



Dilute Combustion Control Using Spiking Neural Networks

Bryan P. Maldonado, Brian C. Kaul, Catherine D. Schuman, Steven R. Young, and J. Parker Mitchell

Oak Ridge National Laboratory

Citation: Maldonado, B.P., Kaul, B.C., Schuman, C.D., Young, S.R., et al., "Dilute Combustion Control Using Spiking Neural Networks," SAE Technical Paper 2021-01-0534, 2021, doi:10.4271/2021-01-0534.

Abstract

Dilute combustion with exhaust gas recirculation (EGR) in spark-ignition engines presents a cost-effective method for achieving higher levels of engine efficiency. At high levels of EGR, however, cycle-to-cycle variability (CCV) of the combustion process is exacerbated by sporadic occurrences of misfires and partial burns. Previous studies have shown that temporal deterministic patterns emerge at such conditions and certain combustion cycles have a significant influence over future events. Due to the complexity of the combustion process and the nature of CCV, harnessing all the deterministic information for control purposes has remained challenging even with physics based 0-D, 1-D, and high-fidelity computational fluid dynamics (CFD) models. In this study, we present a data-driven approach to optimize the combustion process by controlling CCV adjusting the cycle-to-cycle fuel injection quantity. Readily available data from in-cylinder

pressure was used to train a spiking neural network (SNN) which learns the optimal way to manage fuel injection in order to reduce CCV while maintaining acceptable levels of fuel consumption. SNNs are particularly well suited for powertrain control applications due to their ability to be deployed on FPGA-based neuromorphic hardware which are small, inexpensive, and have a low power demand. The high-performance computing (HPC) resources of Oak Ridge National Laboratory were used to run an evolutionary-based training approach for choosing the best SNN configuration that minimizes the size of the network while achieving the desired goal. The neuromorphic hardware with the optimized SNN deployed was connected to the rapid prototyping engine control system for real-time control implementation and tested on a single cylinder version of a GM LNF 4-cylinder engine. The results show a significant reduction of CCV with a small percentage of additional fuel used to stabilize the charge.

Introduction

Charge dilution, through either exhaust gas recirculation (EGR) or excess air (lean operation), is an established path to increasing the efficiency of spark-ignition (SI) engines through both reduction of pumping losses and thermodynamic (γ) effects. Stoichiometric operation with EGR, in particular, allows for improved efficiency while maintaining compatibility with three-way catalyst (TWC) after-treatment systems to control criteria emissions. However, at high dilution levels, combustion stability is limited by poor flame initiation (misfires) [1] or propagation (partial burns) [2]. It has been shown that the cycle-to-cycle variations beyond the dilute limits for both lean and EGR dilution exhibit deterministic patterns along with stochastic variations [3,4,5,6,7]. Cycle-to-cycle spark timing control has been previously explored in order to prevent partial burns and misfires, but such strategies mainly focus on the stochastic properties observed during long periods of time [8,9]. The deterministic components of CCV are attributed to a nonlinear dependence of flame initiation and flame speed on composition, with feed-forward communication between cycles due to residual gas composition [10]. This presence of determinism implies that

active fuel injection quantity control to stabilize combustion at the dilute limit should be possible. Past experiments have verified that beyond the dilute limit, the sensitivity of combustion to changes in fueling increases significantly, such that only small control perturbations to the cycle fuel injection quantity are required to effect changes in the combustion event [11].

There were early demonstrations of CCV control for lean operation [12,13,14], though these were limited by the computational power of engine controllers at the time and the ability to accurately predict next-cycle combustion variations. In particular, while simple models have been able to produce similar dynamics to those observed in engine data [10], these are not predictive. On the other hand, more physically detailed computational fluid dynamics (CFD) models are too computationally expensive for real-time control applications [15]. Some attempts have been made to apply artificial neural networks (ANNs) in a "black-box" solution for both lean [16,17,18,19] and EGR [20,21,22,23] dilution. These efforts trained ANN predictors based on the simple model of Daw et al. [10], and coupled that with an ANN controller. While the predictors accurately learned the magnitude of CCV, they

did not fully capture the relevant cycle-to-cycle dynamics, limiting the effectiveness of the controllers.

More recently, the authors have further improved the Daw model by adding cycle-resolved dependence of residual gas fraction (in addition to composition) on the prior combustion event [24]. Initially, a parametric sigmoid combustion efficiency function was calibrated to match the dynamic response and the stochastic properties of the gross heat release [25]. The model was further improved by considering a hybrid physics-based/machine learning (ML) approach where a nonparametric kernel density estimator (KDE) was used to model the stochastic properties of combustion efficiency [26]. The latter model is able to fully capture the cycle-to-cycle dynamics while remaining computationally inexpensive and suitable for control applications.

In this paper, spiking neural networks (SNNs) were developed and trained based on this new hybrid model to implement next-cycle control of combustion variability for high-EGR operation. Spiking neural networks are a neural network type that takes more inspiration from biological brains than traditional artificial neural networks. In particular, they include a notion of time in how they process information, making them well-suited for processing temporal data or performing control tasks, which should offer an advantage over traditional ANNs for CCV control based on prior-cycle events. We have not compared with a traditional ANN approach on this task as that would require utilizing a non-traditional, reinforcement learning-based training approach. SNNs can also be very efficiently implemented in neuromorphic hardware, allowing for small, low-power hardware deployment. These networks were then deployed and evaluated experimentally for real-time combustion stability control.

The remaining sections of this paper are organized as follows. The experimental engine setup, including the rapid prototyping engine controller, is presented. The characteristics of combustion CCV are discussed for a variety of conditions with high levels of EGR. The control-oriented combustion model based on the hybrid physics-based/ML approach and used for offline simulations is presented. A brief introduction to SNNs and their applicability to powertrain control are discussed. The methodology for offline control design is given and the resulting SNNs candidates for online feedback control are presented. The deployment of the selected SNNs on a field-programmable gate array (FPGA) and its interaction with the engine controller is described. Finally, results of the system under closed-loop control over the cycle-to-cycle fuel injection quantity are presented.

Experimental Setup

The data presented in this study were collected from a single-cylinder version of a 2.0 L GM Ecotec LNF engine. The experimental engine is gasoline, spark-ignited, direct injection, with four valves per cylinder (two intake and two exhaust), and variable cam timing. The combustion chamber geometry and the operating conditions considered for this work are detailed in Table 1. The engine speed was kept constant at 1500 rpm with an engine dynamometer. Airflow was kept constant at a

TABLE 1 Engine characteristics and operating conditions.

Displaced Volume	500 cc
Stroke	86 mm
Bore	86 mm
Connecting Rod	145 mm
Compression Ratio	9.2:1
Intake Valve Closing (IVC)	-100 °aTDC
Exhaust Valve Opening (EVO)	150 °aTDC
Exhaust Valve Closing (EVC)	357 °aTDC
Engine Speed	1500 rpm
Global Equivalence Ratio	$\lambda = 1$ (open loop)
Start of Injection	-280 °aTDC
Indicated Load (IMEP)	7.7 bar
Coolant/Oil Temperature	70 °C
EGR Cooler Temperature	50 °C
Intake Manifold Temperature	35 °C

UT-Battelle, LLC.

medium load for all experimental conditions using a mass flow controller. A wide-band exhaust oxygen sensor was utilized to adjust the feedforward fuel quantity to maintain a stoichiometric charge. The engine was fitted with an external, cooled, low-pressure EGR system. The EGR molar fraction in the intake manifold was calculated using oxygen measurements reported by a wide-band intake oxygen sensor. The EGR mass fraction was calculated using the mole fraction and the molecular weights of air and EGR. In-cylinder pressure was measured using a flush-mounted piezoelectric pressure transducer from Kistler (6125C). Research-grade E10 gasoline known as RD5-87, which was designed to be representative of regular-grade market gasoline, was utilized. In addition, exhaust gas analyzers were utilized to measure the concentrations of O₂, CO, CO₂, NO_x, and total hydrocarbons (THC). Emissions data were mainly used to estimate the average combustion efficiency at each condition.

An open LabVIEW-based engine control system implemented on National Instruments Compact RIO (NI cRIO) hardware was used for engine controls, along with an in-house open, LabVIEW-based Oak Ridge Combustion Analysis System (ORCAS) for data acquisition and analysis. The ORCAS software was customized for this application to allow relevant combustion feedback parameters to be calculated immediately after EVO to provide feedback for next-cycle control. The SNN-based controller was deployed on a μ Caspien FPGA [27], which was connected to the host computer via Universal Serial Bus (USB). A sketch of the engine architecture, as well as the rapid prototyping controller, is depicted in Figure 1.

Cycle-to-Cycle Variability in Dilute Combustion

At the constant medium speed/load condition of 1500 rpm and 7.7bar indicated mean effective pressure (IMEP), the spark advance and EGR fraction were manipulated to explore different CCV characteristics. Figure 2 shows the

FIGURE 1 Experimental engine setup with rapid prototyping engine controller.

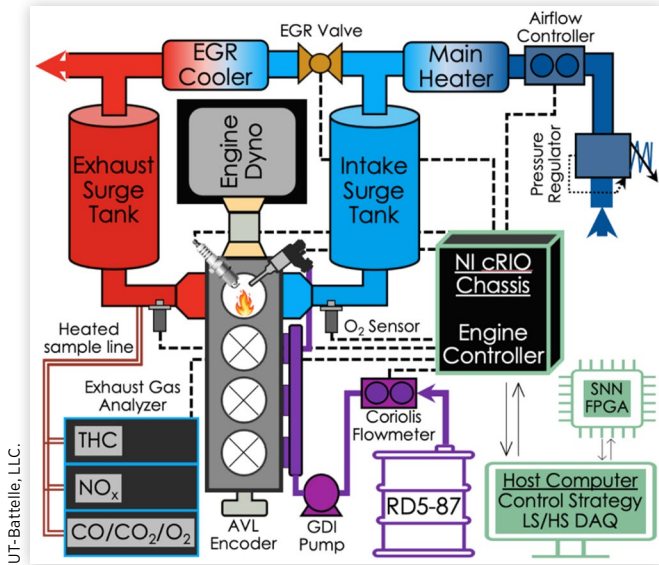
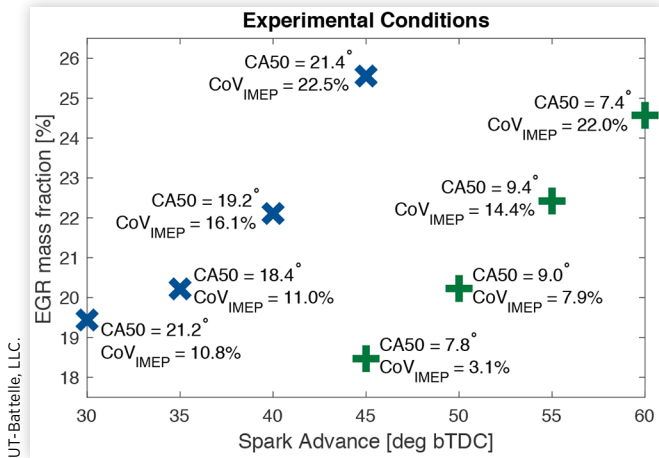


FIGURE 2 Spark/EGR sweep at constant engine speed and load. Blue (x) markers correspond to conditions with retarded spark timing while green (+) markers indicate conditions with optimal combustion phasing.



experimental conditions explore in this study. A total of 3,000 engine cycles were recorded at each condition in order to present reliable statistics. The experimental conditions are divided into two groups based on their combustion phasing, measured by the crank angle where 50% fuel mass is burned (CA50). The green (+) markers in Figure 2 represent the conditions where CA50 is optimal (close to 8 °aTDC). As expected, the spark timing needs to be advanced as the EGR fraction increases to compensate for the slow combustion kinetics and maintain optimal phasing at high dilution [6]. The CCV is typically quantified by the coefficient of variation (CoV) of IMEP. Industry standards limit the amount of EGR by considering a maximum of CoV_{IMEP} = 3%. For the set of experiments with optimal combustion phasing, the condition with the lowest EGR fraction of approximately 19% was considered as

the baseline, and no control action was applied to it. For the remaining conditions, Figure 2 shows that the CoV of IMEP rapidly increases as EGR increases. The second set of conditions are pictured by the blue (x) markers. Here, the spark timing has been retarded to maintain a combustion phasing of CA50 ≈ 20 °aTDC. The conditions at retarded spark were chosen to validate the controller at conditions outside the training data set. Retarded combustion phasing conditions are relevant during cold starts where high gas enthalpy is used to warmup the aftertreatment system [28,29].

Previous studies have used return maps of the gross heat release to identify deterministic patterns in combustion CCV required for developing effective control algorithms [5,6]. At each combustion cycle k , the gross heat release (Q_{gross}) was calculated from a single-zone analysis of a closed system (IVC to EVO), neglecting the effects of blow-by and crevice losses [30]:

$$Q_{gross}[k] = \int_{IVC}^{EVO} \frac{1}{\gamma_\theta - 1} V \frac{dP[k]}{d\theta} + \frac{\gamma_\theta}{\gamma_\theta - 1} P[k] \frac{dV}{d\theta} + \frac{dQ_{HT}[k]}{d\theta} \quad (1)$$

Here, V is the in-cylinder volume, $P[k]$ is the in-cylinder pressure during cycle k , θ is the crank angle position, and γ_θ is the polytropic coefficient which changes linearly with respect to θ from $\gamma_C = 1.42$ during compression to $\gamma_E = 1.28$ after the combustion is completed. The last term represents the convective heat transfer to the cylinder walls which was calculated as:

$$\frac{dQ_{HT}[k]}{dt} = HT_{mult} A_{cyl} h_{Woschni}[k] (T[k] - T_{wall}) \quad (2)$$

The rate of heat transfer depends on the cylinder wall area A_{cyl} , the Woschni heat transfer coefficient $h_{Woschni}[k]$, and the difference between the cylinder wall temperature $T_{wall} = 500$ K (assumed to be constant) and the bulk gas temperature at cycle k calculated as:

$$T[k] = \frac{P[k]V}{R(M_{fuel}[k] + M_{air}[k] + M_{inert}[k])} \quad (3)$$

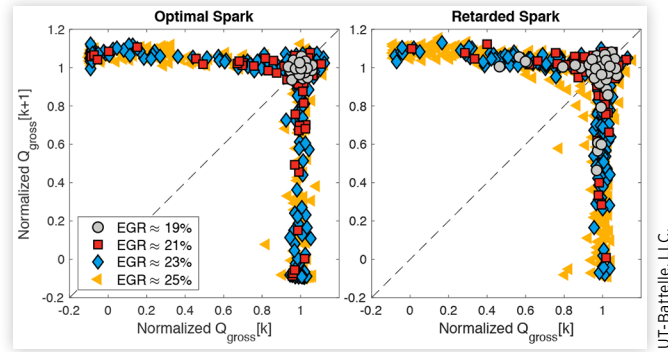
The constant parameter $R = 287$ J/kgK corresponds to the ideal gas constant for dry air. At cycle k , the total in-cylinder mass is composed of the masses of fuel $M_{fuel}[k]$, air $M_{air}[k]$, and burned (inert) gas $M_{inert}[k]$. In addition, a heat transfer multiplier $HT_{mult} \in [0.6, 0.75]$, which decreases with advanced spark timing, was used to calibrate the calculated value of combustion efficiency

$$\eta_c[k] = \frac{Q_{gross}[k]}{M_{fuel}[k] Q_{LHV}} \quad (4)$$

and match the measured value from the exhaust gas analyzer. The RD5-87 fuel has a lower heating value of $Q_{LHV} = 41.61$ MJ/kg, as measured by ASTM method D240. Calculation of the total amount of fuel, air, and inert gas at any cycle k will be discussed in the following section.

Figure 3 shows the return maps for the normalized gross heat release at various levels of EGR. The left plot shows the return maps for the conditions at optimal spark advance. The baseline condition at around 19% EGR fraction shows a completely random behavior centered at the nominal value.

FIGURE 3 Return maps for normalized gross heat release at various levels of EGR for optimal spark conditions (left) and retarded spark conditions (right).



As EGR levels increase, however, misfire and partial burns occur sporadically, and the return map presents two asymmetric arms extending towards low-energy events. Similarly, the right plot shows the return maps for the retarded spark conditions. In this case, the baseline case already presents some partial-burn events, which corroborates the high CoV of IMEP value of 10.8% indicated in Figure 2. As EGR levels increase, the high values of CoV are mainly due to partial burns. In contrast, at optimal spark conditions there is a higher density of low-energy events compared with medium-energy events. For instance, at 23% EGR (blue diamonds), the return map at optimal spark presents more events at low energy ($-0.2 < Q_{\text{gross}} < 0.3$) than events at medium energy ($0.4 < Q_{\text{gross}} < 0.7$). On the other hand, the return map at retarded spark with similar EGR levels has a higher concentration of events at medium energies. This is the main difference between the CCV characteristics at optimal and retarded spark timings.

The similarity between all conditions at high levels of EGR, however, consists of the deterministic patterns that emerge when low-energy events occur. The asymmetric nature of the return maps, favoring high-energy events after (low-energy) partial burns and misfires, indicates the existence of prior-cycle correlation. In the following sections, such a correlation will be used to design a next-cycle control strategy to manipulate the fuel injection quantity and reduce the combustion CCV. As the controller was realized by a SNN, the training was performed offline using evolutionary-based learning. Offline simulation of the cycle-to-cycle combustion events was done using a control-oriented model where the residual gas from one cycle to the next induces a dynamic coupling.

Control-Oriented Combustion Model

The control-oriented model is based on a physics-based approach where the total in-cylinder mass in a given cycle is composed of the residual gas from the previous cycle and the fresh mass admitted to the cylinder:

$$\begin{bmatrix} M_{\text{fuel}} \\ M_{\text{air}} \\ M_{\text{inert}} \end{bmatrix}_{k+1} = X_{\text{res}}[k] \begin{bmatrix} 1 - \eta_c[k] & 0 & 0 \\ -AFR_s \eta_c[k] & 1 & 0 \\ (1 + AFR_s) \eta_c[k] & 0 & 1 \end{bmatrix} \begin{bmatrix} M_{\text{fuel}} \\ M_{\text{air}} \\ M_{\text{inert}} \end{bmatrix}_k + \begin{bmatrix} 0 \\ 1 \\ \frac{X_{\text{EGR}}}{1 - X_{\text{EGR}}} \end{bmatrix} m_{\text{air}} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} m_{\text{fuel}}[k] \quad (5)$$

$$Q_{\text{gross}}[k] = \eta_c[k] M_{\text{fuel}}[k] Q_{\text{LHV}}$$

The state of the system consists of the total in-cylinder masses of fuel, air, and inert burned gas at any given cycle $x_k = [M_{\text{fuel}} \ M_{\text{air}} \ M_{\text{inert}}]_k^T$. The constant parameter $AFR_s = 14.7$ is the stoichiometric air-to-fuel ratio. It is assumed that the fresh air mass m_{air} and the EGR mass fraction X_{EGR} remain constants at a given experimental condition. The control command corresponds to the injected fuel mass $u_k = m_{\text{fuel}}[k]$ and will be adjusted in a cycle-to-cycle basis by the SNN controller. The model parameters $X_{\text{res}}[k]$ and $\eta_c[k]$ represent the residual gas fraction and the combustion efficiency, which vary at each cycle. Recall that $\eta_c[k]$ can be calculated from the heat release analysis previously discussed. The residual gas fraction was estimated assuming an isentropic exhaust process from EVO to EVC:

$$X_{\text{res}}[k] = \frac{V_{\text{EVC}}}{V_{\text{EVO}}} \left(\frac{P_{\text{EVC}}[k]}{P_{\text{EVO}}[k]} \right)^{1/\gamma} \quad (6)$$

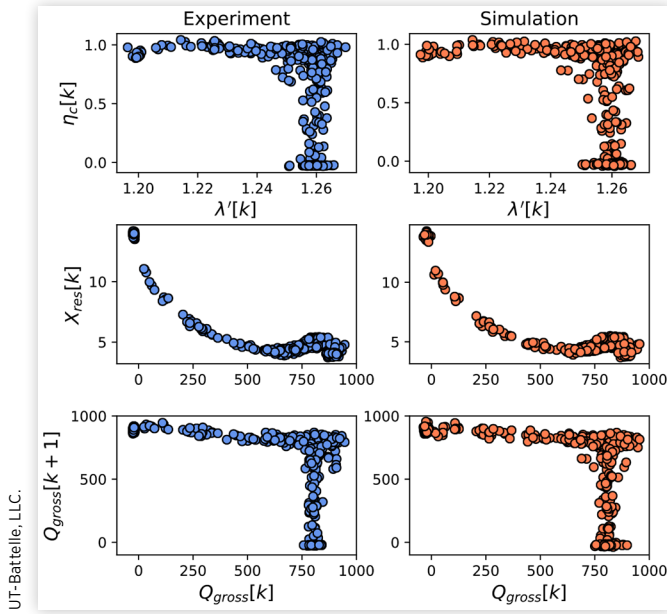
A detailed discussion of the method used for system identification based on the control-oriented model and the heat release analysis discussed in the previous section can be found in Maldonado and Kaul [25]. During the online implementation, however, the in-cylinder pressure at EVC was replaced by the average exhaust manifold pressure ($P_{\text{EVC}}[k] = \bar{P}_{\text{exh}} \approx 1$ bar) in order to terminate the calculations at EVO instead of EVC. This modification allowed the engine controller enough time to communicate with the $\mu\text{Caspian}$ FPGA and to issue the appropriate control command before the beginning of the next cycle.

For offline simulation and training of the SNNs, the model parameters $X_{\text{res}}[k]$ and $\eta_c[k]$ were calculated as functions of the state $x_k \in \mathbb{R}^3$. To reduce the dimensionality of the problem, consider defining the gas-fuel equivalence ratio as a proxy for the in-cylinder composition [31]:

$$\lambda'[k] = f_{\lambda}(x_k) = \frac{M_{\text{air}}[k] + M_{\text{inert}}[k]}{M_{\text{fuel}}[k]} \cdot \frac{1}{AFR_s} \quad (7)$$

The top-left plot of Figure 4 shows that the combustion efficiency can be written as a function of the gas-fuel equivalence ratio: $f_{\eta}: \lambda' \rightarrow \eta_c$. Similarly, the second-left plot shows that the residual gas fraction depends on the gross heat release $f_X: Q_{\text{gross}} \rightarrow X_{\text{res}}$, which intrinsically depends on the system states. The mappings f_{η} and f_X , however, present a stochastic nature that can be modeled by an appropriate choice of random variables. Under a probabilistic framework, the model parameters $X_{\text{res}}[k]$ and $\eta_c[k]$ can be seen as random variables

FIGURE 4 Comparison between open-loop experimental (left) and simulated (right) model parameters using the hybrid modeling approach.



sampled from the following conditional probability density functions (PDFs):

$$\eta_c[k] \sim f_{\eta|\lambda}(\eta|\lambda'[k]) \text{ and } X_{res}[k] \sim f_{X|Q}(X_{res}|Q_{gross}[k]) \quad (8)$$

Given that such distributions cannot be accurately regressed by standard parametric PDFs, the nonparametric kernel density estimator (KDE) was used since it provides accurate estimates for arbitrary distributions [32]. The unsupervised KDE for the conditional density $f_{X|Y}(X=x|Y=y)$ using N number of observations $\{X_i, Y_i\}_{i=1 \leq i \leq N}$ can be calculated as follows:

$$\hat{f}_{X|Y}(x|Y=y) = \frac{1}{h_X} \frac{\sum_{i=1}^N K\left(\frac{X_i - x}{h_X}\right) K\left(\frac{Y_i - y}{h_Y}\right)}{\sum_{j=1}^N K\left(\frac{Y_j - y}{h_Y}\right)} \quad (9)$$

where $K(z) = (2\pi)^{-0.5} e^{-0.5z^2}$ is the Gaussian kernel function and h_X, h_Y are the bandwidths hyperparameters chosen using the maximum-likelihood leave-one-out cross-validation.

The inverse cumulative distribution function (CDF) sampling was used to obtain the cycle-to-cycle values of $\eta_c[k]$ and $X_{res}[k]$ during simulations. The vector of uniform random variables $w_k = [w_\eta[k] \ w_X[k]]^T \sim \mathcal{U}(0,1)$ introduces the uncertainty observed in the data. Simulations were completed by evaluating the inverse CDF at the corresponding random variable as follows:

$$\begin{aligned} \eta_c[k] &= \hat{F}_{\eta|\lambda}^{-1}(w_\eta[k]|\lambda'[k]) \\ X_{res}[k] &= \hat{F}_{X|Q}^{-1}(w_X[k]|Q_{gross}[k]) \end{aligned} \quad (10)$$

Ultimately, this modeling approach renders a nonlinear stochastic system for offline simulations of the form:

$$x_{k+1} = f(x_k, u_k, w_k), \quad w_k \sim \mathcal{U}(0,1) \quad (11)$$

The right column of Figure 4 shows the simulated values for the model parameters $X_{res}[k]$ and $\eta_c[k]$. The bottom row compares the return maps for the measured (left) and simulated (right) gross heat release. The hybrid approach utilizing the physics-based model and the machine learning-based KDE for the model parameters provided accurate values and dynamic behavior. A more detailed discussion regarding the model can be found in Maldonado et al. [26].

In order to identify the parameters for offline simulations, standard dyno instrumentation such as airflow, oxygen sensors, and engine speed is required. In addition, in-cylinder pressure data are required for estimating the cycle-to-cycle energy release. Data for the offline combustion model were collected from a single operating condition using 5,000 engine cycles, a value sufficient to capture the dynamic properties of the system and statistically characterize it. The offline simulations were used to train the SNNs in closed-loop to reduce combustion CCV and, simultaneously, to optimize the network structure and parameters before FPGA deployment.

Spiking Neural Networks for Control

Like traditional artificial neural networks, spiking neural networks are composed of neurons and synapses. However, both neurons and synapses in SNNs incorporate a notion of time into how they process information, taking inspiration from biological neurons and synapses. In particular, synapses in an SNN can have different *delay* values, which determine how long it takes for charge to travel along that synapse. Neurons in SNNs accumulate charge over time and fire when their threshold is reached. Unlike traditional ANNs, fires in neurons throughout an SNN are asynchronous, occurring whenever each individual neuron reaches its threshold. Because of this temporal processing capability, spiking neural networks can be extremely well-suited to data analysis of temporal data or for control tasks. In particular, with respect to control tasks, an SNN accumulates information from all previous observations over the course of simulation to inform the current action taken, rather than making a decision based on a fixed number of previous observations. In this work, we use an integrate-and-fire neuron model. The charge accumulation for one of our neurons is given below.

$$A_j(t) = A_j(t-1) + \sum_{i=1}^N w_{i,j} S_i(t - d_{i,j}) \quad (12)$$

In this equation, $A_j(t)$ is the amount of charge on neuron j , at time t , $w_{i,j}$ is the weight of the synapse between neuron i and j (or 0 if no such synapse exists), and $d_{i,j}$ is the delay of the synapse between neuron i and j . $S_i(t)$, a binary function that says whether or not neuron i spikes at t , is calculated as follows:

$$S_i(t) = \begin{cases} 1 & \text{if } A_i(t) > \tau_i \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

where τ_i is the threshold of neuron i .

SNNs can be very efficiently implemented in neuromorphic hardware [33]. Neuromorphic hardware systems are brain-inspired custom hardware, where the computing hardware itself is made up of neurons and synapses. Because of their asynchronous, event-driven nature, SNNs as implemented on neuromorphic hardware are often extremely energy efficient. However, custom-chip neuromorphic hardware systems are not yet available on the market, though there are several research chips such as Intel's Loihi [34] and IBM's TrueNorth [35]. In this work, we implemented the neuromorphic architecture on an FPGA. Such custom-made architecture deployed on a commercial off-the-shelf FPGA will be referred to as μ Caspian [27]. However, this same architecture could be custom fabricated; in a custom chip version, we would expect to see at least an order of magnitude improvement in energy efficiency.

The low size, weight, and power (SWaP) demand of neuromorphic hardware is attractive for powertrain applications. For flexibility, the development board has both USB and direct input/output (I/O) interfaces. When using the USB interface, the board consumes around 500 mW including I/O and power regulation losses. However, removing USB and extra development circuitry, μ Caspian could approach a total power consumption on the order of 10-20 mW. Of that, about 6 mW is necessary for the static and dynamic power of the FPGA logic.

Offline SNN Training

To design and train a spiking neural network for this task, we utilized a training approach called Evolutionary Optimization for Neuromorphic Systems (EONS) [36]. EONS utilizes a genetic or evolutionary approach to determine both the structure (number of neurons and synapses, and how they are connected) and the parameters (weights of synapses $w_{i,j}$, delays of synapses $d_{i,j}$, and thresholds of neurons τ_i). This

process begins with a population of M randomly initialized potential SNN solutions. Each of the SNNs in the population is evaluated on the task and assigned a score (fitness value) for how well it performs. Then, a selection process is conducted to preferentially select better performing SNNs to serve as parents for the next generation. Reproduction operators such as duplication, crossover or recombination, and mutation are enacted on the parent networks to produce children. This evaluation, selection, and reproduction process is repeated until a fixed amount of time has elapsed or the desired performance level is reached. A pictorial description of the EONS training is presented in Figure 5. EONS and the simulation of the μ Caspian are custom software systems implemented in C++. In this work, we utilized Python bindings on top of the C++ implementation to integrate with the control-oriented combustion model developed in Python.

For this study, the hybrid control-oriented model was used to simulate the engine behavior as part of the evaluation process. At any cycle k , the model outputs the states of the system, namely $x_k = [M_{\text{fuel}} \ M_{\text{air}} \ M_{\text{inert}}]^T_k$. Before passing the current state value to the SNN, the state space was discretized. Each component was classified into one of 10 equal-sized bins in order to create input neurons. Neurons in the input layer are indicated with index values. Neurons 0-9 correspond to the bins for the amount of in-cylinder fuel M_{fuel} , neurons 10-19 correspond to bins for the amount of in-cylinder air M_{air} , and neurons 20-29 correspond to the amount of inert gas M_{inert} . After the inputs propagate through the network, the output layer produces one of 11 fixed fuel injection quantities (u_k), as indicated in Table 2. The possible control commands were chosen to favor values close to the nominal fuel injection quantity to avoid large deviations from stoichiometric conditions.

The objective function in this optimization seeks to minimize the amount of combustion CCV, while simultaneously minimize the additional amount of fuel needed to stabilize the charge. To that end, consider defining the following fitness function:

FIGURE 5 EONS workflow: Multiple SNNs are run in parallel using the Summit supercomputer at ORNL. The fitness values for each network are estimated using the hybrid control-oriented combustion model. The best networks are used to generate the next generation of SNNs and improve the fitness value. After N iterations, the best SNN is deployed on the FPGA to be used by the engine controller during the online implementation.

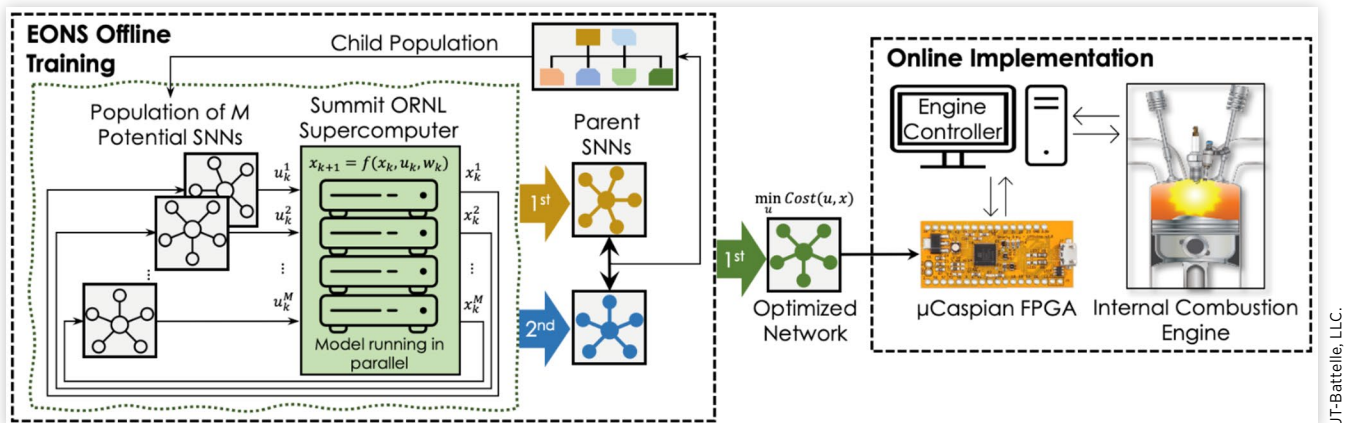


TABLE 2 SNN output layer (fuel injection quantity command)

Neuron Index	Fuel Control (u_k)	Relative value
30	20.63 mg	$m_{\text{fuel},0}$
31	20.42 mg	$m_{\text{fuel},0} - 1\%$
32	20.68 mg	$m_{\text{fuel},0} + 0.25\%$
33	20.73 mg	$m_{\text{fuel},0} + 0.50\%$
34	20.78 mg	$m_{\text{fuel},0} + 0.75\%$
35	20.83 mg	$m_{\text{fuel},0} + 1\%$
36	20.88 mg	$m_{\text{fuel},0} + 1.25\%$
37	20.94 mg	$m_{\text{fuel},0} + 1.50\%$
38	20.99 mg	$m_{\text{fuel},0} + 1.75\%$
39	21.04 mg	$m_{\text{fuel},0} + 2\%$
40	21.25 mg	$m_{\text{fuel},0} + 3\%$
41	21.45 mg	$m_{\text{fuel},0} + 4\%$

UT-Battelle, LLC.

$$Cost(x_1, \dots, x_{5000}) = \sum_{k=1}^{5000} \sigma_f \left(\frac{m_{\text{fuel},\text{in}}[k]}{m_{\text{fuel},0}} - 1 \right)^2 + \sigma_Q \left(\frac{Q_{\text{gross}}[k]}{Q_{\text{gross},0}} - 1 \right)^2 + \sigma_n g(Q_{\text{gross}}[k]) \quad (14)$$

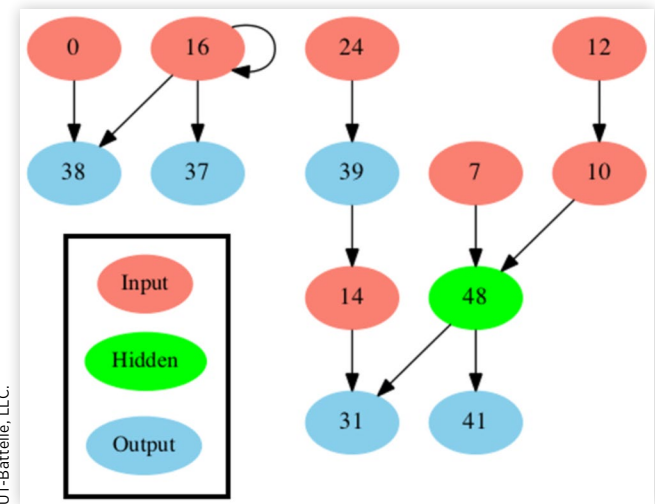
Here, $m_{\text{fuel},0}$ and $Q_{\text{gross},0}$ are the nominal values of fuel quantity and gross heat release, considered as constants. The function $g(\cdot)$ penalizes the occurrence of partial burns and misfires as follows:

$$g(Q_{\text{gross}}[k]) = \begin{cases} 0 & \text{if } Q_{\text{gross}}[k] \geq 645 \text{ J} \\ 1 & \text{if } Q_{\text{gross}}[k] < 645 \text{ J} \end{cases} \quad (15)$$

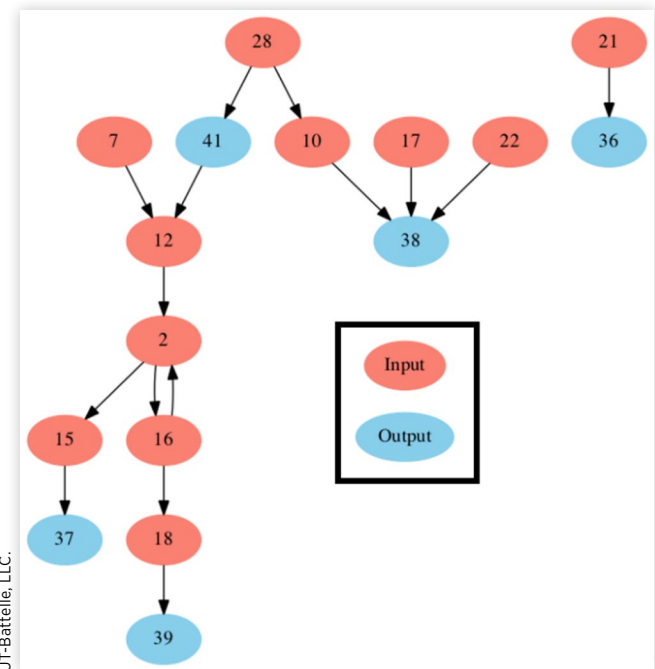
where $Q_{\text{gross}} = 645 \text{ J}$ corresponds to the threshold of 80% energy release with respect to the nominal value $Q_{\text{gross},0}$. The weighting factors reflect the importance of each term in the cost function and were chosen as $\sigma_f = 10^4$, $\sigma_Q = 1$, and $\sigma_n = 500$. The proposed objective function was specifically designed to penalize events that contribute to high CoV values. This presents an advantage over traditional proportional-integral (PI) control where the fuel command can only target the average value of Q_{gross} instead of its CoV.

For each SNN evaluation during the EONS training, 5,000 cycles were simulated, and the value of the cost function was calculated. The networks with the minimum cost were chosen to produce the next generation. More details about the offline control design can be found in Schuman et al. [37]. To speed up evolution, EONS was run on the Summit supercomputer at Oak Ridge National Laboratory. The training was run on 28 nodes of Summit, each of which ran an individual EONS process. This process was equivalent to running 28 standard workstations in parallel. Each of Summit's 28 nodes has 42 cores. As such, at this scale, it was possible to evaluate up to 1,176 SNNs simultaneously. Each of the individual EONS processes utilized a population size of 100. The initial population was made up of networks with 3 hidden neurons and six randomly initialized synapses (connecting randomly selected input, hidden, and output neurons in the network). The crossover rate, which determines how often crossover occurs, was

set at 0.5. The mutation rate was set at 0.9. The mutations correspond to changes in the network such as adding a neuron or synapse, or changing a parameter value. Rather than having a fixed number of generations, training was executed for two hours. This resulted in approximately 80 generations of evolution for each of our 28 runs. Over the course of training, neurons and synapses are added and removed, and parameter values are changed, customizing the network topology to the task. However, based on the initial random SNNs, very different SNNs can result from this evolution. Three such networks are shown in Figure 6, Figure 7, and Figure 8. The red neurons correspond to the input layer (x_k), the blue

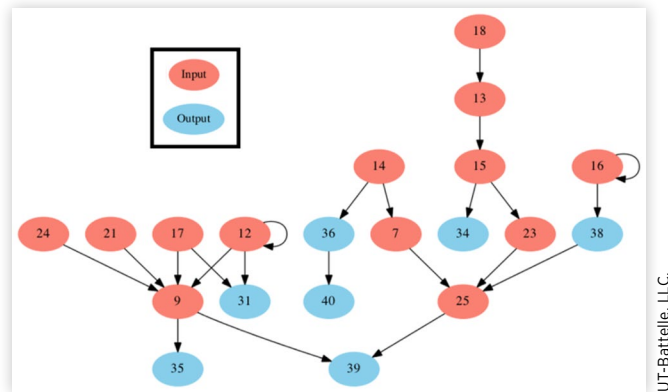
FIGURE 6 Network graph for SNN 1. Red neurons are inputs, blue neurons are outputs, and green neurons are hidden.

UT-Battelle, LLC.

FIGURE 7 Network graph for SNN 2. Red neurons are inputs and blue neurons are outputs.

UT-Battelle, LLC.

FIGURE 8 Network graph for SNN 3. Red neurons are inputs and blue neurons are outputs.



neurons correspond to the output layer (u_k), and the green neurons correspond to a hidden layer.

Online Control Implementation

After the EONS offline training phase was completed, the optimized networks were deployed onto the μ Caspian neuro-morphic hardware architecture [27]. The implementation was performed on a Lattice iCE40 UP5k FPGA, which provides real-time control decisions at very low size, weight, and power (SWaP). The selected FPGA was connected over a 1-MBaud universal asynchronous receiver-transmitter (UART) to a Teensy 4.0 ARM-based microcontroller. The microcontroller was in charge of receiving the engine state variables (x_i) from the LabVIEW ORCAS implementation over USB in a command string and generating the necessary spiking packets to send to the FPGA. The FPGA then executed the SNN model and sent the output spikes back to the microcontroller. These output spikes were decoded into the control command by selecting the output neuron which spiked the most (winner takes all), and the resulting decision for the next combustion cycle was sent back to LabVIEW ORCAS engine controller. Note that the deployment of the SNN occurs solely on the μ Caspian FPGA, independent of the choice of the engine control unit (ECU). Therefore, the SNN controller is inherently flexible for real-time implementation in modern ECUs capable of cylinder pressure-based control [38,39,40].

Additionally, the microcontroller has the capability to load different SNN models onto μ Caspian as desired. For this study, the three aforementioned networks were stored in the firmware such that a different network can be selected between each test using a simple command to the microcontroller. A future system could add the capability to automatically pick a specialized network depending on operating conditions.

The ORCAS combustion calculations, the SNN calculations, and the USB communication overhead with the μ Caspian FPGA combined to a duration of ~ 2 ms. Due to particular limitations of the NI cRIO ECU hardware implementation on this experimental setup, however, the ORCAS

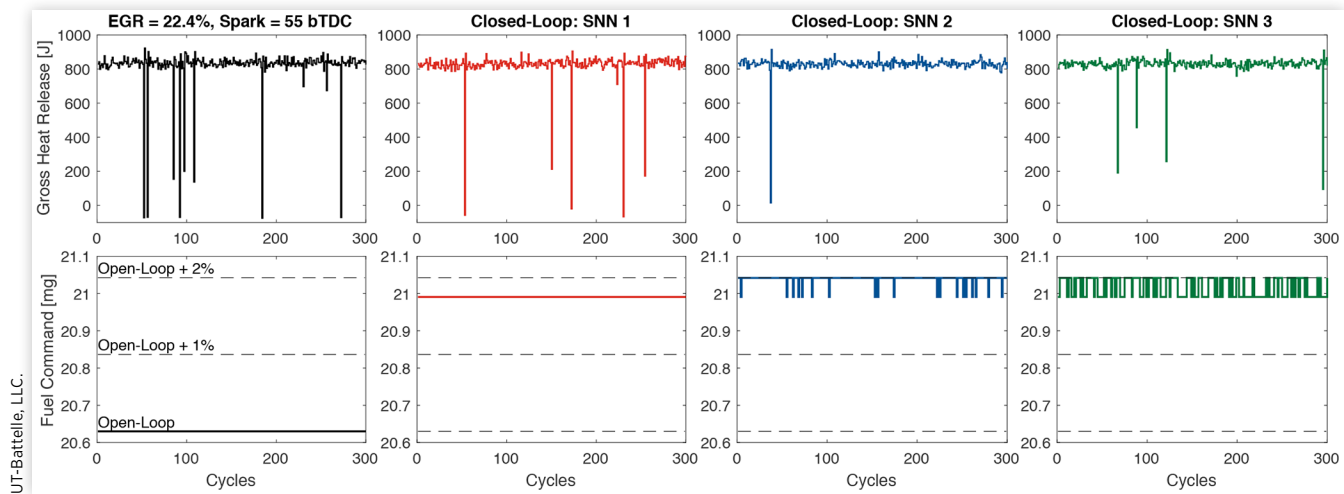
data acquisition system was operated on a separate PC, and analog communications were required to transfer the updated fuel command to the ECU as a voltage input which was acquired on a 4 ms loop. This means that up to 6 ms maximum elapsed time could be required between the end of combustion and the start of fuel injection for the next cycle. If the combustion analysis and SNN calculations were implemented onboard an integrated ECU without communication overhead, the time required for all calculations between the end of combustion and start of injection would be cut to likely < 1 ms.

At an engine speed of 1500 rpm, this implies that for this implementation, the fuel command needs to be calculated $\sim 60^\circ\text{CA}$ before the desired start of injection (-280°aTDC). Recall that all the calculations needed to determine the model parameters $X_{\text{res}}[k]$ and $\eta_c[k]$ can be completed at EVO (150°aTDC), leaving enough room for real-time cycle-to-cycle fuel control at 1500 rpm in this configuration; higher engine speeds could be easily achieved with a more fully-integrated ECU. It should also be noted that while computational power requirements could be expected to linearly scale with number of cylinders, the current combustion model and SNN training do not consider cylinder-to-cylinder interactions, which can have significant impacts on the system dynamics [41,42]. Further development would be needed to capture these additional dynamics in multi-cylinder engines.

Results

The EONS offline training was performed using the hybrid control-oriented combustion model calibrated for a condition similar to the one at 22.4% EGR and 55°bTDC spark advance shown in Figure 2. The resulting optimized networks were used for cycle-to-cycle closed-loop fuel control of the experimental engine. Figure 9 shows the time series of the experimental gross heat release (top row) and fuel command (bottom row) for the targeted condition during open-loop (OL) and closed-loop operation. In general, the resulting SNNs determined that a fuel increase of around 2% can significantly reduce the combustion CCV. This implies that the SNNs learned the benefits of enriching the charge without being explicitly informed. In addition, note that the controller does not fully utilize the +4% fuel enrichment that could potentially yield a more stable charge. Rather, the SNN controller found a balance between fuel enrichment and CCV reduction with +2% additional fuel. The property that small percentages of additional fuel can be enough to observe a significant reduction of CCV during operation with high levels of EGR has been previously reported by Jatana and Kaul [11]. Therefore, the SNNs decided to enrich just enough to reduce the number of partial burns and misfires by half (from 56 in open-loop to 29 with SNN₁, 22 with SNN₂ and 26 with SNN₃) without incurring as high a fuel penalty. Note that SNN₁ achieves this by maintaining a constant command. Hence, a constant feed-forward fuel command could generate similar results as the feedback controller using this SNN. However, the activation observed with controllers SNN₂ and SNN₃ result in a lower number of misfires and CoV overall. This implies that only

FIGURE 9 Time series of gross heat release (top row) and fuel injection quantity (bottom row) during a 300-cycle window of the three selected SNNs compared with open-loop case.

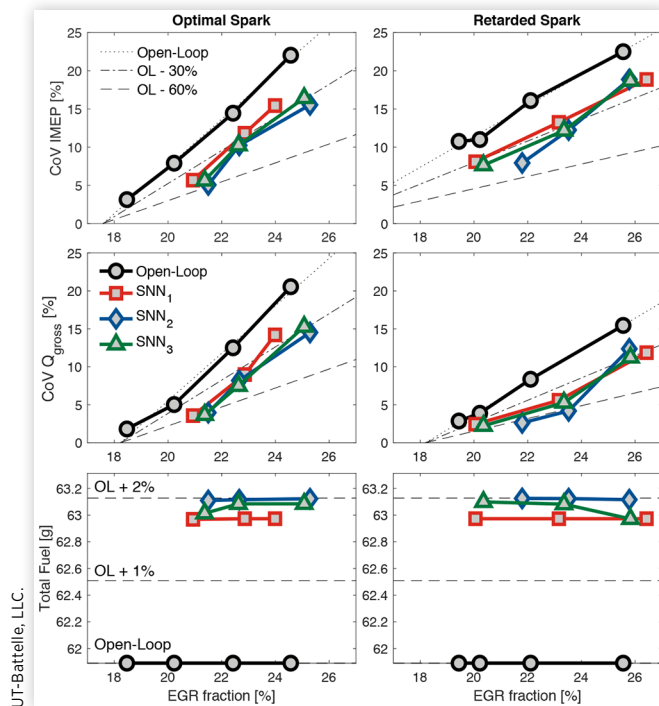


the feedback control law can exploit the deterministic properties of the system.

Even though the SNN controllers were trained under a single operating condition, no further calibration was performed and the same controllers were tested at all experimental conditions showed in Figure 2 with optimal spark and $EGR > 20\%$. Here, it was assumed that the conditions with optimal spark timing share similar CCV characteristics. The left column of Figure 10 summarizes the results of the closed-loop SNN controllers and compares them against the open-loop baseline. Each condition corresponds to an individual experiment of 3,000 engine cycles. The top-left plot shows the CoV of IMEP values at different EGR levels. Note that the open-loop experiments have an almost linear increasing trend, captured by the black dotted line. Under this assumption, the dash-dotted and dashed black lines represent linear trends with 30% and 60% reduction in CoV of IMEP with respect to the baseline, respectively. Closed-loop experiments at different EGR levels were recorded for SNN₁, SNN₂, and SNN₃. Note that, for EGR levels near 21%, the CCV reduction gets closer to 60% relative to open-loop. On the other hand, conditions around 25% EGR show relative reduction closer to 30%. The second plot on the left column shows the reduction in CoV of Q_{gross} which is the proxy variable for CCV introduced in the cost function. Similar trends are observed, implying that IMEP and Q_{gross} have similar CCV characteristics at optimal spark timing. The bottom-left plot shows the total fuel injected during the experiment. As previously mentioned, all three SNNs increased the nominal fuel injection quantity by 2%. Note that SNN₂ used slightly more fuel than SNN₃, however, both generated similar CCV reductions for $EGR < 24\%$. On the other SNN₁ uses the lowest amount of fuel overall, which translates to the lowest overall CCV reduction.

In order to test the robustness and usefulness of the SNN controllers without further calibration, closed-loop experiments were performed at retarded spark conditions. Recall that the CCV characteristics between advanced and retarded spark are different [6]. The top-right and second-right plots show the CoV of IMEP and CoV of Q_{gross} , respectively.

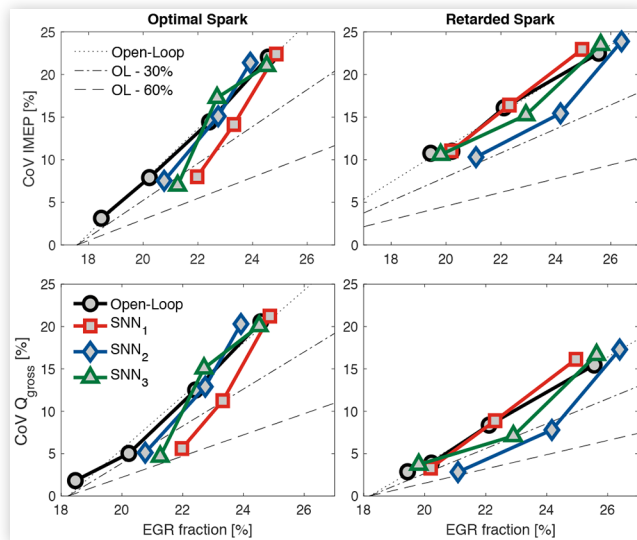
FIGURE 10 CoV of IMEP (top), CoV of gross heat release (middle) and total fuel command (bottom) for 3000 experimental engine cycles at optimal spark (left) and retarded spark (right) conditions.



Note that the open-loop CoV is significantly different in absolute value between IMEP and Q_{gross} . This is a consequence of the retarded phasing causing slow but fully burned combustion events, reducing IMEP but not changing significantly Q_{gross} . For retarded spark conditions, larger relative CoV reduction is observed for Q_{gross} (near 60% for $EGR < 24\%$) compared with the IMEP reduction (close to 30% overall).

The experimental results presented show that a 2% fuel increase by the SNN controller can yield up to 60% relative

FIGURE 11 CoV of IMEP (top) and CoV of gross heat release (bottom) for experimental conditions with closed-loop SNN fuel control based on CCV reduction and PI control based on $\lambda = 1$ target.



UT-Battelle, LLC.

CoV of Q_{gross} reduction for advanced and retarded spark conditions close to the stability limit ($\text{EGR} < 22\%$). As EGR levels increase, the SNN controller is less effective at avoiding misfires and partial burns, shifting the relative CoV of Q_{gross} reduction to 30%. The fuel increase introduced by the controller, however, reduces the global air-fuel equivalence ratio from stoichiometric $\lambda = 1$ to rich $\lambda = 0.98$. While this does indicate success of the SNNs in learning a means of reducing CCV, it is also a naïve solution that does not necessarily indicate that cycle-to-cycle dynamics have been accurately learned and are being taken advantage of, as was the goal.

In order to assess the effectiveness of the SNN controllers at reducing CCV through next-cycle control action while maintaining net stoichiometric conditions, a PI controller was used to maintain $\lambda = 1$ based on the exhaust oxygen sensor. The PI controller was designed to have a slow time constant to guarantee a bandwidth separation with respect to fast cycle-to-cycle SNN controller, similar to a master-slave decentralized control strategy [43]. The bandwidth separation ensures that the two controllers do not interfere with each other's goals.

Experiments at Stoichiometric Conditions

Figure 11 shows the summary of the experiments where the PI controller changed the nominal fuel quantity while the SNN controller performed small cycle-to-cycle adjustments. The top row shows the CoV of IMEP while the bottom row shows the CoV of Q_{gross} for optimal (left) and retarded (right) spark conditions. A plot for fuel consumption is not included since the PI controller guaranteed an average of $\lambda = 1$ during the 3,000 engine cycles recorded. Compared with Figure 10, we observe a more modest reduction of CoV for $\text{EGR} < 24\%$ while at the highest EGR conditions the SNN controllers were

unable to reduce the CoV during stoichiometric conditions. This indicates that for very high EGR levels the CoV reduction observed previously was primarily due to enrichment. Nonetheless, there is a net benefit of using the SNN controllers close to the unstable limit, indicating that they did successfully learn and utilize cycle-to-cycle combustion dynamics in addition to using enrichment.

During optimal spark conditions, SNN_1 and SNN_3 achieved the highest CoV reduction close to the stability limit. On the other hand, SNN_2 is the only network that provides a nonzero benefit for $\text{EGR} < 22\%$ at retarded spark conditions. The results suggest that distinct SNN topologies may be needed for operating conditions with different CCV characteristics, depending on combustion phasing. In general, close to the misfire limit, we observe a 30% CoV of IMEP reduction while the CoV of Q_{gross} reduction is closer to 60%. Given the appropriate SNN, however, this result seems to be independent of spark advance.

These results show that SNNs are a promising approach for active combustion stability control at the dilute limit. Several SNNs were able to reduce CCV near the dilute stability limit through next-cycle control action without net enrichment. It is also apparent that care must be taken to avoid unintended naïve solutions such as enrichment; this can be accomplished through carefully selecting the criteria used for the design and optimization process of the SNNs.

Summary/Conclusions

A complete methodology is described to design and implement a cycle-to-cycle controller to reduce combustion CCV in diluted combustion using spiking neural networks. A control-oriented model was presented that utilizes a physics-based approach to define the cycle-to-cycle combustion dynamics and a machine learning-based approach to regress the model parameters from engine data. The model was used for offline simulations in order to train SNNs for optimal fuel injection quantity control. The EONS offline training methodology was employed, which generates multiple generations of SNNs with optimized topologies. A cost function was defined that penalizes deviations from the nominal fuel command, penalizes deviations from the nominal gross heat release (proxy variable for CCV), and penalizes the occurrence of misfires and partial burns. Oak Ridge National Laboratory's Summit supercomputer was used to speed up that EONS offline training using a variety of initial SNN populations. Three SNNs with optimized topology and minimum cost were considered for online implementation. The selected SNNs were deployed in a FPGA-based $\mu\text{Caspian}$ neuromorphic hardware with low SWaP demand and connected to the engine controller via USB. A two-way communication was established between the FPGA and LabVIEW-based engine controller that was fast enough to allow next-cycle control of the fuel injection quantity based on previously observed combustion events. Closed-loop experiments were recorded close to the stability limit (where the CoV of IMEP is slightly higher than 3%) and during very high levels of EGR (CoV of IMEP higher than 20%). For all different EGR levels, two spark

timing conditions were considered: 1) advanced spark for optimal CA50 and 2) retarded spark mimicking instabilities at cold-start conditions.

The results showed that the SNNs quickly learned that enriching the charge can yield large CCV reductions with a relatively low fuel penalty. In this case, a 2% fuel enrichment resulted in nearly 60% CoV of IMEP reduction for conditions close to the stability limit. This solution, however, is not practical for a sustained operation since it will render the TWC ineffective. A slow PI controller was implemented, along with the SNN controller, to adjust the nominal fuel injection quantity. This control strategy guarantees that, on average, a stoichiometric operation is maintained. Experimental results showed that next-cycle control effects are significant for CoV of IMEP reduction close to the stability limit, where a 30% reduction is reported. At high levels of EGR, however, next-cycle control shows almost no additional benefit, indicating that enrichment has a bigger effect at reducing CCV. Similar results were observed for both optimal and retarded spark timings.

The results presented show the potential of SNN-based control, however, this strategy might not be globally optimal, and further research is needed to determine the maximum CCV reduction using next-cycle fuel injection control. Suboptimality might arise from the following aspects in the present study:

- The control-oriented model depends on the data collected at open-loop conditions, which explores only a subset of the state and control spaces.
- The EONS training was constrained by available HPC resources. In this study, EONS ran for 2 hours, which might not be enough to find the global optimal solution.
- The weighting factors σ_p , σ_Q , and σ_n were not optimized. Appropriate hyperparameter optimization can avoid the SNN solution of enriching the charge.
- The experimental condition used for offline training does not exactly match all the different conditions tested during online implementation. In general, a global optimal solution could be achieved if the SNN is retrained at each experimental condition.

Future research will focus on hyperparameter optimization and online learning/training in order to reduce overall fuel enrichment and explore global optimal solutions.

References

1. Lian, H., Martz, J.B., Maldonado, B.P., Stefanopoulou, A.G. et al., "Prediction of Flame Burning Velocity at Early Flame Development Time With High Exhaust Gas Recirculation and Spark Advance," *ASME. J. Eng. Gas Turbines Power* 139(8):082801, 2017 August, doi:10.1115/1.4035849.
2. Quader, A.A., "What Limits Lean Operation in Spark Ignition Engines - Flame Initiation or Propagation?" SAE Technical Paper 760760, 1976, doi:10.4271/760760.
3. Finney, C.E.A., Kaul, B.C., Daw, C.S., Wagner, R.M. et al., "Invited Review: A Review of Deterministic Effects in Cyclic Variability of Internal Combustion Engines," *International Journal of Engine Research* 16:366-378, 2015, doi:10.1177/1468087415572033.
4. Daw, C.S., Kennel, M.B., Finney, C.E.A., and Connolly, F.T., "Observing and Modeling Nonlinear Dynamics in an Internal Combustion Engine," *Physical Review E* 57(3):2811-2819, 1998 Mar, doi:10.1103/PhysRevE.57.2811.
5. Kaul, B.C., Wagner, R.M., and Green, J.B.J., "Analysis of Cyclic Variability of Heat Release for High-EGR GDI Engine Operation with Observations on Implications for Effective Control," *SAE International Journal of Engines* 6(1):132-141, 2013, doi:10.4271/2013-01-0270.
6. Maldonado, B., Stefanopoulou, A., Scarcelli, R. and Som, S. "Characteristics of Cycle-to-Cycle Combustion Variability at Partial-Burn Limited and Misfire Limited Spark Timing under Highly Diluted Conditions," in *Proceedings of the ASME 2019 Internal Combustion Engine Division Fall Technical Conference*, 2019, Chicago, Illinois, USA: ASME. V001T03A018, doi:10.1115/ICEF2019-7256.
7. Maldonado, B.P. and Stefanopoulou, A.G. "Non-Equiprobable Statistical Analysis of Misfires and Partial Burns for Cycle-to-Cycle Control of Combustion Variability," in *Proceedings of the ASME 2018 Internal Combustion Engine Division Fall Technical Conference*, 2018, San Diego, California, USA: ASME. V002T05A003, doi:10.1115/ICEF2018-9540.
8. Maldonado, B.P., Li, N., Kolmanovsky, I., and Stefanopoulou, A.G., "Learning Reference Governor for Cycle-to-cycle Combustion Control with Misfire Avoidance in Spark-ignition Engines at High Exhaust Gas Recirculation-Diluted Conditions," *International Journal of Engine Research* 21(10):1819-1834, 2020, doi:10.1177/1468087420929109.
9. Maldonado, B.P., and Stefanopoulou, A.G., "Cycle-to-Cycle Feedback for Combustion Control of Spark Advance at the Misfire Limit," *ASME. J. Eng. Gas Turbines Power* 140(10):102812, 2018 October, doi:10.1115/1.4039728.
10. Daw, C.S., Finney, C.E.A., Green, J.B.J., Kennel, M.B. et al., "A Simple Model for Cyclic Variations in a Spark-Ignition Engine," SAE Technical Paper 962086, 1996, doi:10.4271/962086.
11. Jatana, G.S., and Kaul, B.C., "Determination of SI Combustion Sensitivity to Fuel Perturbations as a Cyclic Control Input for Highly Dilute Operation," *SAE International Journal of Engines* 10(3), 2017, doi:10.4271/2017-01-0681.
12. Davis, L.I.J., Daw, C.S., Feldkamp, L.A., Hoard, J.W., Yuan, F. and Connolly, F.T., inventors; "Method of Controlling Cyclic Variation in Engine Combustion," Patent 5,921,221, 1999 Jul 13.
13. Daw, C.S., Green, J.B.J., Wagner, R.M., Finney, C.E.A. et al., "Controlling Cyclic Combustion Variations in Lean-Fueled Spark-Ignition Engines," *AIP Conference Proceedings* 622:265-277, 2002, doi:10.1063/1.1487542.
14. Davis, L.I.J., Feldkamp, L.A., Hoard, J.W., Yuan, F. et al., "Controlling Cyclic Combustion Variations in Lean-Fueled Spark-Ignition Engines," SAE Technical Paper 2001-01-0257, 2001, doi:10.4271/2001-01-0257.

15. Scarcelli, R., Sevik, J., Wallner, T., Richards, K. et al., "Capturing Cyclic Variability in Exhaust Gas Recirculation Dilute Spark-Ignition Combustion Using Multicycle RANS," *ASME. J. Eng. Gas Turbines Power* 138(11):112803, 2016 November, doi:[10.1115/1.4033184](https://doi.org/10.1115/1.4033184).
16. Vance, J.B., He, P., Kaul, B.C., Sarangapani, J., and Drallmeier, J.A., "Neural Network-based Output Feedback Controller for Lean Operation of Spark Ignition Engines," in *Proceedings of the American Controls Conference*, 2006, Minneapolis, doi:[10.1109/ACC.2006.1656497](https://doi.org/10.1109/ACC.2006.1656497).
17. Vance, J.B., Kaul, B.C., Jagannathan, S., and Drallmeier, J.A., "Output Feedback Controller for Operation of Spark Ignition Engines at Lean Conditions Using Neural Networks," *IEEE Transactions on Control Systems Technology* 16:214-228, 2008 Mar, doi:[10.1109/TCST.2007.903368](https://doi.org/10.1109/TCST.2007.903368).
18. Vance, J., Kaul, B., Jagannathan, S., and Drallmeier, J., "Neuro Emission Controller for Minimising Cyclic Dispersion in Spark Ignition Engines with EGR Levels," *International Journal of General Systems* 38(1):45-72, 2009, doi:[10.1080/03081070802193028](https://doi.org/10.1080/03081070802193028).
19. Vance, J.B., Singh, A., Kaul, B.C., Jagannathan, S., and Drallmeier, J.A., "Neural Network Controller Development and Implementation for Spark Ignition Engines with High EGR Levels," *IEEE Transactions on Neural Networks* 18(4):1083-1100, 2007 Jul, doi:[10.1109/TNN.2007.899199](https://doi.org/10.1109/TNN.2007.899199).
20. Singh, A., Vance, J.B., Kaul, B.C., Sarangapani, J., and Drallmeier, J.A., "Neural Network Control of Spark Ignition Engines with High EGR Levels," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 2006, Vancouver, 4978-4985, doi:[10.1109/IJCNN.2006.247201](https://doi.org/10.1109/IJCNN.2006.247201).
21. Shih, P., Kaul, B.C., Jagannathan, S., and Drallmeier, J.A., "Reinforcement-Learning-Based Dual-Control Methodology for Complex Nonlinear Discrete-Time Systems with Application to Spark Engine EGR Operation," *IEEE Transactions on Neural Networks* 19(8):1369-1388, 2008 Jul 15, doi:[10.1109/TNN.2008.2000452](https://doi.org/10.1109/TNN.2008.2000452).
22. Shih, P., Kaul, B.C., Jagannathan, S., and Drallmeier, J.A., "Reinforcement-Learning-Based Output-Feedback Control of Nonstrict Nonlinear Discrete-time Systems with Application to Engine Emission Control," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 1162-1179, 2009 Mar 24, doi:[10.1109/TSMCB.2009.2013272](https://doi.org/10.1109/TSMCB.2009.2013272).
23. He, P., Bui, H., Jagannathan, S., Kaul, B., and Drallmeier, J., "Neuro Emission Controller for Minimizing Cyclic Dispersion in Spark Ignition Engines with High EGR Levels," . In: *Intelligent Engineering Systems through Artificial Neural Networks*, (ASME Press, 2004), 421-426.
24. Maldonado, B.P., and Kaul, B.C., "Evaluation of Residual Gas Fraction Estimation Methods for Cycle-to-Cycle Combustion Variability Analysis and Modeling," *International Journal of Engine Research* 1-16, 2021 Jan, doi:[10.1177/1468087420983087](https://doi.org/10.1177/1468087420983087).
25. Maldonado, B.P. and Kaul, B.C., "Control-Oriented Modeling of Cycle-to-Cycle Combustion Variability at the Misfire Limit in SI Engines," in *Proceedings of the ASME 2020 Dynamic Systems and Control Conference*, 2020, Pittsburgh, PA, USA, 1-11.
26. Maldonado, B.P., Kaul, B.C., Schuman, C.D., Young, S.R., and Mitchell, J.P., "Next-Cycle Optimal Fuel Control for Cycle-to-Cycle Variability Reduction in EGR-Diluted Combustion," *IEEE Control Systems Letters* 1-6, 2020 Dec, doi:[10.1109/LCSYS.2020.3046433](https://doi.org/10.1109/LCSYS.2020.3046433).
27. Mitchell, J.P., Schuman, C.D., and Potok, T.E., "A Small, Low Cost Event-Driven Architecture for Spiking Neural Networks on FPGAs," in *ICONS 2020: International Conference on Neuromorphic Systems*, 2020, 1-4, doi:[10.1145/3407197.3407216](https://doi.org/10.1145/3407197.3407216).
28. Maldonado, B., Bieniek, M., Hoard, J., Stefanopoulou, A. et al., "Modelling and Estimation of Combustion Variability for Fast Light-off of Diesel Aftertreatment," *International Journal of Powertrains* 9(1-2):98-121, 2020, doi:[10.1504/IHPT.2020.108423](https://doi.org/10.1504/IHPT.2020.108423).
29. Bieniek, M., Stefanopoulou, A., Hoard, J., Maldonado, B.P. et al., "Retard to the Limit: Closed-Loop COVIMEP Control for Aggressive Exhaust Heating," *IFAC-PapersOnLine* 52(5):624-629, 2019, doi:[10.1016/j.ifacol.2019.09.099](https://doi.org/10.1016/j.ifacol.2019.09.099).
30. Heywood, J.B., *Internal Combustion Engine Fundamentals* 2nd Edition (New York: McGraw-Hill Education, 2018).
31. Shayler, P., Winborn, L., Hill, M., and Eade, D., "The Influence of Gas/Fuel Ratio on Combustion Stability and Misfire Limits of Spark Ignition Engines," SAE Technical Paper [2000-01-1208](https://doi.org/10.4271/2000-01-1208), 2000, doi:[10.4271/2000-01-1208](https://doi.org/10.4271/2000-01-1208).
32. Murphy, K.P., *Machine Learning: A Probabilistic Perspective* (Cambridge, MA: The MIT Press, 2012).
33. Schuman, C.D., Potok, T.E., Patton, R.M., Birdwell, J.D., Dean, M.E., Rose, G.S., et al. "A Survey of Neuromorphic Computing and Neural Networks in Hardware," arXiv preprint [arXiv:1705.06963](https://arxiv.org/abs/1705.06963), 2017.
34. Davies, M., Srinivasa, N., Lin, T.H., China, G. et al., "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," *IEEE Micro* 38(1):82-99, 2018, doi:[10.1109/MM.2018.112130359](https://doi.org/10.1109/MM.2018.112130359).
35. Merolla, P.A., Arthur, J.V., Alvarez-Icaza, R., Cassidy, A.S. et al., "A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface," *Science* 345(6197):668-673, 2014, doi:[10.1126/science.1254642](https://doi.org/10.1126/science.1254642).
36. Schuman, C.D., Mitchell, J.P., Patton, R.M., Potok, T.E. and Plank, J.S., "Evolutionary Optimization for Neuromorphic Systems," in *NICE '20: Proceedings of the Neuro-Inspired Computational Elements Workshop*; 2020, doi:[10.1145/3381755.3381758](https://doi.org/10.1145/3381755.3381758).
37. Schuman, C.D., Young, S.R., Mitchell, J.P., Johnston, J.T., Rose, D., Maldonado, B.P., et al. "Low Size, Weight, and Power Neuromorphic Computing to Improve Combustion Engine Efficiency," in *2020 11th International Green and Sustainable Computing Workshops (IGSC)*, 2020, Pullman, WA, USA, 1-8, doi:[10.1109/IGSC51522.2020.9291228](https://doi.org/10.1109/IGSC51522.2020.9291228).
38. Kosowatz, J., "Rekindling the Spark," *Mechanical Engineering Magazine* 139(11):28-33, 2017 Nov, doi:[10.1115/1.2017-Nov-1](https://doi.org/10.1115/1.2017-Nov-1).
39. Gordon, D., Wouters, C., Wick, M., Lehrheuer, B. et al., "Development and Experimental Validation of a Field Programmable Gate Array-based In-Cycle Direct Water Injection Control Strategy for Homogeneous Charge Compression Ignition Combustion Stability," *International Journal of Engine Research* 20(10):1101-1113, 2019, doi:[10.1177/1468087419841744](https://doi.org/10.1177/1468087419841744).

40. Luo, Y., Maldonado, B., Liu, S., Solbrig, C. et al., "Portable In-Cylinder Pressure Measurement and Signal Processing System for Real-Time Combustion Analysis and Engine Control," *SAE Int. J. Adv. & Curr. Prac. in Mobility* 2(6):3432-3441, 2020 Apr, doi:10.4271/2020-01-1144.
41. Daw, C.S., Green, J.B.J., Wagner, R.M., Finney, C.E.A., and Connolly, F.T., "Synchronization of Combustion Variations in a Multicylinder Spark Ignition Engine," *Proceedings of the Combustion Institute*. 28:1249-1255, 2000, doi:10.1016/S0082-0784(00)80337-9.
42. Daw, C.S., Finney, C.E.A., Kaul, B.C., Edwards, K.D., and Wagner, R.M., "Characterizing Dilute Combustion Instabilities in a Multi-cylinder Spark-Ignited Engine Using Symbolic Analysis," *Philosophical Transactions of the Royal Society A* 373:20140088, 2014, doi:10.1098/rsta.2014.0088.
43. Maldonado, B.P., Zaseck, K., Kitagawa, E., and Stefanopoulou, A.G., "Closed-loop Control of Combustion Initiation and Combustion Duration," *IEEE Transactions on Control Systems Technology* 28(3):936-950, 2019 Mar 5, doi:10.1109/TCST.2019.2898849.

Contact Information

Bryan Maldonado

Engine Technologies Research Group
Energy Science and Technology Directorate
Oak Ridge National Laboratory
2360 Cherahala Blvd, Knoxville, TN 37932
maldonadobp@ornl.gov

Acknowledgments

This research was supported by the DOE Office of Energy Efficiency and Renewable Energy (EERE), Vehicle Technologies Office, under the guidance of Gurpreet Singh and Michael Weismiller, and used resources at the National Transportation Research Center, a DOE-EERE User Facility at Oak Ridge National Laboratory. This work was supported in part by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory.

This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

This research used resources of the Compute and Data Environment for Science (CADES) at the Oak Ridge National

Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Definitions/Abbreviations

°aTDC - Crankshaft degrees after Top Dead Center

°bTDC - Crankshaft degrees before Top Dead Center

°CA - Crank Angle degrees

ANN - Artificial Neural Network

CCV - Cycle-to-Cycle Variability

CDF - Cumulative Distribution Function

CFD - Computational Fluid Dynamics

CoV - Coefficient of Variation

ECU - Engine Control Unit

EGR - Exhaust Gas Recirculation

EONS - Evolutionary Optimization for Neuromorphic Systems

EVO - Exhaust Valve Opening

FPGA - Field-Programmable Gate Array

HPC - High-Performance Computing

IMEP - Indicated Mean Effective Pressure

I/O - Input/Output

IVC - Intake Valve Closing

KDE - Kernel Density Estimator

ML - Machine Learning

OL - Open-Loop

ORCAS - Oak Ridge Combustion Analysis System

PI - Proportional-Integral

PDF - Probability Density Function

SNN - Spiking Neural Network

SWaP - Size, Weight, and Power

TDC - Top Dead Center

THC - Total Hydrocarbon Content

TWC - Three-Way Catalyst

UART - Universal Asynchronous Receiver-Transmitter

USB - Universal Serial Bus