Optimal Design of Automotive Hybrid Powertrain Systems

Ryan Fellini, Nestor Michelena, Panos Papalambros, and Michael Sasena
Department of Mechanical Engineering & Applied Mechanics
The University of Michigan, Ann Arbor, Michigan 48109, United States
{rfellini,nestorm,pyp,msasena}@umich.edu
http://ode.engin.umich.edu

Abstract

Alternative powertrains for automotive applications aim at improving emissions and fuel economy. Lack of experience with these relatively new technologies makes them ideal applications for computer-based modeling and simulation studies. There is a variety of configurations, control strategies, and design variable choices that can be made. If mathematical models exist, rigorous optimization techniques can be used to explore the design space. This paper provides an overview of a design environment for alternative powertrains that has these characteristics: modularity, allowing a system to be built by combining components; flexibility, allowing different levels of fidelity and different existing codes to be used; and, rigor, since it is based on mathematical methods of decision making. A simple application to a hybrid diesel-electric powertrain is included.

1. Introduction

Automotive engine emissions are recognized as a major source of concentrated pollution, particularly in urban areas. Although the internal combustion engine remains the dominant prime mover for technological and cost reasons, alternative propulsion systems have been under consideration for some time and have recently become available in commercial products. Much of the present motivation for such products derives from government regulations and corporate recognition of public consciousness. Nevertheless, the automotive industry widely recognizes that widespread use of alternative powertrains will be inevitable at some point in the future.

Given present limitations in technologies such as batteries and fuel cells, the most viable powertrain alternatives are hybrid configurations that include a relatively small internal combustion engine. Since practical experience with such configurations is limited and hardware prototypes are expensive, use of mathematical modeling and simulation techniques is particularly attractive for exploring preliminary designs and studying trade-offs.

The research effort outlined in this article aims at developing a modular simulation and design environment that can be used to study a variety of vehicles and powertrain configurations. Simulations are based on physical models and experimental validations. Optimization algorithms are used to drive design iterations in search of the best possible design according to some criterion. An object-oriented software architecture allows easy integration of legacy codes and new models in familiar environments such as MATLAB or in one based on CORBA-compliant distributed objects.

Hybrid powertrain simulation software includes codes such as ADVISOR [3], from the U.S. National Renewable Energy Laboratory (NREL), and proprietary software used by the U.S. automotive industry. These simulations are integrated with high-fidelity engine simulations, particularly for advanced diesel engines, and special-purpose optimization algorithms. Vehicle handling simulations and advanced light-weight body structure designs are also available and should be gradually integrated to build an overall analysis and design environment for hybrid electric vehicles (HEV).

We present the basic elements of this research effort to date and some illustrative results.

2. Software design environments

Hybrid powertrains are complex electromechanical systems that can be configured in a variety of ways. To examine design scenarios prior to building physical prototypes requires a sophisticated simulation and computing environment. Such a design environment should allow for easy integration of engineering models and simulations, along with the capabilities to perform visualization and optimization studies. In our efforts to build a design environment for hybrid electric vehicles, we have developed two practical approaches for analysis and design tool integration: the first one uses distributed object technology whereas the second is based on the commercial MATLAB-SIMULINK computing environment.

2.1 Object-oriented, CORBA-based environment

Object-Oriented Programming (OOP) is now recognized as an important development in software engineering. OOP gives the programming team a language in which software can be developed with a rigid structure that effects reusable and encapsulated code. As a result many individuals working on a project can share code without having to be aware of details of internal data structures and procedures.

Recently the advantages of OOP have been incorporated into the development of libraries for mathematical optimization. Such libraries include the Hilbert Class Library (HCL) [6] and the Template Numerical Toolkit (TNT) [12], which are collections of fundamental mathematical objects that can be used for designing and developing mathematical programming-based software. Other libraries that also include a number of optimization algorithms are the CWP Object-Oriented Optimization Library (COOOL) [4] and Opt++ [8].

An object-oriented design framework has been developed at the University of Michigan, building on some of the basic concepts in COOOL and Opt++. This framework [14] specifies formal interfaces between several types of distributed components. The components include subsystem models, design models, search engines, design model partitioning, design coordination, and user interface, Figure 1. The diagram shows only the subclasses of the design model class and the possible multiplicity of search engines, analysis models, and design submodels. The object-oriented design of this framework allows further addition of other components.

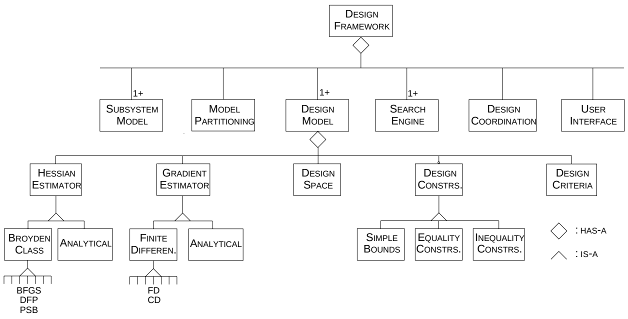


Figure 1: Object-oriented optimization framework

This object-oriented optimization framework is taken one step further with the introduction of the CORBA standard into the architecture. CORBA (Common Object Request Broker Architecture) is an industry standard for distributed, heterogeneous, object-oriented applications, which is open, robust, interoperable, multiplatform and multivendor supported. CORBA uses the Interface Definition Language (IDL) to specify attributes, operations and parameters of each operation that a server provides in the implementation of a given object. CORBA objects can be accessed by remote clients via method invocations, Figure 2. Both the language and the compiler used to create server objects are totally transparent to clients. Clients do not need to know where the distributed object resides, what operating system it executes on and how the server object is implemented. Clients only need to know the interface their server objects publish. CORBA allows encapsulated engineering analysis objects to be placed on a distributed computing environment and the creation of libraries of such objects.

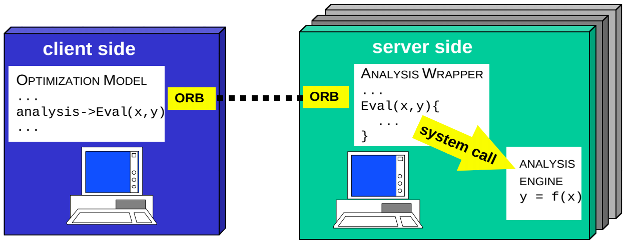


Figure 2: CORBA distributed network call

These computing infrastructure properties are important in an actual vehicle design environment where different users, possibly from different supplier companies, are tying their respective simulation modules together for an overall system study.

2.2 MATLAB-based environment

MATLAB has several attractive features as an environment to develop optimization tools: namely, platform independence, an optimization toolbox, a high level programming language, and sophisticated visualization tools. MATLAB's wide use in academic and industry sites is another attractive feature.

A MATLAB design optimization framework is built by expanding the library of algorithms in the optimization toolbox and by developing interfaces that produce and parse simulation input and output files and system-level scripts that enable distributed analysis by means of remote process calls.

In our present effort, existing simulations in MATLAB for hybrid vehicles, engines, and other vehicle components, are augmented with improved models and integrated with mathematical optimization algorithms. There are important practical issues related to this integration, but they will not be addressed here.

3. Hybrid electric vehicle models

Various computer programs exist for the simulation of vehicle powertrains, in particular hybrid electric powertrains. Such software tools include SIMPLEV, CarSim, HVEC, CSM HEV, V-Elph and, most recently, ADVISOR (for more details see [15]). The US government under the initiative called “Partnership for a New Generation of Vehicles” (PNGV) has sponsored both public and proprietary work in this area. An example is the proprietary System Analysis Toolkit (SAT) developed by a consortium of US automotive companies. Each major automotive company has its own specialized, proprietary modeling software. ADVISOR [3], a public domain program, was selected as the basic simulation used to estimate design metrics in the study described later in this article.

3.1 Description of ADVISOR simulation

ADVISOR is a MATLAB/SIMULINK-based, feedbackward simulation of hybrid electric powertrains. ADVISOR allows quick analysis of the performance, emissions, and fuel economy of conventional, electric, and hybrid vehicles. The component models in ADVISOR are empirical, relying on input/output relations measured in the laboratory, and quasi-static, using data collected in steady state tests and correcting them for transient effects such as the rotational inertia of drivetrain components.

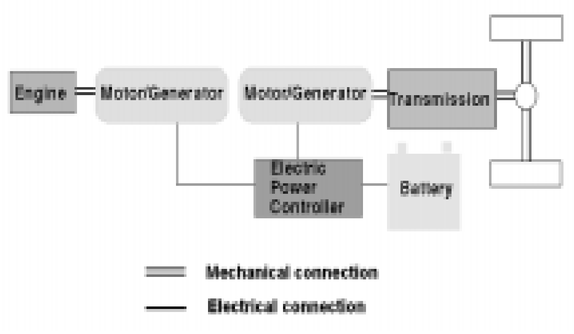


Figure 3: HEV series configuration

The series hybrid configuration, Figure 3, consists of an electric motor driving the wheels, receiving power from an onboard battery pack. This is essentially the configuration used in a battery-driven electric vehicle. The hybrid addition is in the form of a fuel converter, which is usually a small internal combustion engine whose main function is to extend vehicle range. When the state of charge of the battery reaches a specified lower limit, the fuel converter engages to recharge the batteries.

The parallel hybrid system, Figure 4, consists of both an electric motor and a fuel converter that can simultaneously or individually drive the vehicle, depending on the control strategy. A common scheme is to use the electric motor for city driving, and both the engine and the motor for highway driving. The engine can then both power the vehicle and recharge the battery pack, using the motor as a generator. Both the series and parallel configurations allow for regenerative braking.

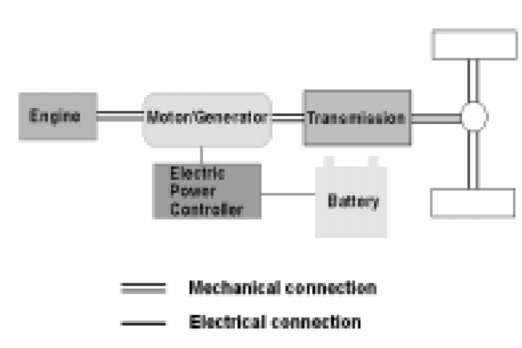


Figure 4: HEV parallel configuration

3.2 Component models

In addition to design studies of overall hybrid powertrain systems that tend to use relatively simple models for vehicle components, we have looked into the integration of system simulations with high-fidelity subsystem and component models, such as models of engines, CVT transmissions, and batteries.

Inclusion of detailed component models extend the set of design variables to subsystems and components. Higher fidelity can be so introduced and decisions can be made at an increased level of detail. Also, a component may be studied in great detail while tracking the impact of local design decisions on the overall system—which is included in the study with much less detail.

3.3 Integration of engine simulation

One important subsystem that can be introduced in greater detail is the internal combustion powerplant. The Turbo Diesel Engine Simulation (TDES) program was selected as the model to be integrated with ADVISOR [1]. The integration increases the accuracy of the engine maps for fuel consumption and performance calculations used within ADVISOR. TDES is a feed-forward simulation derived from fundamental thermodynamic equations that calculates engine properties at each crank degree angle. TDES integration extends the set of design variables to include quantities such bore, stroke, valve events, and manifold geometry. TDES was wrapped as a CORBA object before being placed in the object-oriented optimization environment described in Section 2.

4. Optimization algorithms

In mathematical optimization, a mathematical model (which could be an implicit computer simulation) is used to evaluate design criteria and constraint functions. Formal methods are employed to search the design space defined by the constraint functions and identify a point or points that maximizes or minimizes the design criteria. This approach presents several challenges when applied to system design. System simulation responses are expensive to compute but also multimodal, noisy and/or discontinuous functions. In these cases, gradient-based optimization algorithms, such as the well-known Sequential Quadratic Programming (SQP) algorithm, can run into trouble because of inaccurate gradient information used to determine search directions and convergence.

Derivative-free optimization techniques can be used to address the above issues. However, they tend to need a much larger number of iterations and/or function evaluations which may make them impractical. A third approach is to use metamodels, namely, simpler models derived from the more complicated ones. These simpler models can then be used for the optimization study. Table 1 gives a rough classification of the algorithms currently used in the hybrid powertrain system design environment.

Algorithm	Advantages	Disadvantages
SQP	• best general purpose code • very efficient on smooth problems	• requires accurate derivatives • only finds local solution
Trajectory	• deals with small scale noise more effectively	• requires derivatives • slow convergence towards solution
Complex	• derivative-free • better chance of getting through local solutions and large noise	• starting points must be feasible • stochastic (random) in nature • slow convergence towards solution
DIRECT	• derivative-free • searches both locally and globally • no starting point or control parameters needed	• lack of formal convergence criteria • suitable only for small number of variables
SMO	• can be efficient • can smooth out data • provides insight to problem	• often fails with multimodal problems • suitable only for small number of design variables

Table 1: Optimization algorithms under study

The SQP algorithm is a standard one, such as the one available in MATLAB's toolbox. The trajectory algorithm is due to Snyman [16] and is less sensitive to derivative accuracies. The complex algorithm is a variant of the classical constraint simplicial algorithm by Box [2]. The DIRECT algorithm is a derivative-free global optimization algorithm based on Lipschitz constants and proposed by Jones [7]. The sequential metamodel optimization (SMO) algorithm was written by Sasena [13] and uses a sequence of metamodels to "zoom in" to the optimum.

As design models become larger, it becomes necessary to try and distribute the design decision-making process by some form of problem partitioning and solution coordination. The advantage of the partitioned design model is that smaller, more manageable, design problems are solved. Same level subproblems may be solved in parallel, using the optimization technique most suitable for the underlying submodel, gaining in robustness, speed and engineering interpretation of results. Formal methods have been developed and implemented that allow the use of coordination strategies such as Sequentially Decomposed Programming[10] and Hierarchical Overlapping Coordination [9]. It is important that such methods are not ad hoc, but provide some assurance of convergence to the system optimum.

5. HEV design studies

A hybrid electric parallel powertrain was selected for the design studies. This configuration is considered to be more efficient than the series one. A commercial implementation is the Toyota Prius. The vehicle we consider in this study is based on the Chevrolet Lumina midsize sedan.

5.1 Problem definition

The optimization problem statement was derived by specifying vehicle mission targets similar to those put forth by the US PNGV.

The mathematical problem statement is as follows:

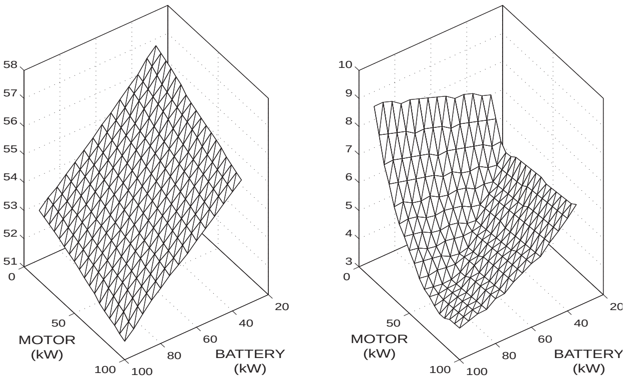
$$
\begin{array}{l}
\text{maximize}\ldots \\
f(x) = \text{mpg (fuel economy)} \\
\text{with respect to}\ldots \\
x = \{\text{engine size, motor size, battery size}\} \\
\text{subject to}\ldots \\
\qquad 0-60\,\text{mph time} \le 12\,\text{s} \\
\qquad 40\text{–}60\ \text{mph (passing time)} \le 5.3\,\text{s} \\
\qquad 0-85\,\text{mph time} \le 23.4\,\text{s} \\
\qquad \text{max acceleration} \ge 0.5\,g \\
\qquad \text{max speed} \ge 85\,\text{mph} \\
\qquad 5\text{s distance} \ge 140\,\text{ft} \\
\qquad 55\text{mph cruise grade} \ge 6.5\,\text{percent}
\end{array}
$$

This model states that we wish to maximize the fuel economy over a combination of the U.S. Federal Urban Driving Schedule (FUDS) and U.S. Federal Highway Driving Schedule (FHDS) subject to some desirable performance requirements. Note that the values used here correspond to just one design scenario. Other scenarios can be examined by changing the bounds on the inequalities or by changing the driving schedule combinations.

5.2 Response surface study

Before the optimization study itself, it is useful to make a quick examination of the mathematical nature of the problem functions. Recall that these are not really known since they are computed implicitly by the simulations.

Plots of the response surfaces for various design variables were produced to further the understanding of the design problem. Figure 5 shows two such responses: fuel economy and 40 to 60 m.p.h. passing time.



FUEL ECONOMY (mpg) PASSING TIME (s)

Figure 5: Response surfaces of parallel HEV

Although the objective function appears to be a smooth surface, zooming in further does find small-scale noise—possibly caused by integration errors within the MATLAB routines. It was also found that most of the constraints were monotonic and contained relatively low levels of noise. However, in some regions the noise can be quite pronounced, which will inevitably cause difficulties with gradient-based optimization algorithms.

5.3 Optimization results

The optimization studies were performed using ADVISOR integrated with TDES. Four SUN Ultra10 workstations were used to generate engine maps by running TDES on the CORBA environment. Given the presence of noise, two derivative-free optimization algorithms were selected to solve the optimization problem, namely DIRECT and Complex.

The baseline design had a 1.45 L diesel engine, a 55 kW electric motor, and a 65 kW battery, with a resulting fuel economy of 43.41 m.p.g. Results of the optimization studies are shown in Table 2. Fuel economy increased by about 12% thanks to a smaller engine, motor, and battery, while still meeting performance constraints.

DIRECT	Complex
100 function calls	91 (443) function calls
x_{engine} = 1.0 L	x_{engine} = 1.0 L
x_{motor} = 41.96 kW	x_{motor} = 42.18 kW
$x_{battery}$ = 53.67 kW	$x_{battery}$ = 53.49 kW
fuel econ = 48.54 m.p.g.	fuel econ = 48.52 m.p.g.

Table 2: Optimization results for ADVISOR + TDES

Both algorithms were able to converge to approximately the same solution; however, DIRECT was most efficient for this particular problem. Complex was able to converge in fewer function calls; the number of function calls in parentheses refers to the total number required, since the search was re-started whenever it became stuck in an infeasible region. Computational time was decreased substantially by running the simulations on distributed workstations.

5.4 Post optimality studies

Post optimality studies allow more insight into the problem by determining the effect of variations in the design variables and parameters on the design criterion and constraints.

5.4.1 Variable sensitivity

The effect of small changes in the design variables on fuel economy was determined by increasing engine, motor and battery sizes (one at a time) by 1% from its optimal value. As a result, fuel economy decreases by 0.23%, 0.02% and 0.06%, respectively. Decreasing any of these design variables resulted in a infeasible design.

5.4.2 Parameter sensitivity

An important aspect of systems design is the robustness of the solution to changes in design parameters. We studied the effect of the following parameters: drag coefficient, shell mass, battery efficiency, and final drive ratio. The nominal values for these parameters were 0.33, 1000kg, 93%, and 3.84, respectively. We re-optimized with the values 0.30, 900kg, 95%, and 4.00, respectively. Figure 6 shows the effect of parameter variation on the optimal values of fuel economy and engine, motor, and battery sizes.

These results show that optimal engine size is virtually insensitive to variations of the four selected design parameters, whereas optimal motor and battery sizes (and fuel economy) are noticeable sensitive to variations in vehicle mass. A small variation in battery efficiency did not seem to affect the design optimum.

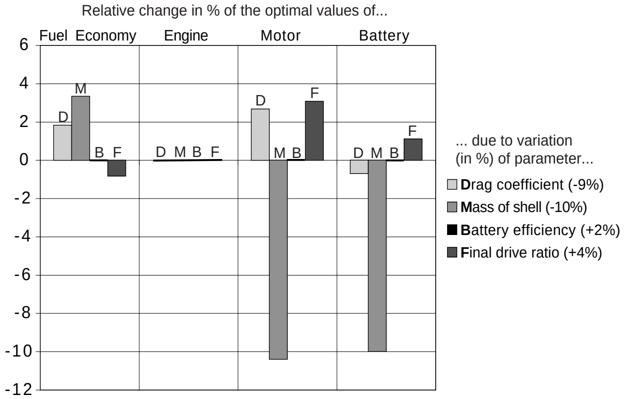


Figure 6: Parameter sensitivity of parallel HEV

6. Discussion

Both MATLAB and the object-oriented, CORBA-based optimization framework enable study of the trade-offs involved in hybrid vehicle design. Next steps in this research effort include: more complicated optimization problem statements to include engine geometry, control strategies, and multiple objectives; refinement of the derivative-free and metamodeling optimization strategies; use of partitioning/coordination strategies for the more complicated models; and, inclusion of additional higher fidelity component models.

Acknowledgment

This research was partially supported by the U.S. Army Automotive Research Center for Modeling and Simulation of Ground Vehicles. This support is gratefully acknowledged. The authors wish to thank D. Assanis, Z. Filipi, and G. Delagrammatikas of the University of Michigan, for their assistance with the TDES model; M. Cuddy and K. Wipke of NREL, for their assistance with ADVISOR; and M. Salman and T. Weber of General Motors, for their advice on hybrid vehicle models.

References

[1] D. Assanis and J. Heywood, “Development and Use of a Computer Simulation of the Turbocompounded Diesel System for Engine Performance and Component Heat Transfer Studies,” *SAE Paper 860329*, 1986.

[2] M.J. Box, “A New Method of Constrained Optimization and a Comparison with Other Methods,” *Imperial Chemical Industries Limited, Central Instrument Research Laboratory*, Bozedown House, Whitchurch Hill, Nr. Reading, Berks, 1965.

[3] M. Cuddy and K. Wipke, “Analysis of the Fuel Economy Benefit of Drivetrain Hybridization”, *SAE Paper 970289*, 1997.

[4] H. Deng, W. Gouveia, and J. Scales, “The CWP Object-Oriented Optimization Library,” *The Leading Edge*, 15(5), pp. 365-369, 1996.

[5] R. Fellini, “Derivative-Free and Global Search Optimization Algorithms in an Object-Oriented Design Framework,” *Master of Science Thesis*, Department of Mechanical Engineering & Applied Mechanics, University of Michigan, Ann Arbor, MI, 1998.

[6] M. Gockenbach and W. Symes, “Hilbert Class Library: A Library of Abstract C++ Classes for Optimization and Inversion,” *Computers and Mathematics with Applications*, 32(6), pp. 1-13, 1996.

[7] D. Jones, C. Perttunen, and B. Stuckman, “Lipschitzian Optimization Without the Lipschitz Constant,” *Journal of Optimization Theory and Applications*, Volume 79, No. 1, October 1993.

[8] J. Meza, “OPT++: An Object-Oriented Class Library for Nonlinear Optimization,” *Sandia Report SAND94-8225*, Albuquerque, NM, 1994.

[9] N. Michelena, H. A. Park, P. Papalambros, and D. Kulkarni, “Hierarchical Overlapping Coordination for Large-Scale Optimization by Decomposition,” to appear in *AIAA Journal*, August, 1998.

[10] S. Nelson II and P. Papalambros, “Sequentially Decomposed Programming,” *AIAA Journal*, 35(7), pp. 1209–1216, July, 1997.

[11] P. Papalambros and D. Wilde, “Principles of Optimal Design: Modeling and Computation,” Cambridge University Press, New York, NY, 1988.

[12] R. Pozo, “TNT: A Template Numerical Toolkit for Linear Algebra,” Mathematical and Computational Sciences Division, National Institute of Standards and Technology, URL: http://math.nist.gov/tnt/, 1997.

[13] M. Sasena, “Optimization of Computer Simulations via Smoothing Splines and Kriging Metamodels,” *Master of Science Thesis*, Department of Mechanical Engineering & Applied Mechanics, University of Michigan, Ann Arbor, MI, 1998.

[14] C. Scheffer, “A Software Framework for Optimal Systems Design,” *Master of Science Thesis*, Department of Mechanical Engineering & Applied Mechanics, University of Michigan, Ann Arbor, MI, 1997.

[15] R. Senger, “Validation of ADVISOR as a Simulation Tool for a Series Hybrid Electric Vehicle Using the Virginia Tech FutureCar Lumina,” *Master of Science Thesis*, Department of Mechanical Engineering, Virginia Polytechnic Institute & State University, Blacksburg, VA, 1997.

[16] J. Snyman., “An Improved Version of the Original Leap-Frog Dynamic Method for Unconstrained Minimization LFOP(b),” Applied Mathematical Modeling, Volume 7, pp. 216-218, 1983.