2018-01-1027 Published 03 Apr 2018

Mission-based Design Space Exploration for Powertrain Electrification of Series Plugin Hybrid Electric Delivery Truck

Daniel Jung and Qadeer Ahmed Ohio State University

Xieyuan Zhang Tsinghua University

Giorgio Rizzoni Ohio State University

Citation: Jung, D., Ahmed, Q., Zhang, X., and Rizzoni, G., "Mission-based Design Space Exploration for Powertrain Electrification of Series Plugin Hybrid Electric Delivery Truck," SAE Technical Paper 2018-01-1027, 2018, doi:10.4271/2018-01-1027.

Abstract

Hybrid electric vehicles (HEV) are essential for reducing fuel consumption and emissions. However, when analyzing different segments of the transportation industry, for example, public transportation or different sizes of delivery trucks and how the HEV are used, it is clear that one powertrain may not be optimal in all situations. Choosing a hybrid powertrain architecture and proper component sizes for different applications is an important task to find the optimal trade-off between fuel economy, drivability, and vehicle cost. However, exploring and evaluating all possible architectures and component sizes is a time-consuming task. A search algorithm, using Gaussian Processes, is proposed that simultaneously explores multiple architecture options, to identify the Pareto-optimal solutions. The search algorithm is designed to carefully select the candidate in each iteration which is most likely to be Pareto-optimal, based on the results from previous candidates, to reduce computational time. The powertrain of a medium-sized series plugin hybrid electric delivery truck with a range extender is optimized for different driving missions. Three different powertrain architectures are included in the design space exploration and the fuel economy is evaluated using a simulation model of the powertrain and Dynamic Programming. Results from the analysis show which ranges of powertrain component sizes are recommended for the different types of driving scenarios.

Introduction

Reducing fuel consumption and emissions is an important problem in the automotive industry and HEV are essential to solve this. HEV with various electrified powertrain architectures have been proposed, for example, different types of series- and parallel hybrids, which have their own advantages. Optimizing the design of the electrified powertrain is often performed based on multiple performance objectives, for example fuel consumption, component cost, battery size, and drivability. Many of the objectives are contradictory meaning that one candidate is never optimal with respect to all objectives, but instead there are different trade-offs between different powertrain candidates [1].

When comparing different segments of the transportation industry, for example, public transports or different sizes of trucks for parcel delivery, it is clear that one powertrain may not be optimal in all situations. Selecting electrified powertrain architecture and component sizes is an important task to find the optimal trade-off between fuel consumption and other performance objectives, for each specific application. The problem of finding the optimal powertrain configuration is referred to as design space exploration which is often formulated as a Multi-Objective Optimization Problem (MOOP). Design space exploration is, in general, a non-convex optimization problem where the design space grows exponentially with the number of components that are optimized.

Fuel consumption can be evaluated by simulating or optimizing the specific powertrain for a given set of driving scenarios representing realistic driving missions. Therefore, evaluating the performance of each powertrain can be time-consuming. In these situations, selecting a suitable search algorithm is important. A design space exploration algorithm is proposed which uses Gaussian Processes to select the powertrain candidate, in each iteration, that is most likely to be Pareto-optimal.

In this work, the powertrain design of a medium-sized delivery truck is considered, see Figure 1. The truck has a series hybrid powertrain with an internal combustion engine as range extender where different powertrain architectures are explored. The powertrain is optimized for different driving cycles to analyze how the driving mission affects the selected powertrain. A simulation model of the powertrain is developed and Dynamic Programming is used to compute the optimal control strategies and fuel consumption for each architecture.

FIGURE 1 The type of medium-sized delivery truck that is considered in the design space exploration.

Design Space Exploration of Hybrid Electric Vehicles

Research on design space exploration for HEV can be traced back to the 1990s, when the Good-Design Seeker, an architecture for exploring large design space of HEV using exhaustive search and dominance filtering, was proposed in [2]. During the past two decades of development, three main approaches are adopted to solve the design optimization problem: (1) exhaustive search, [3, 4, 5]; (2) derivative-free algorithms including genetic algorithm [6], particle swarm optimization [7], DIRECT [8], and the Nelder-Mead Simplex algorithm, [9]; (3) gradient-based algorithms including convex optimization [10] and sequential quadratic programming [11].

While the optimization on HEV design level can be solved separately, there is a trend of considering the HEV powertrain optimal control as part of the design space as well. One typical example is to consider the sizing and control variables as a whole and optimize this bi-level system design using an iterative convex optimization and dynamic programming procedure, [12]. Meanwhile, the cost function, referring to the objective of search, has been extended from simply maximizing fuel economy to a comprehensive evaluation of emission, hybridization cost, operation cost (both gasoline and electricity), efficiency and battery life-cycle, thus resulting in different types of output: an optimized solution or a series of Pareto-fronts.

However, the vast majority of previous design space exploration research is based on a fixed HEV topology, which limited the scope of design optimization to sizes of certain components and the final drive ratio. Among those who take into consideration the generation and selection of different topologies, few explore the variety of series hybrid topologies [13] compared with the power-split architecture with planetary gear sets [14, 15]. In this article, a general framework for HEV design space exploration, which is using the "topology-design-control" system as the design space while specifying in detail the elements of input, constraint, cost function and output, is proposed based on a thorough overview of related research.

Problem Description

The main objective in this paper is to investigate the impact of specified driving missions on how to design the electrified powertrain of a Hybrid Electric Delivery Truck. The electrified powertrain is fixed to be a series plug-in hybrid with an internal combustion engine as a range extender. However, there are multiple architecture options and component sizes that can be used within this framework. The powertrain is here optimized with respect to fuel economy, vehicle weight, and battery size.

The complexity of implementing control algorithms for different architectures is not considered in this case study. Fuel economy is evaluated using Dynamic Programming and a simulation model of each powertrain architecture. Since different candidates will have various trade-offs of the objective functions, the goal is to explore the Pareto-front to understand the optimal trade-offs between the objective functions and to see which powertrain architectures and component sizes that are relevant in different situations.

The different driving missions are represented by driving cycles which describe the general characteristics for each driving mission, for example, average and maximum speed and the frequency of starts and stops. One powertrain can be optimized for multiple driving cycles simultaneously, but in this study only one is considered at the time.

To solve the design space exploration problem, a search algorithm is necessary that can choose candidates from different architectures based on which is more likely to be Pareto-optimal. Design space exploration is, in general, a non-linear problem meaning that an exhaustive search approach is necessary to guarantee that the global optimum is found. However, since the search space grows exponentially with the number of components, an exhaustive search is only feasible if the search space is not too large.

Evaluating the objective functions of each powertrain candidate is considered computationally costly, due to heavy simulations and optimizations of the powertrain fuel consumption. Therefore, the search algorithm should carefully select which candidates to evaluate to reduce the search time. The candidate selection procedure is allowed to take time if it helps to reduce the overall time of the design space exploration. A heuristic search algorithm is proposed which is able to explore relevant parts of the design space from multiple architectures while reducing the number of evaluated candidates.

Powertrain Architectures

Three different candidate powertrain architectures are considered in this case study. Schematics for the three powertrains are shown in Figures 2-4. In all three architectures, the gen-set consists of an internal combustion engine driving a generator to generate electric power that is used to recharge the battery or powering the motor. The red arrows in the schematics represent electric power flows and blue arrows mechanical power flows. The gen-set can provide electric power to both the battery and the electric motors and is assumed to, when turned on, provide electric power by working at constant optimal speed and torque.

The first candidate powertrain architecture has a more conventional drivetrain with a gearbox and torque converter

FIGURE 2 A schematic of an electrified series powertrain architecture with gearbox and torque converter.

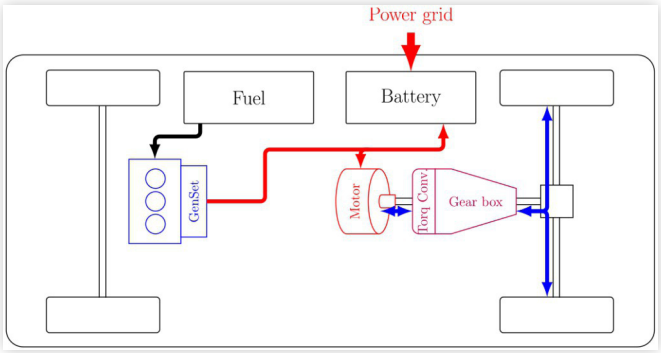


FIGURE 3 A schematic of an electrified series powertrain architecture with direct-drive.

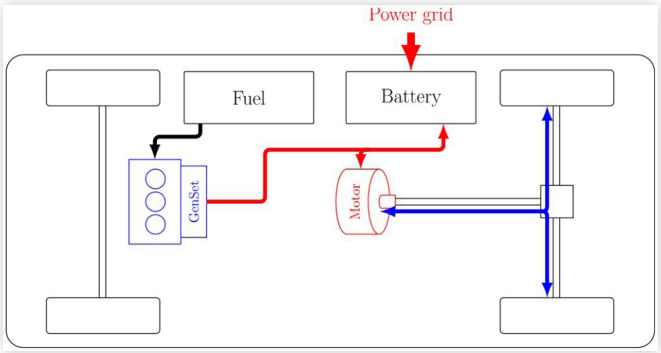
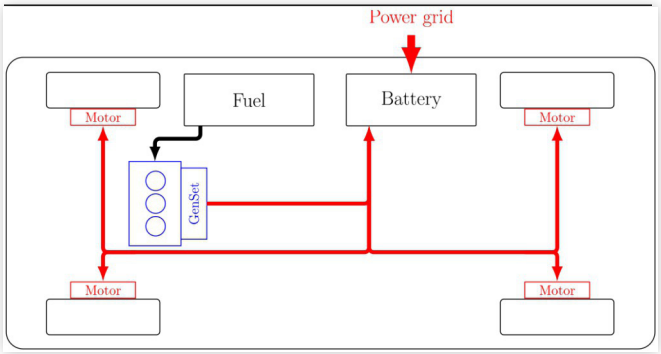


FIGURE 4 A schematic of an electrified series powertrain architecture with four in-wheel motors.



as shown in Figure 2. This architecture has the largest degree-of-freedom in the optimization, compared to the other alternatives, where gear shifting and mode controlling of the torque converter are part of the optimization. However, the additional components of the drivetrain add up to the vehicle weight.

The second architecture in Figure 3 is a direct-drive, and the third architecture in Figure 4 has no mechanical drivetrain but instead four in-wheel motors to drive the vehicle. In the third architecture, the power request is, in the analysis, distributed equally among the four motors.

Powertrain Modeling

All powertrain schematics described in the previous section are based on the same model structure. The difference is mainly based on which powertrain components are connected to each other. All component models are presented here.

Each powertrain is evaluated for a given driving cycle and a backward simulation model is implemented for evaluating fuel economy using Dynamic Programming [16]. The vehicle model is assumed to perfectly follow the predefined velocity profile $v(t)$. The required traction force provided by the electric motor at the wheel is computed as

$$F_{trac}(t) = m\dot{v}(t) + F_a(t) + F_r(t) + F_m(t)$$

whereas m is vehicle mass and $v(t)$ is vehicle speed as function of time t. The friction terms are given by

$$F_a(t) = 0.5\rho_a c_d A_f v(t)^2,$$

$$F_r(t) = c_r mg\cos(\theta(t)), \qquad v(t) > 0$$

$$F_m(t) = mg\sin(\theta(t)),$$

where $\theta(t)$ is road grade, g is the gravity constant, ρ_a is the ambient air density, c_d is the air drag coefficient, A_f is the vehicle frontal area, and c_r is the friction coefficient. The vehicle parameters ρ_a, c_d, and A_f, and the friction coefficient c_r, are equal for all architectures.

The wheel speed and torque are computed as

$$T_{wh}(t) = rF_{trac}(t), \quad v(t) > 0$$

$$\omega_{\text{th}}(t) = v(t)/r$$

where r is the wheel radius.

The final differential and the gearbox are modeled as

$$T_{in}(t) = \begin{cases} \dfrac{T_{out}(t)\eta}{\gamma} \text{ if } T_{out}(t) \le 0 \\ \dfrac{T_{out}(t)}{\eta\gamma} \text{ otherwise} \end{cases}$$

$$\omega_{in} = \omega_{out}\gamma$$

where γ is the gear ratio, η is the efficiency, the subscripts *in* refers to torque and speed at the motor size of the component and *out* refers to the wheel side. In the gearbox model, the efficiency η is assumed constant for all gears but the gear ratio γ is a function of selected gear. A summary of the vehicle parameters is shown in Table 1.

The torque converter is modeled as a static function using the model structure proposed in [17]. The torque amplification and fluid coupling modes are modeled using second-order polynomials in the form

$$T_p(t) = a_0\omega_p^2(t) + a_1\omega_p(t)\omega_t(t) + a_2\omega_t^2(t)$$

$$T_t(t) = b_0\omega_p^2(t) + b_1\omega_p(t)\omega_t(t) + b_2\omega_t^2(t)$$

TABLE 1 Some vehicle model parameters.

Parameter	Value
ρ_a	1.13
c_d	0.75
A_f	5.70
c_r	0.0070
η (final differential)	0.934
η (gearbox)	0.92
γ (gearbox, ratios)	(3.45, 2.25, 1.41, 1.00)

with different parameters a_0, a_1, a_2 for each mode, where subscript t refers to the turbine side and p refers to the pump side. The torque converter lockup mode is modeled as

$$T_p(t) = T_t(t)$$

$$\omega_p(t) = \omega_t(t)$$

and activating the torque converter lockup mode is controlled by an external input.

The electric power request of the electric motor is modeled using a static map to describe motor efficiency as function of motor torque and speed as

$$P_{EM}(t) = T_{EM}(t)\omega_{EM}(t)\eta_{EM}\left(T_{EM}(t),\omega_{EM}(t)\right)$$

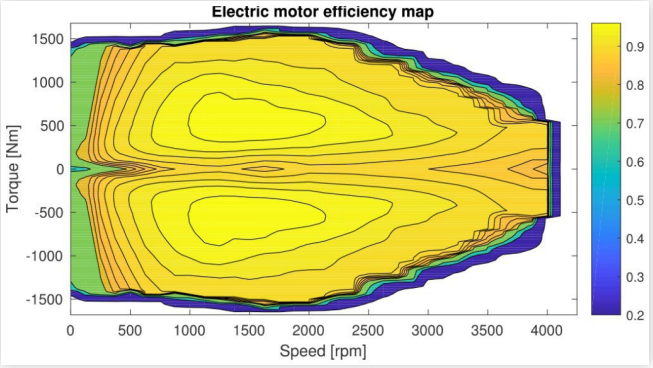
When P_{EM} is negative, i.e. when T_{EM} is negative, the motor works as a generator. A reference efficiency map $\eta_{EM}(T_{EM},\omega_{EM})$ is shown in Figure 5. The map is then scaled to represent different motor sizes.

The required electric power of the motor P_{EM} can be provided either from the battery P_{batt} or from the gen-set P_{gen}. The power split is assumed ideal, i.e.

$$P_{batt}(t) = P_{EM}(t) - P_{gen}(t)$$

A positive value of P_{batt} represents power provided by the battery and a negative value corresponds to power used to charge the battery. The internal combustion engine in the gen-set is assumed, when turned on, to operate at a given constant operating point. Thus, the gen-set works in two modes and is assume to either provide a given amount of electric power P_{gen} when turned on or nothing when turned off.

FIGURE 5 An efficiency map of the electric motor.



The model of the battery pack, models the State-of-Charge (SOC) dynamics of each cell

$$P_{cell}(t) = \frac{P_{batt}(t)}{N_p N_s}$$

where N_p is number of batteries in parallel and N_s is number of batteries in series in the battery pack.

The battery cell SOC dynamics is modeled as

$$SOC(t+1) = SOC(t) - \frac{I_{cell}(t)}{Q_{cell}}$$

where Q_{cell} is the cell capacity and the cell current I_{cell} is computed as

$$I_{cell}(t) = \eta_{batt}\left(V_{oc}(SOC(t)) - \sqrt{V_{oc}(SOC(t))^2 - 4r_{batt}(SOC(t))P_{cell}}\right)$$

where V_{oc} is cell open circuit voltage and $r_{batt}(SOC)$ is the battery cell internal resistance.The cell open circuit voltage V_{oc} is computed as a function of SOC and is given by

$$V_{oc}(t) = V_0 + \alpha_0\left(1 - e^{-\beta SOC(t)}\right) + \kappa SOC(t) + \zeta\left(1 - e^{-\frac{\epsilon}{1-SOC(t)}}\right)$$

where V_0, α_0, β, κ, ζ, and ϵ, are parameters.

Dynamic Programming

Dynamic Programming (DP) is an exhaustive search algorithm that uses a discretized state vector, backward induction, and the Bellman equation to find the optimal solution [16]. It has previously been successfully used for evaluating optimal fuel consumption of HEV, see for example [18]. However, because of the computational cost, DP has mainly been used for off-line optimization. However, there are applications where DP has been used for on-line control, see for example [19].

For architecture one, shown in Figure 2, the model has four states and three control inputs. The four states are battery state-of-charge, selected gear, engine on/off mode, and torque converter lockup on/off mode, and the control inputs are gear request, engine on/off request, and torque converter lockup mode request. Excessive gear changing, turning on the engine, and changing the mode of the torque converter are penalized as a small additional fuel cost each time any of the states are changing (except for the engine which is only penalized when turned on). The other two architectures have the same states and inputs as architecture one except the ones related to the gear and torque converter. Fuel economy is here evaluated by optimizing the powertrain control when the vehicle follows a given velocity profile.

A Search Algorithm for Design Space Exploration

The design space exploration problem can be formulated as a Multi-Objective Optimization Problems (MOOP) [20]. MOOP

are optimization problems where the objective is to simultaneously minimize a set of M objective functions

$$\min_{\bar{x}} \left(g_1(\bar{x}), g_2(\bar{x}), \ldots, g_M(\bar{x}) \right)$$

$$\text{subject to } \bar{x} \in X$$

where $g_i(\bar{x})$ is an objective function, and X denotes the design search space.

However, in many cases these objective functions are contradictory meaning that if a new solution reduces the value of one objective function it will increase the value of another. Thus, different solutions will have various trade-offs between the objective functions. A candidate is called Pareto-optimal if there are no other candidates that dominates the solution, i.e. if another candidate is better with respect to all objective functions [20]. The Pareto-optimal candidates represent the optimal trade-offs between the different objective functions. The set of all Pareto-optimal solutions is referred to as the Pareto-front. Thus, identifying the Pareto-front gives useful information about which candidates, in this case powertrain architectures and component sizes, that give the optimal trade-off with respect to the different objective functions.

Pareto-front Exploration

The approach proposed here to explore the Pareto-front is to formulate a single-objective function which quantifies how close a candidate $\bar{x} \in X$ with the set of M objective functions

$$\bar{y} = \left(g_1(\bar{x}), g_2(\bar{x}), \ldots, g_M(\bar{x}) \right)$$

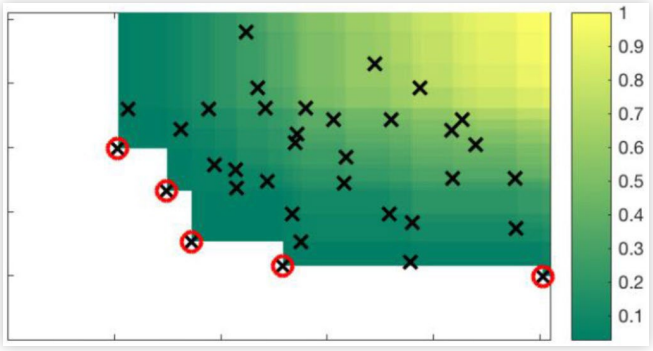
is to be Pareto-optimal as

$$h\left(\bar{y}; \hat{Y}\right) = \frac{\left| \left\{ \tilde{y} \in \hat{Y} : \tilde{y}_i \le \bar{y}_i, \tilde{y}_i \ne \bar{y}_i, \forall i = 1, \ldots, n \right\} \right|}{\left| \hat{Y} \right|} \quad (1)$$

where $|\bullet|$ denotes the number of elements in the set. The set $\hat{Y}$ contains the vectors $\bar{y}$ for all evaluated candidates. The single-objective function $h\left(\bar{y}; \hat{Y}\right)$ measures the ratio of other candidates that dominates the given candidate. An advantage of the proposed measure (1) is that it is normalized with respect to the value ranges of the different objectives. Note that for Pareto-optimal candidates, the single-objective function $h\left(\bar{y}; \hat{Y}\right) = 0$. An example of a set of evaluated candidates and the value of the single-objective function is shown in Figure 6. The white area represents where $h\left(\bar{y}; \hat{Y}\right) = 0$. The Pareto-optimal candidates are marked with circles.

Since the true Pareto-front is not known beforehand (without evaluating the whole design space) it is approximated in each iteration by computing it based on the already evaluated candidates $\hat{Y}$. The idea of the search algorithm is to sequentially search for candidates that minimizes (1). Since the search focuses on candidates that are most likely to be Pareto-optimal, the Pareto-front will improve as new candidates are evaluated in the design space.

FIGURE 6 An example of a set of evaluated candidates with two objective functions where the colors represent the value of the single-objective function.



Candidate Selection using Gaussian Processes

A search algorithm is necessary to identify which candidates in the design space to evaluate next in each iteration. A machine learning method called Gaussian Processes [21] is here proposed to identify which candidate to evaluate based on the previously evaluated candidates. Gaussian Processes have previously been proposed to solve global optimization problems, see for example [22]. Here, the idea is to estimate $h\left(\bar{y}; \hat{Y}\right)$ for candidates that have not yet been evaluated. If any candidate is estimated to have $h\left(\bar{y}; \hat{Y}\right) = 0$, within some confidence interval, the candidate is considered relevant to be evaluated because it can improve the Pareto-front. If it is estimated that $h\left(\bar{y}; \hat{Y}\right) > 0$, i.e. the candidate is not predicted to have better performance than the "so far" Pareto-optimal solutions, that candidate will not be considered for evaluation.

Gaussian Processes A Gaussian Process (GP) is a non-parametric function that can be used to model a spatially correlated function $f(u)$. It uses observations of the function input u and output y to compute an estimate $y = \hat{f}(u)$. A GP is defined by its mean function $\mu(u)$ and covariance function $k(u, u')$, which is also called a kernel function.

$$f(u) \sim GP\left(\mu(u), k(u, u')\right)$$

The conditional distribution $p\left(\hat{f} | f\right)$ is computed as

$$A = K(u, u') K(u, u)^{-1}$$

$$P = K(u', u') - K(u', u) K(u, u)^{-1} K(u, u')$$

The covariance matrix $K(u, u')$ is given by

$$K(u, u') = \begin{pmatrix} k(u_1, u_1') & \cdots & k(u_1, u_M') \\ \vdots & \ddots & \vdots \\ k(u_N, u_1') & \cdots & k(u_N, u_M') \end{pmatrix}$$

There are many different kernel functions to model spatial correlation. Here, the exponential kernel is used,

$$k(u, u') = \sigma^2 \exp\left(-\frac{r}{\ell} \right),$$

where $r = \sqrt{(u-u')^T (u-u')}$ is the Euclidean distance and σ and ℓ are tuning parameters.

The matrix P gives a confidence measure of the estimate, meaning that the estimate $\hat{f}$ has greater uncertainties at points where the diagonal elements of P is large. By assuming that the true function f exceeds a lower bound of the confidence interval, it is possible to determine which points are most likely to be a global minimum. Since the objective is to find Pareto-optimal candidates by finding candidates such that $h = 0$, new candidates are selected such that the lower bound of the confidence interval is lower than zero. As more candidates are evaluated, the confidence interval close to the evaluated candidates will shrink and new candidates will be selected in other parts of the design space which has a lower confidence interval. This will help balance the search between global exploring of different areas of the design space which have not yet been explored and local exploring where many Pareto-optimal candidates have been found.

Candidate Selection Gaussian Processes are used to estimate the function which maps a specific candidate $\bar{x}$, i.e., a set of component sizes for a given architecture, to an estimate of the distance to the Pareto-front. A GP model is estimated for each powertrain architecture where the evaluated candidates, i.e. their component configurations, are input and the corresponding values $h\left(\bar{y};\hat{Y}\right)$ are outputs in the training data. Note that, the values of the single-objective function $h\left(\bar{y};\hat{Y}\right)$ are based on all evaluated candidates for all architectures and not only the candidates from each architecture.

In each iteration of the search algorithm, the GP models are updated for all architectures. Then, the GP model for each architecture is used to estimate for all remaining candidates which are likely to be Pareto-optimal, i.e. the candidates which have a confidence interval that includes zero. A new candidate $\bar{x}$ is then selected from the architecture where the lower bound of the confidence interval has the lowest value.

If a candidate $\bar{x}$ cannot be evaluated or is infeasible, i.e. it cannot fulfill the specified performance requirements, the single-objective function (1) has no value. These candidates are ignored when evaluating the single-objective function $h\left(\bar{y};\hat{Y}\right)$. However, note that it is likely that Pareto-optimal solutions are close to the set of infeasible candidates which means that the search algorithm is likely to select infeasible candidates. To avoid that the search algorithm focuses the search among the infeasible set, a penalty is added based on the distance from each candidate to the infeasible candidates. This is done by generating a second GP model for each architecture which is used to add a penalty to the estimated lower bound based on how close it is to evaluated candidates which where feasible and infeasible. The penalty is scaled to make sure that the search avoids candidates that are likely to be infeasible but still evaluate those that are feasible. A GP model $p = \hat{f}_p(\bar{x})$ is trained for each architecture using evaluated candidates $\bar{x}$ as input where infeasible candidates have output one and feasible candidates have output minus one. Then, the penalty is computed as the maximum of the estimated output for a new candidate $\hat{p} = \hat{f}_p(\bar{x}_{new})$ and zero to have a value that is non-negative and should not penalize candidates far from the infeasible candidates.

Initialization In order to make predictions which candidate to evaluate in each iteration requires there is a set of already evaluated candidates from all architectures. Therefore, the search algorithm is initialized by evaluating a randomly selected set of candidates from each architecture. The results from these candidates are then used to predict which next candidate to evaluate by the search algorithm.

Search Algorithm Summary

The design space exploration search algorithm can be summarized in the following steps:

1. Define a search space including multiple architectures $X = X_1 \cup X_2 \cup \ldots \cup X_q$ where q is the number of architectures.
2. Select randomly for each architecture a number of candidates and evaluate the objective functions for these candidates.
3. Train a GP model for each architecture using the evaluated candidate component sizes as input and the value of the single-objective function (1) as output.
4. Train another GP for all architectures where feasible candidates are mapped to the value zero and infeasible candidates to a penalty value γ. This is used to penalize candidates that are likely to be infeasible.
5. Use the GP models to predict a lower bound of the confidence interval for all other candidates and the infeasibility penalty.
6. Select the candidate that has the lowest lower bound (after adding the infeasibility penalty). If the lower bound is larger than 0, then stop the search.
7. Otherwise, evaluate the selected candidate and go to step 3.

Some analyses of the design space exploration algorithm are presented in [23].

Powertrain Optimization for Hybrid Electric Delivery Truck

The electrified powertrain optimization is performed for the delivery truck case study using the proposed search algorithm described in the previous section. The results from the optimization are analyzed and the performance for the different architecture candidates are compared.

Design Space Exploration

The proposed design space exploration algorithm is used to optimize the powertrain based on three different driving missions defined by the driving cycles: NREL (National Renewable Energy Laboratory) PG&E (Pacific Gas and Electric Company) Utility truck driving cycle, see Figure 7,

FIGURE 7 The velocity profile of the NREL PG&E Utility truck driving cycle.

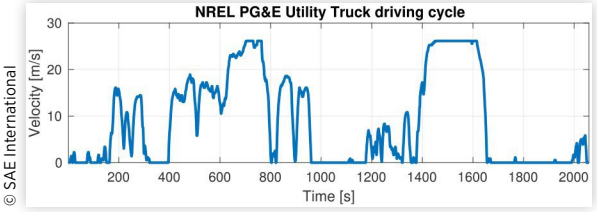


FIGURE 8 The velocity profile of the NREL Baltimore Parcel Delivery driving cycle.

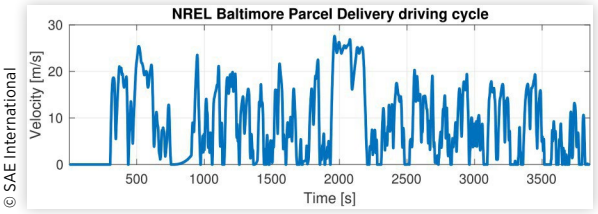
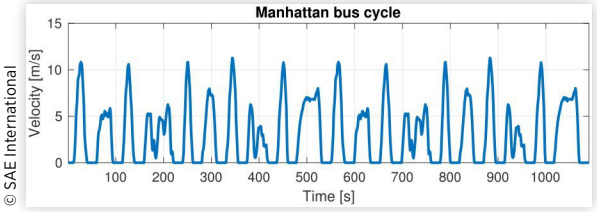


FIGURE 9 The velocity profile of the Manhattan bus cycle.



the NREL Baltimore Parcel Delivery driving cycle, see Figure 8, and the Manhattan bus driving cycle, see Figure 9 [24]. The roads in all driving cycles are here assumed to be flat. The NREL PG&E Utility truck driving cycle and the NREL Baltimore Parcel Delivery driving cycle have similar velocity characteristics but the first one has longer stops and the second one has a time duration and driving distance which are twice as long. The Manhattan bus cycle is shorter than the other driving cycles, and has a significantly slower velocity profile, including both lower maximum and average speed. A suitable powertrain is expected to differ for the three driving cycles because of the different driving mission characteristics.

The design spaces for the different architectures are defined as shown in Table 2. Note that architecture three has a different reference motor map compared to architectures one and two. There is a limited set of component sizes, or differential ratios, for each component. The number of dimensions of the design spaces for architectures one and two are three, including battery size (by varying the number of batteries in parallel), motor size, and differential ratio, and two for architecture three, including only battery size and motor size. The total number of candidates for each architecture is shown in Table 2, which gives a total number of 16107 candidates for the complete design search space.

The results from the three design space explorations are summarized in Table 3 and the Pareto-fronts are visualized in Figures 10-12. The candidates from each architecture are

TABLE 2 A summary of the design space for each architecture.

Powertrain component	Architecture 1	Architecture 2	Architecture 3
Battery size	72 – 216 kW	72 – 216 kW	72 – 216 kW
Electric motor (Relative to reference map)	90 – 200 %	90 – 200 %	100 – 400 % (smaller reference map)
Differential ratio	4.23 – 6.14	4.23 – 16.93	-
Number of candidates	9177	6279	651

© SAE International

TABLE 3 A summary of the design space exploration for the three driving cycles

Driving cycle	# Eval	# Infeas	# Pareto
NREL PG&E Utility truck	1635	153	89
NREL Baltimore Parcel Delivery	1830	150	169
Manhattan bus cycle	123	13	33

© SAE International

FIGURE 10 The Pareto-front for the design space exploration using the NREL PG&E Utility truck driving cycle. Candidates from different architectures are color-coded and the Pareto-optimal candidates are highlighted.

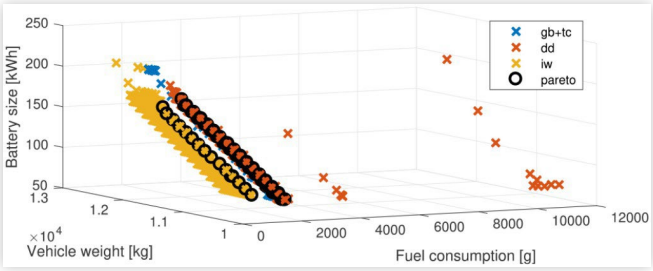
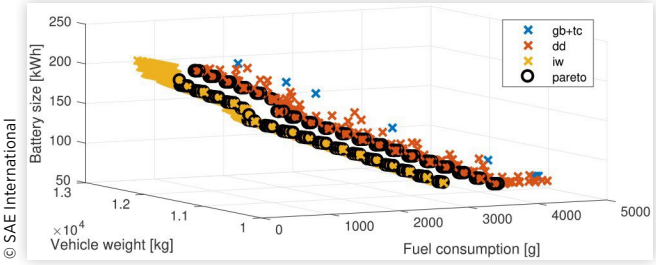


FIGURE 11 The Pareto-front for the design space exploration using the NREL Baltimore Parcel Delivery driving cycle.



color-coded in the figures and the Pareto-optimal solutions are highlighted where architecture one (gb+tc) is blue, architecture two (dd) is orange, and architecture three (iw) is yellow. Some candidates have been evaluated which are located far from the Pareto-front. However, most candidates are close to be Pareto-optimal which indicates that the proposed search algorithm is able to identify the relevant part of the design space. Around 10% of the evaluated candidates turned out to

FIGURE 12 The Pareto-front for the design space exploration using the Manhattan bus cycle. Note that only candidates from architecture two (direct-drive) are Pareto-optimal.

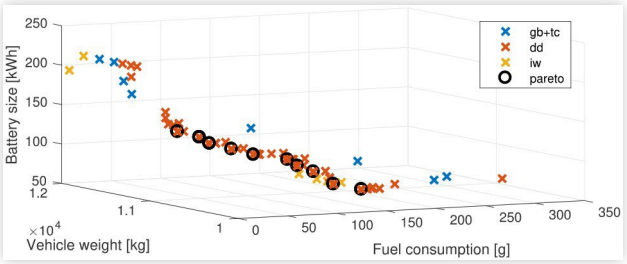


FIGURE 13 Analysis of the component-sizes of the Pareto-optimal candidates from architectures two (upper plots) and three (lower plots), with respect to fuel consumption when driving the NREL PG&E Utility truck driving cycle.

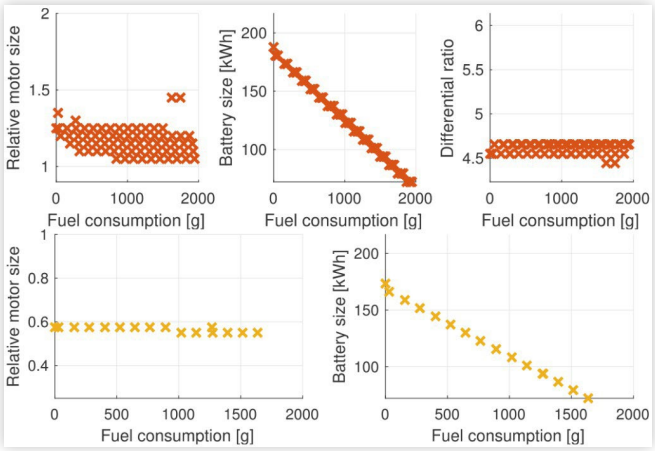


FIGURE 14 Analysis of the component-sizes of the Pareto-optimal candidates from architectures two (upper plots) and three (lower plots), with respect to fuel consumption when driving the NREL Baltimore Parcel Delivery driving cycle.

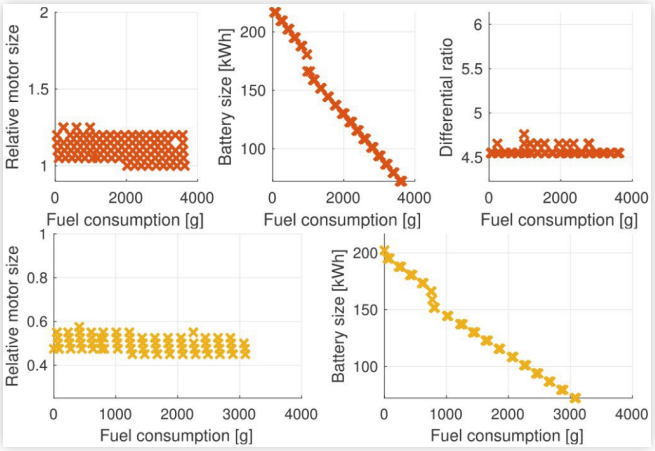
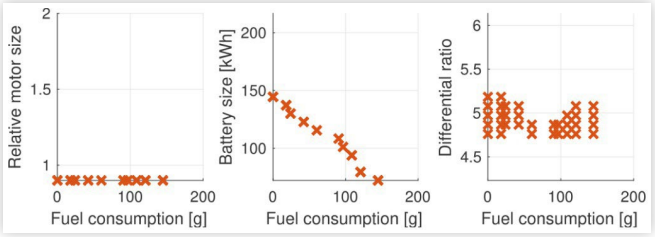


FIGURE 15 Analysis of the component-sizes of the Pareto-optimal candidates three with respect to fuel consumption when driving the Manhattan bus cycle.



be infeasible which is expected since the Pareto-optimal candidates are likely to have components that are just enough to fulfill the requirements of each driving cycle.

The proposed search algorithm finished the design space exploration after evaluating approximately 11% of all candidates in the design space. However, in the third case study, using the Manhattan bus cycle, the search stopped already after evaluating less than 0.1% of the candidates which can be explained by that the Pareto-front is dominated by one architecture. Monte Carlo simulations of the search algorithm, with different initial starting points of the search, gave similar Pareto-fronts, including similar numbers of evaluated candidates and the numbers of Pareto-optimal candidates. This indicates that the proposed search algorithm is robust and able to find the Pareto-front.

Analyzing the Results from the Electrical Powertrain Optimization

The most visible trade-off for all architectures in the three case studies is between fuel consumption and battery size since the fuel consumption is reduced if a larger battery pack is used. When comparing the different Pareto-fronts for the three driving scenarios, it is visible that a longer driving cycle requires a larger the battery size to reduce fuel consumption to a certain level. For example, a 50% larger battery pack is required to fulfill Baltimore Parcel Delivery driving cycle with zero emissions compared to the Manhattan bus cycle. Candidates from architecture one, i.e. the powertrain including a torque converter and gearbox, are never Pareto-optimal in any of the three design space explorations. An explanation is that candidates from architecture one is, in general, heavier than candidates from the other two architectures and also have a lower powertrain efficiency. Also, the in-wheel motors do not need really as large battery pack as the direct-drive even though the difference is not that significant.

To visualize the range of component sizes for the Pareto-optimal candidates from architecture two and three, they are plotted in Figures 13-15, respectively, with respect to fuel consumption for each of the three case studies. When comparing the different figures, it is clear that the minimum size of the electric motor among the Pareto-optimal candidates increases as fuel consumption goes to zero, especially for the direct-drive. This can be explained by that a larger battery size is required to reduce the fuel consumption, which

increases the total vehicle weight and, thus, requires a larger motor. This is most visible in the Figure 13 when optimizing for the NREL PG&E Utility Truck driving cycle, where the minimum relative motor size is almost 25% larger for zero fuel consumption compared to 1 kg of fuel consumption. The minimum required motor size also depends on the required traction force characteristics of the driving cycle. When comparing the results from the NREL driving cycles with the Manhattan bus cycle, which has lower top speed and average speed, a 20% smaller engine size is sufficient to fulfill the driving cycle.

There is also a visible difference in optimal final differential ratios for the three different driving cycles. The Pareto-optimal solutions for the Manhattan bus cycle have a higher differential ratio range compared to the NREL cycles. Vehicles developed primarily for urban driving, for example the Nissan Leaf, are expected to have a smaller operating range, mainly including the lower speed range and also lower torque demands. In these situations, it is more optimal to have a higher differential ratio to make the motor operate more efficient while a lower ratio is better when the driving missions have more varying speeds and torque demands.

Concluding Remarks and Future Works

The proposed design space exploration framework, for multiple powertrain architecture candidates and multiple component sizes, is applied to optimizing a medium-sized delivery truck. Characteristics of the driving mission have a significant effect on how the powertrain should be sized. One of the general conclusions of the analysis is that the direct-drive powertrain and in-wheel motors have a lower fuel consumption for a given size of the battery, compared to a more conventional drivetrain, because of reduced vehicle weight and lower losses in the drivetrain. The results also indicate that a higher differential gear ratio is more suitable for urban driving scenarios compared to driving scenarios with more varying driving conditions. For future works, the aim is to expand the design space exploration by adding additional powertrain architectures and including more components in the optimization. Multiple driving cycles should also be taken into consideration when evaluating fuel consumption to improve robustness. It should also be investigated how to improve the performance of the search algorithm, for example using parallel computations.

References

1. Silvas, E., Hofman, T., Murgovski, N., Etman, L.P., and Steinbuch, M., "Review of optimization strategies for system-level design in hybrid electric vehicles," *IEEE Transactions on Vehicular Technology* 66(1):57-70, 2017.
2. Josephson, J.R., Chandrasekaran, B., Carroll, M., Iyer, N., et al. "An architecture for exploring large design spaces." AAAI/IAAI 143150, 1998.
3. Rizzoni, G., Josephson, J.R., Soliman, A., Hubert, C. et al., "Modeling, simulation, and concept design for hybrid-electric medium-size military trucks," *Proc. SPIE* 5805:1-12, 2005.
4. Vora, A.P., Jin, X., Hoshing, V., Guo, X. et al., "Simulation Framework for the Optimization of HEV Design Parameters: Incorporating Battery Degradation in a Lifecycle Economic Analysis," *IFAC-PapersOnLine* 48(15):195-202, 2015.
5. Ebbesen, S., Elbert, P., and Guzzella, L., "Engine downsizing and electric hybridization under consideration of cost and drivability," *Oil & Gas Science and Technology-Revue d'IFP Energies nouvelles* 68(1):109-116, 2013.
6. Donateo, T., Serrao, L., and Rizzoni, G., "Multi-objective optimization of a heavy-duty hybrid electric vehicle," *Strings* 1(4.0):1, 2007.
7. Zhou, Q., Zhang, W., Cash, S., Olatunbosun, O. et al., "Intelligent sizing of a series hybrid electric power-train system based on Chaos-enhanced accelerated particle swarm optimization," *Applied Energy* 189:588-601, 2017.
8. Silvas, E., Bergshoeff, E, Hofman, T., and Steinbuch, M., "Comparison of bi-level optimization frameworks for sizing and control of a hybrid electric vehicle," *Vehicle Power and Propulsion Conference (VPPC)*, 2014 IEEE, 1-6. IEEE, 2014.
9. Shankar, R., Marco, J., and Assadian, F., "The novel application of optimization and charge blended energy management control for component downsizing within a plug-in hybrid electric vehicle," *Energies* 5(12):4892-4923, 2012.
10. Pourabdollah, M., Silvas, E., Murgovski, N., Steinbuch, M. et al., "Optimal sizing of a series PHEV: Comparison between convex optimization and particle swarm optimization," *IFAC-PapersOnLine* 48(15):16-22, 2015.
11. Bayrak, A.E., Kang, N., and Papalambros, P.Y., "Decomposition-based design optimization of hybrid electric powertrain architectures: Simultaneous configuration and sizing design," *Journal of Mechanical Design* 138(7):071405, 2016.
12. Pourabdollah, M., Murgovski, N., Grauers, A., and Egardt, B., "An iterative dynamic programming/convex optimization procedure for optimal sizing and energy management of PHEVs," *IFAC Proceedings* 47(3):6606-6611, 2014.
13. Pisu, P., Koprubasi, K., and Rizzoni, G., "Energy management and drivability control problems for hybrid electric vehicles," In Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on, 1824-1830. *IEEE*, 2005.
14. Bayrak, A.E., Yi, R., and Papalambros, P.Y., "Topology Generation for Hybrid Electric Vehicle Architecture Design," *Journal of Mechanical Design* 138(8):081401, 2016.
15. Zhuang, W., Zhang, X., Peng, H., and Wang, L., "Simultaneous optimization of topology and component sizes for double planetary gear hybrid powertrains," *Energies* 9(6):411, 2016.
16. Guzzella, L. and Sciarretta, A., "*Vehicle propulsion systems (Vol. 1)*," (Berlin Heidelberg, Springer-Verlag, 2007).
17. Kotwicki, A. J., "Dynamic models for torque converter equipped vehicles," SAE Technical Paper 820393, 1982, 10.4271/820393

18. Sundstrom, O. and Guzzella, L., "A generic dynamic programming Matlab function," in Control Applications, (CCA) & Intelligent Control, (ISIC), 2009 IEEE, 1625-1630, 2009.
19. Hellström, E., Ivarsson, M., Åslund, J., and Nielsen, L., "Look-ahead control for heavy trucks to minimize trip time and fuel consumption," *Control Engineering Practice* 17(2):245-254, 2009.
20. Deb, K., "Multi-objective optimization," . In: *Search methodologies*. (US, Springer, 2014), 403-449.
21. Rasmussen, C. and Williams, C., "*Gaussian processes for machine learning*," . Vol. 1 (MIT press Cambridge, 2006).
22. Osborne, M. A., Garnett, R., and Roberts, S. J., "Gaussian processes for global optimization". In 3rd international conference on learning and intelligent optimization (LION3), 1-15, 2009 January.
23. Jung, D., Ahmed, Q., and Rizzoni, G., "Design Space Exploration for Powertrain Electrification using Gaussian Processes", In American Control Conference 2018, Milwaukee, USA.
24. Kelly, K., Prohaska, R., Ragatz, A., and Konan, A., "NREL DriveCAT - Chassis Dynamometer Test Cycles. National Renewable Energy Laboratory," http://www.nrel.gov/transportation/drive-cycle-tool, 2016.

Contact Information

Daniel Jung
Center for Automotive Research, The Ohio State University, Columbus, Ohio, 43212, USA.
(jung.693@osu.edu, daniel.jung@liu.se)

ISSN 0148-7191