**Proceedings of ICEF07**
**2007 Fall Technical Conference of the ASME Internal Combustion Engine Division**
**October 14-17, 2007, Charleston, South Carolina, USA**

# ICEF2007-1603

# REAL-TIME, SELF-LEARNING OPTIMIZATION OF DIESEL ENGINE CALIBRATION

**Andreas A. Malikopoulos**
Department of Mechanical Engineering
University of Michigan
Ann Arbor, Michigan 48109, U.S.A
amaliko@umich.edu

**Dennis N. Assanis**
Department of Mechanical Engineering
University of Michigan
Ann Arbor, Michigan 48109, U.S.A
assanis@umich.edu

**Panos Y. Papalambros**
Department of Mechanical Engineering
University of Michigan
Ann Arbor, Michigan 48109, U.S.A
pyp@umich.edu

## ABSTRACT

Compression ignition engine technologies have been advanced in the last decade to provide superior fuel economy and high performance. These technologies offer increased opportunities for optimizing engine calibration. Current engine calibration relies on deriving static tabular relationships between a set of steady-state operating points and the corresponding optimal values of the controllable variables. The values of these tabular relationships are interpolated to provide values of the controllable variables for each operating point while the engine is running. These values are controlled by the electronic control unit to achieve desirable engine behavior, for example in fuel economy, pollutant emissions, and engine acceleration performance. These strategies, however, are less effective during transient operation. Use of learning algorithms is an alternative approach that treats the engine as an "autonomous" system, namely, one capable of learning its optimal calibration for both steady-state and transient operating points in real time. In this approach, while the engine is running the vehicle, it progressively perceives the driver's driving style and eventually learns to operate in a manner that optimizes specified performance indices. Major challenges to this approach are problem dimensionality and learning time. This paper examines real-time, self-learning calibration of a diesel engine with respect to two controllable variables, i.e., injection timing and VGT vane position, to minimize fuel consumption. Some promising simulation-based results are included.

Keywords: diesel engine calibration, engine management system, injection timing, VGT blade position, fuel economy, stochastic optimal control, sequential decision-making under uncertainty

## 1. INTRODUCTION

Advanced compression ignition engine technologies, such as fuel injection systems, variable geometry turbocharging (VGT), exhaust gas recirculation (EGR), and variable valve actuation (VVA), have alleviated the traditional disadvantages of diesel engines, and facilitated their use in the passenger vehicle market. These technologies provide an increased number of engine controllable variables that can be used for engine calibration to optimize one or more engine performance criteria, e.g., fuel economy, pollutant emissions, or engine acceleration performance. Current engine calibration methods generate a static tabular relationship between the optimal values of the controllable variables and the corresponding steady-state engine operating points to achieve optimal performance with respect to the specified criteria. This relationship is incorporated into the electronic control unit (ECU) that aims to maintain performance optimality. While the engine is running, values in the tabular correlations are being interpolated to provide the values of the controllable variables for each operating point.

Design of Experiments (DoE) techniques [1-4] have been widely employed as the baseline method in implementing engine calibration. The major objective of DoE is to expedite dynamometer tests, using a smaller subset of tests. This subset is utilized either in implementing engine calibration experimentally [5, 6] or in developing mathematical models for evaluating engine output. The latter employs equation-based or artificial neural network (ANN) models [7-9]. Using these models, appropriate optimization algorithms [10] can determine the relationships between the steady-state operating points and the values of the controllable variables [11-15]. These methods, however, seldom guarantee optimal engine calibration for the entire operating domain, especially during transient operation [16]. The latter often constitutes the largest segment of engine operation compared to the steady-state one [17, 18]. Fuel consumption and emissions during transient operation are extremely complicated [18], and are highly dependent on engine calibration [19, 20]. Atkinson *et al.* [16] implemented a dynamic model-based calibration system to provide optimal calibration for transient engine operation of particular driving cycles. Pre-specifying the entire transient engine operation as imposed by different driving cycles and deriving the optimal values of the controllable variables associated with transient operating points is not possible in practice, thus preventing *a priori* optimal calibration.

As an effort to address these issues, an alternative approach was implemented recently, which treats the engine as a controlled stochastic system, and engine operation as a Markov Decision Process [21]. Engine calibration is formulated as a sequential decision-making problem under uncertainty. A learning algorithm was implemented allowing the engine to learn the optimal values of the controllable variables in real time while running the vehicle. As the engine is running the vehicle, it progressively perceives the driver's driving style and eventually learns to operate in a manner that optimizes specified performance indices, e.g., fuel economy, emissions, or engine performance with respect to the engine controllable variables. Consequently, optimal calibration is achieved for steady-state and transient engine operating points resulting from the driver's driving style. The engine's ability to learn its optimum calibration is not limited, however, to a particular driving style. The engine can learn to operate optimally for different drivers if they indicate their identity before starting the vehicle. The engine can then adjust its operation to be optimal for a particular driver based on what it has learned in the past regarding his/her driving style. A major challenge to this approach is the increase of the problem's dimensionality when more than one controllable variable is considered.

This paper introduces a decentralized learning method to include two or more controllable variables in real-time, self-learning optimization of engine calibration. In decentralized learning, the learning algorithm no longer considers all combinations of values of the controllable variables. Instead, it establishes a hierarchy of the variables, and learns the optimal values of each variable in parallel. In particular, the algorithm is employed to derive the optimal values of one controllable variable with respect to the sequence of state transitions imposed by the driver's driving style. At the same time, the algorithm is engaged separately to derive the optimal values of the second controllable variable with respect to the optimal policy as being learned for the first one. In case of more than two controllable variables the algorithm is employed to derive the optimal values of the third variable with respect to the second one and so forth.

In the following section the mathematical framework of considering the engine operation as a Markov Decision Process (MDP) is reviewed. The decentralized engine learning process is introduced in Section 3. The effectiveness of the method is demonstrated in Section 4 through simulation of a diesel engine calibration with respect to the injection timing and VGT vane position. Results are discussed in Section 5, and conclusions are presented in Section 6.

## 2. MODELING ENGINE OPERATION AS A MARKOV DECISION PROCESS

In implementing self-learning optimization for engine calibration in real time, the engine is treated as a controlled stochastic system, and engine operation is modeled as a Markov Decision Process (MDP) [22]. The engine performance indices, e.g., fuel economy, emissions, and engine acceleration performance, are considered controlled random functions. The objective is to select the optimal control policy (optimum values of the controllable variables) for the sequences of engine operating point transitions, corresponding to the driver's driving style, that optimize one or more engine performance indices (random functions). The problem of engine calibration is thus formulated as a sequential decision-making problem under uncertainty.

The MDP provides the mathematical framework for modeling such problems [23]. It comprises a decision maker (engine), states (engine operating points), actions (controllable variables), the transition probability matrix (driver), the transition reward matrix (engine performance indices), and optimization criteria (e.g., maximizing fuel economy, minimizing pollutant emissions, maximizing engine performance). In this framework, the engine (decision maker) is faced with the problem of influencing the performance indices over time by selecting optimal actions (values of the controllable variables). The objective of the engine is to select the course of action (control policy) which optimizes one or more engine performance indices.

Following the exposition in [21], a discrete-time, stochastic controlled MDP is defined as the tuple:

$$\mathcal{X}_n = \{\mathcal{S}, \mathcal{A}, \mathbf{P}(\cdot,\cdot), \mathbf{R}(\cdot,\cdot)\} \qquad (1)$$

where $\mathcal{S} = \{s_i \mid i = 1,2,...,N\}$, $N \in \mathcal{N}$ denotes a finite state space, $\mathcal{A} = \bigcup_{s_i \in \mathcal{S}} A(s_i)$ stands for a finite action space, $\mathbf{P}(\cdot,\cdot)$ is the transition probability matrix, and $\mathbf{R}(\cdot,\cdot)$ is the transition reward matrix. The decision-making process occurs at each of a sequence of stages $\kappa = 0,1,2,...,M$, $M \in \mathcal{N}$. At each stage, the decision maker observes a system's state $s_i \in \mathcal{S}$, and executes an action $\alpha \in A(s_i)$, from the feasible set of actions $A(s_i) \subseteq \mathcal{A}$ at this state. At the next stage, the system transits to the state $s_j \in \mathcal{S}$ imposed by the conditional probabilities $p(s_j \mid s_i, \alpha)$, designated by the transition probability matrix $\mathbf{P}(\cdot,\cdot)$. The conditional probabilities of $\mathbf{P}(\cdot,\cdot)$, $p : \mathcal{S} \times \mathcal{A} \rightarrow [0,1]$, satisfy the constraint

$$\sum_{j=1}^{N} p(s_j \mid s_i, \alpha) = 1. \tag{2}$$

Following this state transition, the decision maker receives a reward associated with the action $\alpha$, $R(s_j \mid s_i, \alpha), R : \mathcal{S} \times \mathcal{A} \rightarrow \mathfrak{R}$ as imposed by the transition reward matrix $\mathbf{R}(\cdot,\cdot)$. The states of an MDP possess the Markov property, stating that the conditional probability distribution of future states of the process depends only upon the current state and not on any past states, i.e., it is conditionally independent of the past states (the path of the process) given the present state.

Mathematically, the Markov property states that

$$\begin{aligned} &p(X_{n+1} = s_j \mid X_n = s_i, X_{n-1} = s_{i-1}, ..., X_0 = s_0) = \\ &= p(X_{n+1} = s_j \mid X_n = s_i). \end{aligned} \tag{3}$$

The solution to an MDP can be expressed as a policy $\pi = \{\mu_0, \mu_1, ..., \mu_M\}$, which provides the action to be executed for a given state, regardless of prior history; $\mu_\kappa$ is a function mapping states $s_i$ into actions $\alpha = \mu_\kappa(s_i)$, such that $\mu_\kappa(s_i) \in A(s_i)$. Such policies are addressed as admissible. Consequently, for any initial state at stage $\kappa = 0$, $s_i^0$, and for any finite sequence of stages $\kappa = 0,1,2,...,M$, $M \in \mathcal{N}$, the expected accumulated value of the rewards of the decision maker is given by the Bellman equation

$$V_\pi(s_l^M) = E\{R(s_j^0 \mid s_i^0, \mu_0(s_i^0)) + \sum_{\kappa=1}^{M} V_\pi(s_j^\kappa), j \in \mathcal{N}\}, \quad \forall s_i, s_j, s_l \in \mathcal{S}. \tag{4}$$

In the finite-horizon context the decision maker should maximize the accumulated value for the next $M$ stages; more precisely, an optimal policy $\pi^*$ is one that maximizes the overall expected accumulated value of the rewards

$$V_{\pi^*}(s_l^M) = \max_{\pi \in \mathcal{A}} V_\pi(s_l^M). \tag{5}$$

Consequently, the optimal policy $\pi^* = \{\mu_0^*, \mu_1^*, ..., \mu_M^*\}$ for the $M$-stage sequence is

$$\pi^* = \arg\max_{\pi \in \mathcal{A}} V_{\pi^*}(s_l^M). \tag{6}$$

The finite-horizon model is appropriate when the decision-maker's "lifetime" is known, namely, the terminal stage of the decision-making sequence. However, in most real-life problems this is not the case; these problems are modeled in the infinite-horizon context, where the overall expected reward is the limit of the corresponding $M$-stage overall reward as $M \rightarrow \infty$:

$$V_{\pi^*}(s_l^M) = \lim_{M \rightarrow \infty} \max_{\pi \in \mathcal{A}} V_\pi(s_l^M). \tag{7}$$

This relation is extremely valuable for various MDP problems, where the terminal stage is unknown; Eq. (7) holds under certain conditions [24].

Dynamic programming (DP) has been widely used for many years as the principal method for solving Markov decision problems [25]. However, DP algorithms require the realization of the transition probability matrix, $\mathbf{P}(\cdot,\cdot)$, and the transition reward matrix, $\mathbf{R}(\cdot,\cdot)$. For complex systems, e.g., an internal combustion engine, with large state space, these matrices can be either impractical or impossible to compute. Alternative approaches for solving Markov decision problems have been developed in the field of reinforcement learning (RL) [26, 27]. RL has aimed to provide simulation-based algorithms for learning control policies of complex systems, where exact modeling is infeasible or expensive [28]. In this framework, the system interacts with its environment in real time, and obtains information enabling it to improve its future performance by means of rewards associated with the control actions taken. This interaction allows the system to learn in real time the course of action (control policy) that optimizes the rewards. The majority of RL algorithms are founded on dynamic programming. They utilize evaluation functions attempting to successively approximate the Bellman equation, Eq. (4). These evaluation functions assign to each state the total reward expected to accumulate over time starting from a given state when a policy $\pi$ is employed. However, in learning engineering systems in which the initial state is not fixed, recursive updates of the evaluation functions to approximate Eq.(4) would demand significant amount of time to achieve the desired system performance.

For the engine calibration problem built upon the MDP theoretical framework, the predictive optimal stochastic control learning algorithm is employed [21]. The algorithm is intended to derive the optimal policy (values of the engine controllable variables) $\pi^* \in \mathcal{A}$, for any initial state (engine operating point). In applying this algorithm to more than one controllable variable, limitations arise due to the requirement for the algorithm to account for all combinations of the controllable variables in a single set of a finite action space $\mathcal{A}$. To overcome this problem the decentralized learning method is implemented.

## 3. DECENTRALIZED LEARNING METHOD FOR TWO OR MORE CONTROLLABLE VARIABLES

While the engine is running the vehicle and interacting with the driver, the probability of engine operating point transitions designate the elements of the transition probability matrix, $\mathbf{P}(\cdot,\cdot)$. The desired engine performance indices, e.g., fuel economy, pollutant emissions, etc, are represented by the elements of the transition reward matrix, $\mathbf{R}(\cdot,\cdot)$. Through this

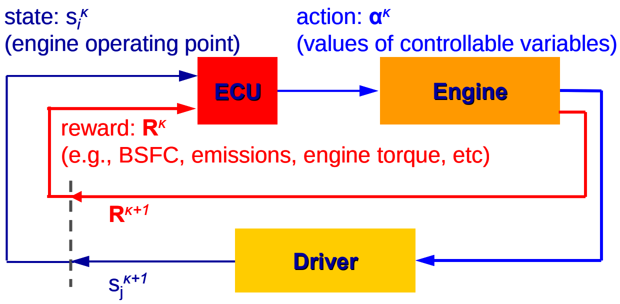


**Figure 1.** Learning process during the interaction between the engine and the driver.

interaction, the driver introduces a state $s_i^\kappa \in \mathcal{S}$ (engine operating point) to the engine's electronic control unit (ECU), and on that basis the ECU selects an action, $\alpha^\kappa \in A(s_i)$ (combination of values of the controllable variables). As a consequence of its action, the ECU receives a numerical reward, $R^{\kappa+1} \in \mathfrak{R}$, and the engine transits to a new state $s_j^{\kappa+1} \in \mathcal{S}$, as illustrated in Figure 1. The algorithm aims to predict the optimal policy (optimal values of the controllable variables) for the sequence of engine operating transitions based on the conditional probabilities of the matrix, $\mathbf{P}(\cdot,\cdot)$. During this process, however, when two or more controllable variables are considered, the combinations of their values can grow intractably, resulting in a huge feasible action set $\mathcal{A} = \bigcup_{s_i \in \mathcal{S}} A(s_i)$.

The decentralized learning method establishes a learning process that enables the derivation of the optimal values of the controllable variables to occur in parallel phases. The algorithm is employed to derive the optimal policy of the one controllable variable with respect to the sequence of state transitions imposed by the driver's driving style. Concurrently, the algorithm is also engaged separately to derive the optimal policy of the second controllable variable with respect to the optimal policy as being learned for the first one. In case of more than two controllable variables the algorithm is employed in a similar fashion, namely, the third variable with respect to the second one and so forth.

In implementing a diesel engine calibration with respect to the injection timing, $\alpha$, and VGT vane position, $\beta$, a feasible set of values, $\mathcal{A}$ and $\mathcal{B}$, for each controllable variable is defined. The decentralized learning enables the engine to implement two different mappings in parallel. In the first, injection timing is mapped to the states as a result of the correspondence of the driver's driving style to particular engine operating points, i.e., $\mathcal{S} \times \mathcal{A} \to \mathfrak{R}$. In the second, VGT is mapped to the injection timing, i.e., $\mathcal{A} \times \mathcal{B} \to \mathfrak{R}$. The learning algorithm utilizes these two mappings to derive the optimal policies, $\pi_\alpha^* \in \mathcal{A}$, and $\pi_\beta^* \in \mathcal{B}$ (optimal values of injection timing and VGT) for the driver's driving style as expressed by the incidence in which particular states or particular sequences of states arise.

The decentralized learning process of the engine transpires at each stage $\kappa$ in conjunction with the injection timing $\alpha \in \mathcal{A}$ taken for each state $s_i \in \mathcal{S}$, and VGT vane position $\beta \in \mathcal{B}$ for each $\alpha \in \mathcal{A}$. At the early stages, and until full exploration of the feasible sets, $\mathcal{A}$ and $\mathcal{B}$, occurs, the mapping from states to probabilities of selecting a particular value of injection timing $\alpha \in \mathcal{A}$, and the mapping from $\alpha \in \mathcal{A}$ to probabilities of selecting VGT $\beta \in \mathcal{B}$ are constant; namely, the values of each controllable variable are selected randomly with the same probability

$$p(\alpha \mid s_i) = \frac{1}{|\mathcal{A}|}, \forall \alpha \in \mathcal{A}, \forall s_i \in \mathcal{S}, \text{ and} \tag{8}$$

$$p(\beta \mid \alpha) = \frac{1}{|\mathcal{B}|}, \forall \beta \in \mathcal{B}, \forall s_i \in \mathcal{S}, \tag{9}$$

$$i = 1, 2, \ldots, N, N = |\mathcal{S}|.$$

Exploration of the entire feasible set for each variable is important to evade sub-optimal solutions. The learning algorithm is thus used after the exploration phase to realize the optimal policies, $\pi_\alpha^*$, and $\pi_\beta^*$ by means of the expected rewards, $V(s_j \mid s_i, a)$ and $V(\alpha_m \mid \alpha_n, \beta)$, generated by the mappings $\mathcal{S} \times \mathcal{A} \to \mathfrak{R}$, and $\mathcal{A} \times \mathcal{B} \to \mathfrak{R}$, respectively. The expected rewards are defined to be

$$V(s_j \mid s_i, a) = p(s_j \mid s_i, a) \cdot R(s_j \mid s_i, a) + \\ + \max_{a \in \mathcal{A}} \left( \frac{\sum_{l=1}^{N} p(s_l \mid s_j, a) \cdot R(s_l \mid s_j, a)}{N} \right), and \tag{10}$$

$$V(\alpha_m \mid \alpha_n, \beta) = p(\alpha_m \mid \alpha_n, \beta) \cdot R(s_j \mid \alpha, \beta) \\ + \max_{\beta \in \mathcal{B}} \left( \frac{\sum_{p=1}^{\Lambda} p(a_p \mid a_m, \beta) \cdot R(a_p \mid a_m, \beta)}{N} \right), \tag{11}$$

$$i, j = 1, 2, \ldots, N, N = |\mathcal{S}|, and$$
$$m, n = 1, 2, \ldots, \Lambda, \Lambda = |\mathcal{A}|.$$

In deriving the optimal policies of the injection timing and VGT in self-learning calibration, which is treated in a stochastic framework, all uncertain quantities are described by probability distributions. The optimal policies, $\pi_\alpha^*$, and $\pi_\beta^*$ are based on the max-min control approach, whereby the worst possible values of the uncertain quantities within the given set are assumed to occur. This is a pessimistic point of view that essentially assures the optimal policies will result in at least a minimum overall reward value. Consequently, being at state

$s_i$, the algorithm predicts the optimal policy $\pi_\alpha^*$ in terms of the values of injection timing $\alpha$ as

$$\pi_\alpha^*(s_i) = \arg\max \left\{ \min_{\substack{s_j \in S \\ a \in A}} V(s_j \mid s_i, a) \right\}, \quad \forall s_i, s_j \in S. \tag{12}$$

For this optimal policy $\pi_\alpha^*$ the algorithm predicts the optimal policy $\pi_\beta^*$ in terms of the values of the VGT vane position $\beta$ as

$$\pi_\beta^*(\alpha) = \arg\max \left\{ \min_{\substack{a \in A \\ \beta \in B}} V(\alpha_m \mid \alpha_n, \beta) \right\}. \tag{13}$$

Employing decentralized learning, the derivation of the optimal values of more than one controllable variable can be achieved while avoiding the problem of dimensionality.

## 4. EXAMPLE

The decentralized learning introduced in the previous section is now applied to a four-cylinder, 1.9L turbocharged diesel engine. The objective is to find the optimal injection timing and VGT vane position while the engine is running the vehicle that maximize the engine brake torque. Injection timing is an important controllable variable in the combustion process, and affects performance and emissions [29]. The major objective of injection timing is to initiate the start of the fuel injection at the crank angle resulting in the maximum brake torque (MBT). It designates the ignition delay defined to be the crank angle between the start of injection (SOI) and the start of combustion (SOC). The VGT technology was originally considered to increase engine brake torque at tip-ins and reduce turbo-lag. VGT has a system of movable guide vanes located on the turbine stator. By adjusting the guide vanes, the exhaust gas energy to the turbocharger can be regulated, and thus the compressor mass airflow and exhaust manifold pressure can be controlled.

The software package enDYNA Themos CRTD by TESIS [30] suitable for real-time simulation of diesel engines is employed. The software utilizes thermodynamic models of the gas path and is well suited for testing and development of electronic control unit (ECU). In the example, the existing static correlation involving injection timing and VGT is bypassed to incorporate the learning method and is used as a baseline comparison. The engine models with the baseline and self-learning calibration are run repeatedly over the same driving style represented by a segment of the FTP-75 driving cycle, illustrated in Figure 2. Every run over this driving style constitutes one complete simulation. Before initiating the first simulation of the engine model, the elements of the transition probability and reward matrix are assigned to be zero. That is, the engine at the beginning has no knowledge regarding the particular driving style and the values of the rewards associated with the controllable variables (injection timing and VGT).

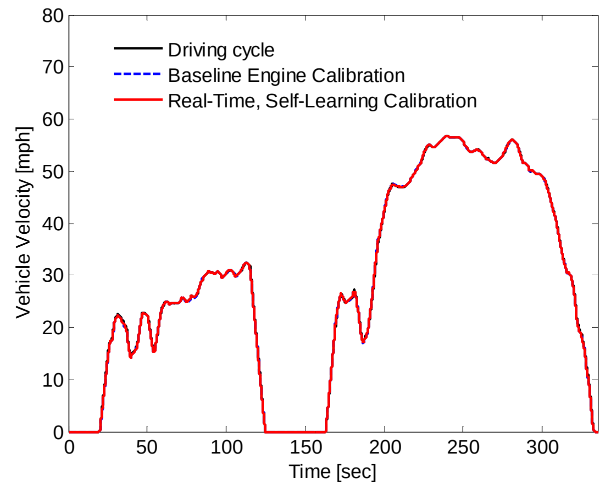


**Figure 2.** Segment of the FTP-75 driving cycle.

## 5. SIMULATION RESULTS

After completing eight simulations with the decentralized learning method, the algorithm specified the optimal values of the injection timing and VGT vane position. The vehicle with the self- learning calibration was able to follow the segment of the driving cycle requiring lower gas pedal position rates for the same engine speed, as illustrated in Figures 3-5. The implication is that the derived policy of injection timing and VGT resulted in higher engine torque compared to the baseline calibration. The injection timing (before top dead center BTDC) for both vehicles is illustrated in Figures 6 and 7. While the baseline calibration interpolates values of the injection timing of steady-state operating points, the injection timing derived by the learning algorithm corresponded to the engine operating point transitions imposed by the driver's driving style, and thus, self-learning calibration was able to capture transient engine operation. Lower gas pedal position rates resulted in reducing the fuel mass injection duration, shown in Figure 8, and consequently, less fuel mass was injected into the cylinders, as illustrated in Figure 9 (in zoom-in for clarity). In the decentralized learning of the engine, the injection timing was mapped to the engine operating points (states) while the VGT vane position was mapped to the optimal injection timing. The derived VGT policy is illustrated in Figures 10 and 11. Both injection timing and VGT were derived from the learning algorithm to maximize the engine torque during the engine operating point transitions.

Having the engine operate at the maximum brake torque, a 9.3% overall improvement of fuel economy was accomplished, as illustrated in Figure 12, compared to the baseline calibration. Figures 13 and 14, show a decrease in the temperature and NOx concentration of the exhaust gas; this is due to the earlier injection determined for the engine operating transitions of the particular driver's style.

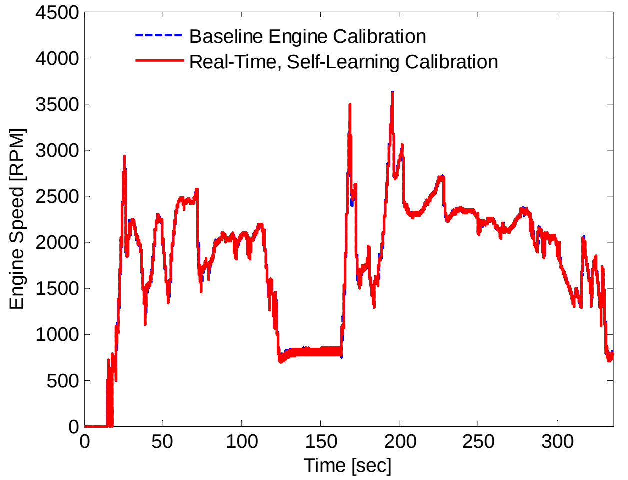


**Figure 3.** Engine speed.

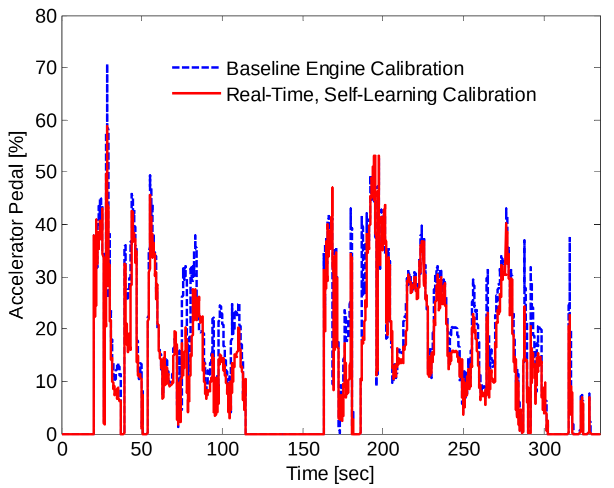


**Figure 4.** Gas-pedal position rate representing a driver's driving style.

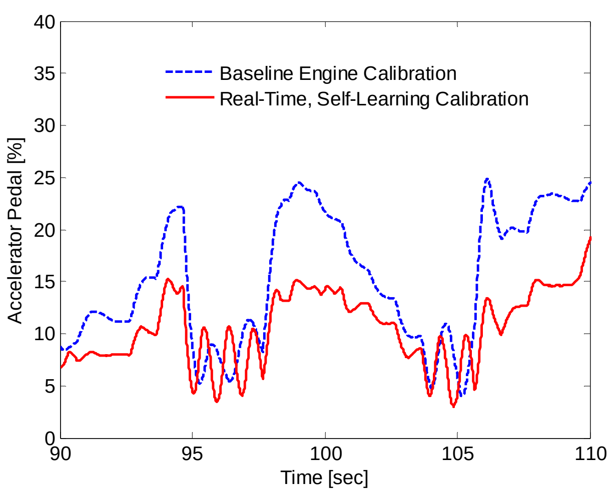


**Figure 5.** Gas-pedal position rate representing a driver's driving style (zoom-in).

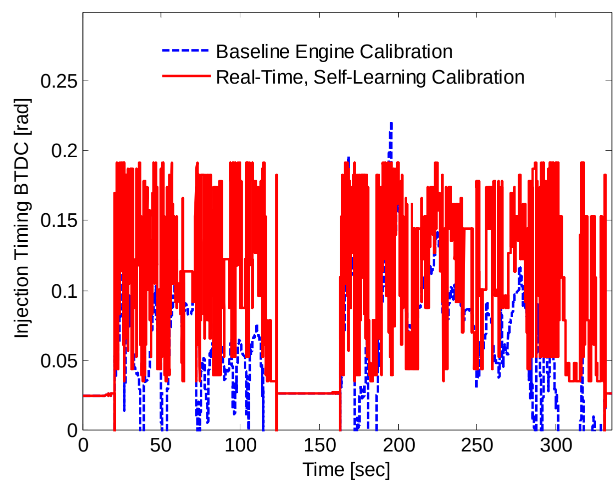


**Figure 6.** Injection timing.

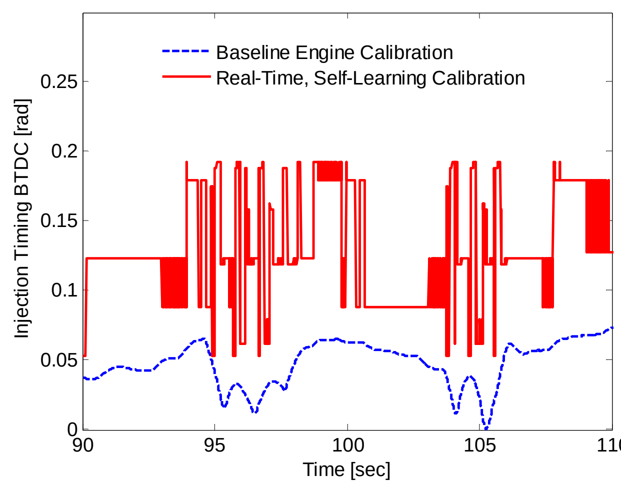


**Figure 7.** Injection timing (zoom-in).

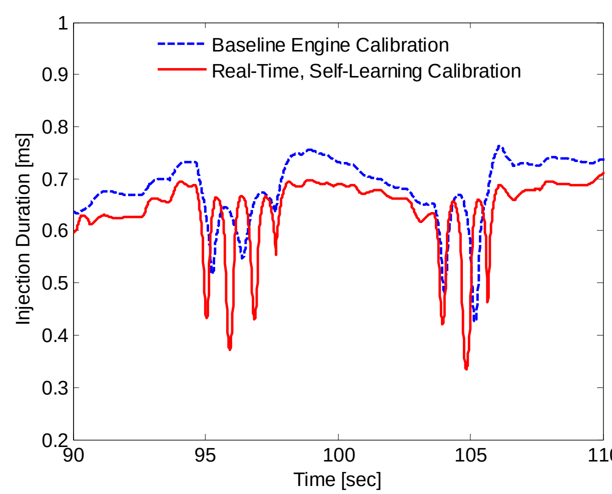


**Figure 8.** Fuel mass injection duration (zoom-in).

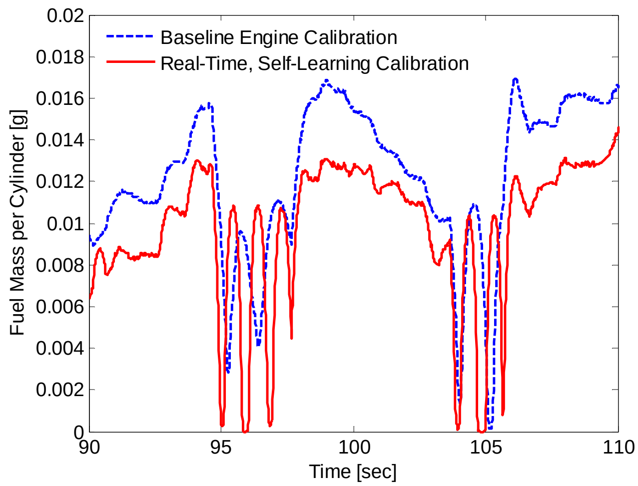


**Figure 9.** Fuel mass injected per cylinder (zoom-in).

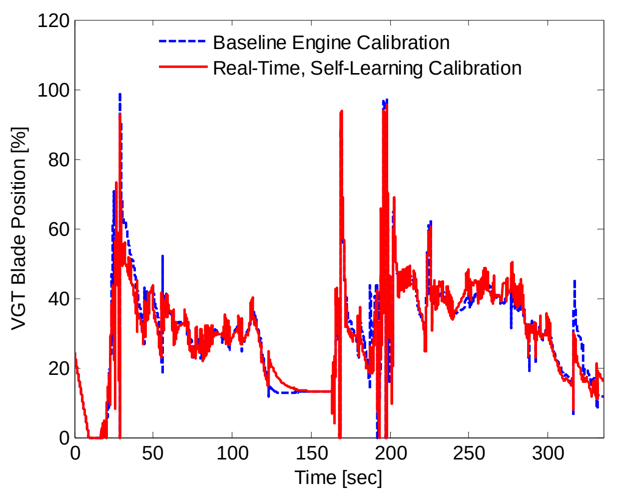


**Figure 10.** VGT vane position.

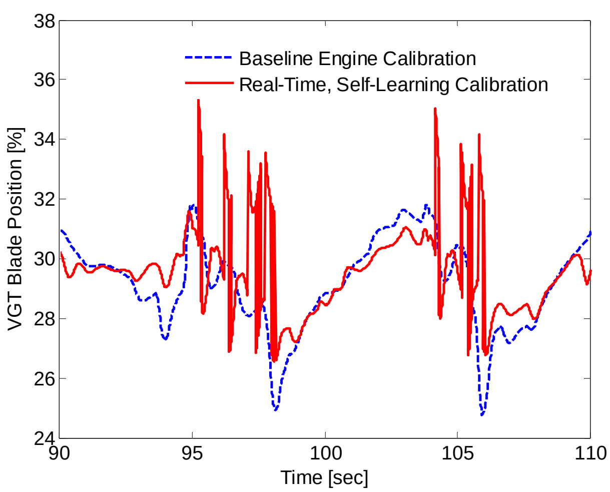


**Figure 11.** VGT vane position (zoom-in).

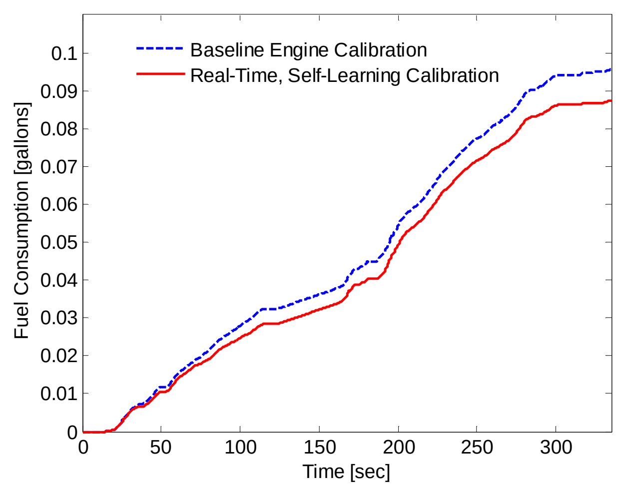


**Figure 12.** Fuel consumption for the driving cycle.

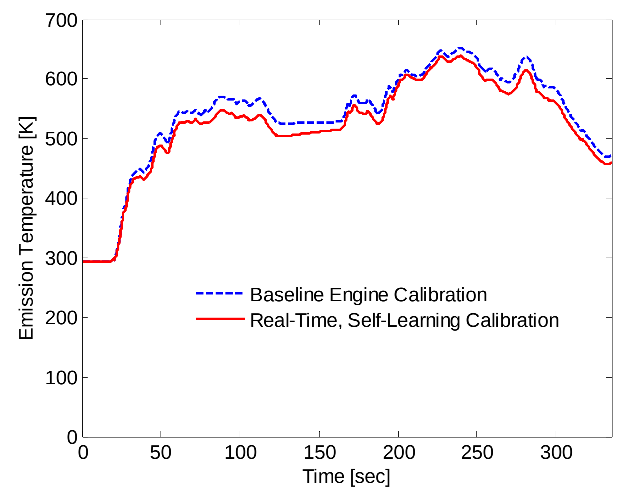


**Figure 13.** Emission temperature in the exhaust manifold.

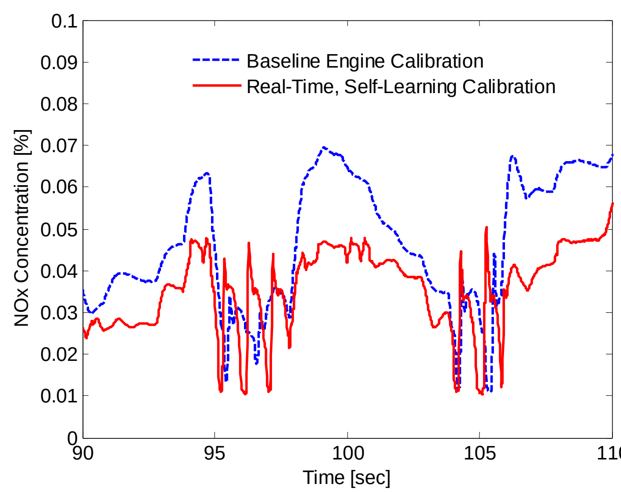


**Figure 14.** NOx concentration of emissions (zoom-in).

## 6. CONCLUDING REMARKS

This paper has presented a decentralized learning method suited to cope with the problem of dimensionality emerged in real-time, self-learning optimization of engine calibration with respect to more than one controllable variable. The method establishes a learning process that enables the derivation of the optimal values of the controllable variables to occur in parallel phases. The derivation of optimal values for more than one controllable variable can be achieved while keeping the problem's dimensionality low. The example presented an application of this method in real-time, self learning calibration of a diesel engine with respect to injection timing and VGT vane position. The engine was able to realize the optimal values of injection timing and VGT for a driving style represented by a segment of the FTP-75 driving cycle, and thus, optimizing fuel economy. Future research should validate this method to more than two controllable variables and the implications in the learning time.

Pre-specifying the entire transient engine operation as imposed by different driving cycles is impractical, and thus, optimal calibration for transient engine operation cannot be implemented *a priori*. The proposed method in conjunction with the previous developed learning algorithm can guarantee optimal calibration for steady-state and transient engine operating points resulting from the driver's driving style. This capability can be valuable in engines utilized in hybrid-electric powertrain configurations when real-time optimization of the power management is considered.

## ACKNOWLEDGMENTS

This research was partially supported by the Automotive Research Center (ARC), a U.S. Army Center of Excellence in Modeling and Simulation of Ground Vehicles at the University of Michigan. The engine simulation package enDYNA Themos CRTD was provided by TESIS DYNAware GmbH. This support is gratefully acknowledged.

## REFERENCES

[1] Brooks, T., Lumsden, G., and H.Blaxill, "Improving Base Engine Calibrations for Diesel Vehicles Through the Use of DoE and Optimization Techniques," Powertrain and Fluid Systems Conference and Exhibition, San Antonio, Texas, USA, September 24-27, 2005, SAE 2005-01-3833.

[2] Clarke, G. M. and Kempson, R. E., *Introduction to the Design and Analysis of Experiments*, Hodder Arnold, November 1996.

[3] Diamond, W. J., *Practical Experiment Designs : for Engineers and Scientists*, 3rd edition, John Wiley & Sons, February 2001.

[4] Hicks, C. R. and Turner, K. V., *Fundamental Concepts in the Design of Experiments*, 5th edition, Oxford University Press, USA, March 1999.

[5] Knafl, A., Hagena, J. R., Filipi, Z. S., and Assanis, D. N., "Dual-Use Engine Calibration: Leveraging Modern Techonologies to Improve Performance-Emissions Tradeoff," SAE Word Congress, Detroit, Michigan, April 11-14, 2005, SAE 2005-01-1549.

[6] Montgomery, D. T. and Reitz, R. D., "Optimization of Heavy-Duty Diesel Engine Operating Parameters Using A Response Surface Method," SAE Transactions-Journal of Fuels and Lubricants, v. 109, 2000, SAE 2000-01-1962.

[7] Brahma, I., He, Y., and Rutland, C. J., "Improvement of Neural Network Accuracy for Engine Simulations," Powertrain and Fluid Systems Conference and Exhibition, Pittsburgh, Pennsylvania, September 27-30, 2003, SAE 2003-01-3227.

[8] Lowe, D. and Zapart, K., "Validation of Neural Networks in Automotive Engine Calibration," Proceedings of the Fifth International Conference on Artificial Neural Networks, pp. 221-6, Cambridge, UK, 1997,

[9] Meyer, S. and Greff, A., "New Calibration Methods and Control Systems with Artificial Neural Networks," SAE 2002 World Congress, Detroit, Michigan, March 4-7, 2002, SAE 2002-01-1147.

[10] Papalambros, P. Y. and Wilde, D. J., *Principles of Optimal Design: Modeling and Computation*, 2nd edition, Cambridge University Press, July 2000.

[11] Guerrier, M. and Cawsey, P., "The Development of Model-Based Methodologies for Gasoline IC Engine Calibration," SAE Transactions-Journal of Engines, v. 113, 2004, SAE 2004-01-1466.

[12] Rask, E. and Sellnau, M., "Simulation-Based Engine Calibration: Tools, Techniques, and Applications," SAE Transactions-Journal of Engines, v. 113, 2004, SAE 2004-01-1264.

[13] Roepke, K. and Fischer, M., "Efficient Layout and Calibration of Variable Valve Trains," SAE Transactions-Journal of Engines, v. 110, 2001, SAE 2001-01-0668.

[14] Stuhler, H., Kruse, T., Stuber, A., Gschweitl, K., Piock, W., Pfluegl, H., and Lick, P., "Automated Model-Based GDI Engine Calibration Adaptive Online DoE Approach," SAE 2002 World Congress, Detroit, Michigan, March 3-7, 2002, SAE 2002-01-0708.

[15] Wu, B., Prucka, R. G., Filipi, Z. S., Kramer, D. M., and Ohl, G. L., "Cam-Phasing Optimization Using Artificial Neural Networks as Surrogate Models~Maximizing Torque Output," SAE Transactions-Journal of Engines, v. 114, 2005, SAE 2005-01-3757.

[16] Atkinson, C. and Mott, G., "Dynamic Model-Based Calibration Optimization: An Introduction and Application to Diesel Engines," SAE World Congress, Detroit, Michigan, April 11-14, 2005, SAE 2005-01-0026.

[17] Rakopoulos, C. D., Giakoumis, E. G., Hountalas, D. T., and Rakopoulos, D. C., "The Effect of Various Dynamic, Thermodynamic and Design Parameters on the Performance of a Turbocharged Diesel Engine Operating under Transient Load Conditions," SAE 2004 World Congress and Exhibition, Detroit, Michigan, April 8-11, 2004, SAE 2004-01-0926.

 

[18] Wijetunge, R. S., Brace, C. J., Hawley, J. G., Vaughan, N. D., Horroocks, R. W., and Bird, G. L., "Dynamic Behavior of a High-Speed, Direct-Injection Diesel Engine," SAE Transactions-Journal of Engines, v. 108, 1999, SAE 1999-01-0829.
[19] Samulski, M. J. and Jackson, C. C., "Effects of Steady-State and Transient Operation on Exhaust Emissions from Nonroad and Highway Diesel Engines," SAE Transactions-Journal of Engines, v. 107, 1998, SAE 982044.
[20] Green, R. M., "Measuring the Cylinder-to-Cylinder EGR Distribution in the Intake of a Diesel Engine During Transient Operation," SAE Transactions-Journal of Engines, v. 109, 2000, SAE 2000-01-2866.
[21] Malikopoulos, A. A., Papalambros, P. Y., and Assanis, D. N., "A Learning Algorithm for Optimal Internal Combustion Engine Calibration in Real Time," Proceedings of the ASME 2007 International Design Engineering Technical Conferences Computers and Information in Engineering Conference, Las Vegas, Nevada, September 4-7, 2007, DETC2007-34718.
[22] Puterman, M. L., *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 2nd Rev. edition, Wiley-Interscience, 2005.
[23] Sennott, L. I., *Stochastic Dynamic Programming and the Control of Queueing Systems*, 1st edition, Wiley-Interscience, September 1998.
[24] Bertsekas, D. P., *Dynamic Programming and Optimal Control (Volumes 1 and 2)*, Athena Scientific, September 2001.
[25] Bertsekas, D. P. and Shreve, S. E., *Stochastic Optimal Control: The Discrete-Time Case*, 1st edition, Athena Scientific, February 2007.
[26] Bertsekas, D. P. and Tsitsiklis, J. N., *Neuro-Dynamic Programming (Optimization and Neural Computation Series, 3)*, 1st edition, Athena Scientific, May 1996.
[27] Sutton, R. S. and Barto, A. G., *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*, The MIT Press, March 1998.
[28] Borkar, V. S., "A Learning Algorithm for Discrete-Time Stochastic Control," *Probability in the Engineering and Information Sciences*, vol. 14, pp. 243-258, 2000.
[29] Heywood, J., *Internal Combustion Engine Fundamentals*, 1st edition, McGraw-Hill, April 1988.
[30] TESIS, <http://www.tesis.de/en/>.