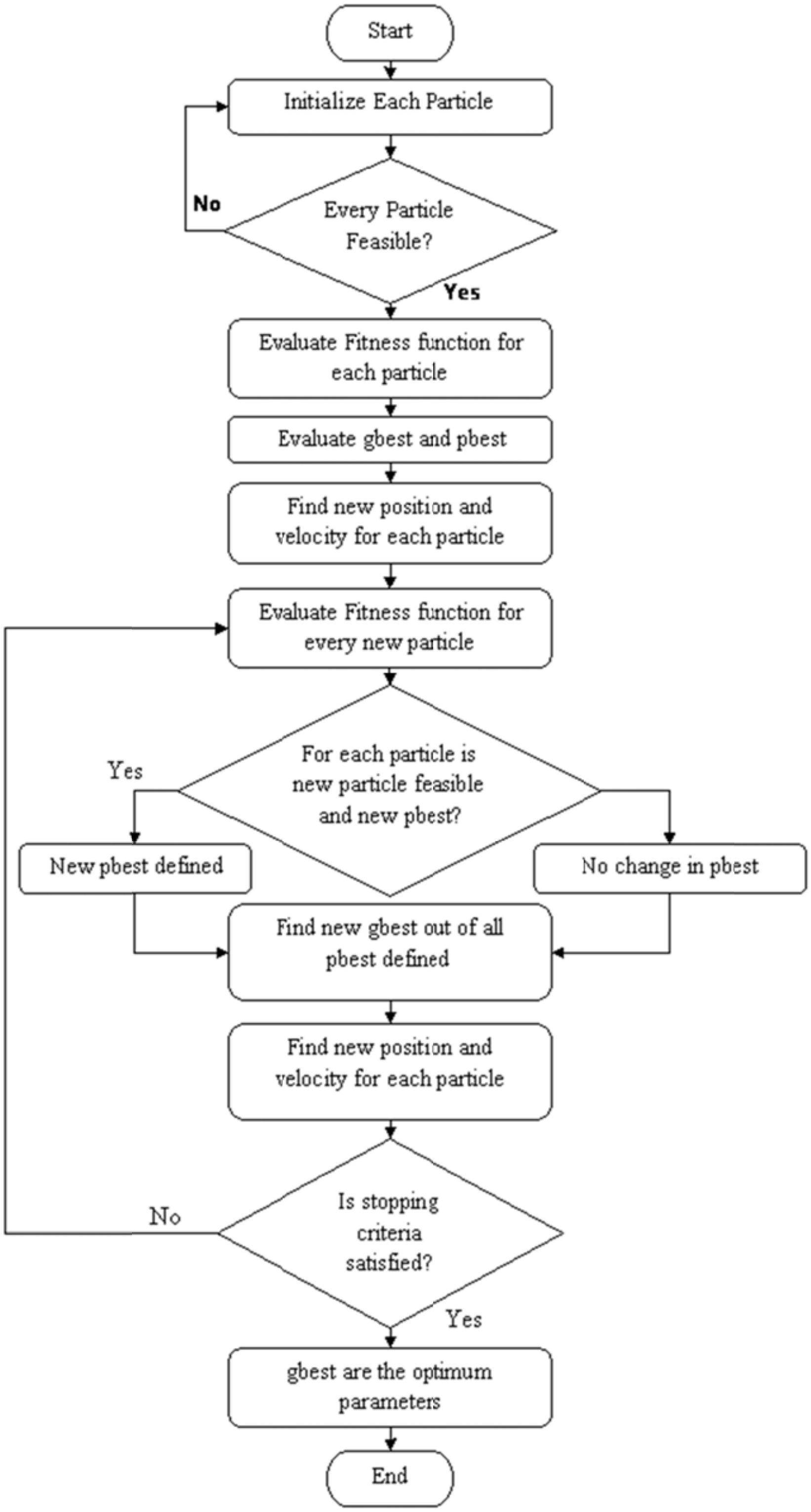
Start
Initialize Each Particle
Every Particle Feasible?
No
Yes
Evaluate Fitness function for each particle
Evaluate gbest and pbest
Find new position and velocity for each particle
Evaluate Fitness function for every new particle
For each particle is new particle feasible and new pbest?
Yes
New pbest defined
No change in pbest
Find new gbest out of all pbest defined
Find new position and velocity for each particle
Is stopping criteria satisfied?
No
Yes
gbest are the optimum parameters
End

Figure 3.1 Flow chart of constrained PSO algorithm

3.2 Baseline Vehicle Specifications and Performance Requirements

In search for the performance requirements, the 2008 Toyota Prius general performance requirements are taken into account. This information can be found in Toyota Motor Company's official website. These performance requirements are necessary to determine the constraints and solution space boundaries for the optimization methodology. Table 3.1 shows the performance requirements for the vehicle. In the following sections of this chapter a brief explanation of the derivation of mathematical constraint equations out of these performance requirements are explained.

Table 3.1 Performance requirements of the baseline vehicle

Performance Requirement	**Value**	**Unit**
Maximum speed	104	mph
Maximum grade at 60 mph speed	6	%
Average time to reach from 0-60 mph	10	seconds

Like every optimization problem, the search for the optimal component sizes of the drivetrain components of PHEV needs an initial start points for each objective function variable. This section describes the details of the specifications of the baseline vehicle that is used in this project in terms of each component. The initial points, that is to say, the baseline specifications are taken from a 2008 Toyota Prius. The values of the parameters are the general public information that can be found on Toyota's official website.

3.2.1 Engine Specifications

As mentioned before in general, the engine model and the specifications of the baseline component used is 2008 Toyota Prius engine whose specifications are given by the following list:

- Inline 4-cylinder Double Overhead Camshaft (DOHC)
- Displacement: 1,497 cc
- Compression Ratio: 13.0:1
- Peak power: 51 kW at 4,500 rpm
- Peak Torque: 82 lb.-ft. at 4,200 rpm

3.2.2 Electric Motor Specifications

The electric motor is the secondary power unit which provides the in city highly efficient operations points by eliminating the inefficient operating regions of the SI engine. This phenomenon makes the motor extremely important. Therefore, the initial points of the motor specifications that are given in the following list are very critical:

- Permanent Magnet
- Capacity: 6.5 amperes
- Peak power: 52 kw at 400 rpm
- Peak Torque: 258 lb./ft. (350 Nm) 0-400 rpm

3.2.3 Energy Storage System Specifications

Unlike the other components the energy storage system used in this project is different than the one used in Toyota Prius. Since Prius is an HEV, its battery specifications are not adequate to be used in this research as a PHEV battery pack. The battery pack that is used in this optimization research is a relatively bigger capacity battery pack. A123 L5 Lithium Ion Nanophosphate battery is used instead of the Prius' OEM battery pack. The specifications are shown in the following list:

- Elements Per module → 10 cells in parallel, 7 series elements
- Number of Modules → 5
- Cell Normal Volt → 3.5
- Cell Max/Min Volt → 3.5/2.5

3.3 Construction of the Methodology

An optimization problem consists of two major parts; first one is the objective function, which is also called the cost, or the fitness function. To be consistent with all the other chapters and sections, in this thesis it is called the objective function. The second part is the constraints, which determine the solution space boundaries. In the following subsections of this section, the mathematical derivation of these objective function and constraint equations are examined in detail.

Below is the general mathematical expression for the optimization problem.

$$\underset{X \epsilon \Omega}{min} F(\boldsymbol{X}) \qquad \boldsymbol{X} = [P_M, P_E, NBM, FC]^T$$

$$s.t. \quad C_u(\boldsymbol{X}) > 0 \quad u = 1, 2, 3 \dots, k$$

where:

- $\boldsymbol{X}$ is the column vector consists of objective function variables
- Ω is the solution space.
- P_M is the power of the electrical motor.
- P_E is the power of the engine.
- NBM is the number of the battery modules.
- FC is the fuel consumption.
- C_u nonlinear functions of the design constraints (performance requirements)

3.3.1 Objective Function

Objective function is constructed by the size parameters of the components that will be optimized. Minimization of this function is required to find the optimal points of the component sizes in the boundaries whilst trying to reach the minimum fuel consumption. In the PSO optimization process the variables provide an objective function value. After adequate amount of iterations, the objective function value starts to converge to some finite number when the optimization tool and the variables to be optimized reach saturation and that point is called the optimum value or the global minima.

The objective function includes the electric motor power, number of battery modules and fuel consumption [6]. All four variables are normalized and weighted for the objective function construction. The normalization process is required to bring down all the variables to one level because the addition of the process requires the addition of all those four major variables in one level. The following equation represents the objective function:

$$F(P_M, P_E, NBM, FC) = w_1 \frac{P_M}{P_{M,max}} + w_2 \frac{P_E}{P_{E,max}} + w_3 \frac{NBM}{NBM_{max}} + w_4 \frac{FC_{max}}{FC} \quad (3.1)$$

Here the parameters representing the components; P_M, in kW, peak power of the electric motor, P_E, in kW, peak power of the engine and NBM, number of battery modules are aimed to be minimized and as it can be seen, they are directly proportional to the objective function value. However, on the other hand, FC, representing the fuel economy in terms of miles per gallon is reversely proportional to aim increasing the mileage of the vehicle. Also, w_i is the weights of the objective function variables, i = 1,2..,4. The weight values are determined via trial and error during the optimization process by observing the convergence rate and various objective function values.

3.3.2 Constraints

Boundaries are determined by the dynamic-equation representation of the performance requirements and design constraints. Maximum possible sizing values of three most significant components, engine, electric motor and the battery, are used in determining the major nonlinear constraints of the optimization problem. In order to size the components, as mentioned earlier, parameters for the sizes of the drivetrain components are chosen as the power of the engine and the motor and for the energy storage system, the capacity converted to number of battery modules. Each parameter is examined in terms of two sections; one is the lower boundary, i.e. the minimum end and the other one is the upper boundary, i.e. the maximum end. A brief description of how the constraints are derived mathematically, are mentioned in the following subsections.

3.3.2.1 Engine Constraints

The minimum power required from the engine can be calculated with mean cruise speed. The following equation represents the minimum engine power:

$$P_{E,min} = \frac{1}{\eta_T}\left(mgfv_1 + {}^1/_2\,\rho C_d A {v_1}^3\right) \tag{3.2}$$

where η_T is the powertrain efficiency, m *is* mass of the vehicle, g is the gravitational acceleration, f is the coefficient of rolling resistance, v_1 is the 6% grade speed, ρ is the air density, C_d is the air drag coefficient, A is the frontal area of the vehicle, v_1 is the mean cruise speed.

The maximum required power for the engine can be determined by either at the maximum cruise speed or on the path with a slope at a constant speed going uphill. These values are computed and the maximum value amongst each other is chosen. Equation 3.3 shows the maximum function and Equation 3.4 and Equation 3.5 show the determination of the two values:

$$P_{E,max} = \max\left(P_{E,1}, P_{E,2}\right) \tag{3.3}$$

$$P_{E,1} = \frac{1}{\eta_T}\left(mgfv_{max} + {}^1/_2\,\rho C_d A {v_{max}}^3\right) \tag{3.4}$$

$$P_{E,2} = \frac{1}{\eta_T}(mgfv_{max}\cos\alpha + mgv_{max}\sin\alpha + {}^1/_2\,\rho C_d A {v_{max}}^3) \tag{3.5}$$

Where η_T is the powertrain efficiency, m *is* mass of the vehicle, g is the gravitational acceleration, f is the coefficient of rolling resistance, v_1 is the 6% grade speed, ρ is the air density, C_d is the air drag coefficient, A is the frontal area of the vehicle, v_1 is the mean cruise speed and α is the grade degree and v_{max} is the maximum cruise speed.

3.3.2.2 Electric Motor Constraints

The peak power of the electric motor is calculated by the maximum acceleration constraint. Equation 3.6 represents the maximum size of the electric motor:

$$P_{M,max} = \frac{1}{2t_f} m{V_f}^2 \tag{3.6}$$

Here t_f is the time required to reach the final speed, m is the mass of the vehicle and V_f is the final speed that is to be reached.

On the other hand, the minimum power of the electric motor that is required for the vehicle to drive at a constant speed on a road with a gradient slope is shown by the following equation:

$$P_{M,min} = mgfv_1 \cos\alpha + mgv_1 \sin\alpha + {}^1/_2\, \rho C_d A {v_1}^3 \tag{3.7}$$

where m *is* mass of the vehicle, g is the gravitational acceleration, f is the coefficient of rolling resistance, v_1 is the 6% grade speed, α is the grade degree, v_1 is the mean cruise speed, ρ is the air density, C_d is the air drag coefficient and A is the frontal area of the vehicle.

3.3.2.3 Energy Storage System Constraints

The energy storage system is the only power resource for the electric motor. Therefore, these two components are bounded to each other in terms of their power requirements. The electric motor's minimum voltage requirement determines the minimum number of battery modules.

$$NBM_{min} = Round\left(\frac{U_{M,min}}{U_{b,min}}\right) \tag{3.8}$$

Here $U_{M,min}$ is the minimum voltage of the motor, $U_{b,min}$ is the minimum voltage of the battery module. $Round$ is the function rounds up the value to the upper integer.

The maximum number of battery modules is derived from the peak power of the electric motor.

$$NBM_{max} = \frac{P_{M,max}}{D_p m_M \eta_T} \tag{3.9}$$

In Equation 3.9, D_p is the specific power of the battery module, m_M is the mass of the battery module, η_T is the efficiency.

4 SIMULATION

The optimization problem in this research is solved using PSAT tool along with MATLAB. This chapter is a detailed description of how the simulation structure of the problem is constructed and the results acquired through the simulation. First section will describe the PSAT/Simulink modeling structure and explains the Matlab script constructed for the optimization tool, which also makes the optimization tool work along with the PSAT software, and in the second section the results of the simulation are illustrated. After finding the optimal sizes of the components a comparison is made with the baseline vehicle model and the optimal sized vehicle in terms of performance values and fuel consumption.

4.1 Simulation Model

4.1.1 PSAT/Simulink

The mathematical model of the vehicle that was explained in the previous chapters of this thesis is constructed as block diagrams in PSAT software [15]. PSAT is a powerful simulation tool based on Matlab/Simulink. The default Toyota Prius vehicle model in PSAT is used in this research to compute major parameter values that are used in the optimization tool. Since Toyota Prius is a HEV, the model had to be modified to represent a PHEV model. Figure 4.1 shows the block diagram configuration of the vehicle model.

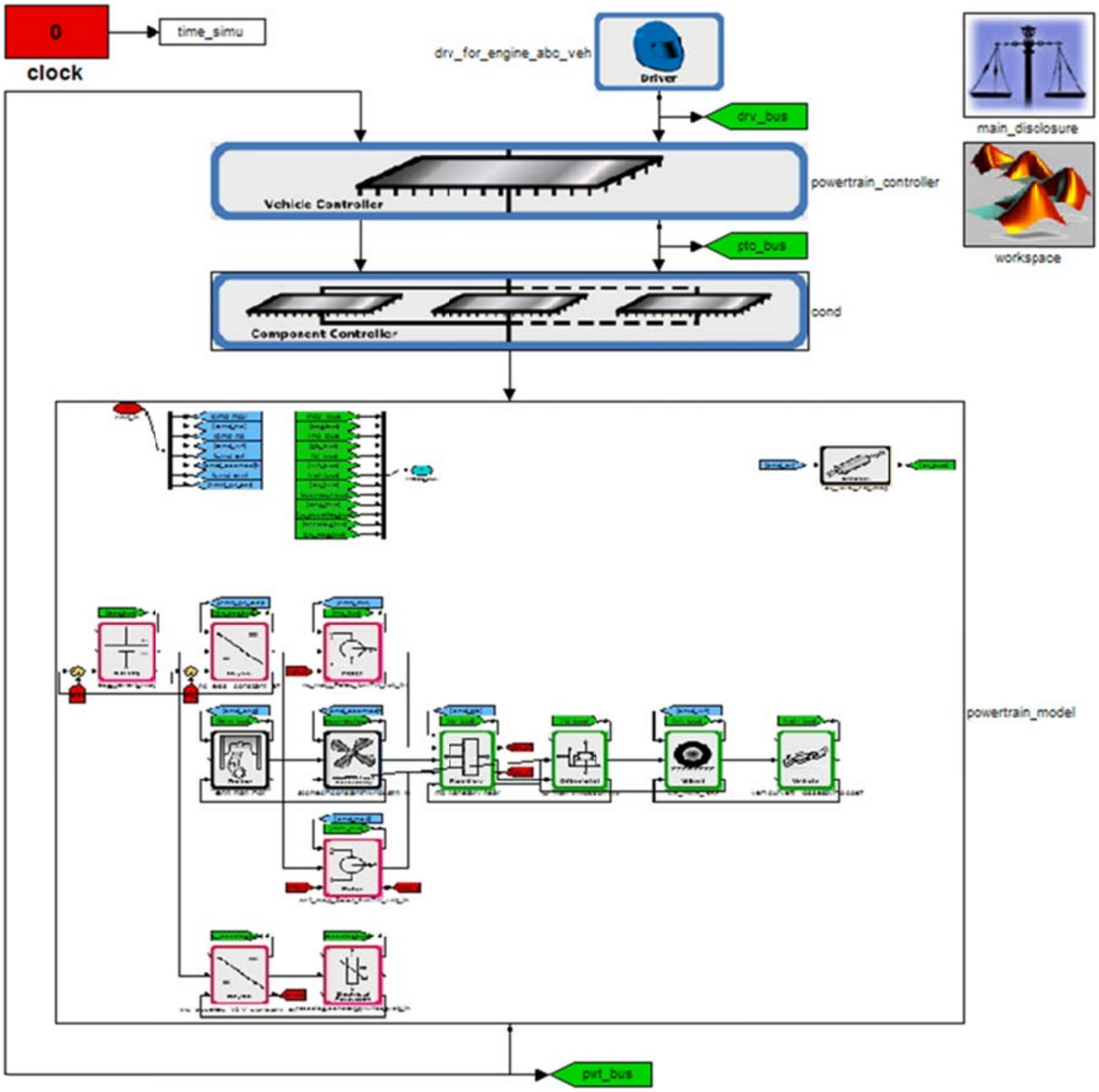

Figure 4.1 PSAT model PHEV vehicle block diagram

In the above figure, the upper portion shows the vehicle controller and the component controller. The middle part is the powertrain model that consists of engine, motor, battery and rest of the powertrain components. The blocks that represent the look up tables are shown as well.

4.1.2 Matlab

This section explains working process as well as the structure of the Matlab script in detail. As mentioned earlier, Matlab is used to join the PSAT/SIMULINK block diagram structure with the PSO optimization tool. The PSO script is embeded in the Matlab script which calls and runs the simulation software PSAT/SIMULINK to compute necessary parameters that is shown in the following parts of this section.

This section is where PSAT/Simulink simulation is used. Since the system is highly nonlinear and the vehicle model is a complicated model, to achieve more realisic results, complete vehicle model simulation is used to determine the fuel consumption and the emission levels according to various random component sizes in their own boundaries. The internal structure of the PSO algorithm is explained in Problem Formulation chapter, under the Particle Swarm Optimization section. The figure below shows 'Evaluate Fitness function for each particle' portion of the constrained PSO algortihm flow chart.

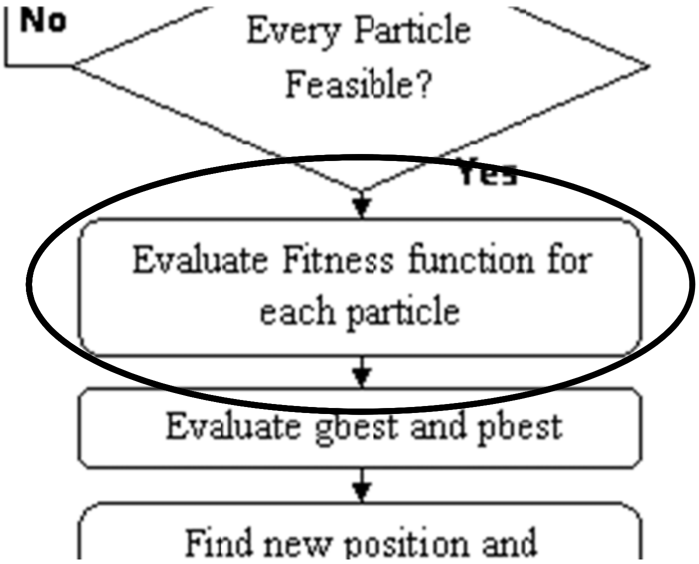

Figure 4.2 'Evaluate Objective Function' portion of PSO algorithm flow chart

The following cycle chart shows the basic correlation between Matlab, PSAT/Simulink. The Matlab script includes the PSO algorithm, it first sends the initial values of the component sizes to the simulation tool, and then the initial fuel consumption and emission level values are computed. According to those initial values, the script sends the randomized component sizes to the simulation tool and this continues, in the order of the PSO flow chart, until every particle converges to the optimum solution. Complete Matlab code can be found in Appendix A.

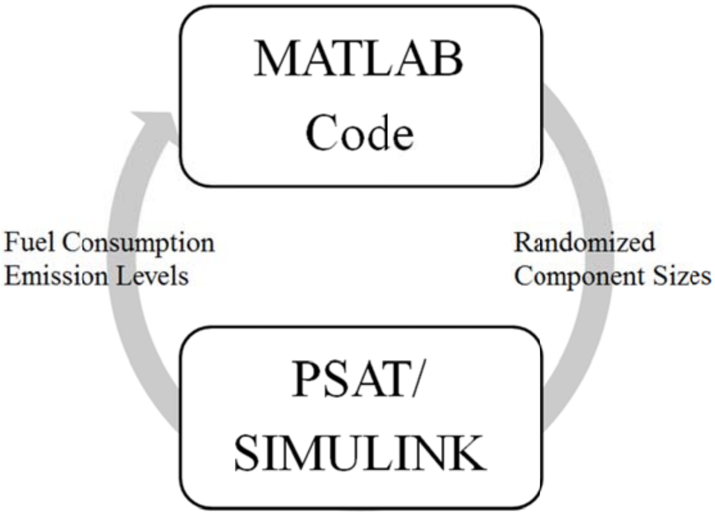

Figure 4.3 Basic Matlab-PSAT correlation cycle chart

4.2 Simulation Results

This constrained optimization problem is solved via the modified PSO algorithm mentioned in the previous sections. The extremes of the problem and the objective function are implemented together into the code. Therefore, in each iteration, for each particle, the simulation runs in PSAT tool according to the parameter values that have been computed via the PSO tool depending on the $gBest$ and $pBest$ values and velocity calculations of the PSO. As the result of the simulation that is done over the PSAT model, the fuel consumption and some other parameters are calculated. The objective function value is computed. This loop continuously repeated until the particles converge to the optimal solution point. Next subsection describes the optimization problem and simulation setup. The results are presented in the second subsection of this section.

4.2.1 Simulation Setup

The initial configuration of the vehicle is very similar to the powertrain configuration of 2008 Toyota Prius, the baseline specification parameters of the vehicle model are shown in the below table that was explained in the Problem Formulation chapter in detail:

Table 4.1 Initial powertrain specs

Component	**Model**
Generator	52 kW (peak) PM Motor
Energy Storage	5 kWH Li Ion Battery
Motor	50 kW PM Motor
Gearbox	Planetary Gear
Engine	57 kW Engine

The values of the boundaries for the objective function variables, which are calculated via dynamic equations of the performance requirements as explained in detail in Chapter 3, are shown in Table 4.2.

Table 4.2 Boundary values for the constraints

Component	Lower	Upper	Unit
P_M	30	75	kW
P_E	40	85	kW
NBM	6	20	-

Since P/HEVs' major focus is urban driving, the simulations are driven for The EPA Urban Dynamometer Driving Schedule (UDDS) for 5 times consecutively. The UDDS drive cycle is of 7.45 miles and 1369 seconds duration. Table 4.3 shows the characteristics of this specific drive cycle and the following figure is a vehicle speed (mph) vs. time (seconds) plot of the UDDS cycle (the velocity profile over time plot of the UDDS cycle):

Table 4.3 UDDS cycle characteristics

	Max	Average	Stand. Dev.	Unit
Speed	56.7	19.57	14.69	mph
Acceleration	1.4752	0.505	0.45	m/s^2

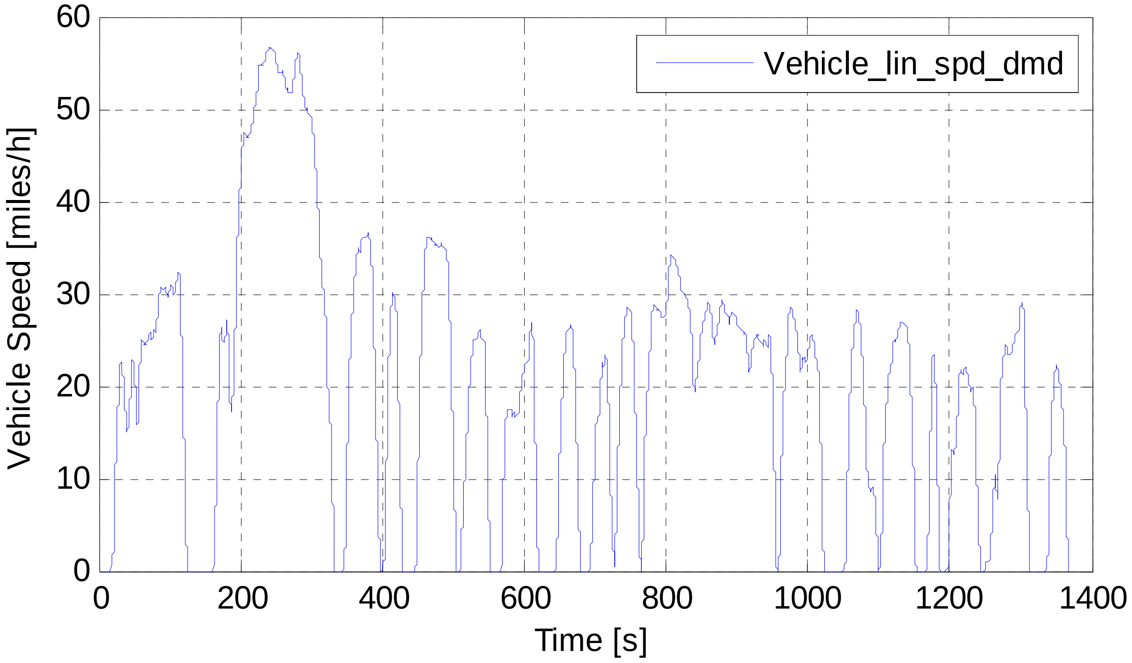

Figure 1.1 UDDS drive cycle velocity profile over time

As described in detail in the Particle Swarm Optimization section of the Problem Formulation chapter, PSO tool needs some parameter values to be determined as explained in the section. The following table represents the values of these PSO tool variables.

Table 4.4 PSO tool parameter values

Parameter	**Value**
c_1	2.6
c_2	1.5
w	0.6
Population Size (# of particles)	10
Maximum iteration number	30

The values of c1, c2 and w parameters are determined by trial and error after running vast number of simulations. Population size and the maximum iteration numbers are restricted because of extremely long simulation hours. However, the advantages of having more particles are explained and examined in the recommendations section of this thesis.

4.2.2 Results

Table 4.5 Comparison of the component sizes and results

Parameter	PSAT	Optimal	unit
P_M	52	58	kW
P_E	57	51	kW
NBM	7	9	-
FC	103.52	134.78	mpg
CO	0	0	g/mile
NOx	0	0	g/mile
HC	0	0	g/mile
$CO2$	86.2	74.8	g/mile

In order to validate the configuration that was found through the optimization process, the default model and the optimized model are simulated in PSAT. Table 4.5 also shows the comparison of these two configurations and their performance results. The default PSAT configuration simulation results are obtained via five consecutive UDDS cycles as well with the component sizes that are determined in PSAT default.

Figure 4.5 and Figure 4.6 shows the comparison plots of the drive cycle torque and velocity demand.

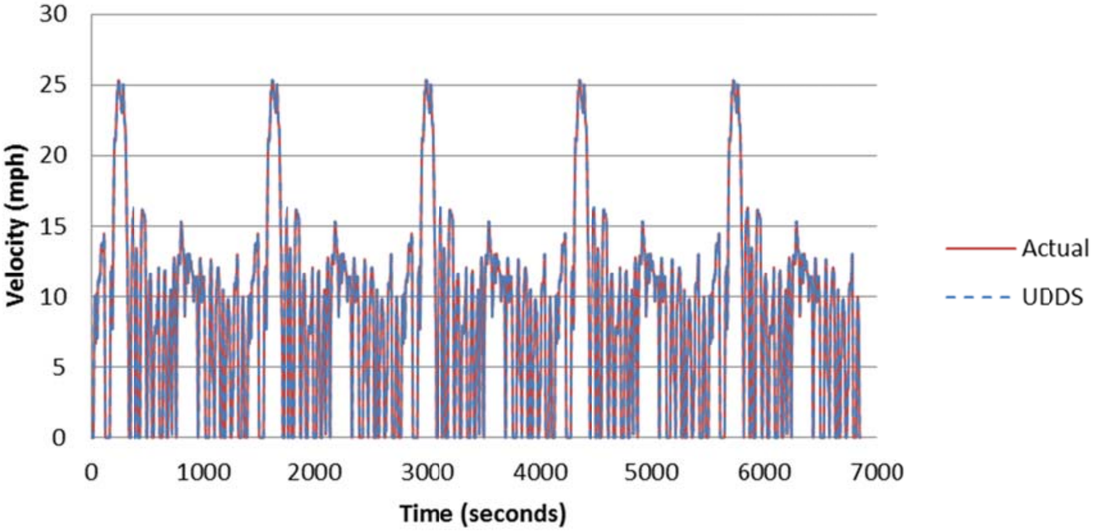

Figure 4.4 UDDS velocity profile vs. actual velocity profile

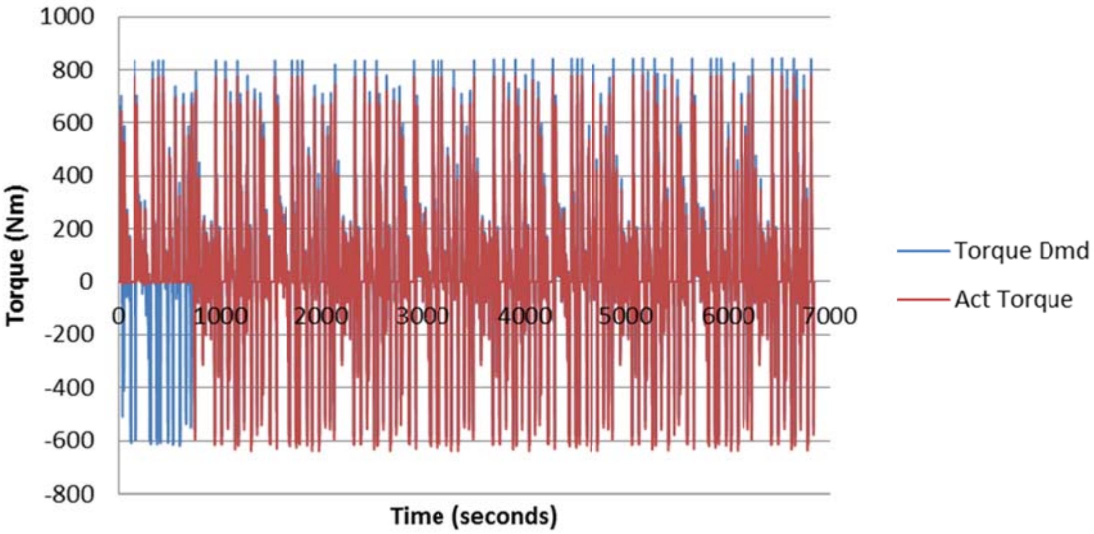

Figure 4.5 Drivecycle torque demand vs. actual final drive torque

Figure 4.7 shows the actual Engine Speed and the Motor Speed correlation. Since these components connected via planetary gear scheme they show similar trend in terms of angular velocity.

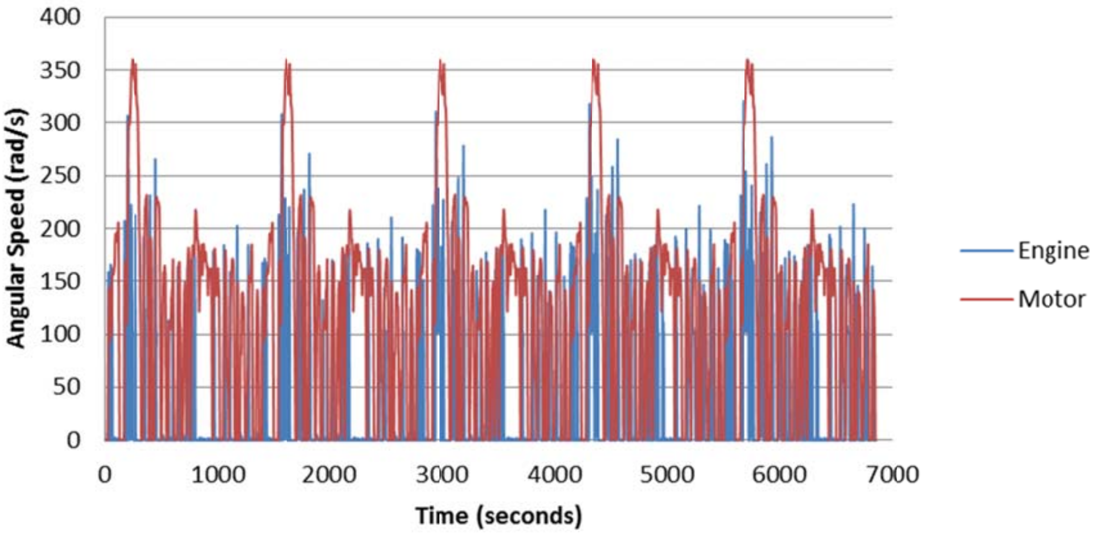

Figure 4.6 Actual engine and motor angular velocity

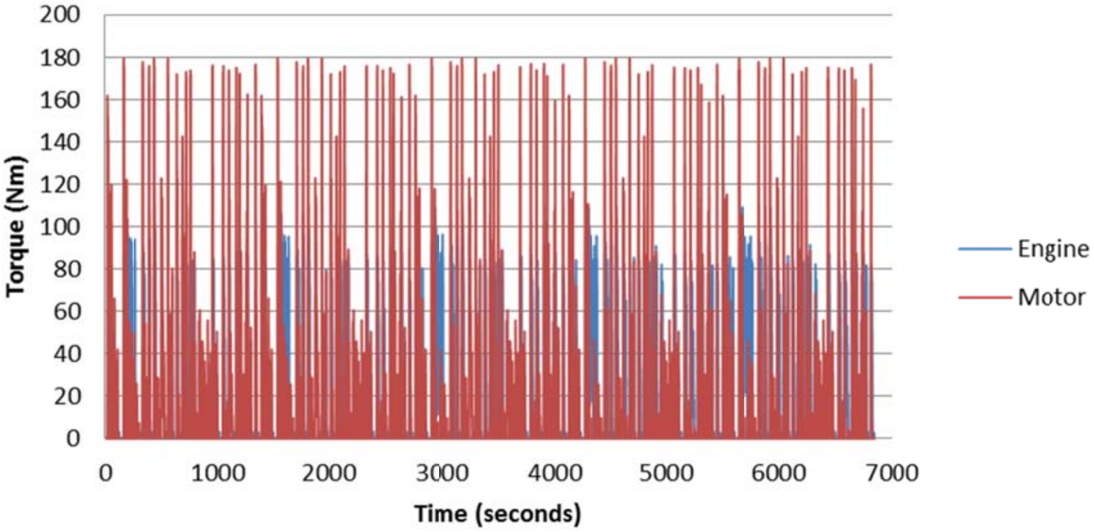

Figure 4.7 Actual engine and motor angular torque

Following two figures show the main objective, enhancing the fuel economy, related results. Figure 4.8, shows the cumulative fuel consumption comparison of the baseline PSAT simulation and PSO component optimized model simulation. The unit of the fuel consumed showed in ‘kg’. The second following figure, Figure 4.9, on the other hand, shows the instantenous fuel consumption in terms of ‘kg/sec’, to show in which parts of the drivecycle the fuel consumption is decreased.

The conversion of the fuel consumption to the fuel economy is given in page 46 of this section in Table 4.5.

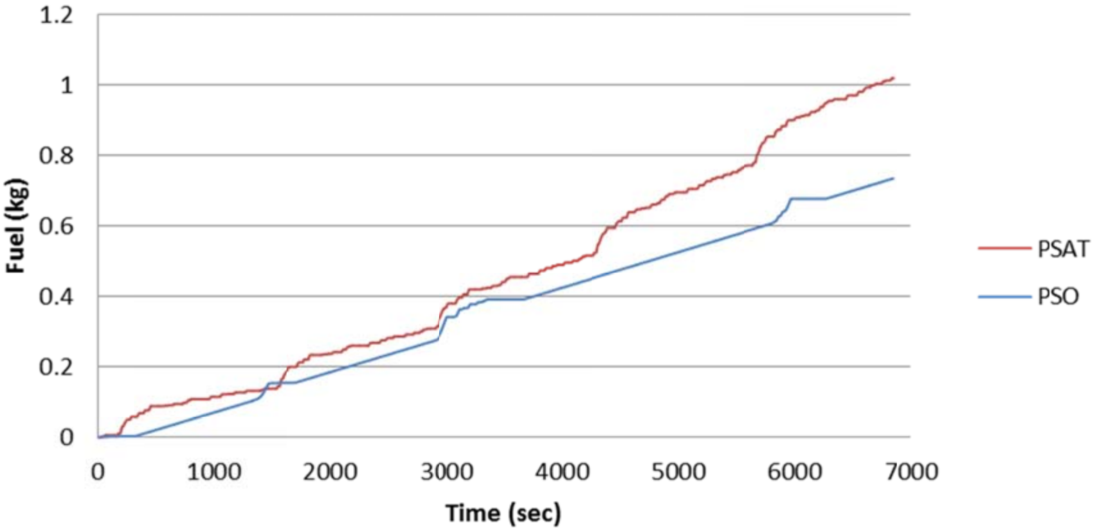

Figure 4.8 Cumulative fuel consumption comparison

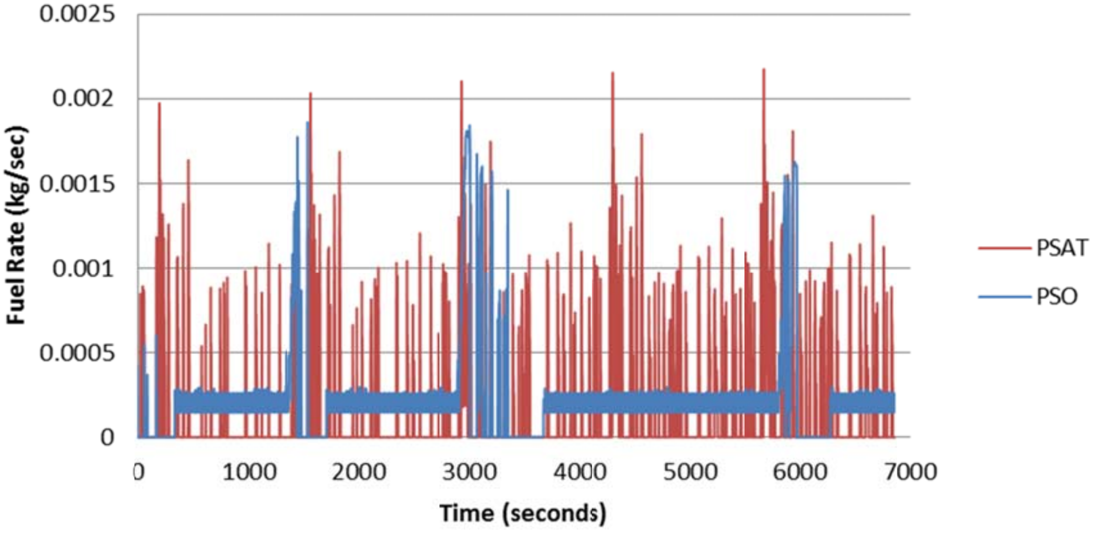

Figure 4.9 Instantaneous fuel consumption comparison

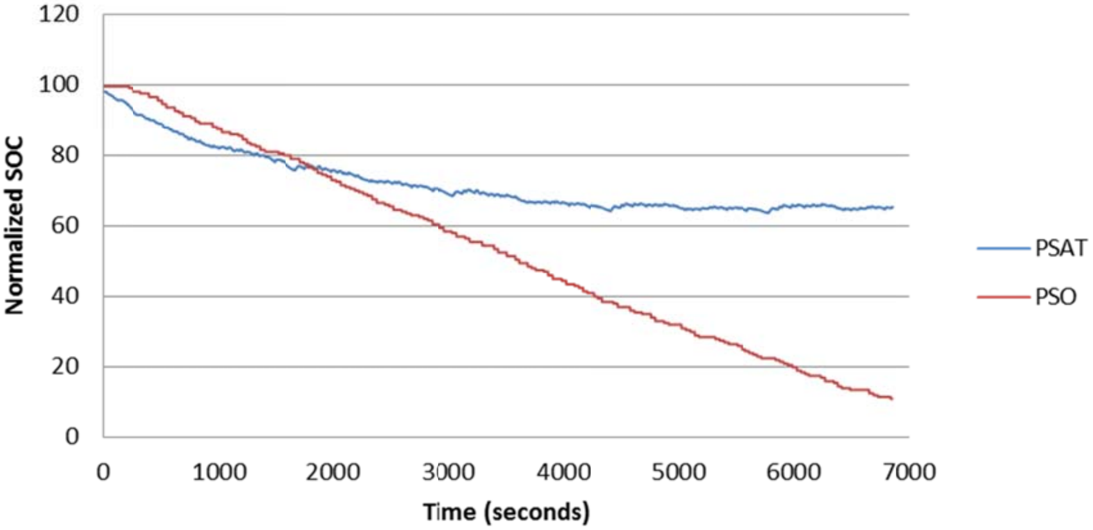

Figure 4.10 State of Charge (SOC) comparison plot

5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The gradient free algorithm, e.g., the particle swarm optimization was used to determine the optimal configuration of the component sizes to achieve a better fuel economy and emission levels. Therefore, a simplified model of a power split plug-in hybrid electric vehicle powertrain was developed for a plug-in hybrid electric vehicle in PSAT. This simplified model was used along with PSO algorithm to determine the optimal sizes of the major components of the vehicle such as, engine power, motor power, and battery energy capacity, constrained by the performance requirements. The computed optimum component sizes were then implemented on the PSAT model. The simulation results from this new configuration were then compared with those from the default PSAT model configuration.

The results show a significant improvement in the fuel economy, almost 30%, of the vehicle with the components that are sized through PSO optimization algorithm compared to the default configuration of the vehicle model. Thus, the main objective of the study, i.e., enhancing the fuel economy, has been achieved. However, since, the measurements of the emission levels are done after the catalytic converter the values of pollutant emissions are significantly too low to compare.

5.2 Recommendations for Future Work

Since the simulation time was significantly long, the optimization problem has been restricted to some certain amount of iterations and parameter values. Therefore, for further enhancement of the system, number of iterations and number of particles that are searching for the global extremes of the scheme can be increased and thus it might be possible to further refine the configuration of the PHEV components.

Furthermore, by implementing different control algorithms to the vehicle model a comparison can be made between the baseline models and the optimized component configurations, also, another comparison amongst each other might lead to a better approach in terms of finding better-tuned component sizes.

Last but not least, there are various kinds of gradient free optimization methodologies, because of the time restrictions and long simulation times, PSO assumed to work better than those other methodologies on PHEV powertrain component sizing systems by researching on the previous work done by different researchers. However, a better practice would be applying those optimization tools on the system and comparing them with PSO results. This might provide more confidence in the algorithm to choose and ensure the most proper optimization tool for such systems.

LIST OF REFERENCES

LIST OF REFERENCES

[1] J.P. Norbye and J. Dunne, "Hybrid Car Ready in 1969" *Popular Science*, p. 86-7, Jul. 1969.

[2] D. Assanis, G. Delagrammatikas, R. Fellini, Z. Filipi, J. Liedtke, N. Sales, and M. Sasene, “An optimization approach to hybrid electric propulsion system design,” *Mech Struct Machines,* 1999, 27(4), p. 393-421

[3] R. Fellini, N. Michelena, P. Papalambros, and M. Sasena, “Optimal design of automotive hybrid powertrain systems,” *Proceedings of International Symposium ON Environmentally Conscious Design and Inverse Manufacturing EcoDesign’99*, p. 400-405.

[4] V. Galdi, L. Ippolito, A. Piccolo, and A. Vaccaro, “A Genetic-Based Methodology for Hybrid Electric Vehicles Sizing,” *Soft Computing 6*, Springer-Verlag 2001.

[5] M. Montazeri-Gh, and A. Poursamad, “Optimization of component sizes in hybrid electric vehicle via genetic algorithms,” *IMECE, ASME International Mechanical Engineering Congress and Exposion,* Orlando, USA, 2005.

[6] Z. Zhengli, Z. Jianwu, and Y. Chengliang, “Optimization Approach for Hybrid Electric Vehicle Powertrain Design,” *Chinese Journal of Mechanical Engineering*, Vol. 18, No. 1, p. 30-36, 2005.

[7] X. Liu, Y. Wu, and J. Duan, “Optimal Sizing of a Series Hybrid Electric Vehicle Using a Hybrid Genetic Algorithm,” *Proceedings of the IEEE, International Conference on Automation and Logistics*, Jinan, China, August 18 – 21, 2007.

[8] A. Hasanzadeh, B. Asaei and A. Emadi. “Optimum design of series hybrid electric buses by genetic algorithm,” *IEEE ISIE 2005,* Dubrovnik, Croatia, p.1465-1470, June 2005.

[9] T. Markel, K. Wipke, "Optimization techniques for hybrid electric vehicle analysis using ADVISOR," *Proceedings of the ASME International Mechanical Engineering Congress and Exposition.* New York, November 11-16, 2001.

[10] X. Wu, B. Cao, J. Wen, and Z. Wang, "Application of Particle Swarm Optimization for Component Sizes in Parallel Hybrid Electric Vehicles," *IEEE Congress on Evolutionary Computation*, p. 2874-2878, 2008.

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

[12] H. Banvait, S. Anwar and Y. Chen, "Plug-in Hybrid Electric Vehicle Energy Management System using Particle Swarm Optimization," *EVS24*, Stavanger, Norway, May 13 - 16, 2009.

[13] H. Banvait, S. Anwar and Y. Chen, "A Rule-Based Energy Management Strategy for Plug-in Hybrid Electric Vehicle," *American Control Conference*, St Louis, Missouri, June 10-12, 2009.

[14] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Neural Networks*, *Proceedings, IEEE International Conference*, Vol. 4, Pages1942-1948, 1995.

[15] Powertrain Systems Analysis Toolkit Documentation, *Argonne National Laboratory,* 2008.

[16] Q. Cao, S. Pagerit, R. B. Carlson, A. Rousseau, PHEV Hymotion Prius Model Validation and Control Improvements, *EVS23*, Anaheim, California, Dec 2-5, 2007.

[17] K. E. Parsopuulos and M. Vrahatis, "Particle Swarm Optimization Method for Constrained Optimization Problems," Intelligent Technologies- Theory and application, IOS Press, p. 214-220, 2002.

[18] X. Wu, B. Cao, J. Wen, and Y. Bian, "Particle Swarm Optimization for Plug-in Hybrid Electric Vehicle Control Strategy Parameter," *IEEE Vehicle Power and Propulsion Conference (VPPC)*, Harbin, China, Sept 3-5, 2008.

[19] Y. Gao and M. Ehsani, "Design and Control Methodology of Plug-in Hybrid Electric Vehicles," *IEEE Vehicle Power and Propulsion Conference (VPPC)*, Harbin, China, September 3-5, 2008.

[20] X. Liu and Y. Wu, and J. Duan, “Optimal Sizing of a Series Hybrid Electric Vehicle Using a Hybrid Genetic Algorithm,” *IEEE International Conference on Automation and Logistics*, Jinan, China, October 2007.

[21] S. Moura and D. Calllaway, H. Fath, and J. Stein, ” Impact of Battery Sizing on Stochastic Optimal Power Management in Plug-in Hybrid Electric Vehicles,” *IEEE International Conference on Vehicular Electronics and Safety*, Columbus, OH, Sept 22-24, 2008.

[22] X. Hu and R. Eberhart. “Solving constrained nonlinear optimization problems with Particle Swarm Optimization,” *Proceedings of the Sixth World Multi Conference on Systemics, Cybernatics and Informatics*, Orlando, Florida 2002.

[23] E. T. Yildiz, Q. Farooqi, S. Anwar, Y. Chen, and Afshin Izadian, “Nonlinear Constrained Component Sizing of a Plug-in Hybrid Electric Vehicle“, *EVS 25*, Shenzhen, China, Nov. 5-9, 2010.

[24] Gregorio Toscano and Pulido C., “A constraint Handling Mechanism for Particle Swarm Optimization,” *Evolutionary Computation*, Vol. 2, p. 1396-1403, 2004.

APPENDICES

Appendix A Matlab Script

```
fcn=@my_function;
clear all
load 5UDDS_PSAT_sim.mat
max_iter=30;
PSAT_step_time=0.1;
num_p = 10;
num_cyc=5;
w=0.3;
for ii = 1:num_p,
  x(1,ii,1)=15+40.*rand;
  x(2,ii,1)=30000+55000.*rand;
  x(3,ii,1)=30000+50000.*rand;
  v(1,ii,1)=15+60.*rand;
  v(2,ii,1)=30000+55000.*rand;
  v(3,ii,1)=20000+50000.*rand;
  pbest_pos(:,ii)=x(:,ii,1);
  %Change_Vars(x(1,ii,1),x(2,ii,1),x(3,ii,1));
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 load 5UDDS_PSAT_sim.mat
  %% Changing Parameter Values Function
%% Files to Run
ess_li_25_616_A123
ess_calculation
ess_cap_erg2pwr_ratio_s_lin
ess.scale.cap_max_des =x(1,ii,1) ;%%%%%%%%%%%%%%%%%%%%%xls read
addition here
ess_cap_erg2pwr_ratio_s_lin
%%ess_old_mass=...
%%num of battery modules

%%% ENG variable change sequence
eng_si_1497_57_US_04Prius

eng_old_mass=eng.init.block_mass+eng.init.tank_mass+eng.init.radiator_m
ass;
 eng_calculation
eng_s_lin
```

```
eng.scale.pwr_max_des = x(2,ii,1);%%%%%%%%%%%%%%%%%%%xls read addition
here
eng_s_lin

%%% Motor variable change sequence
mc_pm_25_50_prius

mc_old_mass=mc.init.motor_mass+mc.init.controller_mass;

mc_pre_calculation
mc_calculation
mc_s
mc.scale.pwr_max_des =x(3,ii,1); %%%%%%%%%%%%%%%%%%%%xls read addition
here
mc_s

% %%% Change the total mass of the vehicle
eng_new_mass=(eng.init.block_mass+eng.init.tank_mass+eng.init.radiator_
mass);
mc_new_mass=(mc.init.motor_mass+mc.init.controller_mass);
veh.init.mass=veh.init.mass+(eng_new_mass-eng_old_mass)+(mc_new_mass-
mc_old_mass);
%% Re-Evaluate Controller files
%%% Propelling
p_stf_split_best_eng_MY04_US_prius
%%% Braking
b_stf_split_best_eng
%% Driveline Calculation
driveline_s
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%
  %sim('PSAT_mdl', [0:0.05:psat.global.gbl_stop_time]);

  sim('PSAT_mdl', [0:0.01:1369*num_cyc]);

        %%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%%


        gal_cons=0.264172052*(max (eng_fuel_cum_simu)/0.749) ;%%
gallons consumed


        mil_cyc=11990*0.000621371192*num_cyc;%%miles driven

        if gal_cons==0
            FC_cyc=190;
        else
        FC_cyc=mil_cyc/gal_cons;
        end
        if FC_cyc>=190
            FC_cyc=190;
        end
```

```
%%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%%

  f(ii,1)=190/FC_cyc+x(1,ii,1)/55+x(2,ii,1)/85000+x(3,ii,1)/80000;
%%%%%%%%%%%%%%%%%%%%%%%%%%%% UPDATEEEEEE
end
pbest(:,1)=f(:,1);%%%pbest(:, this dimension remain 1 always)
[gbest,l]=min(f(:,1));gbest_pos(:,1)=x(:,l,1);
for  kk= 2:max_iter,
    for ii=1:num_p,
        % finding new positions for each type
        for d=1:3
            x(d,ii,kk)=x(d,ii,kk-1)+v(d,ii,kk-1);
            if x(1,ii,kk)<=0
               x(1,ii,kk)=1;
            end
            if x(2,ii,kk)<=0
               x(2,ii,kk)=1000;
            end

            if x(3,ii,kk)<=0
               x(3,ii,kk)=1000;
            end
        end
        v(1,ii,kk)=w.*(v(1,ii,kk-1)+2*rand(1)*(pbest_pos(1,ii)-
x(1,ii,kk-1))+2*rand(1)*(gbest_pos(1,kk-1)-x(1,ii,kk-1)));
        v(2,ii,kk)=w.*(v(2,ii,kk-1)+2*rand(1)*(pbest_pos(2,ii)-
x(2,ii,kk-1))+2*rand(1)*(gbest_pos(2,kk-1)-x(2,ii,kk-1)));
        v(3,ii,kk)=w.*(v(3,ii,kk-1)+2*rand(1)*(pbest_pos(3,ii)-
x(3,ii,kk-1))+2*rand(1)*(gbest_pos(3,kk-1)-x(3,ii,kk-1)));

        %Change_Vars (x(1,ii,kk),x(2,ii,kk),x(3,ii,kk));
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 load 5UDDS_PSAT_sim.mat
            %% Changing Parameter Values Function
%% Files to Run
ess_li_25_616_A123
ess_calculation
ess_cap_erg2pwr_ratio_s_lin
ess.scale.cap_max_des =x(1,ii,kk) ;%%%%%%%%%%%%%%%%%%%%%%xls read
addition here
ess_cap_erg2pwr_ratio_s_lin
%%ess_old_mass=...
%%num of battery modules

%%% ENG variable change sequence
eng_si_1497_57_US_04Prius

eng_old_mass=eng.init.block_mass+eng.init.tank_mass+eng.init.radiator_m
ass;

eng_calculation
eng_s_lin
```

```
eng.scale.pwr_max_des = x(2,ii,kk);%%%%%%%%%%%%%%%%%%%xls read addition
here
eng_s_lin

%%% Motor variable change sequence
mc_pm_25_50_prius

mc_old_mass=mc.init.motor_mass+mc.init.controller_mass;

mc_pre_calculation
mc_calculation
mc_s
mc.scale.pwr_max_des =x(3,ii,kk); %%%%%%%%%%%%%%%%%%%%xls read addition
here
mc_s

%%% Change the total mass of the vehicle
new_mass=(eng.init.block_mass+eng.init.tank_mass+eng.init.radiator_mass
+mc.init.motor_mass+mc.init.controller_mass);
veh.init.mass=veh.init.mass+new_mass-(eng_old_mass+mc_old_mass);
%% Re-Evaluate Controller files
%%% Propelling
p_stf_split_best_eng_MY04_US_prius
%%% Braking
b_stf_split_best_eng
%% Driveline Calculation
driveline_s
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%
        %sim('PSAT_mdl', [0:0.05:psat.global.gbl_stop_time]);
        sim('PSAT_mdl', [0:0.01:1369*num_cyc]);
        ess_soc_simu %%to check if the variable is changing every
iteration
        %%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%%


        gal_cons=0.264172052*(max (eng_fuel_cum_simu)/0.749) ;%%
gallons consumed


        mil_cyc=11990*0.000621371192*num_cyc;%%miles driven

        if gal_cons==0
            FC_cyc=190;
        else
        FC_cyc=mil_cyc/gal_cons;
        end
        if FC_cyc>=190
            FC_cyc=190;
        end
        %%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%%
```

```
f(ii,kk)=190/FC_cyc+x(1,ii,1)/55+x(2,ii,1)/85000+x(3,ii,1)/80000;
%%%%%%%%%%%%%%%%%%%%%%%%%%%% UPDATEEEEEE,NORMALIZE

        if pbest(ii,1)> f(ii,kk)
            pbest(ii,1)=f(ii,kk);
            pbest_pos(:,ii)=x(:,ii,kk);
        end
    end
    [gbest,l]=min(pbest(:,1));%%%%%%%%%%%%GLOBAL BEST IS THE BEST AMONG
THE PBEST VALUES, SINCE THE PBEST VALUES ARE STORING THE BEST EACH
PARTICLE FOUND SO FAR
    gbest_pos(:,kk)=pbest_pos(:,l);
end
```

Appendix B Objective Function Values

```
x(:,:,1) =

  1.0e+004 *

  Columns 1 through 6

    0.0053    0.0018    0.0029    0.0052    0.0016
0.0027
    5.5336    6.1931    8.2896    6.3140    5.5967
6.5776
    4.4382    4.4351    7.7678    6.5107    6.2254
7.4570

  Columns 7 through 10

    0.0051    0.0041    0.0040    0.0021
    4.4109    6.0055    8.4235    7.4601
    7.8340    4.2566    4.0358    5.3825


x(:,:,2) =

  1.0e+005 *

  Columns 1 through 6

    0.0007    0.0005    0.0006    0.0011    0.0005
0.0009
    1.1736    1.3180    1.5515    1.1432    1.1445
1.0732
    0.7204    1.0727    1.2846    0.9768    0.9454
1.1452

  Columns 7 through 10

    0.0010    0.0006    0.0010    0.0004
    0.8320    1.0289    1.6298    1.5277
    1.3965    0.8062    0.8397    1.0558
```

```
x(:,:,3) =

  1.0e+005 *

  Columns 1 through 6

    0.0006    0.0007    0.0006    0.0012    0.0006
0.0011
    1.3612    1.5276    1.7468    1.2961    1.3306
1.1949
    0.8032    1.2614    1.2530    0.9708    0.9476
1.1048

  Columns 7 through 10

    0.0012    0.0005    0.0011    0.0005
    0.9830    1.1646    1.8173    1.7441
    1.4190    0.9270    0.9896    1.1647


x(:,:,4) =

  1.0e+005 *

  Columns 1 through 6

    0.0003    0.0003    0.0004    0.0005    0.0004
0.0007
    0.9884    1.0946    1.5917    1.0184    1.1626
1.0121
    0.6563    0.7525    0.9140    0.4836    0.7923
0.9522

  Columns 7 through 10

    0.0009    0.0003    0.0009    0.0004
    0.8408    0.8488    1.3817    1.3055
    0.8536    0.6585    0.7185    0.9167


x(:,:,5) =
  1.0e+005 *

  Columns 1 through 6

   0.0002    0.0000    0.0001    0.0000    0.0001
0.0005
   0.3978    0.7032    0.6357    0.3394    0.7706
0.7988
   0.3517    0.3681    0.2497    0.0638    0.4894
0.4557

  Columns 7 through 10

   0.0002    0.0000    0.0001    0.0002
   0.3781    0.3274    1.0679    0.3938
   0.0907    0.2711    0.2564    0.3119


x(:,:,6) =

  1.0e+004 *

  Columns 1 through 6

   0.0015    0.0001    0.0001    0.0001    0.0001
0.0006
   0.0568    2.4570    0.1000    0.5457    2.1168
4.2054
   1.5540    0.1000    0.1000    0.1000    2.9378
1.9948

  Columns 7 through 10

   0.0001    0.0001    0.0001    0.0001
   0.1000    0.1000    6.8828    0.1000
   0.1000    0.1000    0.1000    0.1000


x(:,:,7) =

  1.0e+004 *
  Columns 1 through 6

    0.0023    0.0001    0.0001    0.0024    0.0001
0.0001
    0.1000    0.9256    0.1000    1.1316    0.1000
2.4824
    1.6778    0.1000    0.6395    5.2664    2.2732
2.6590

  Columns 7 through 10

    0.0001    0.0025    0.0001    0.0001
    0.1000    0.8336    3.5347    0.6699
    1.6111    0.1000    0.9652    0.0197


x(:,:,8) =

  1.0e+005 *

  Columns 1 through 6

    0.0003    0.0001    0.0001    0.0005    0.0001
0.0000
    0.4241    0.4296    0.6867    0.6333    0.2733
0.2937
    0.3454    0.3325    0.2549    1.1599    0.2995
0.5738

  Columns 7 through 10

    0.0000    0.0004    0.0003    0.0001
    0.2460    0.3623    0.2597    0.5128
    0.8559    0.0417    0.1926    0.3834


x(:,:,9) =

  1.0e+005 *

  Columns 1 through 6

    0.0004    0.0002    0.0003    0.0006    0.0002
```

```
0.0001
   0.9824    1.0338    1.0395    1.1377    0.6920
0.5359
   0.5290    0.7478    0.8223    1.3310    0.4120
0.7601

  Columns 7 through 10

   0.0001    0.0005    0.0006    0.0002
   0.7488    0.9101    0.4685    1.2130
   1.2411    0.2726    0.3159    0.8373


x(:,:,10)
=

  1.0e+005 *

  Columns 1 through 6

   0.0005    0.0003    0.0004    0.0005    0.0003
0.0003
   1.2283    1.3732    1.1573    1.2839    0.9686
0.7633
   0.6516    0.9399    1.1067    0.8764    0.6056
0.8832

  Columns 7 through 10

   0.0001    0.0004    0.0006    0.0002
   1.1913    1.2496    1.0057    1.4880
   1.2499    0.6542    0.4171    1.0606


x(:,:,11)
=

  1.0e+005 *

  Columns 1 through 6

   0.0005    0.0003    0.0004    0.0003    0.0003
0.0004
```

```
    1.0562    1.1999    1.0709    1.0394    1.0233
0.9488
    0.6655    0.7551    1.1635    0.6641    0.7268
0.7960

  Columns 7 through 10

    0.0003    0.0003    0.0005    0.0003
    1.1197    1.2153    1.2649    1.2276
    1.0119    0.8826    0.5308    0.9645


x(:,:,12)
=

  1.0e+005 *

  Columns 1 through 6

    0.0003    0.0002    0.0003    0.0001    0.0002
0.0004
    0.6396    0.6992    0.8159    0.5514    0.8079
0.9118
    0.6119    0.3437    0.6362    0.4079    0.7299
0.6355

  Columns 7 through 10

    0.0004    0.0003    0.0004    0.0003
    0.3739    0.9635    1.1283    0.8182
    0.4954    0.8340    0.5677    0.5098


x(:,:,13)
=

  1.0e+004 *

  Columns 1 through 6

    0.0028    0.0006    0.0020    0.0013    0.0015
0.0026
    0.7037    2.4423    7.2063    3.2813    3.2017
```

```
8.3998
    3.9732    0.1000    2.1176    2.4012    5.9819
4.7682

  Columns 7 through 10

    0.0044    0.0026    0.0038    0.0019
    0.1000    6.6168    5.3296    3.0218
    0.1000    6.4864    4.7369    2.1458


x(:,:,14)
=

  1.0e+004 *

  Columns 1 through 6

    0.0021    0.0001    0.0007    0.0019    0.0009
0.0001
    0.1000    0.9022    6.6185    2.9514    1.1440
5.5540
    1.8720    0.1000    0.6317    2.8767    4.9581
4.0967

  Columns 7 through 10

    0.0040    0.0029    0.0028    0.0013
    0.1000    2.4410    2.0609    1.0194
    0.1000    2.1502    3.1128    1.4087


x(:,:,15)
=

  1.0e+004 *

  Columns 1 through 6

    0.0022    0.0010    0.0005    0.0028    0.0007
0.0001
    2.6734    2.7938    6.0653    3.3966    0.8579
4.1131
```

```
   1.4978    1.5604    2.7878    5.3828    4.2651
3.7874

  Columns 7 through 10

   0.0036    0.0032    0.0018    0.0011
   4.6060    0.6307    1.8291    1.0546
   2.6114    0.1000    2.3614    2.5748


x(:,:,16)
=

  1.0e+004 *

  Columns 1 through 6

   0.0038    0.0027    0.0017    0.0035    0.0012
0.0006
   5.5708    9.2767    5.8977    6.5851    3.2401
4.4262
   3.1471    3.6535    8.2301    7.5794    4.5627
4.8890

  Columns 7 through 10

   0.0027    0.0036    0.0017    0.0012
   7.8363    2.5459    5.1950    5.6604
   6.6198    0.6193    2.8996    4.5841


x(:,:,17)
=

  1.0e+005 *

  Columns 1 through 6

   0.0005    0.0004    0.0003    0.0005    0.0002
0.0002
   0.9045    1.3239    0.6444    1.0401    0.7290
0.5534
   0.4180    0.5542    1.3588    0.8376    0.5647
```

```
0.6289

  Columns 7 through 10

    0.0002    0.0003    0.0002    0.0002
    0.9380    0.6539    0.9314    0.8235
    1.0512    0.3313    0.4152    0.6269


x(:,:,18)
=

  1.0e+005 *

  Columns 1 through 6

    0.0005    0.0004    0.0004    0.0005    0.0002
0.0003
    1.0329    1.1785    0.7437    1.1326    0.9298
0.7656
    0.5538    0.6478    1.3274    0.7021    0.6309
0.7798

  Columns 7 through 10

    0.0003    0.0003    0.0002    0.0003
    0.8544    0.8367    1.1305    0.9550
    1.1819    0.6531    0.5166    0.6913


x(:,:,19)
=

  1.0e+005 *

  Columns 1 through 6

    0.0004    0.0003    0.0004    0.0004    0.0002
0.0005
    0.7874    0.8261    0.7781    0.9621    0.9150
0.8797
    0.6169    0.5987    0.7941    0.3825    0.6595
0.8555
```

```
  Columns 7 through 10

    0.0004    0.0002    0.0003    0.0003
    0.6675    0.8747    0.9838    0.9278
    1.0312    0.7938    0.5488    0.6304


x(:,:,20)
=

  1.0e+004 *

  Columns 1 through 6

    0.0025    0.0019    0.0026    0.0025    0.0021
0.0043
    4.7361    3.5996    7.3021    5.9519    6.4451
8.3905
    5.5105    4.8872    3.2775    1.7398    5.7131
7.2613

  Columns 7 through 10

    0.0045    0.0022    0.0037    0.0028
    4.6452    7.3144    7.1950    7.7790
    4.5810    6.6383    5.1960    4.0500


x(:,:,21)
=

  1.0e+004 *

  Columns 1 through 6

    0.0026    0.0010    0.0018    0.0018    0.0016
0.0030
    2.5110    0.7719    6.6023    3.4460    4.1298
6.6190
    3.9369    3.5566    1.2057    1.9836    4.3967
5.0163
```

```
  Columns 7 through 10

    0.0032    0.0021    0.0040    0.0019
    3.6483    6.3622    4.8809    5.7762
    0.1000    5.0038    3.7335    2.9322


x(:,:,22)
=

  1.0e+004 *

  Columns 1 through 6

    0.0038    0.0006    0.0015    0.0020    0.0012
0.0006
    1.9473    2.4096    5.9252    2.8079    3.3102
5.1312
    2.4113    2.8476    1.3192    3.8939    3.9785
3.5000

  Columns 7 through 10

    0.0016    0.0021    0.0036    0.0008
    3.3524    5.1983    3.9364    4.5548
    0.1000    2.2306    2.8729    2.9919


x(:,:,23)
=

  1.0e+004 *

  Columns 1 through 6

    0.0052    0.0008    0.0016    0.0034    0.0011
0.0001
    4.4048    5.8298    6.6058    3.6845    3.7462
4.6028
    2.0985    3.0054    3.1139    7.7027    4.4173
3.8533

  Columns 7 through 10
```

```
   0.0011    0.0027    0.0030    0.0004
   4.5906    4.6610    5.0912    4.8002
   5.2192    1.1892    2.9158    4.9480


x(:,:,24)
=

  1.0e+005 *

  Columns 1 through 6

   0.0006    0.0001    0.0002    0.0004    0.0001
0.0000
   0.7609    0.9384    0.7011    0.5574    0.4768
0.4904
   0.3910    0.3463    0.5140    1.0415    0.4722
0.5114

  Columns 7 through 10

   0.0002    0.0003    0.0002    0.0001
   0.6572    0.5214    0.7752    0.6014
   1.2340    0.1139    0.4286    0.5960


x(:,:,25)
=

  1.0e+005 *

  Columns 1 through 6

   0.0005    0.0002    0.0003    0.0004    0.0002
0.0001
   0.8836    1.0760    0.7990    0.7770    0.5993
0.6249
   0.5512    0.4732    0.7897    1.0876    0.5649
0.7043

  Columns 7 through 10
```

```
   0.0005    0.0004    0.0002    0.0002
   0.7865    0.6169    0.9404    0.7913
   1.4917    0.2936    0.5249    0.6115


x(:,:,26)
=

  1.0e+005 *

  Columns 1 through 6

   0.0004    0.0003    0.0003    0.0004    0.0002
0.0003
   0.7906    1.0041    0.8429    0.8568    0.7150
0.7526
   0.6408    0.5808    0.9622    0.6845    0.6316
0.7565

  Columns 7 through 10

   0.0007    0.0004    0.0002    0.0003
   0.6854    0.6745    0.9161    0.8891
   1.0375    0.6321    0.5510    0.5381


x(:,:,27)
=

  1.0e+004 *

  Columns 1 through 6

   0.0019    0.0026    0.0029    0.0037    0.0022
0.0041
   5.8303    8.6117    8.1272    8.2314    7.3999
8.0688
   5.9048    5.9168    8.4716    2.5952    6.4889
7.7379

  Columns 7 through 10

   0.0073    0.0035    0.0033    0.0038
```

```
    5.7577    6.8929    7.5235    8.3699
    4.3998    7.8654    5.0191    4.0551


x(:,:,28)
=

  1.0e+004 *

  Columns 1 through 6

    0.0013    0.0023    0.0026    0.0036    0.0017
0.0037
    3.7344    6.0664    6.9169    5.7993    6.2926
7.2287
    4.1373    5.2432    6.9601    0.3856    6.2907
6.7329

  Columns 7 through 10

    0.0070    0.0025    0.0044    0.0022
    4.2966    6.5288    6.0551    7.2746
    1.1081    6.4992    3.7342    3.4758


x(:,:,29)
=

  1.0e+004 *

  Columns 1 through 6

    0.0013    0.0017    0.0023    0.0037    0.0012
0.0027
    3.0764    4.3263    6.0803    3.5986    4.9030
5.9897
    1.9046    4.3962    5.5071    2.4690    5.9130
5.8370

  Columns 7 through 10

    0.0030    0.0021    0.0042    0.0000
    3.4889    6.0882    5.6682    5.9125
```

```
   1.2838    3.3244    2.7654    3.5205


x(:,:,30)
=

  1.0e+004 *

  Columns 1 through 6

   0.0032    0.0011    0.0021    0.0033    0.0010
0.0009
   3.5644    3.9379    5.5450    3.1582    4.3037
5.1214
   1.4304    3.6035    4.6426    6.0252    5.7555
5.1504

  Columns 7 through 10

   0.0000    0.0019    0.0040    0.0001
   3.3327    5.6521    6.4933    5.1792
   3.6137    0.9583    2.8141    4.1345
```