Optimal torque control of an integrated starter–generator using genetic algorithms

Jeffery McGehee and Hwan-Sik Yoon

Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering published online 30 September 2014
DOI: 10.1177/0954407014548739

Original Article

Proc IMechE Part D:
J Automobile Engineering
1–10

DOI: 10.1177/0954407014548739
pid.sagepub.com

Optimal torque control of an integrated starter–generator using genetic algorithms

Jeffery McGehee[1] and Hwan-Sik Yoon[2]

Abstract

A mild hybrid electric vehicle is a type of hybrid electric vehicle with a simplified hybrid drivetrain mechanism. These mild hybrid electric vehicles use an integrated starter–generator to assist the internal-combustion engine simply rather than to drive the vehicle independently of the internal-combustion engine. In a previous study, a lookup- table-based control scheme was proposed for optimal control of the integrated starter–generator and the internal-combustion engine on a mild hybrid vehicle. In order to investigate further the optimality of the integrated starter–generator control lookup table, a widely used global optimization technique called a genetic algorithm is utilized in this study. The constructed optimal lookup tables are implemented in MATLAB/Simulink together with a mild hybrid vehicle model, which is based on an US Environmental Protection Agency light-duty vehicle model. The simulation results show that the genetic algorithm can construct an optimal lookup table with better performance characteristics than the weighted cost function method.

Keywords

Mild hybrid electric vehicle, integrated starter-generator, lookup-table-based control, genetic algorithm, optimal torque control

Date received: 6 February 2014; accepted: 29 July 2014

Introduction

Many alternative powertrain technologies are being actively researched recently owing to the increasingly stringent emission regulations as well as the rising cost of petroleum-based fuels. One of these technologies is the hybrid electric vehicle (HEV). By combining two different power sources with different capabilities (an internal-combustion engine (ICE) and an electric motor), it is possible to obtain a higher overall vehicle efficiency.[1] This makes an HEV a very attractive method for improving the fuel economy while maintaining a vehicle performance similar to that of a conventional vehicle.

HEVs can be categorized by their 'degree' of hybridization into two basic classes: full hybrids and mild hybrids. A full HEV is able to drive the wheels with the electric motor only, the ICE only, or a combination of both. A mild hybrid electric vehicle (MHEV) is equipped with electric hardware similar to that of a full hybrid, but this hardware is not designed to drive the wheels on its own. Instead, the electric hardware is used to provide an assistive torque which allows the engine to be turned off and restarted quickly while the vehicle is braking or stopped. It is also possible to use the electric drive components to assist the ICE when the ICE requires extra power as demanded by the driver. This paper focuses on this mild hybrid architecture.

Currently, there exist a broad knowledge base in the control of HEVs.[2,3] They can be mainly classified into three different groups: rule-based algorithms, dynamic programming approaches, and equivalent consumption minimization strategies (ECMSs). Rule-based algorithms control predefined energy paths based on event-triggered rules that depend on the driver inputs and the battery's state of charge (SOC).[4,5] Dynamic programming approaches are global optimization methods

[1]Variable, Inc., Chattanooga, Tennessee, USA
[2]Department of Mechanical Engineering, The University of Alabama, Tuscaloosa, Alabama, USA

Corresponding author:
Hwan-Sik Yoon, Department of Mechanical Engineering, The University of Alabama, Box 870276, Tuscaloosa, AL 35487-0276, USA.
Email: hyoon@eng.ua.edu

that guarantee optimal solutions for given cost functions by solving the control problem in a backward recursive way.[6,7] Finally, ECMSs consider the electric power as another type of fuel consumption to produce an optimal control solution in an integrated manner.[8,9] Similarly to dynamic programming, ECMSs are also global optimization methods. Despite the plethora of control methods for HEVs, only a few control algorithms have been developed specifically for MHEVs.

In the early days of mild hybrid vehicle development, the electric components were not used to their full potential with respect to the energy efficiency. In fact, they were only used to allow the ICE to be turned off during braking and stop-and-go traffic situations. Recently, however, researchers have been finding methods to utilize the electric hardware in ways that will allow optimum energy efficiency. This involves the derivation of sophisticated control methods, which require an advanced understanding of how the electric components and the ICE of the vehicle interact. Research into these control methods is a relatively new field, but several control techniques for this problem have already been established. For example, McGehee and Yoon[10] proposed a lookup-table-based control scheme for optimal control of the integrated starter–generator (ISG) and the ICE on a mild hybrid vehicle. The simulation results show that the optimally controlled mild hybrid vehicle has a better fuel efficiency with comparable drivability when compared with a simple intuitive rule-based control strategy. However, the study raised the question that the constructed lookup table may not an optimal table.

In order to investigate optimality of the ISG control lookup table further, a widely used global optimization technique called a genetic algorithm (GA) is utilized in this paper. The constructed lookup tables are implemented in MATLAB/Simulink together with a mild hybrid vehicle model and evaluated against the Federal Test Procedure (FTP) driving cycle. The simulation results show that the GA enables construction of an optimal lookup table for control of the ISG torque. The resulting vehicle performance characteristics are better than those obtained by the previous weighted cost function method. Currently, no previous research is available in the literature regarding the use of GAs for the lookup-table-based control of HEVs or MHEVs. Therefore, this research is intended to lay a foundation for further exploration in this area. As such, the GAs proposed in this research will provide a good starting point for further development of this type of framework.

Mild hybrid electric vehicle model

For performance evaluation as well as controller development of a mild hybrid vehicle, a conventional light-duty vehicle model developed by the US Environmental Protection Agency (EPA) was modified to include the necessary components for the mild hybrid drivetrain.

Table 1. Parameters of the EPA ALPHA model with small car option.

Mass of the vehicle	1191 kg
Frontal area of the vehicle	2.291 m^2
Drag coefficient of the vehicle	0.29
Peak power of the engine	82.3 kW
Peak torque of the engine	145 N m
Base fuel economy	36.16 mile/gal

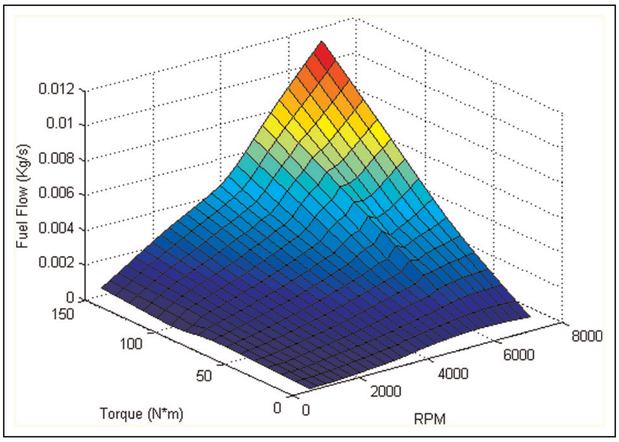

Figure 1. Engine fuel map of the EPA ALPHA model with a small car option.
RPM: r/min.

EPA light-duty vehicle model

The EPA model called ALPHA is a full-vehicle model developed to simulate various types of light-duty vehicle.[11] Among several options for vehicle configurations available in ALPHA, this research is focused on a small passenger car model. The vehicle specifications for the small car are shown in Table 1. This model is a non-hybrid full-vehicle simulator built in MATLAB/Simulink.

The ALPHA model employs various lookup tables for engine control algorithms, which is advantageous in the controller design for two reasons. First, models based on lookup tables can run certain driving-cycle simulations in a relatively short amount of time, which allows rapid development of the control algorithm. Second, the values in the lookup tables can be used in the formulation of the control scheme. As an example, the engine fuel map of the ALPHA model, which is a necessary part of the current control design, is shown in Figure 1.

It should be noted that the ALPHA model is not a high-fidelity physics-based model. This means that the simulation results may not be as accurate as those for other high-fidelity vehicle models. However, for the purposes of controller development and performance improvement evaluation for an MHEV, which is the focus of this research, the accuracy of this model is deemed acceptable.

Mild hybrid electric vehicle model

For the purpose of this study, the baseline ALPHA model was modified to include the necessary electric components of an MHEV based on the belt-driven alternator configuration. These include an electric motor–starter and an enlarged battery pack. Currently, there exist two major types of motor in hybrid drivetrains: induction motors and d.c. motors.[12,13] Because the focus of this research is control system design rather than vehicle modeling, the electric motor or ISG was modeled as a simple permanent-magnet d.c. motor.

For the enlarged battery pack, the lithium-ion battery model developed by Tremblay and Dessaint[14] was adopted as this is the current trend in hybrid vehicles.[15] The charge equation and discharge equation for the lithium-ion battery model are

$$V_{batt} = E_0 - iR - K\frac{C}{it - 0.1C}i^* - K\frac{C}{C - it}it + A\,\mathrm{e}^{-Bit} \tag{1}$$

and

$$V_{batt} = E_0 - iR - K\frac{C}{C - it}(it + i^*) + A\,\mathrm{e}^{-Bit} \tag{2}$$

respectively, where V_{batt} is the battery voltage (V), $E_0 = 3.336$ V is the battery's constant voltage, $K = 0.0076$ V/Ah is the polarization constant, $C = 2.3$ Ah is the battery capacity, i is the battery current (A), $it = \int i\mathrm{d}t$ is the actual battery charge (Ah), i* is the filtered current (A), $R = 0.002\ \Omega$ is the internal resistance, $A = 0.26422$ V is the exponential zone amplitude and $B = 26.5487\ \mathrm{A}^{-1}\ \mathrm{h}^{-1}$ is the exponential zone time constant inverse.

For the mild hybrid model, the desired torque and the rotational speed of the motor are used as the inputs, and the corresponding current and voltage are determined and provided by the battery. For control strategy implementation, it is assumed that the motor can safely add or subtract a maximum torque of 50 N m to or from the system. The original ALPHA model includes an electric motor that adds a torque load to the crankshaft based on the state of the vehicle's electronic components. Therefore, for model integration, it is required only to add these new components to the electric system model in a manner that would allow the torque of the ISG to be added to or subtracted from the crankshaft torque.

Since the ALPHA model is not a high-fidelity physics-based vehicle model, the focus of this research is on the *relative* performance improvements by different control strategies, and not on the *absolute* values of the performance metrics. Therefore, the system parameters used in the simulations are primarily intended to estimate the approximate performance of such vehicles.

Integrated starter–generator control based on the lookup table

Control strategy

When a vehicle is in operation, the powertrain's main function is to propel the vehicle at the velocity desired by the driver. Since the actual driving force to a vehicle system is a torque produced by the crankshaft, matching of the vehicle speed to a desired velocity profile is equivalent to matching the powertrain output torque to a desired torque profile. In the mild hybrid architecture, the torque produced by the powertrain is the sum of the torque from the ISG and the torque from the ICE. In order to ensure that the torque demand from the driver is met, the relationship

$$T_{demand} = T_{ISG} + T_{ICE} \tag{3}$$

should be satisfied. By applying the equality constraint shown in equation (3) to a given torque demand, is is found that there is only one independent variable, which is either T_{ISG} or T_{ICE}. In this study, the ISG torque is chosen to be the design variable that needs to be optimized. Then, the ICE torque is determined by equation (3).

Determining the optimal ISG torque

For optimal control of the ISG torque, the cost function should be a function of the fuel consumption rate and the SOC of the battery according to[10]

$$\text{Cost function} = f(\dot{m}_{fuel}, \mathrm{SOC}) \tag{4}$$

where the mass flow rate $\dot{m}_{fuel}$ of the fuel is determined by the ICE torque and the engine speed (r/min) as

$$\begin{aligned}\dot{m}_{fuel} &= \text{fuel\_rate}(T_{ICE}, \mathrm{r/min})\\ &= \text{fuel\_rate}'(T_{ISG}, \mathrm{r/min})\end{aligned} \tag{5}$$

Note that, in equation (5), the relationship in equation (3) is used to rewrite the fuel flow rate as a function of the ISG torque instead of the ICE torque.

Meanwhile, the battery's SOC is related to the initial SOC minus the time integral of the discharged current, which is a function of the ISG torque. More specifically,

$$\mathrm{SOC} = \frac{C \times \mathrm{SOC}_{init} - \int_0^t i(T_{ISG})\,\mathrm{d}t}{C} \tag{6}$$

Therefore, by substituting equations (5) and (6) into equation (4), it can be seen that the cost function is eventually a function of only one design variable, namelyT_{ISG}; thus,

$$\text{Cost function} = f'(T_{ISG}) \tag{7}$$

All the other variables are either given or determined from the dynamics of the system.

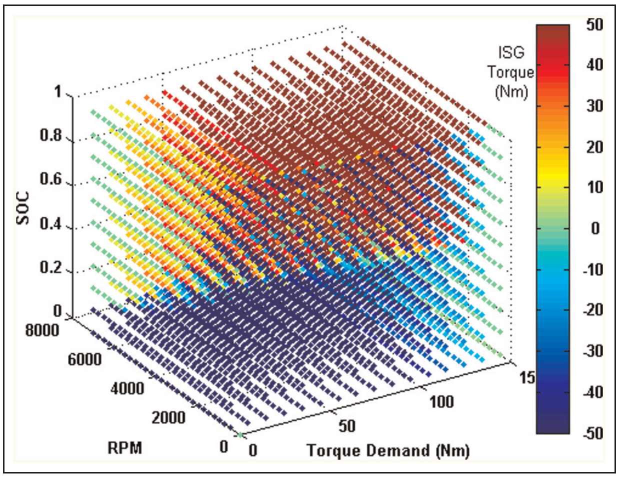

Figure 2. Optimal ISG torque map.[10]
SOC: state of charge; RPM: r/min; ISG: integrated starter–generator.

Performance evaluation

By minimizing the cost function with respect to T_{ISG} at each engine state consisting of three operating variables (T_{demand}, r/min and SOC), a three-dimensional (3D) map or lookup table for the optimal values of T_{ISG} can be obtained from a previous study, as shown in Figure 2.[10] While its performance was better than a simple rule-based strategy, the resulting lookup table was optimum only at each powertrain state and not over a driving cycle. Thus, the purpose of this research is to improve further the lookup table by applying a global optimization algorithm called the GA where the lookup table in Figure 2 is used as the starting point from which GA optimization will begin.

The control method based on the 3D lookup table can be tested on the mild hybrid vehicle model. To compare the performance of the control method with those of other methods, vehicle performance simulations are conducted for the FTP driving cycle. While simulations allow many comparisons to be made, there are three main metrics of interest, as shown in the previous study:[10]

(a) the fuel economy (miles/gal);
(b) the change ΔSOC in the battery's state of charge;
(c) the drivability.

While the fuel economy is one of the most important metrics for any vehicle, maintaining the minimal ΔSOC is important for HEVs because otherwise the control effort would deplete the battery to improve the fuel economy.[10,16] Although the first metric and the second metric can be easily quantified using the simulation result, the third metric, namely drivability, needs to be defined. This can be achieved by comparing the actual vehicle speed for the FTP simulations with the desired speed of the FTP cycle. Then, it is assumed that any deviation in the actual vehicle speed greater than 2% from the desired speed is considered undesirable. Finally, the ratio of the time that the vehicle spends outside this 2% error range to the total drive time will be used as the drivability metric VE according to

$$\mathrm{VE} = \frac{\sum \mathrm{time}|_{error \geqslant 2\%}}{\mathrm{totaltime}} \tag{8}$$

Genetic algorithm for lookup table optimization

The goal of GA-based optimization for MHEV control is to create an optimal 3D lookup table for the ISG torque values. The optimal values can be determined by minimizing the cost function given in equation (7) at any given powertrain state defined by the specific engine speed, the driver's torque demand, and the battery's SOC. In general, GAs consist of the following three steps.[17]

1. *Step 1*. Form the initial population.
2. *Step 2*. Perform a fitness evaluation.
3. *Step 3*. Create a new population.

These steps are executed iteratively with the children formed in Step 3 being used as the parents in the next iteration. Because the parents of each child are selected according to their fitness, the population will evolve to become 'fitter' over time. Based on this general algorithm, the population of parents is chosen to be an array of 3D lookup tables similar to that shown in Figure 2.

First genetic algorithm

The algorithm presented in this section is the first attempt to apply the GA to optimize the 3D lookup table for ISG control.

Form the initial population. The first part of any GA is to form the initial population. Because of computational constraints, a population size of 20 members was chosen. It is considered that this population size is sufficiently large to test the validity of GA-based optimization for MHEV control. For the 'seed' lookup table, the table produced by the weighted cost function in the previous study was employed.[10] However, the seed table was not produced using weights that would yield zero ΔSOC. This provides an opportunity to analyze the GA's ability to reduce ΔSOC. Each consecutive member $\mathbf{M}_i$ of the population is formed by multiplying the original lookup table $\mathbf{T}^{seed}$ by $\mathbf{T}^{rand}$, which is a 3D array of equal dimensions with randomly generated elements between -1 and 1 according to

$$\mathbf{M}_i = \mathbf{T}^{seed} \circ \mathbf{M}_i^{rand}|_{-1}^{1}, \quad i = 2, \ldots, 20 \tag{9}$$

where $\circ$ represents the entrywise matrix product called the Hadamard product. Equation (9) shows how the initial population is formed for the first iteration of the GA. Each consequent population is formed from Step 2 and Step 3 of the previous iteration.

Perform a fitness evaluation. After the population is formed, each member undergoes a fitness evaluation. For the current application, the evaluation is the vehicle simulation performance over the FTP driving cycle. As previously explained, the performance metrics are chosen to be the fuel economy, the change in the battery's SOC, and the drivability, which are denoted by MPG, ΔSOC and VE respectively. Because there are three performance metrics, a weighted linear combination of the three metrics is calculated and assigned to each member as a single score. However, it is determined that ΔSOC greater than 20%, or VE greater than 8%, receives a score of 0 as an indication of inappropriateness. Thus, the score function is defined as

$$\text{Score} = \begin{cases} \text{if } |\Delta\text{SOC}| > 0.2 \text{ or VE} > 0.08, 0 \\ \text{otherwise, MPG} - 10|\Delta\text{SOC}| - \text{VE} \end{cases} \tag{10}$$

Note that the weighting factors are arbitrarily determined by trial and error and do not necessarily represent optimal values.

Create a new population. The final step in the algorithm is to sort the members based on the score obtained by the scoring function. Using a technique called the elite pass-through method, the two highest-scoring lookup tables pass through to the next generation without any modification. Then, a probability factor is assigned to each member on the basis of its score. This probability factor determines the statistical likelihood that the particular member will reproduce. For the algorithm, the weights are assigned as shown in Table 2.

To form the remainder of the next generation, a 'father' and 'mother' are chosen from the current population using a random function based on the probability factor assigned to the members. This means that every member has the chance to reproduce, but the higher-scored members are more likely to be selected. When two parents are chosen, a chance of 0.1% was used for the child to be a mutation. This produces a new table with completely random values between −50 N m and 50 N m. If the mutation does not occur, a child is produced by averaging the values from the mother and father table. This is repeated until a new population of 20 members is formed. The new population then becomes the initial population for the next iteration. The algorithm iterates until the performance of the top performer converges and no more improvement is observed.

Table 2. Reproduction probability based on the fitness rank in the percentile.

Percentile	Probability factor
80–100	0.7
60–80	0.6
40–60	0.5
20–40	0.4
0–20	0.3

Second genetic algorithm

For the second algorithm, several changes were made to the first algorithm based on a literature review and observations of the results from the first algorithm. Changes were made to the initial population, the fitness function, and the reproduction and mutation strategies.

Form the initial population. The first member of the initial population was chosen to be the same lookup table obtained from the weighted cost function. In order to introduce more randomness into the initial population, however, the remainder of the population is formed with completely random values within the ISG torque range from −50 N m to 50 N m. Each remaining member $\mathbf{M}_i$ of the population is thus formed as

$$\mathbf{M}_i = \mathbf{T}_i^{rand}|_{-50}^{50}, \quad i = 2, \ldots, 20 \tag{11}$$

Equation (11) shows how the initial population is formed for the first population of the second GA. Each consequent population is formed from Step 3 of the previous iteration.

Perform a fitness evaluation. Although the weighted linear combination of the performance metrics used in the first algorithm is still acceptable, a slight modification is made for better GA performance. The scoring function is chosen as

$$\text{Score} = \begin{cases} \text{if } \Delta\text{SOC} > 0, \text{MPG} - 5|\Delta\text{SOC}| - \text{VE} \\ \text{if } \Delta\text{SOC} < 0, \text{MPG} - 50|\Delta\text{SOC}| - \text{VE} \end{cases} \tag{12}$$

This scoring function allows each member to receive a score relative to its performance rather than assigning a score of zero to any member that performs 'poorly' as is done in the first algorithm.

Create a new population. Like the first algorithm, the final step starts with sorting the members based on the fitness score. Two top-scored members are passed through to the next generation, just as in the first algorithm. The probability factors also remain the same. To form the remainder of the next generation, a father and a mother are chosen from the current population using a random function. This is where the similarities between the first algorithm and the second algorithm end. In the second algorithm, the lookup tables are partitioned into multiple 'cells' to mix the mother's traits and the father's traits instead of simply averaging the mother and the father as a whole as in the first algorithm. Cells are used in GAs to mimic an organism.[18]

In this algorithm, the parents are divided into eight equal-sized cells, and a child is formed by combining the cells of each parent stochastically. This procedure is described by

$$\underset{j=1,\dots,8}{\mathrm{cell}_j^{child}} = \begin{cases} \mathrm{cell}_j^{mother} & \\ \mathrm{cell}_j^{father} & \text{with equal probabilities} \\ \mathrm{average}(\mathrm{cell}_j^{mother}, \mathrm{cell}_j^{father}) & \end{cases} \tag{13}$$

This means that, with equal probabilities, each cell of the child will be the same respective cell from the father or the mother, or it will be an average of the mother's corresponding cells and the father's corresponding cells. This is a slightly modified form of the uniform crossover method.[18]

The method for introducing mutation into the population is also modified for the second algorithm. Mutations are made to occur at parts *within* the child instead of creating an entirely new child. Specifically, for each child, a probability of 1% is used for each value from the lookup table to be swapped for a random value between −50 N m and 50 N m. This essentially guarantees that each child is subject to some degree of mutation since the size of the lookup table is $27 \times 16 \times 11$.

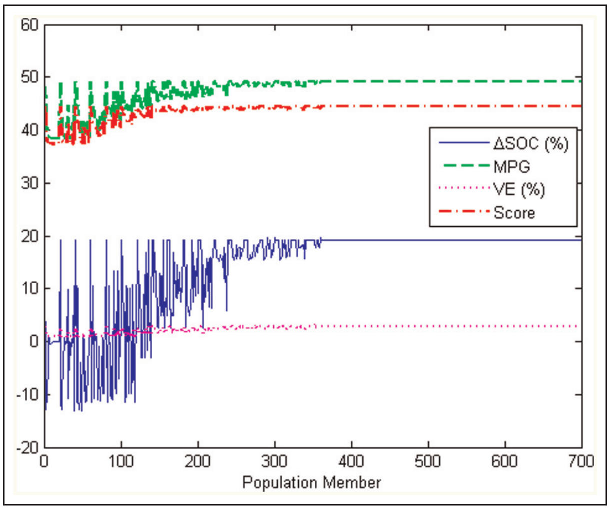

Figure 3. Evolution of the population performance in the first GA.
ΔSOC: change in the state of charge of the battery; MPG: fuel economy (miles/gal); VE: drivability.

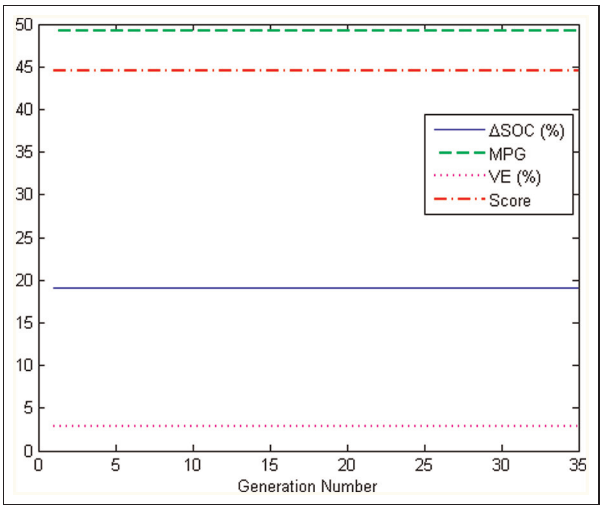

Figure 4. Evolution of the top performers in the first GA.
ΔSOC: change in the state of charge of the battery; MPG: fuel economy (miles/gal); VE: drivability.

Simulation results

First genetic algorithm

The results of the first GA were not satisfactory. Upon the start of the optimization process, the GA converged quickly to the original lookup table used as the seed. While this result was undesired, two observations could be made from it.

1. The lookup table based on the weighted cost function is likely to be a local optimum in the optimization space.
2. The search space created by the GA was not sufficiently large to explore beyond the local maximum, causing very quick convergence.

It was possible to extract some information on the evolution of the population from the data that were collected from the first algorithm. Figure 3 shows the evolution of the population's performance in the three selected metrics and the overall score. It can be seen that, although the population has a fairly wide initial variance, it quickly converges to the values that were produced by the seed lookup table. Also, Figure 4 shows that the top performer does not change in each generation, which means that the original seed lookup table remains as the top performer throughout the entire optimization process. This indicates that the initial population and their children were not capable of moving further out of the local optimum and failed to reach another optimum in the optimization space.

The idea of having a 'dominant' member in the initial population is desirable because it can provide the GA with a good starting point in the optimization procedure. However, when every member of the population is derived in a similar way or all members are derived from the single dominant member, the population lacks diversity. This lack of diversity greatly reduces the potential search area of the GA within the optimization space.

Creating a fitness function that is a linear combination of various performance metrics is a common practice in GAs.[17] However, assigning a score of zero to any member outside a specific range on a given metric

Table 3. Comparison of the simulation results obtained by different methods.

Performance metric	Value for the following		
	Second GA	First GA	Weighted cost function[10]
MPG	48.28 miles/gal	49.21 miles/gal	47.40 miles/gal
ΔSOC	0.09%	19.14%	0.2%
VE	1.02%	2.84%	6.54%

GA: genetic algorithm; MPG: fuel economy; ΔSOC: change in the state of charge of the battery; VE: drivability.

will cause the stochastic method used in parent selection to be ineffective. In other words, if a member is classified only as 'bad', it is impossible to know how bad this member really is. If multiple members are classified as bad, they are viewed the same by the weighted parent selection algorithm. This can cause the diversity of the population to be eliminated rapidly.

Averaging parents is not a proper technique in GAs. Children should be produced using a crossover method in which cells from the mother and the father are assembled to form a new population member.[18] When averaging of the parents is coupled with the aforementioned initial population and fitness function, it can be seen that convergence will occur quickly, and the search area will be localized. Another weakness of the mating algorithm was the implementation of the mutation phenomenon. Mutation should occur within certain cells of a child rather than causing the creation of an entire mutant child.[18]

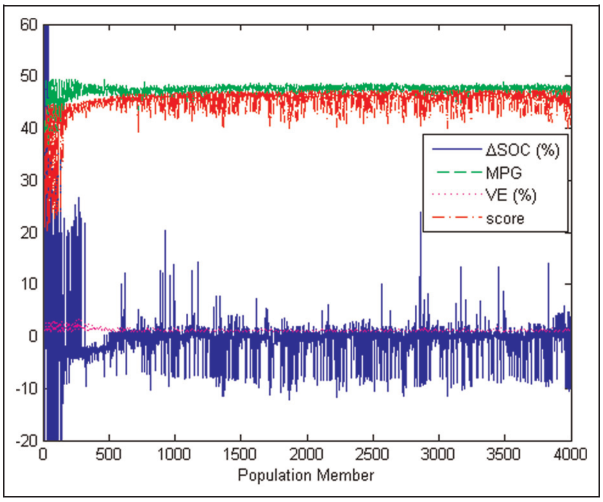

Figure 5. Evolution of the population performance in the second GA.
ΔSOC: change in the state of charge of the battery; MPG: fuel economy (miles/gal); VE: drivability.

Results of second genetic algorithm

Unlike the first GA, the second GA was able to produce a lookup table which is very different from that created by the weighted cost function. The simulation results obtained for the FTP cycle using the optimized lookup table from the second GA are compared with those obtained from the seed table from the first GA and the weighted cost function from the previous research in Table 3.

The evolution of the population for the second GA is shown in Figure 5. When compared with Figure 3 of the first algorithm, it can be seen that, even after 200 generations, the second GA is still exploring the optimization space while the first GA stopped searching after approximately 20 generations. This is due to the change in the mating algorithm and the mutation algorithm, which nearly guarantee that a child will never be the same as either of its parents. Also, it can be noted that the initial population of this algorithm is much more diverse than that of the first algorithm, which allows a much larger initial search area within the optimization space.

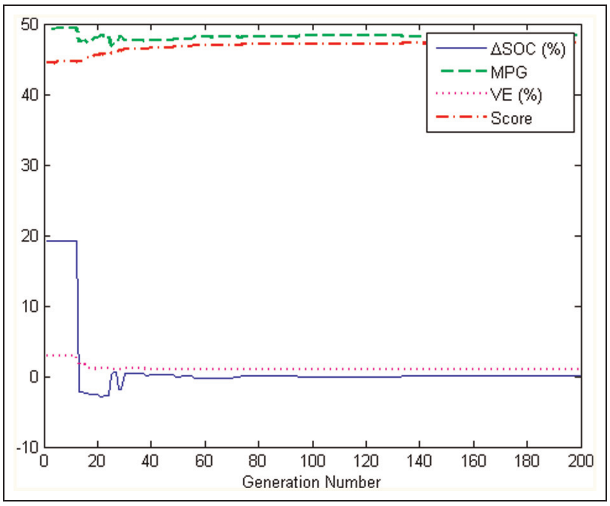

Figure 6. Evolution of the top performers in the second GA.
ΔSOC: change in the state of charge of the battery; MPG: fuel economy (miles/gal); VE: drivability.

Figure 6 shows a plot of the evolution of the top performer for the second GA. Unlike the case of the first GA, the performance does change from generation to generation, implying that the search space has expanded past the local optimum represented by the seed table. It can also be seen that most of the performance gains were made in the first 40 generations although the performance improvement does not end completely until near generation 200. This may seem ideal but, as seen with the first GA, fast convergence in a complex optimization space could mean that the algorithm is trapped in a local optimum.

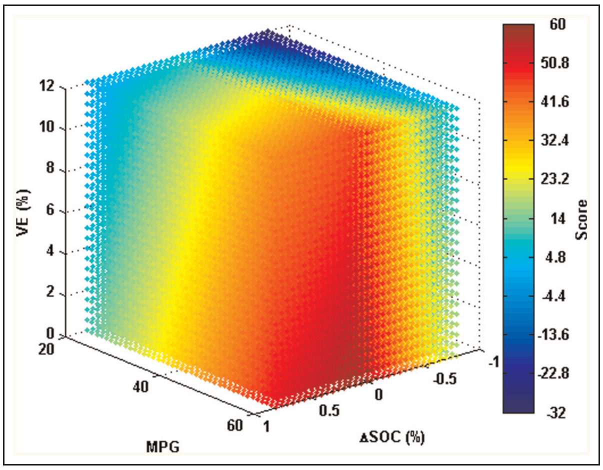

Figure 7. Fitness score function for the second GA.
VE: drivability; MPG: fuel economy (miles/gal); ΔSOC: change in the state of charge of the battery.

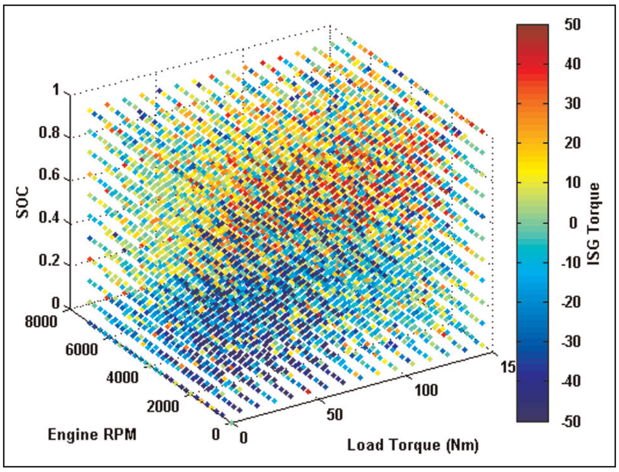

Figure 8. Optimized ISG torque map for the second GA.
SOC: state of charge; RPM: r/min; ISG: integrated starter–generator.

Using the scoring function, the top performer in generation 200 of the second GA received a score of 47.5. In order to gain an understanding of why this score is reached, the scoring function in equation (12) is plotted for ranges of ΔSOC, MPG and VE in Figure 7. In the plot, colored points in the 3D space represent scores as shown in the color bar. Because the conditionality of equation (12) is based on the sign of ΔSOC, the function is not symmetric with respect to the plane of ΔSOC = 0. The score of 47.5 achieved by the second GA is in the high range of this graph. However, it is shown that the score gradient in the MPG–ΔSOC plane is quite steep as |ΔSOC| increases, especially for negative values of ΔSOC. This means that the algorithm is relatively unwilling to trade an increase in ΔSOC for an increase in the MPG. However, the score gradient in the MPG–VE plane is less steep, indicating that the algorithm is more willing to trade an increase in the speed error for an increase in the MPG. Both of these characteristics are shown in Table 3.

Another comparison was made between the seed lookup table shown in Figure 2 and the algorithm's top-performing table shown in Figure 8. While the seed map is fairly 'predictable' in its distribution of ISG torque values, the algorithm's top-performing table has many sections that appear to be random. This can be explained in terms of the vehicle's operating point in the design space, as shown in Figure 9. For the FTP cycle, many of the regions in the lookup table are not occupied by the vehicle. Therefore, the GA has no 'reason' to change any of these values that are not utilized by the vehicle during the FTP driving cycle. Therefore, it can be said that the GA 'trains' an optimal lookup table for a particular driving cycle, which was the FTP driving cycle in this case.

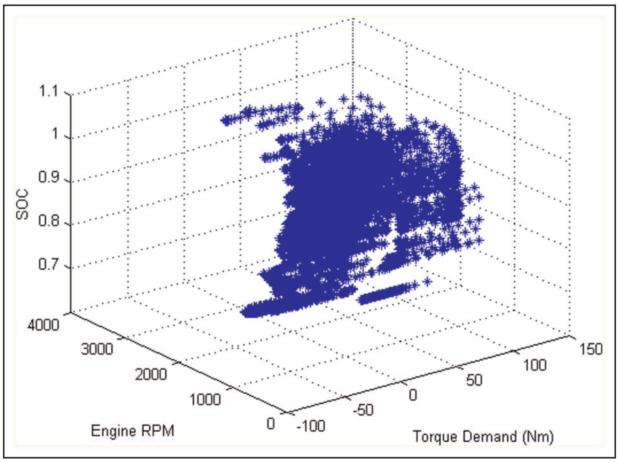

Figure 9. Vehicle's operating points for the FTP cycle simulations.
SOC: state of charge; RPM: r/min.

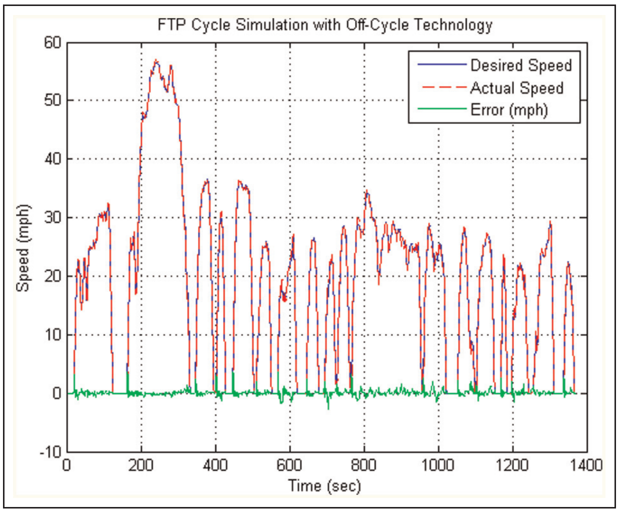

Figure 10. Profile of the FTP velocity of the top performer for the second GA.
mph: miles/h.

Simulation results of the top performer for the second GA

The profile of the vehicle velocity for the FTP cycle is shown in Figure 10 and the profiles of the ISG torque

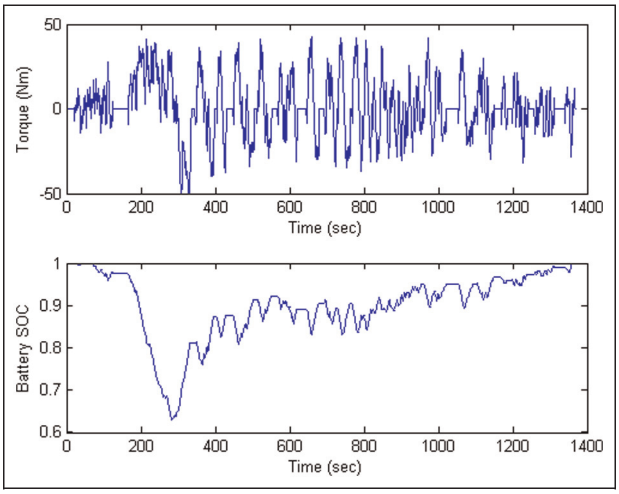

Figure 11. Profiles of the ISG torque and SOC of the top performer of the second GA.
SOC: state of charge.

and SOC are shown in Figure 11. In Figure 10, it can be seen how well the actual vehicle speed follows the desired vehicle speed with minimal error. It can also be seen that the lookup table created by the second GA allows the SOC to drop to its lowest point during the longest portion of acceleration in the driving cycle (between 150 s and 250 s). This is an effect which occurs because the GA is trained on the FTP cycle alone. The battery usage seems to be quite optimal for the FTP cycle but, if the battery's SOC was diminished in this way in an unknown driving cycle, it could result in significant battery depletion. Therefore, it will be necessary to optimize the lookup table for multiple driving cycles in order to account for such adverse situations.

Conclusion and future work

In this paper, two GAs were developed and applied to optimize a lookup table for control of the ISG for an MHEV. Although the first algorithm could not improve the lookup table obtained by a weighted cost function from a previous study, the second algorithm was able to show a significant capability to optimize further the lookup table for MHEV control. When the MHEV model was simulated using the optimized lookup table for the FTP driving cycle, the resulting vehicle performance characteristics turned out to be better than those obtained by the previous weighted cost function method. Therefore, it can be concluded that GAs can be applied to optimize lookup tables for control of MHEVs or HEVs for improved fuel economy and drivability. From the perspective of application, the lookup-table-based MHEV control method is a simpler approach which can be easily implemented by the engine control unit of a production vehicle compared with other types of control method.

However, similar to the dynamic programming approaches, the lookup table optimized by the GA is dependent on a particular driving cycle. If the driving cycle is not known a priori, optimality is not guaranteed and it may perform even more poorly than a simple rule-based control algorithm. For example, when the battery's SOC was examined during the FTP cycle, it was observed that the SOC dropped significantly at a certain point, which could result in significant battery depletion. Therefore, as future work, it will be necessary to optimize the lookup table for multiple driving cycles in order to account for such adverse situations.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Declaration of conflict of interest

The authors declare that there is no conflict of interest.

References

1. Ehsani M and Gao Y. Hybrid drivetrains. *Handbook of automotive power electronics and motor drives*. Boca Raton, Florida: CRC Press, 2005, pp. 37–54.
2. Wirasingha SG and Emadi A. Classification and review of control strategies for plug-in hybrid electric vehicles. *IEEE Trans Veh Technol* 2011; 60(1): 111–122.
3. Overington S and Rajakaruna S. Review of PHEV and HEV operation and control research for future direction. In: *2012 3rd IEEE international symposium on power electronics for distributed generation systems*, Aalborg, Denmark, 25–28 June 2012, pp. 385–392. New York: IEEE.
4. Jalil N, Kheir N and Salman M. A rule-based energy management strategy for a series hybrid vehicle. In: *American control conference*, Albuquerque, New Mexico, USA, 4–6 June 1997, Vol 1, pp. 689–693. New York: IEEE.
5. Baumann B, Washington G, Glenn B and Rizzoni G. Mechatronics design and control of hybrid electric vehicles. *IEEE/ASME Trans Mechatronics* 2000; 5: 58–72.
6. Brahma A, Guezennec Y and Rizzoni G. Optimal energy management in series hybrid electric vehicles. In: *American control conference*, Chicago, Illinois, USA, 28–30 June 2000, Vol 1, pp. 60–64. New York: IEEE.
7. Gong Q and Li Y. Trip based power management of plug-in hybrid electric vehicle with two-scale dynamic programming. In: *IEEE vehicle power propulsion conference*, Arlington, Texas, USA, 9–12 September 2007, pp. 12–19. New York: IEEE.
8. Paganelli G, Ercole G, Brahma A et al. General supervisory control policy for the optimization of charge-sustaining hybrid electric vehicles. *JSAE Rev* 2001; 22(4): 511–518.
9. Pisu P and Rizzoni G. A comparative study of supervisory control strategies for hybrid electric vehicles. *IEEE Trans Control System Technol* 2007; 15: 506–518.
10. McGehee J and Yoon H. An optimal powertrain control strategy for a mild hybrid electric vehicle. SAE paper 2013-01-0482, 2013.
11. Advanced light-duty powertrain and hybrid analysis (ALPHA) tool user's guide for off-cycle credit evaluation.

Report EPA-420-B-12-051, Assessment and Standards Division, Office of Transportation and Air Quality, US Environmental Protection Agency, http://www.epa.gov/oms/climate/documents/420b12051.pdf (Accessed: 29/7/2014).
12. Walters JE, Krefta RJ, Gallegos-Lopez G and Fattic GT. Technology considerations for belt alternator starter systems. SAE paper 2004-01-0566, 2004.
13. Emadi A. *Handbook of automotive power electronics and motor drives*. Boca Raton, Florida: CRC Press, 2005.
14. Tremblay O and Dessaint L. Experimental validation of a battery dynamic model for EV applications. In: *24th international battery, hybrid and fuel cell electric vehicle symposium and exposition*, Stavanger, Norway, 13–16 May 2009; World Electric Veh J 2009; 3: 1–10. Brussels: *AVERE*.
15. Marangoni G and Beghi A. *Battery management system for Li-ion batteries in hybrid electric vehicles*. Master's Thesis, University of Padova, Padova, Italy, 2010.
16. Lammers MFA. *Impact of mild-hybrid functionality on fuel economy and battery lifetime*. Master's Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2006.
17. Mitchell M. *An introduction to genetic algorithms*. Cambridge, Massachusetts: MIT Press, 1999.
18. Sastry K, Goldberg D and Kendall G. Genetic algorithms. In: *Search methodologies: introductory tutorials in optimization and decision support techniques*. New York: Springer, pp. 97–125.