Real-Time Self-Learning Optimization of Diesel Engine Calibration

Andreas A. Malikopoulos
e-mail: amaliko@umich.edu

Dennis N. Assanis
e-mail: assanis@umich.edu

Panos Y. Papalambros
e-mail: pyp@umich.edu

Department of Mechanical Engineering,
University of Michigan,
Ann Arbor, MI 48109

Compression ignition engine technologies have been advanced in the past decade to provide superior fuel economy and high performance. These technologies offer increased opportunities for optimizing engine calibration. Current engine calibration methods rely on deriving static tabular relationships between a set of steady-state operating points and the corresponding values of the controllable variables. While the engine is running, these values are being interpolated for each engine operating point to coordinate optimal performance criteria, e.g., fuel economy, emissions, and acceleration. These methods, however, are not efficient in capturing transient engine operation designated by common driving habits, e.g., stop-and-go driving, rapid acceleration, and braking. An alternative approach was developed recently, which makes the engine an autonomous intelligent system, namely, one capable of learning its optimal calibration for both steady-state and transient operating points in real time. Through 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 criteria. The major challenge to this approach is problem dimensionality when more than one controllable variable is considered. In this paper, we address this problem by proposing a decentralized learning control scheme. The scheme is evaluated through simulation of a diesel engine model, which learns the values of injection timing and variable geometry turbocharging vane position that optimize fuel economy and pollutant emissions over a segment of the FTP-75 driving cycle. [DOI: 10.1115/1.3019331]

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 have 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, and engine acceleration. Current engine calibration methods generate a static tabular relationship between the 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 relationships are being interpolated to provide the values of the controllable variables for each operating point.

These methods, however, seldom guarantee optimal engine calibration for the entire operating domain, especially during transient operation [1]. The latter often constitutes the largest segment of engine operation compared with the steady-state one. Fuel consumption and emissions during transient operation are extremely complicated and are highly dependent on engine calibration [2,3]. Research efforts in addressing transient operation have focused on simulation-based methods to derive calibration maps for transients of particular driving cycles. However, prespecifying 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 are not possible in practice, thus preventing a priori optimal calibration.

To address these issues, an alternative approach was implemented recently, which treats the engine as a controlled stochastic system and the engine operation as a Markov decision process (MDP) [4]. Engine calibration is formulated as a centralized decision-making problem under uncertainty. The predictive optimal stochastic control algorithm (POSCA) [5] was developed, allowing the engine to learn the values of the controllable variables in real time. 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 criteria. Consequently, optimal calibration is achieved for steady-state and transient engine operating points designated by 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 centralized decision-making approach is the increase in the problem's dimensionality when more than one controllable variable (decision maker) is considered. Decentralized decision making requiring limited information is a highly desirable feature in this situation. It is necessary when complete information among decision makers, which is required in centralized decision making, is impractical due to the increase in the problem's dimensionality. Mathematical learning theory has been developed in systems to address the modeling and control aspects of sequential decision making under uncertainty [6–8]. Learning automata have been applied to network routing in which decentralization is attractive and large uncertainties are present [9,10]. The resulting system performance has demonstrated that decentralized learning schemes can be successful, while the problem's dimensionality remains tractable.

Manuscript received March 18, 2008; final manuscript received October 5, 2008; published online December 19, 2008. Review conducted by Christopher J. Rutland. Paper presented at the 2007 Fall Conference of the ASME Internal Combustion Engine Division (ICEF2007), Oct. 14–17, 2007, Charleston, SC.

The problem of decentralized control for a class of large scale interconnected dynamic systems in continuous time domain was studied by Wu [11]. In this offline approach, it is assumed that the considered systems are linear time varying, and the interconnections between each subsystem are unknown. Szer and Charpillet [12] proposed a model-free distributed reinforcement learning (RL) algorithm that utilizes communication to improve learning among the decision makers in a Markov decision process formalism. Scherrer and Charpillet [13] developed a general iterative heuristic approach in which at each decision epoch the focus is on a subgroup of decision makers, and their policies, given the rest of the decision makers, have fixed plans. Beynier and Mouaddib [14] introduced the notion of expected opportunity cost to better assess the influence of a local decision of an agent on the others. An iterative version of the algorithm was implemented to incrementally improve the policies of agents leading to higher quality solutions in some settings. Yagan and Chen-Khong [15] implemented a model-free coordinated reinforcement learning for decentralized optimal control, assuming that each decision maker can partially observe the state condition. This decentralized scheme is suited for partially observable Markov decision processes. Shen et al. [16] developed a decentralized Markov game model to estimate the belief among the decision makers. In the proposed model, the model-free Q-learning algorithm was employed to adjust dynamically the payoff function of each player. Wheeler and Narenda [17] employed a game-theoretic approach and developed a decentralized learning control scheme in finite Markov chains with unknown transition probabilities and costs. In this scheme, the decision makers demonstrate a myopic behavior; namely, they are unaware of the surrounding world. In attempting to improve his/her performance, each decision maker selects a control action, observes the corresponding cost associated with the occupied state, and then updates the action. Although many of these algorithms address the decentralized learning problem, their use of the accumulated data acquired over the learning process is inefficient, and they require a significant amount of experience to achieve acceptable performance. This requirement arises due to the formation of these algorithms in deriving control policies without learning the system models en route. In addition, the requirement of real-time derivation of the values of the engine controllable variables imposes an additional computational burden in implementing such control schemes.

This paper proposes a decentralized learning control scheme that can utilize efficiently the accumulated data acquired from the engine output to achieve acceptable performance in a real-time implementation. The proposed scheme differs from Wheeler and Narenda's in the sense that the decision makers (engine controllable variables) do not demonstrate a myopic behavior explicitly. On the contrary, a random hierarchy among them is assumed, based on which each one observes the control actions of the other.

The remainder of this paper is organized as follows: In Sec. 2, the mathematical framework of modeling the engine operation as a MDP is reviewed. The decentralized learning control scheme is introduced in Sec. 3. The effectiveness of the method is demonstrated in Sec. 4 through simulation of a diesel engine calibration with respect to the injection timing and VGT vane position over a segment of the FTP-75 driving cycle. Conclusions are drawn in Sec. 5.

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 MDP. The engine performance criteria, 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 criteria (random functions). The problem of engine calibration is thus formulated as a centralized sequential decision-making problem under uncertainty.

The MDP [18] provides the mathematical framework for modeling such problems [19]. It comprises a decision maker (controller), 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, and maximizing engine acceleration). In this framework, the controller (decision maker) is faced with the problem of influencing engine operation over time by selecting optimal actions.

Following the exposition in Ref. [5], a discrete-time stochastic controlled MDP is defined as the tuple,

$$s_k = \{\boldsymbol{S}, \boldsymbol{A}, \mathbf{P}(\cdot,\cdot), \mathbf{R}(\cdot,\cdot)\} \tag{1}$$

where $\boldsymbol{S}=\{1,2,\ldots,N\}$, $N \in \mathbb{N}$ denotes a finite state space, $\boldsymbol{A} = \cup_{s_k \in \boldsymbol{S}} A(s_k)$ 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 decision epochs $k=0,1,2,\ldots,M$, $M \in \mathbb{N}$. At each epoch, the decision maker observes a system's state, $s_k=i \in \boldsymbol{S}$, and executes an action, $a_k \in A(s_k)$, from the feasible set of actions, $A(s_k) \subseteq \boldsymbol{A}$, at this state. At the next epoch, the system transits to the state $s_k = j \in \boldsymbol{S}$ imposed by the conditional probabilities $p(s_{k+1}=j|s_k = i,a_k)$, designated by the transition probability matrix $\mathbf{P}(\cdot,\cdot)$. The conditional probabilities of $\mathbf{P}(\cdot,\cdot)$, $p:\boldsymbol{S}\times\boldsymbol{A}\rightarrow[0,1]$, satisfy the constraint

$$\sum_{j=1}^{N} p(s_{k+1}=j|s_k=i,a_k) = 1 \tag{2}$$

Following this state transition, the decision maker receives a reward associated with the action a_k, $R(s_{k+1}=j|s_k=i,a_k)$, where $R:\boldsymbol{S}\times\boldsymbol{A}\rightarrow\mathbb{R}$, as imposed by the transition reward matrix $\mathbf{R}(\cdot,\cdot)$. The states of a MDP possess the Markov property, stating that the conditional probability distribution of future states of the process depends only on 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

$$p(s_{k+1}|s_k,s_{k-1},\ldots,s_0) = p(s_{k+1}|s_k) \tag{3}$$

The solution to a MDP can be expressed as a policy $\pi = \{\mu_0,\mu_1,\ldots,\mu_M\}$, which provides the action to be executed for a given state, regardless of prior history; μ_κ is a function mapping states s_k into actions $a_k=\mu_\kappa(s_k)$, such that $\mu_\kappa(s_k) \in A(s_k)$. Such policies are addressed as admissible. Consequently, for any initial state at decision epoch $k=0$, s_0, and for any finite sequence of epochs $k=0,1,2,\ldots,M$, $M \in \mathbb{N}$, the expected accumulated value of the rewards of the decision maker is given by

$$J^{\pi}(s_0) = \underset{\substack{s_k \in \boldsymbol{S} \\ a_k \in A(s_k)}}{E} \left\{ R_M(s_M) + \sum_{k=0}^{M-1} R_k(s_{k+1}=j|s_k=i,a_k) \right\}$$
$$\forall i,j \in \boldsymbol{S}, \quad \forall\, a_k \in A(s_k) \tag{4}$$

where $R_M(s_M)$ is the reward at the final state. In the finite-horizon context, the decision maker should maximize the accumulated value for the next M decision epochs; more precisely, an optimal policy π^* is one that maximizes the overall expected accumulated value of the rewards,

$$J^{\pi^*}(s_0) = \max_{\pi \in \boldsymbol{A}} J^{\pi}(s_0) \tag{5}$$

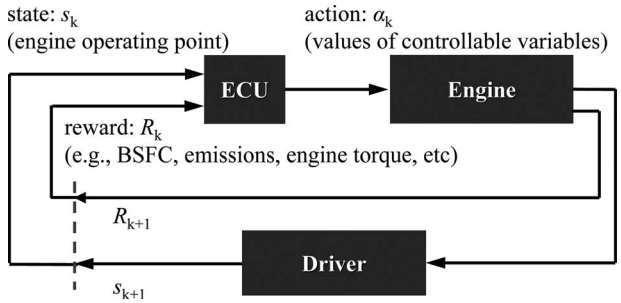

Fig. 1 Learning process during the interaction between the engine and the driver

Consequently, the optimal policy $\pi^*=\{\mu_0^*,\mu_1^*,\ldots,\mu_M^*\}$ sequence is given by

$$\pi^* = \arg\max_{\pi\in A} J^{\pi}(s_0) \tag{6}$$

Dynamic programming (DP) has been widely used for many years as the principal method for solving Markov decision problems [20]. 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 RL [21,22]. RL has aimed to provide simulation-based algorithms for learning control policies of complex systems, where exact modeling is infeasible or expensive [23]. 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 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 π 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 [24].

For the engine calibration problem built upon the MDP theoretical framework, the POSCA is employed [5]. POSCA is intended to derive the control policy (values of the engine controllable variables) for any initial state (engine operating point). In applying this algorithm to more than one controllable variable, however, 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 $\boldsymbol{A}$. To overcome this problem, the decentralized learning control scheme is implemented.

3 Decentralized Learning Control Scheme

While the engine is running the vehicle and interacting with the driver, the probability of engine operating point transitions designates the elements of the transition probability matrix, $\mathbf{P}(\cdot,\cdot)$. The desired engine performance criteria, e.g., fuel economy and pollutant emissions, are represented by the elements of the transition reward matrix, $\mathbf{R}(\cdot,\cdot)$. Through this interaction, the driver introduces a state $s_k \in \boldsymbol{S}$ (engine operating point) to the engine's ECU, and on that basis the ECU selects an action, $a_k \in A(s_k)$ (combination of values of the controllable variables). As a consequence of its action, the ECU receives a numerical reward, $R_{k+1} \in \mathbb{R}$, and the engine transits to a new state, $s_{k+1} \in \boldsymbol{S}$, as illustrated in Fig. 1. POSCA aims to compute the control policy (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 space $\boldsymbol{A}=\cup_{s_k\in\boldsymbol{S}}A(s_k)$.

The decentralized learning control scheme proposed in this paper establishes a learning process that enables the derivation of the values of the controllable variables to occur in parallel phases [25]. A random hierarchy among them is assumed, based on which each one observes the control actions of the other. In particular, POSCA is employed to derive the control actions of the first member in the hierarchy of decision makers with respect to the sequence of state transitions. At the same time, the algorithm is engaged separately to derive the control actions of the second member in the hierarchy of decision makers with respect to the policy as learned from the first one. Similarly, the algorithm is employed to derive the control actions of the third decision maker with respect to the second one, and so forth.

For instance, in implementing a diesel engine calibration with respect to the injection timing, α, and VGT vane position, β, a feasible set of values, $\boldsymbol{A}$ and $\boldsymbol{B}$, 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., $\boldsymbol{S}\times\boldsymbol{A}\rightarrow\mathbb{R}$. In the second, VGT is mapped to the injection timing, i.e., $\boldsymbol{A}\times\boldsymbol{B}\rightarrow\mathbb{R}$. The learning algorithm utilizes these two mappings to derive the control policies, $\pi_\alpha^*\in\boldsymbol{A}$ and $\pi_\beta^*\in\mathbf{B}$ (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 decision epoch k in conjunction with the injection timing $a_k\in\boldsymbol{A}$ taken for each state $s_k=i\in\boldsymbol{S}$ and the VGT vane position $\beta_k\in\boldsymbol{B}$ for each $a_k\in\boldsymbol{A}$. At the early epochs, and until full exploration of the feasible sets $\boldsymbol{A}$ and $\boldsymbol{B}$ occurs, the mapping from states to probabilities of selecting a particular value of injection timing $a_k\in\boldsymbol{A}$ and the mapping from $a_k\in\boldsymbol{A}$ to probabilities of selecting VGT $\beta_k\in\boldsymbol{B}$ are constant; namely, the values of each controllable variable are selected randomly with the same probability,

$$p(a_k|s_k)=\frac{1}{|\boldsymbol{A}|},\quad \forall\, a_k\in\boldsymbol{A},\quad \forall\, s_k\in\boldsymbol{S} \tag{7}$$

and

$$p(\beta_k|a_k)=\frac{1}{|\boldsymbol{B}|},\quad \forall\, a_k\in\boldsymbol{A},\quad \forall\, \beta_k\in\boldsymbol{B} \tag{8}$$

Exploration of the entire feasible set for each variable is important to evade suboptimal solutions. POSCA is thus used after the exploration phase to realize the policies π_α^*, and π_β^* by means of the expected values of the rewards, $V(s_{k+1}|s_k,a_k)$ and $V(a_{k+1}|a_k,\beta_k)$, generated by the mappings $\boldsymbol{S}\times\boldsymbol{A}\rightarrow\mathbb{R}$ and $\boldsymbol{A}\times\boldsymbol{B}\rightarrow\mathbb{R}$, respectively. The expected values of the rewards are defined to be

$$\begin{aligned} V(s_{k+1}=j|s_k=i,a_k) :=\ & p(s_{k+1}=j|s_k=i,a_k)\cdot R(s_{k+1}=j|s_k=i,a_k) \\ & + \max_{a_{k+1}\in\boldsymbol{A}}\left[\sum_{l=1}^{N} p(s_{k+2}=l|s_{k+1}=j,a_{k+1})\right. \\ & \left.\cdot\, R(s_{k+1}=l|s_k=j,a_{k+1})\right], \\ & i,j=1,2,\ldots,N,\quad N=|\boldsymbol{S}| \end{aligned} \tag{9}$$

and

$$V(a_{k+1}=m|a_k=n,\beta_k) := p(a_{k+1}=m|a_k=n,\beta_k)\cdot R(a_{k+1}=m|a_k=n,\beta_k) + \max_{\beta_{k+1}\in \boldsymbol{B}}\left[\sum_{p=1}^{\Lambda} p(a_{k+2}=p|a_{k+1}=m,\beta_{k+1}) \cdot R(a_{k+2}=p|a_{k+1}=m,\beta_{k+1})\right], \quad m,n=1,2,\ldots,\Lambda, \quad \Lambda=|\boldsymbol{A}| \tag{10}$$

In deriving the control 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 control policies π^*_α, and π^*_β are computed by utilizing the max-min control approach, whereby the worst possible values of the uncertain quantities within the given set are assumed to occur. This essentially ensures that the control policies will result in at least a minimum overall reward value. Consequently, at state $s_k=i$, POSCA computes the control policy π^*_α in terms of the values of injection timing α as

$$\pi^*_\alpha(s_k) = \underset{\bar{\mu}_k(s_k)\in A(s_k)}{\arg\max} \min_{s_{k+1}\in \boldsymbol{S}} [V(s_{k+1}=j|s_k=i,a_k)], \quad \forall\, i,j \in \boldsymbol{S} \tag{11}$$

For the control policy π^*_α, POSCA computes the control policy π^*_β of the values of the VGT vane position β as

$$\pi^*_\beta(a_k) = \underset{\beta_k(a_k)\in \boldsymbol{B}(a_k)}{\arg\max} \min_{a_{k+1}\in \boldsymbol{A}} [V(a_{k+1}=m|a_k=n,\beta_k)], \quad \forall\, m,n \in \boldsymbol{A} \tag{12}$$

Employing this decentralized learning control scheme, derivation of the values of more than one controllable variable can be achieved, while the problem's dimensionality remains tractable.

4 Real-Time Self-Learning Injection Timing and VGT in a Diesel Engine

The decentralized learning control scheme introduced in the previous section is applied here to a four-cylinder, 1.9 l turbocharged diesel engine. The objective is to find the injection timing and VGT vane position while the engine is running the vehicle that maximize the engine brake torque. Consequently, the controller's inputs are the engine operating points and brake torque, while the outputs are the injection timing and VGT vane position. Injection timing is an important controllable variable in the combustion process and affects performance and emissions [26]. The major objective of injection timing is to initiate 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 to 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 [27] 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 ECU. In the example, the existing static correlation involving injection timing and VGT is bypassed to incorporate the decentralized learning control scheme 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 Fig. 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 the value of 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).

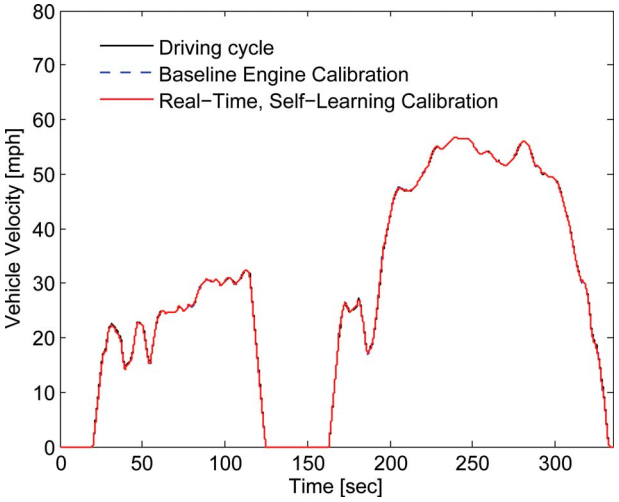

Fig. 2 Segment of the FTP-75 driving cycle

After completing the learning process, the decentralized control scheme specified the 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 Figs. 3–5. The implication is that the derived policy of injection timing and VGT resulted in higher engine torque compared with the baseline calibration. The injection timing (before top dead center (BTDC)) for both vehicles is illustrated in Figs. 6 and 7. While the baseline calibration interpolates values of the injection timing of steady-state operating points, the injection timing derived by the decentralized scheme 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. Drivability issues that may be raised in such a noisy injection timing response could be addressed by tightening the allowable space of two successive control actions (injection timing values). Lower gas-pedal position rates resulted in reducing the fuel mass injection duration, shown in Fig. 8, and consequently, less fuel mass was injected into the cylinders, as illustrated in Fig. 9 (in

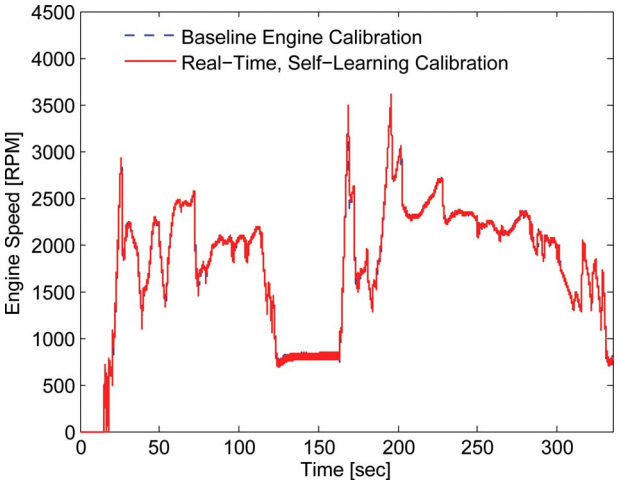

Fig. 3 Engine speed

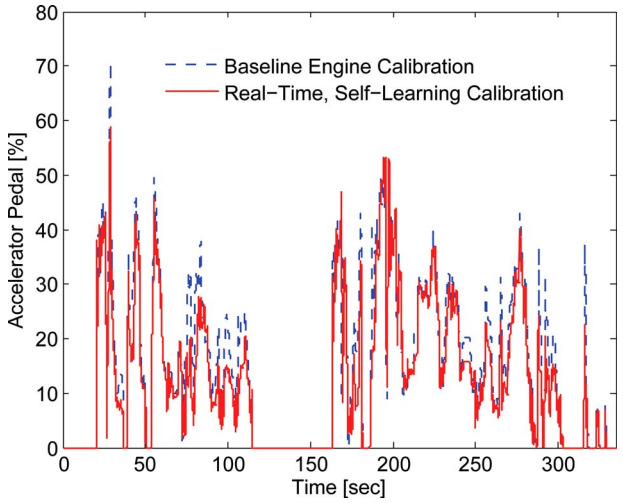

Fig. 4 Gas-pedal position rate representing a driver's driving style

zoom-in for clarity). In the decentralized learning of the engine, the injection timing was mapped to the engine operating points

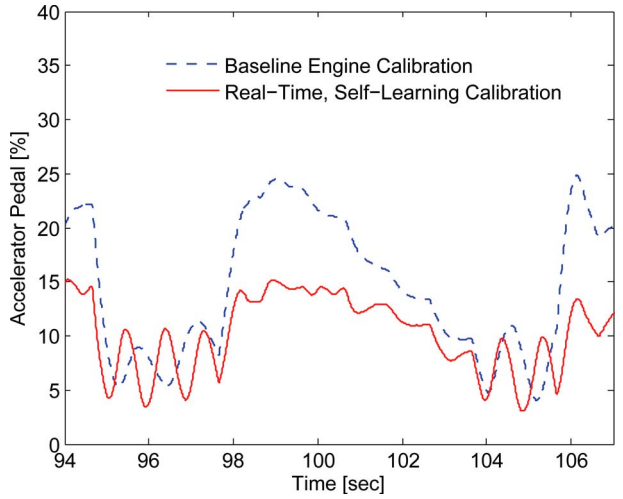

Fig. 5 Gas-pedal position rate representing a driver's driving style (zoom-in)

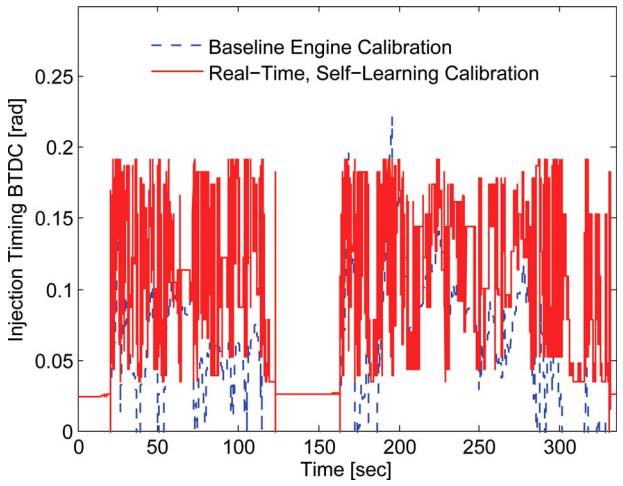

Fig. 6 Injection timing

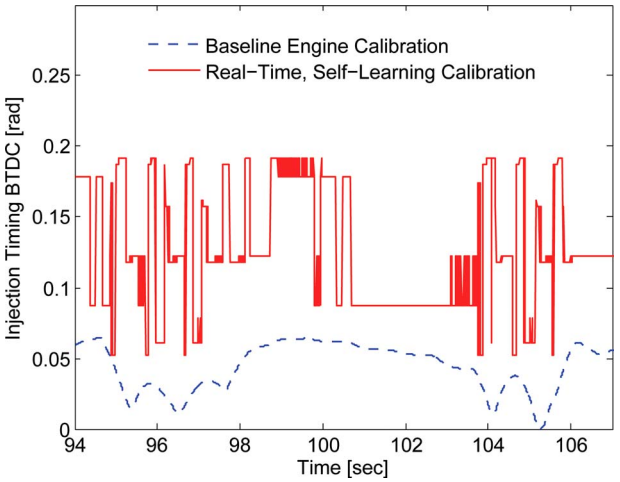

Fig. 7 Injection timing (zoom-in)

(states), while the VGT vane position was mapped to the injection timing. The derived VGT policy is illustrated in Figs. 10 and 11. Having the engine operate at the maximum brake torque, a 9.3% overall improvement of fuel economy was accomplished, as illustrated in Fig. 12, compared with the baseline calibration.

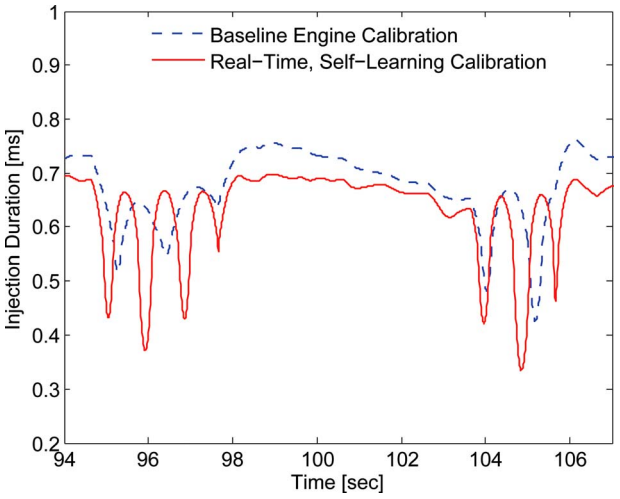

Fig. 8 Fuel mass injection duration (zoom-in)

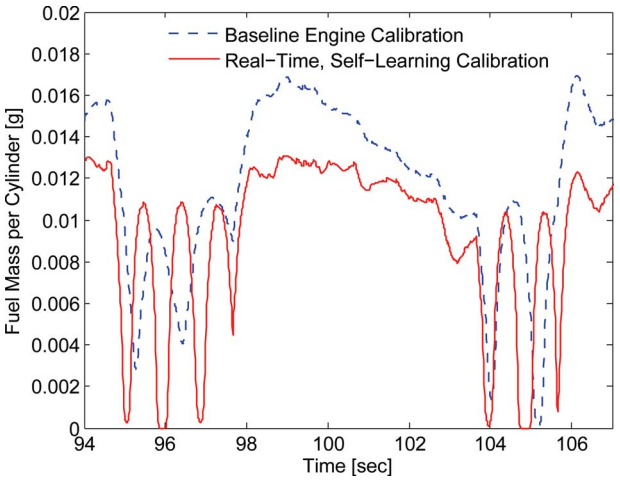

Fig. 9 Fuel mass injected per cylinder (zoom-in)

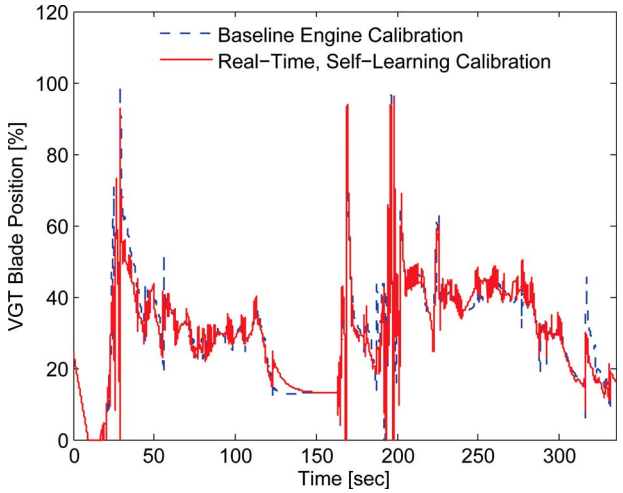

Fig. 10 VGT vane position

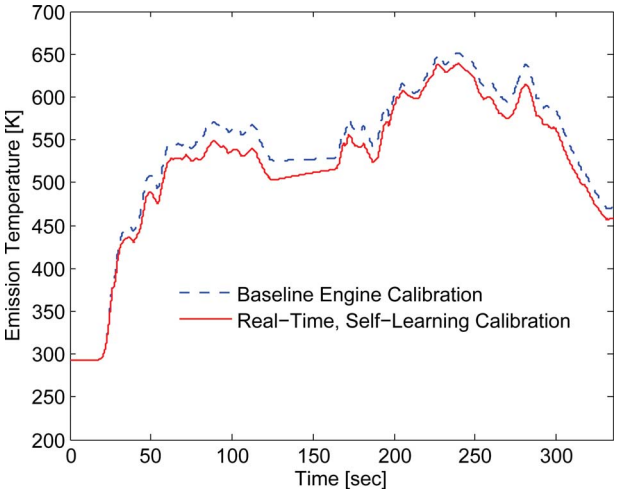

Fig. 13 Emission temperature in the exhaust manifold

At constant engine speed and constant fuel mass per cycle, for a given injection duration, and at fixed brake mean effective pressure (BMEP), if the injection timing is advanced from the MBT, then brake specific fuel consumption (BSFC) is decreased and NO_x emissions are increased. In our case, however, fuel mass per cycle, injection duration, and BMEP all vary, and thus, this behavior is altered. Since the injection duration is decreased, ignition delays decrease as well. Less time is available for fuel-air mixing, resulting in a less intense premixed burn and a lower premixed spike in the heat release rate curve. Cylinder bulk gas temperatures are therefore decreased, and so is NO_x production, as illustrated in Figs. 13 and 14.

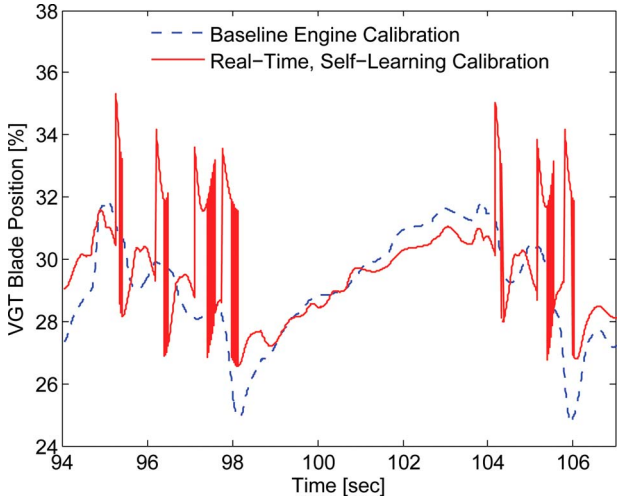

Fig. 11 VGT vane position (zoom-in)

5 Concluding Remarks

This paper presented a decentralized learning control scheme that addresses the problem dimensionality in the centralized decision-making approach as employed in making the engine into an autonomous intelligent system. In this scheme, a learning process is established, which enables the derivation of the values of the controllable variables to occur in parallel phases. The values for more than one controllable variable can thus be determined while keeping the problem's dimensionality tractable. The example presented an application of this scheme to a diesel engine with respect to injection timing and VGT vane position. The engine was able to realize the values of injection timing and VGT for a driving style represented by a segment of the FTP-75 driving cycle. Future research should validate this method for more than two controllable variables and the implications for the required learning time.

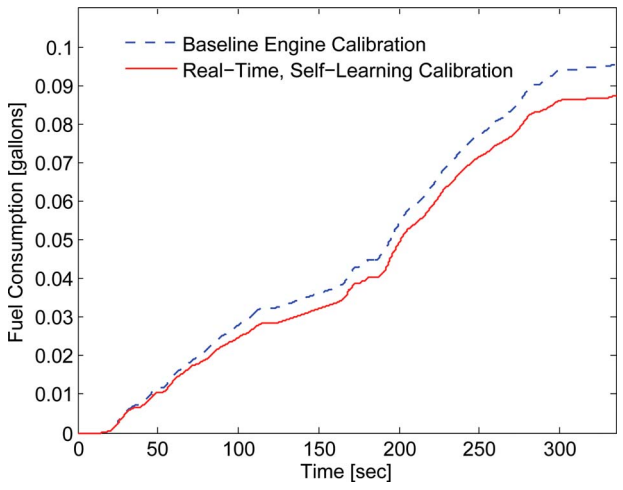

Fig. 12 Fuel consumption for the driving cycle

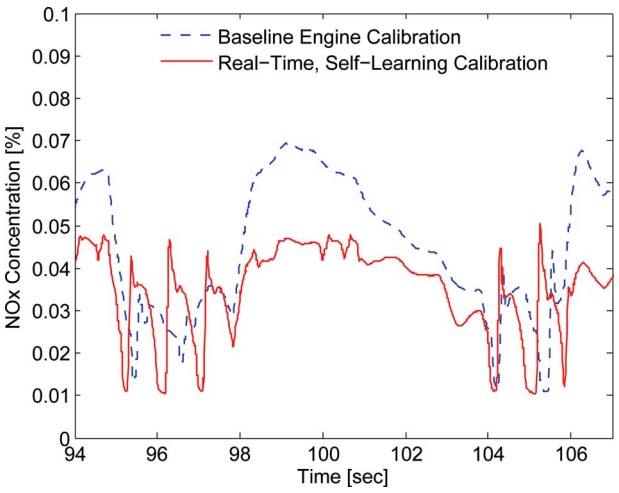

Fig. 14 NO_x concentration of emissions (zoom-in)

The decentralized control scheme, in conjunction with POSCA, can guarantee optimal calibration for steady-state and transient engine operating points designated by the driver's driving style. The ultimate goal of this approach is to fully exploit the engine's given technology in terms of the optimum specified performance criteria, e.g., fuel economy, pollutant emissions, and engine acceleration, that can be achieved. It aims to provide an answer to the following question: "For an engine with a given technology, what are the optimum performance criteria that a driver can get with respect to his/her driving habits?"

The long-term potential benefits of this approach are substantial. True fuel economy of vehicles will be increased while meeting emission standard regulations; drivers will be able to evaluate their driving behavior and learn how to improve fuel economy and reduce emissions by modifying it. This capability can be also especially appealing in engines utilized in hybrid-electric powertrain systems when real-time optimization of the power management is considered.

Acknowledgment

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] Atkinson, C., and Mott, G., 2005, "Dynamic Model-Based Calibration Optimization: An Introduction and Application to Diesel Engines," SAE World Congress, Detroit, MI, Apr. 11–14, Paper No. SAE 2005-01-0026.

[2] Samulski, M. J., and Jackson, C. C., 1998, "Effects of Steady-State and Transient Operation on Exhaust Emissions From Nonroad and Highway Diesel Engines," SAE Transactions-Journal of Engines, Vol. 107.

[3] Green, R. M., 2000, "Measuring the Cylinder-to-Cylinder EGR Distribution in the Intake of a Diesel Engine During Transient Operation," SAE Transactions-Journal of Engines, Vol. 109.

[4] Malikopoulos, A. A., 2008, "Real-Time, Self-Learning Identification and Stochastic Optimal Control of Advanced Powertrain Systems," Ph.D. thesis, Department of Mechanical Engineering, University of Michigan, Ann Arbor.

[5] Malikopoulos, A. A., Papalambros, P. Y., and Assanis, D. N., 2007, "A Learning Algorithm for Optimal Internal Combustion Engine Calibration in Real Time," *Proceedings of the ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Las Vegas, NV, Sept. 4–7.

[6] Bush, R. R., and Mosteler, F., 1958, *Stochastic Models for Learning*, Wiley, New York.

[7] Atkinson, R. C., Bower, G. H., and Crothers, E. J., 1965, *An Introduction To Mathematical Learning Theory*, Wiley, New York.

[8] Tsypkin, Y. Z., 1971, *Adaptation and Learning in Automatic Systems*, Academic, New York.

[9] Narendra, K. S., and Wheeler, R. M., Jr., 1983, "N-Player Sequential Stochastic Game With Identical Payoffs," IEEE Trans. Syst. Man Cybern., **13**, pp. 1154–1158.

[10] Srikantakumar, P. R., and Narendra, K. S., 1982, "A Learning Model for Routing in Telephone Networks," SIAM J. Control Optim., **20**, pp. 34–57.

[11] Wu, H., 2007, "Decentralized Iterative Learning Control for a Class of Large Scale Interconnected Dynamical Systems," J. Math. Anal. Appl., **327**, pp. 233–245.

[12] Szer, D., and Charpillet, F., 2004, "Improving Coordination With Communication in Multi-Agent Reinforcement Learning," *Proceedings of the 16th IEEE International Conference on Tools With Artificial Intelligence, ICTAI 2004*, Boca Raton, FL, pp. 436–440.

[13] Scherrer, B., and Charpillet, F., 2002, "Cooperative Co-Learning: A Model-Based Approach for Solving Multi-Agent Reinforcement Problems," *Proceedings of the 14th IEEE International Conference on Tools With Artificial Intelligence*, Washington, DC, pp. 463–468.

[14] Beynier, A., and Mouaddib, A.-I., 2006, "An Iterative Algorithm for Solving Constrained Decentralized Markov Decision Processes," *Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference*, Boston, MA, Paper No. AAAI-06/IAAI-06, pp. 1089–1094.

[15] Yagan, D., and Chen-Khong, T., 2007, "Coordinated Reinforcement Learning for Decentralized Optimal Control," *Proceedings of the 2007 First IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, Honolulu, HI (IEEE Catalog No. 07EX1572), pp. 296–302.

[16] Shen, D., Chen, G., Cruz, J. B., Jr., Kwan, C., and Kruger, M., 2007, "An Adaptive Markov Game Model for Threat Intent Inference," *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, pp. 1–13.

[17] Wheeler, R., and Narenda, K., 1986, "Decentralized Learning in Finite Markov Chains," IEEE Trans. Autom. Control, **31**(6), pp. 519–526.

[18] Puterman, M. L., 2005, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 2nd revised ed., Wiley-Interscience, New York.

[19] Sennott, L. I., 1998, *Stochastic Dynamic Programming and the Control of Queueing Systems*, 1st ed., Wiley-Interscience, New York.

[20] Bertsekas, D. P., and Shreve, S. E., 2007, *Stochastic Optimal Control: The Discrete-Time Case*, 1st ed., Athena Scientific, Nashua, NH.

[21] Bertsekas, D. P., and Tsitsiklis, J. N., 1996, *Neuro-Dynamic Programming* (Optimization and Neural Computation Series 3), 1st ed., Athena Scientific, Nashua, NH.

[22] Sutton, R. S., and Barto, A. G., 1998, *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*, MIT, Cambridge, MA.

[23] Borkar, V. S., 2000, "A Learning Algorithm for Discrete-Time Stochastic Control," Probability in the Engineering and Information Science, **14**, pp. 243–258.

[24] Malikopoulos, A. A., Papalambros, P. Y., and Assanis, D. N., 2007, "A State-Space Representation Model and Learning Algorithm for Real-Time Decision-Making Under Uncertainty," *Proceedings of the 2007 ASME International Mechanical Engineering Congress and Exposition*, Seattle, WA, Nov. 11–15.

[25] Malikopoulos, A. A., Assanis, D. N., and Papalambros, P. Y., 2007, "Real-Time, Self-Learning Optimization of Diesel Engine Calibration," *Proceedings of the 2007 Fall Technical Conference of the ASME Internal Combustion Engine Division*, Charleston, SC, Oct. 14–17.

[26] Heywood, J., 1988, *Internal Combustion Engine Fundamentals*, 1st ed., McGraw-Hill, New York.

[27] TESIS, http://www.tesis.de/en/