UNIVERSITY OF DEBRECEN
FACULTY OF ENGINEERING
DEPARTMENT OF
MECHANICAL ENGINEERING

OPTIMIZING THE COMBUSTION SYSTEM EFFICIENCY AND CONTROLLING EMISSIONS IN THE CLOUD-CONNECTED GASOLINE ENGINE USING DEEP REINFORCEMENT LEARNING METHOD

MASTER THESIS

Name
Mohamed Abdelmaksoud Abdalla.
Production Engineering Specialization.

Debrecen
202

Table of Contents

Thesis Task Points

Title: Optimizing the Combustion System Efficiency and Controlling Emissions in the Cloud-Connected Gasoline Engine Using Deep Reinforcement Learning Method

Task:

i. Define the combustion emissions, control problems, and system boundaries precisely.
ii. Specify the measurement objects, constraints, and evaluation metric for optimization.
iii. Specify the required data signals, interfaces, and pre-processing pipeline for control.
iv. Design the Deep Reinforcement Learning control framework with a development and validation plan.

List of Abbreviations

AFR	Air-Fuel Ratio
BMEP	Brake Mean Effective Pressure
BSFC	Brake Specific Fuel Consumption
CA50	Crank Angle at 50% Mass Fraction Burned
CO	Carbon Monoxide
CoV	Coefficient of Variation
DDPG	Deep Deterministic Policy Gradient
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
DT	Digital Twin
ECU	Engine Control Unit
EGO	Exhaust Gas Oxygen Sensor
EGR	Exhaust Gas Recirculation
EMS	Engine Management System
FTP	Federal Test Procedure

HIL	Hardware-in-the-Loop
HCCI	Homogeneous Charge Compression Ignition
ICE	Internal Combustion Engine
IMEP	Indicated Mean Effective Pressure
KI	Knock Intensity
LHV	Lower Heating Value
MAP	Manifold Absolute Pressure
MBC	Model-Based Control
MDP	Markov Decision Process
MIL	Model-in-the-Loop
MPC	Model Predictive Control
MVEM	Mean-Value Engine Model
Nox	Nitrogen Oxides
OBD	On-Board Diagnostics
PFI	Port Fuel Injection
RDE	Real Driving Emissions
RL	Reinforcement Learning
RPM	Revolutions Per Minute
SAC	Soft Actor-Critic
SIL	Software-in-the-Loop
TD3	Twin Delayed Deep Deterministic Policy Gradient
TWC	Three-Way Catalyst
VVT	Variable Valve Timing
WCET	Worst-Case Execution Time
WLTC	Worldwide Harmonised Light Vehicle Test Cycle.

List of Tables

List of Figures

Introduction

The internal combustion engine (ICE) is an integral part of the world's transportation infrastructure, especially considering that hybrids will be dominating the transport market for many decades to come [1]. Even though technology has advanced to the point that the ICE powers only a minority of new cars produced annually around the globe, further refinement of this type of engine still holds great significance for the environment and energy security. In light of tightening regulations according to the Euro 7 standard and similar legislation around the globe, the necessity for the reduction of carbon dioxide, nitrogen oxides, and carbon monoxide while maintaining the engine's efficiency is imminent [2].

Concerning the control challenge, the combustion process in a spark-ignition engine is recognized as one of the most complex, highly multivariate, and nonlinear systems that engineers are required to work with. On the other hand, the manipulated variables or control variables, that is, the spark advance (θ) and the air/fuel ratio (λ), have to be tuned to optimize the conflicting thermodynamic objectives, which are impossible to be optimized together.

Thesis Structure

This paper implements V-model development process by dividing the thesis into four chapters:

Chapter 1: Literature Review, A systematic review of engine control system history, theory of reinforcement learning, and digital twin technology is done in order to identify the research gap this thesis aims to fill.

Chapter 2: Methodology, It explains the combined research method, such as the MDP formulation, the Digital Twin protagonistic, the SAC algorithm setting, and the reward function design.

Chapter 3: Results & Analysis, It discusses the experimental results from the three-stage MATLAB implementation, including the training dynamics, drive-cycle benchmarking, robustness analysis, and HIL validation.

Chapter 4: Discussion & Conclusion, It brings together the study, explains the use of intelligent combustion control, recognizes the limitations, and suggests future research directions.

The overall research framework and the logical sequence from problem definition to solution development and validation are depicted in Figure 1. The sequence moves from problem formulation to Digital Twin development, DRL training, multi-stage validation, and final analysis.

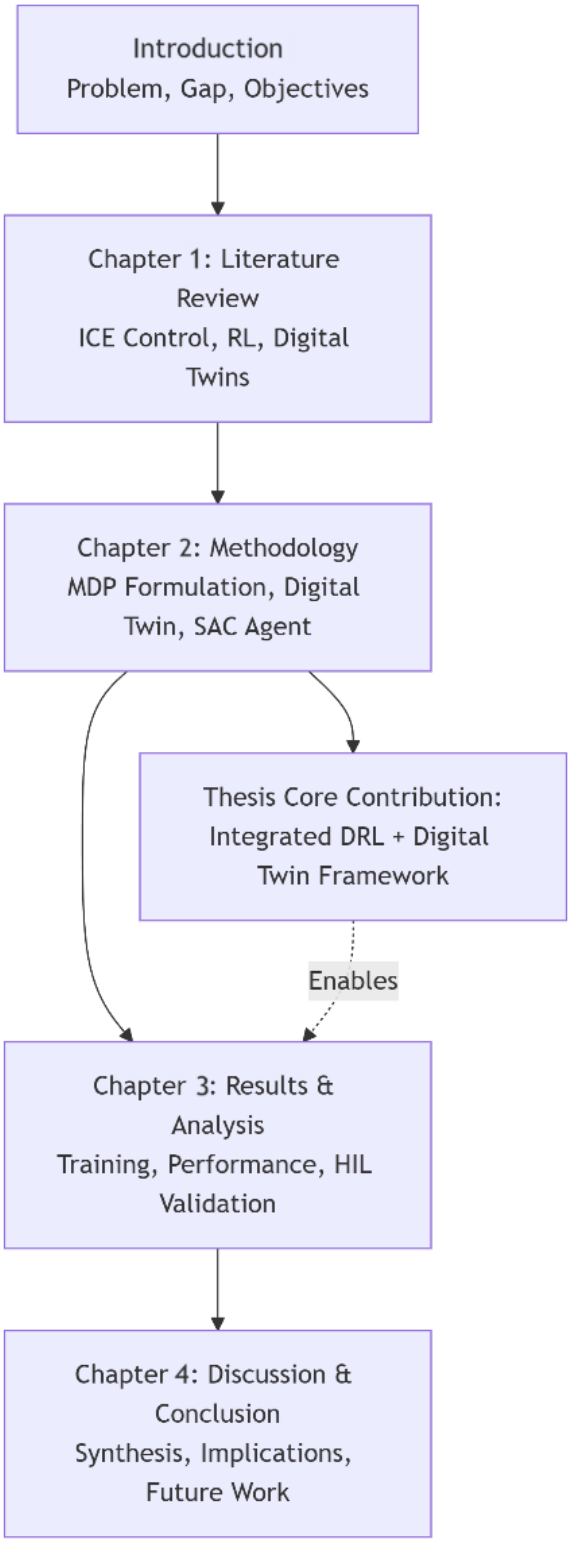

Figure 1: Thesis Structure and Research Design Flow[by myself].

1 Literature Review

This chapter critically reviews the historical development of engine control systems, reinforcement learning theory and digital twin technology. It documents the changes in control approaches from classical methods to the current intelligent control paradigms and establishes the technical foundation for the DRL, Digital Twin framework put forward in this thesis. The initial part is a review of the literature which finally leads to the identification of specific research gaps by the author that are addressed in this work[by myself].

1.1 Historical Evolution of Engine Control Systems

Engine management systems have also witnessed a paradigm shift from conventional mechanical to highly advanced electronic and, later, even intelligence-based cyber-physical systems. The need to comply with growing strict emission regulation norms and, simultaneously, achieve fuel economy and performance improvements is mainly responsible for these developments [3, 4].

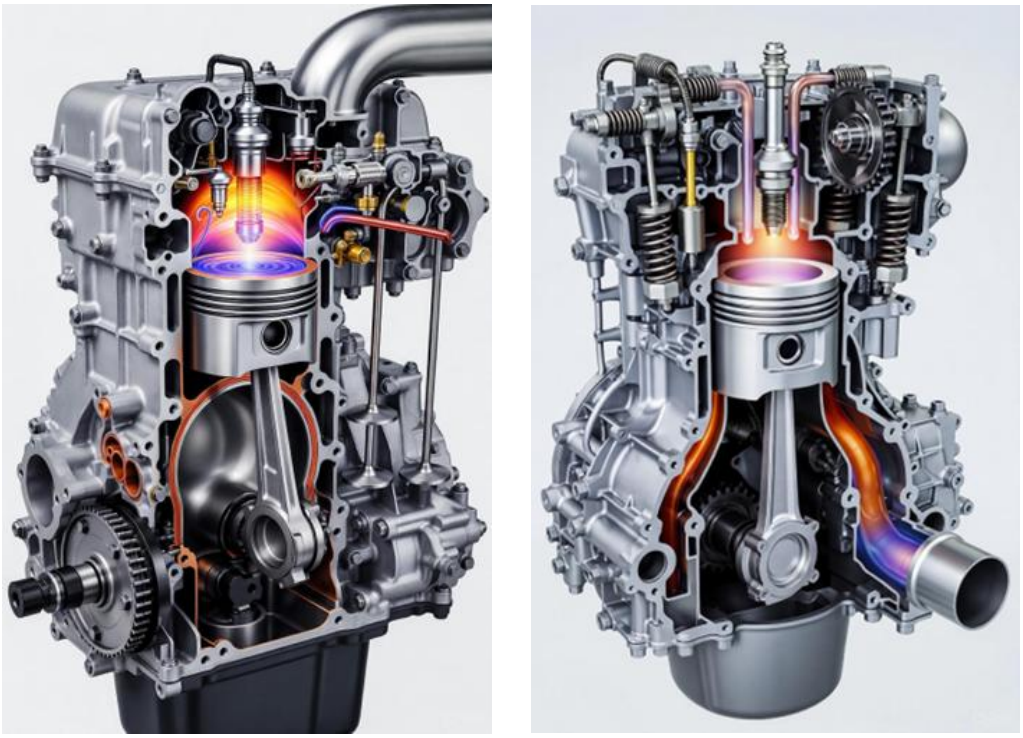

Figure 2: Development of the structure of the engine control system, Left: Original mechanical engine with fixed timing and carburetor , Right: Modern Engine Control Unit with sensor and actuator networks, indicating the shift to an electronic and digital system[3].

In the earliest internal combustion engines, as shown in Figure 2 (left), the ignition timing was fixed and controlled by mechanical systems such as distributors and carburetors. In such systems, it was not possible to optimize the parameters in an automatic manner depending upon the operating conditions present during the running of the engine [3]. The Oil Crisis of the 1970s gave the impetus to the use of microprocessors in internal combustion engines. The systems developed by the company Bosch in this period, such as their Jetronic and Motronic systems, marked the first important steps toward the development of internal combustion engine management systems with two or three-dimensional calibration maps in which the optimum actuator parameters (ignition timing and fuel injection duration) are expressed as a function of the relevant parameters such as the engine speed and the engine load [3].

The 1980s and 1990s marked an era of strong environmental regulations, with the result that the use of exhaust gas oxygen sensors (EGO) and three-way catalysts (TWC) became common, requiring strong control of the stoichiometric air fuel ratio ($\lambda \approx 1.0$) [3,4]. There also began the introduction of On-Board Diagnostics (OBD), which added even further to the increased system complexity. Improved processing power of the engine electronic control unit led to real-time control of an ever-increasing number of actuators, such as VVT, EGR, and turbocharging [3,22].

Despite being robust and reliable, the map-based control paradigm shows some inherent drawbacks [3]. According to Guzzella and Onder [3], static maps for calibration are no more than optimal solutions to trade-offs, fixed on certain test points and less adaptable to practical situations, including the following:

i) Variations in fuel properties
ii) Fluctuations of ambient conditions
iii) An aging engine and degrading components
iv) Stochastic, dynamic driving behaviors [3].

This inflexibility is one of the key reasons for the gap existing between laboratory certification and actual real-driving emissions (RDE) [2, 38].
This summarization of the evolutionary process and its challenges is presented in Table 1.

Table 1: Evolution of Engine Control: Technological Enablers and Persistent Limitations [3].

Era	Control Paradigm	Key Technological Enablers	Inherent Limitations & Challenges
Pre-1970s	Mechanical Control	Carburetors, Centrifugal Advance	No adaptation, fixed optimization, sensitive to wear.
1970s-1980s	Analog electronic	Electronic Ignition, Early Microprocessors	Limited parameters, open-loop operation, coarse control.

1990s-2000s	Digital map-based	µP-based ECUs, O_2 Sensors, OBD-I/II	Static calibration, extensive/expensive testing, and poor transient adaptation.
2000s-2010s	Model-Based Control (MBC)	Increased computing, physical/empirical models	Model inaccuracy, calibration complexity, and computational burden
2010s-Present	Adaptive and Learning-Based	Connectivity, ML algorithms, cloud computing	Certification, interpretability, data requirements, and real-time safety.

1.2 Advanced Control Strategies in Internal Combustion Engines

1.2.1 Model-Based Control Approaches

The limitations of static map-based control have spurred considerable research into Model-Based Control (MBC) approaches. Of these, Model Predictive Control (MPC) has emerged as a prime approach to engine applications due to its inherent capability to handle multivariable systems with constraints [8].

MPC uses a plant model illustrated in Figure 3 to forecast future behavior of the system over a finite horizon, and it solves an online optimization problem to compute a sequence of controls that satisfy certain criteria. Normally, the first control is applied in each problem formulation before the problem is resolved again (receding horizon control) [8].

The importance of engine path control, torque control, and combustion timing is pointed out as areas that utilize MPC by Norouzi et al. [8]. The benefit of model predictive methods relates to their ability to explicitly account for engine constraint conditions represented by knock constraint limits, exhaust temperature limits, and saturated actuator limits [8]. Real-time Nonlinear Model Predictive Control (NMPC) of real-world Fast Multi-Input Multi-Output (MIMO) optimization problems, however, is considered a challenge due to computational complexities that may result in model simplifications affecting engine control performance [8]. The general architecture of an MPC for engine management is illustrated in Figure 3[8].

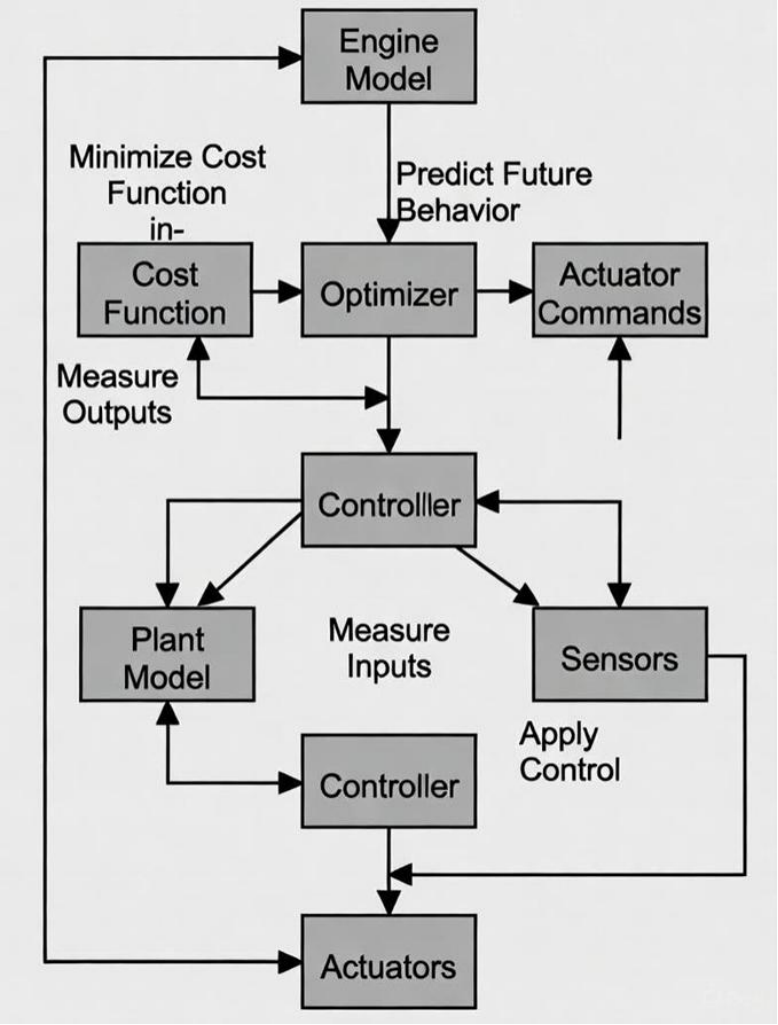

Figure 3: Model Predictive Control (MPC) Architecture for Engine Management[by myself].

1.2.2 Intelligent Control Systems

In addition to MBCs, computational intelligence has led to the introduction of other control techniques like fuzzy logic control and artificial neural networks (ANNs) in engine control. Fuzzy Logic Controllers (FLCs) are based on the expertise developed into linguistic control, where control is possible even when there are no mathematical models [5]. They have been applied in idle speed control, gear shifting, and emission control [5].

More pertinent to this thesis is the role of neural networks as universal function approximators in designing control systems. ANNs are employed as nonlinear virtual sensors, such as virtual pressure in the cylinders and emissions, and as part of inverse models in controllers [5][44]. Nonetheless, this type of model is typically a supervision-learning model and is based on big datasets with no natural ability for sequential decision-making, a major advantage of reinforcement learning [9] [33].

1.3 Reinforcement Learning in Engineering Applications

1.3.1 Theoretical Background and Algorithms

Reinforcement learning (RL) provides a mathematical model through which optimal behavior can be discovered by an agent through trial-and-error interaction in the environment that is modeled as a Markov Decision Process (MDP) [9, 34]. A standard MDP is described by a 5-tuple (S, A, P, R, γ) where:

i) S: State Space
ii) A: Action Space
iii) P: Transition probability function
iv) R: Reward function
v) γ: Discount factor

The agent is aimed at discovering a policy π(a|s) that will enable it to accumulate the maximal expected sum of discounted rewards over time, which can be represented by the Bellman equation:

$$Q^*(s,a) = \mathbb{E}_{s' \sim P}\left[r(s,a) + \gamma \max_{a'} Q^*(s',a')\right] \quad (1)$$

Basic algorithms like Q-learning and Temporal Difference (TD) learning led to the ability to estimate the model-free value function, thereby establishing the foundation of RL in real-world applications [29].

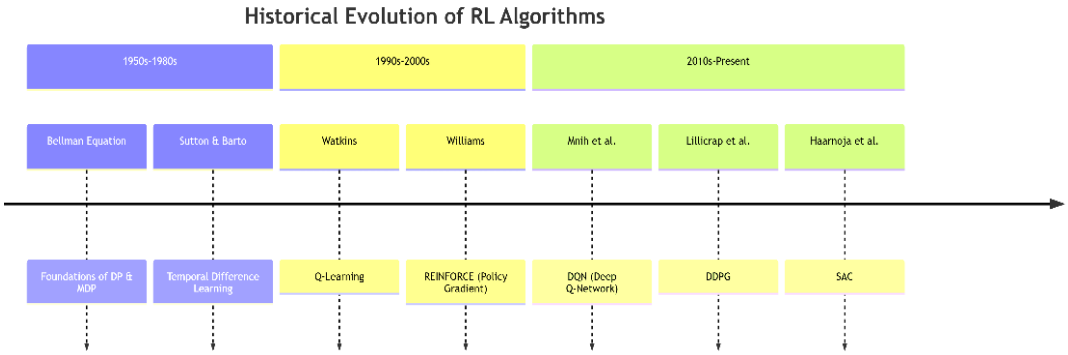

Figure 4: Historical evolution of reinforcement learning algorithms. [from dynamic programming (1950s) to deep reinforcement learning (2013-present)].

The timeline shown in figure 4 reflects the developments from the initial dynamic programming methods (Bellman, 1950s) to temporal difference learning (Sutton, 1980s), Q-learning (Watkins, 1992), policy gradient methods (1990s-2000s), and finally, the deep reinforcement learning revolution starting with DQN (2013) and continuing through actor-critic methods like DDPG (2015) and SAC (2018)[by myself].

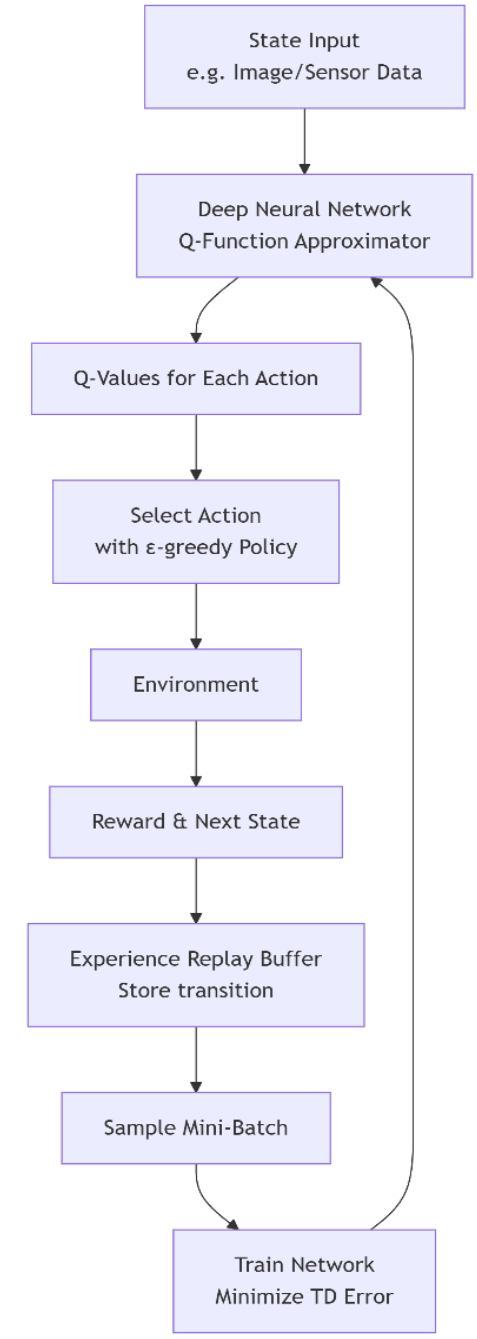

Figure 5: Deep Q-Network (DQN) architecture with experience replay buffer[by myself].

The critical breakthrough was the integration of deep neural networks as function approximators, leading to Deep Reinforcement Learning (DRL), The Deep Q-Network (DQN) algorithm proved the potential for DRL and reached human-like levels of proficiency in complex tasks, specifically for Atari games [29]. DQN brought into existing algorithms methods for stabilizing learning (experience replay and target networks). Figure 5 presents the principal makeup of a DQN agent [29].

According to Figure 5, depicting the use of a deep neural network that acts as a Q-function approximator, which basically means that it takes a state (usually an image or sensor data) as input and outputs the Q-values corresponding to each feasible action. The experience replay buffer is a storage of past transitions (s, a, r, s'), which are randomly selected during the training phase so that the temporal correlations are broken, and the learning process is stabilized [29].

1.3.2 Policy Search Approaches to Continuous Control

In continuous control problems, such as actuation of an engine (spark timing and Lambda are continuous variables), value-based approaches, including DQN, prove suboptimal. However, policy gradient methods, which represent a parameterized policy and optimize it, fare better. The algorithm, Reinforce, exemplifies a straightforward realization of policy gradients, which, owing to Monte-Carlo estimation, has high variance [9].

Actor-Critic architectures are a blend of both paradigms: an actor network parameterizes the policy $\pi(a|s)$, while a critic network estimates the value function V(s) or the Q-function Q(s,a), which serves as a lower-variance baseline for policy updates , Figure 6 depicts the typical actor-critic framework [33].

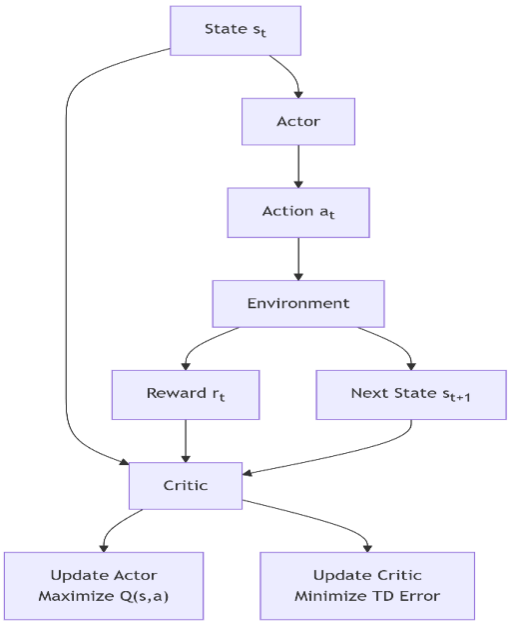

Figure 6: Actor-critic architecture of reinforcement learning with the interaction between actor and critic networks [by myself].

Figure 6 presents the interaction between the actor (policy network) and the critic (value network). The actor determines actions given the current state, the environment provides rewards and consequent states, and the critic appraises the actor's choice by estimating state or state-action values. Both networks are simultaneously updated using gradient descent: the actor is modified to maximize the expected return, and the critic to minimize the prediction error.

The Deep Deterministic Policy Gradient (DDPG) algorithm generalized this approach for continuous action spaces with deterministic policies [27]. Later developments introduced the Twin Delayed DDPG (TD3) approach to tackle the issue of bias from overestimation by using double Q-learning with clipped values, while the Soft Actor-Critic (SAC) is a more recent state-of-the-art algorithm for its ability to incorporate entropy maximization as part of its formulation [10][17].

1.3.3 Automotive Systems Applications

RL techniques have been used in many fields related to automotive engineering. A major application field is the energy management of hybrid cars, in which the RL agent learns the optimal strategy of splitting the powers, depending on the driving conditions [13][44]. Figure 7 exemplifies the use of RL for hybrid vehicle energy management from the power split control perspective [13][44].

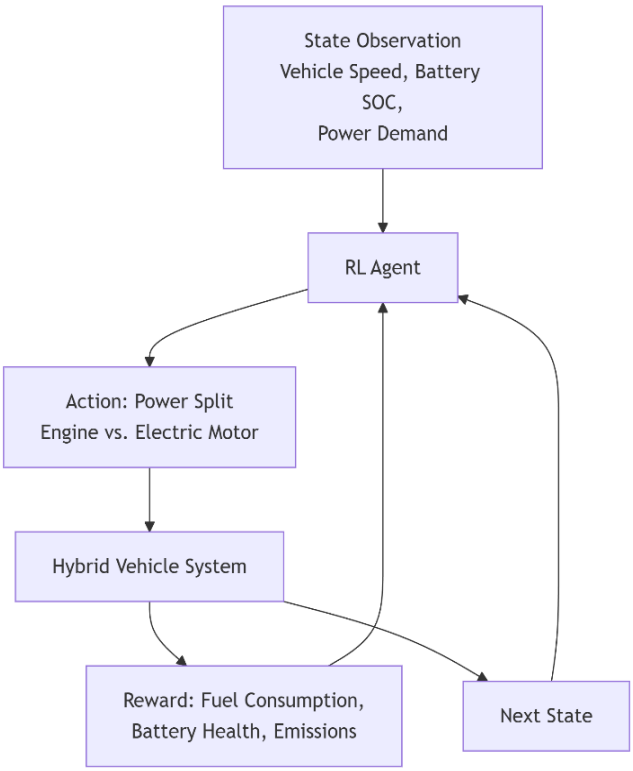

Figure 7: Reinforcement learning for the energy control system in a hybrid vehicle. Power Splitting: ICE, Electric Motor [by myself].

The sketch (Figure 7) shows an RL agent managing the power split between an internal combustion engine and an electric motor in a hybrid vehicle. The agent identifies the vehicle speed, battery state of charge, and power demand to choose the most efficient mode that minimizes fuel consumption while keeping the battery charge at optimal levels [by myself].

Moreover, RL serves as the basis for motion planning in self-driving cars, adaptive cruise control, transmission shift timing, and thermal management system optimization tasks [13].

Major difficulties identified in Automobile RL applications are:

i) Sample inefficiency of Pure Reinforcement Learning
ii) It is a critical requirement for safe exploration as well as a requirement at run-time.
iii) Simulation-to-reality (sim2real) transfer gaps

These concerns have led to research efforts on safe exploration strategies [15, 42], transfer learning, as well as domain randomization approaches to simulate exposure to diverse environments to make learning more robust to transfer from simulation to the physical domain [6][36]. The combination of these techniques into an automotive RL framework is depicted in Figure 8, which sketches the entire route from simulation to real deployment [6][36].

A flowchart (see Figure 8) demonstrating the whole process of developing RL controllers for automotive applications: initiating with high-fidelity simulation and synthetic data generation, going through training with domain randomization and safety constraints, then transferring to Hardware-in-the-Loop (HIL) testing, and finally deploying to real vehicles with continuous monitoring and online adaptation [by myself].

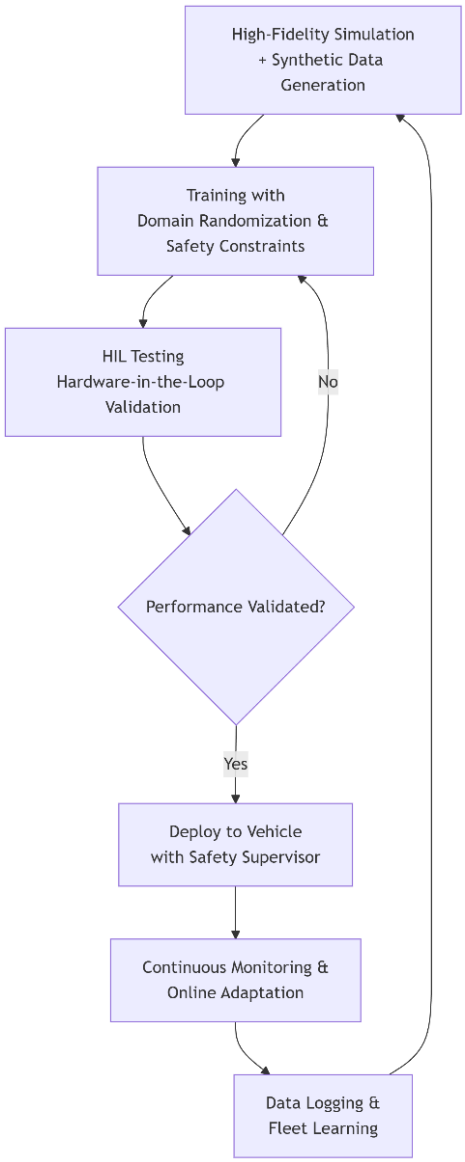

Figure 8: Automotive RL development workflow from simulation to HIL simulation and final implementation [by myself].

1.4 Deep Reinforcement Learning in Combustion Control

1.4.1 Current State of the Art

The use of Deep Reinforcement Learning (DRL) for internal combustion engine control is a relatively new but very fast-expanding field. Initially, when the researchers were limited by the computer power and the algorithms were not efficient, they concentrated on very simplified problems with tabular methods or

elementary function approximators and low-fidelity engine models [5]. These first experiments with DRL showed that RL can be a good alternative to a (P)ID controller, for example, the regulation of an air-fuel ratio within a simulated environment [5].

Nowadays, DRL is transitioning to advanced actor-critic algorithms such as Deep Deterministic Policy Gradient (DDPG) and Soft Actor-Critic (SAC), which are being exposed to multi-actuator control problems where the simultaneous optimization of spark timing, fuel injection, variable valve timing, and exhaust gas recirculation is done in high-fidelity simulation environments [5][40].

Moreover, the research has been extended to advanced combustion modes like Homogeneous Charge Compression Ignition (HCCI) and Reactivity Controlled Compression Ignition (RCCI), which are very difficult to control because they are extremely sensitive to operating conditions and have high cycle-to-cycle variations. The investigations have proven that DRL can be used to stabilize these difficult combustion processes while at the same time handling the complex efficiency-emissions trade-offs [5][39].

1.4.2 Integration Challenges

Presently, the top-end in DRL for engine control showcases several recurring themes as well as substantial unresolved challenges:

(a) *Algorithmic Preference:* Actor-critic methods, particularly the Soft Actor-Critic (SAC) and Twin Delayed DDPG (TD3), have been most frequently used in recent years for their higher sample efficiency and better stability in continuous action spaces [10][17]. Besides, their off-policy characteristic enables them to learn from past experiences, which makes them a perfect match for scenarios where data gathering is either costly or time-consuming [10][17].

(b) *Safety and Constraint Handling:* Since engine operation is a safety-critical task, the emphasis is heavily placed on safe learning approaches. For instance, methods such as Constrained Policy Optimization (CPO) directly integrate safety constraints into the optimization objective [40]. On the other hand, a well-thought-out reward structure where a large penalty is given to the violation of constraints has been massively used; however, this can result in overly cautious policies if the balance is not properly handled [42].

(c) *The Centrality of Simulation:* To train the models, high-fidelity simulation is utilized almost universally, as working with the real engine is too expensive, dangerous, and time-consuming [6][7]. In this way, the effectiveness of training becomes reliant on the simulation model's reliability, which leads to the critical problem of the reality gap, the difference between simulation and actual performance [6][7].

(d) *The Reality Gap and Transfer Learning:* A policy learnt entirely in simulation does not usually work when the policy is transferred to real hardware because the real system has characteristics that have not been

modelled, such as sensor noise and actuator delays. Therefore, it is still one of the key hurdles for actual deployment. Currently, the solutions are domain randomization (learning a policy over a range of simulated environments whose parameters have been changed at random) [6][36], system identification (the simulation model is improved step by step based on the real data), and transfer learning methods that adapt a policy to the real system [6].

1.5 Digital Twin Technology in Automotive Applications

1.5.1 Conceptual Foundations and Definitions

Originally, the Digital Twin concept was introduced by Grieves and Vickers as a tool for product lifecycle management [26]. Since then, it has expanded its scope and is now considered one of the main enablers of Industry 4.0 and cyber-physical systems[7][45].Offer an elaborate definition of a Digital Twin as a "comprehensive, multi-physics, multi-scale, probabilistic simulation of a complex product that employs the most accurate physical models, sensor updates, fleet history, etc., to represent the life of the corresponding physical twin." This definition, among other things, highlights that a DT is always evolving and updated with data, acts as a system, and is more than just a simple, static CAD model or a single simulation [7][45].

1.5.2 Implementation Architectures and Technologies

The deployment of a Digital Twin of an automotive system, in most cases, follows a tripartite layered architecture integrating:

(a) The Physical Layer (physical sensors/actuators on an actual engine),
(b) The Communication Layer (IoT protocols, 5G, CAN),
(c) The Virtual Layer (the simulation model and analytics) [7][37].

Cloud computing services (like AWS, Azure) play a critical role as they offer on-demand storage and computational power necessary to run elaborate simulations and support learning algorithms [7]. Edge computing is the other dimension that localizes the execution of time-sensitive operations, aiming to minimize delay [12][45].

1.5.3 Applications in Powertrain Development and Optimization

Currently, the automotive industry leverages Digital Twins at different stages of the product lifecycle:

(a) *Design & Development:* Facilitating virtual prototyping and decreasing the reliance on physical samples [7][37].

(b) *Calibration:* An extensive exploration of operating conditions in a virtual environment (much faster than on a dynamometer) is made possible, although the hardware validation is still required for the final stage [7].
(c) *Predictive Maintenance & Health Management:* Exploiting usage data to foresee malfunctions and enhance maintenance intervals [7].
(d) *Fleet Learning:* By collecting the aggregated, anonymized data of entire vehicle fleets to keep the Digital Twin model updated and, in this way, control strategies deployed across the fleet are informed and improved [7][23]. Figure 9 conveys a visualization of such a connected architecture [7][23].

The architecture demonstrates the two-way data exchange between the physical car (edge) and the digital twin (cloud), which is facilitated by the IoT connectivity. The digital twin gets updated by the physical asset's data, and at the same time, the digital twin can send back insights and upgraded software for deployment in the car [by myself].

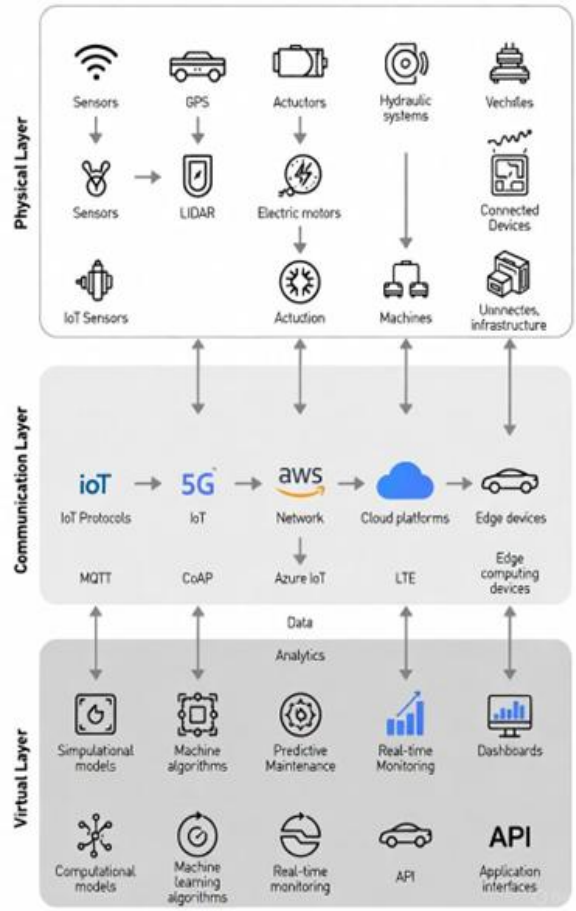

Figure 9: Framework for a Cloud-Connected Automotive Digital Twin[from the internet 12/03/2026].

1.6 Integration of DRL and Digital Twins

1.6.1 Synergistic Potential

The combination of DRL and Digital Twins brings together two highly complementary technologies that result in a potent synergy for the optimization of highly complex systems such as IC Engines. A Digital Twin is basically a synthetic environment, accurate and safe to run experiments in the real world, which supports the training of DRL methods, which are sample-inefficient by nature. Additionally, it gives the opportunity of endless, riskless testing not only at normal operating conditions but also including the potential failures that would be too dangerous or at least destructive to be tried practically [6][7].

On the other hand, DRL offers the "intelligent, decision-making" core to the Digital Twin. To put it simply, a DT can do a simulation and forecasting; however, it is through DRL that the DT is enabled to come up with, by itself, the control policies that it can then apply in the real world, i.e., to the physical engine. This redefines the DT from merely a reflector of reality to an optimizer [7].

1.6.2 Technical Challenges and Solutions

The main technical challenges of the integration can be:

(a) *Model Fidelity & the Reality Gap:* In the first place, the performance of the DRL policy can only be as good as the Digital Twin based on which it was trained. It is therefore essential to use methods like domain randomization (training on different model parameter values), online system identification, and residual learning (the RL agent learns the discrepancy between the model and the real-world and thus how to compensate for it) that can enable a smooth and seamless transfer [6][36].

(b) *Computational Cost:* Running detailed simulations and training DRL are computationally expensive tasks. Besides, cloud-based parallelization and multi-fidelity modeling strategies (using fast, low-fidelity models for the initial exploration phase and calling upon high-fidelity models for final validation) are very helpful [7].

(c) *Data Latency and Security:* If a cloud-connected control is to work in real-time, then communication delay must be dealt with by edge computing; also, fully functional cybersecurity measures must be installed and always maintained [45].

1.7 Research Gaps and Future Directions

1.7.1 Identified Research Gaps

From the review of literature, this work defines and aims to resolve the following highly interrelated research gaps:

(a) *No Integrated, Validated Frameworks:* Although various elements (DRL for control, high-fidelity simulation, cloud architecture) are available, the literature hardly presents a fully integrated, experimentally validated framework, which combines a cloud-connected Digital Twin with a DRL controller for multi-objective, transient engine control, following a strict V-model development process (SIL→MIL→HIL) [7].

(b) *Insufficient Multi-Timescale Optimization:* Research articles mostly concern optimization from cycle to cycle or in steady state. A strategy for multi-timescale optimization that covers a single combustion event to the entire driving cycle and even adapts the fleet at a lifetime level has not been adequately developed [7].

(c) *Certification and Interpretability Challenges:* It is well-known that the "black-box" character of deep neural networks significantly complicates their certification in safety-critical areas such as engine control. On one hand, research on the use of explainable AI (XAI) methods together with safety assurance cases for DRL policies is extremely limited, but on the other hand, it represents a very important factor of industrial penetration [42].

(d) *Limited Real-World Hardware Validation:* Most of the outstanding results are found in simulations. Without any doubt, experiments on HIL and engine-dynamometer validations of DRL controllers are essential to demonstrate real-world feasibility and conclusively close the sim2real gap [6].

1.7.2 Future Research Directions

Besides solving the mentioned problems, the great potential of future research entails:

(a) *Meta-Learning & Transfer Learning:* First and foremost, the fast adaptation of the developed agents to varying engine types, fuel blends, or deteriorated parts drastically cuts down the need for re-calibration [6].

(b) *Multi-Agent Reinforcement Learning (MARL):* Facilitating the collaboration of several DRL agents that control respective vehicle subsystems (engine, transmission, brakes, thermal management) to achieve whole-vehicle optimization [13].

(c) *Integration with Vehicle-to-Everything (V2X):* The use of connectivity and outside data (topography, traffic, weather) by vehicles to provide predictive, eco-driving optimizations [7][13].

(d) *Lifelong Learning:* Systems that keep on learning and adjusting from operational data throughout the vehicle life, thereby compensating for aging and sustaining the level of performance [7].

1.8 Summary and Thesis Positioning

The initial phase of this literature review has involved crafting the historical background, laying the theoretical bases, and defining the cutting-edge of engine control, DRL, and Digital Twin technology. The discussion indicates that although remarkable progress has been accomplished in each field separately, their combination into a single, experimentally confirmed framework for intelligent combustion control is clearly a research opportunity with considerable impact [by myself].

In this thesis I am trying to fill the recognized research gap, and do so by a novel method that:

i) Simply presents the combustion control problem as a constrained MDP [by myself].
ii) Constructs a Digital Twin of high-fidelity, cloud-connected, where both training and validation can be performed [by myself].
iii) Deploys a SAC agent of the state-of-the-art with a carefully thought-out multi-objective reward function and trains it [by myself].
iv) Confirms the effectiveness of the controller through a thorough, multi-phase protocol (SIL, MIL, HIL) along with a standard baseline, thereby demonstrating performance, robustness, and real-time capability convincingly [by myself].

2 Methodology

This chapter presents the integrated research methodology for the identified gaps discussed in Chapter 1. The methodology and development process follow the systematic V-model approach, starting from problem definition through implementation to thorough validation, incorporating principles from control theory [3][8], reinforcement learning [9][10], and digital twin technology [7][37].

2.1 Overall Research Design and Approach

This study uses a sequential mixed-method design with the quantitative computer simulation as the base component, followed by rigorous validation protocols. The overall framework, presented by Figure 10, is a design-simulate-validate paradigm that parallels the V-model development process for automotive software [3][12].

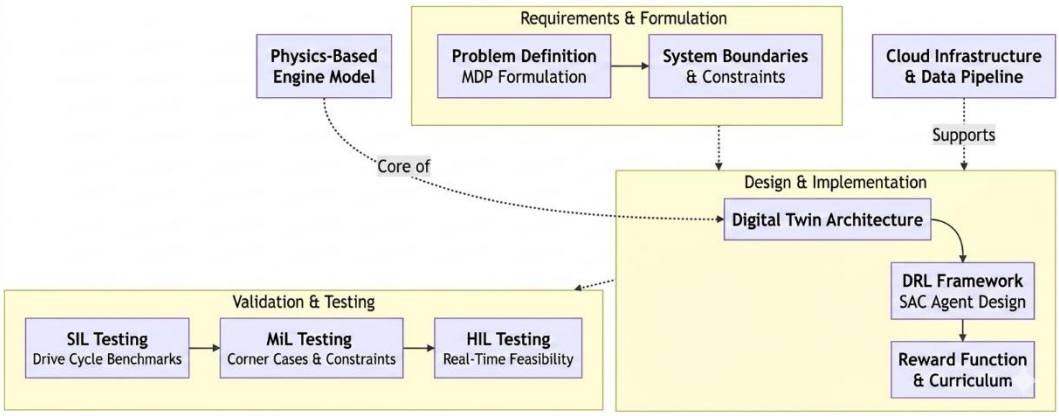

Figure 10: Overall Research Methodology Framework following the V-model development process, showing a systematic flow from requirements and problem formulation through design, implementation, training, and comprehensive validation [from the internet 23/02/2026].

Figure 10 shows the systematic flow from requirements and problem formulation (left side) through design, implementation, and training, to comprehensive validation and testing (right side). The Digital Twin and DRL at the center are the main components of the framework [by me explaining the graph].

The method is based on three mutually supporting pillars:

(a) *Digital Twin Development:* A detailed physical and mathematical model of an engine is created along with a cloud-connected software architecture, which together form the major development and safe exploration environment (Sections 2.4, 2.5) [by myself].

(b) *Deep Reinforcement Learning Implementation:* The Soft Actor-Critic (SAC) agent is designed and trained for continuous, multi-objective

control, employing a carefully controlled curriculum learning strategy (Sections 2.2, 2.5) [by myself].

(c) *Validation & Feasibility Analysis:* A comprehensive validation protocol, Software-in-the-Loop (SIL), Model-in-the-Loop (MIL), and Hardware-in-the-Loop (HIL) is performed to benchmark the performance, robustness, and real-time computational feasibility (Section 2.5.4) [by myself].

This combined method guarantees that the control system developed is theoretically well-founded, verifiable, and suitable for practical implementation [by myself].

2.2 Task 1: Problem Definition and System Boundaries

2.2.1 Formulation for Markov Decision Process (MDP)

The problem of combustion control has been precisely formulated as a finite-horizon, partially observable Markov Decision Process (MDP), characterized by the tuple (S, A, P, R, γ) (see [9][34]). Such a formulation underpins the DRL agent learning process mathematically.

(a) State Space Definition (S):

The state vector $\mathbf{s}_t \in \mathcal{S}$ should contain sufficient information for policy learning and, at the same time, be measurable on a real engine. Accordingly, initially, by substantive control documentation and observability analysis [3, 4], a 7-dimensional state is defined at time step *t* as:

$$\mathbf{s}_t = \left[\tilde{N}_t, \tilde{P}_{\text{man},t}, \tilde{T}_{\text{cool},t}, \widetilde{\text{CA50}}_{t-1}, \widetilde{\text{IMEP}}_{t-1}, \widetilde{\text{NOx}}_{t-1}, \eta_{\text{ind},t-1}\right]^T \quad (2)$$

Each term is normalized to the range [0,1] through operational limits (Table 2). Furthermore, the addition of the last cycle outputs ($\text{CA50}_{t-1}, \text{IMEP}_{t-1}, \text{NOx}_{t-1}$) puts in a temporal history, thus quite effectively dealing with the partial observability of the combustion state [9].

(b) *Action Space Specification ($\mathcal{A}$):*

The action space is limited to changing the primary combustion actuators within a certain range and having continuous values:

$$\mathbf{a}_t = [\Delta\theta_{\text{SA}}, \Delta\lambda]^T \in \mathcal{A} \quad (3)$$

Extremely strict rules are given step by step to operate the system safely when implementing the actions [3][4][42]:

i) Per-cycle adjustments: $\Delta\theta_{\text{SA}} \in [-5°, +5°]$, $\Delta\lambda \in [-0.02, +0.02]$

ii) Absolute limits: $\theta_{SA} \in [-10°, 35°]$ BTDC, $\lambda \in [0.95, 1.05]$

(c) *Reward Function (ℛ):*

The reward function expresses the multi-objective nature of the optimization problem by emphasizing the trade-offs among efficiency, emissions, and constraint satisfaction. It incorporates individual components whose sum equals one [9]:

$$r_t = w_1 \cdot \frac{\eta_{ind,t}}{\eta_{ref}} - w_2 \cdot \frac{\dot{m}_{NOx,t}}{\dot{m}_{NOx,ref}} - w_3 \cdot \frac{\dot{m}_{CO,t}}{\dot{m}_{CO,ref}} - w_4 \cdot \left(\frac{CA50_t - CA50_{opt}}{\Delta CA50_{max}}\right)^2 - w_5 \cdot I_{knock} - w_6 \cdot I_{misfire} - w_7 \cdot || \boldsymbol{a}_t - \boldsymbol{a}_{t-1} ||_2 + r_{constraints} \quad (4)$$

where I are indicator functions for violations, and $r_{constraints}$ additionally penalizes the agent for exceeding the limits of CoV of IMEP, exhaust temperature, or torque error. The weights w_i were adjusted with the help of multi-objective optimization so as to be indicative of each term's priority level (see Table 3) [9].

(d) *Discount Factor (γ):* The discount factor γ = 0.99 is applied, thus emphasizing the importance of long-term performance throughout a timespan of approximately 100 engine cycles that is considered sufficient for capturing the transient drive cycle dynamics [9].

2.2.2 Delimitation of System Boundary

To keep the main combustion control problem manageable and solve it clearly, the combustion control domain was narrowed down so that the focus and computational tractability were maintained [4][14].

Included Elements:

(a) *Plant Model:* A detailed and accurate model of which is an average-value, 1.8L, 4-cylinder, port fuel injection PFI, gasoline engine model with submodels for crank-angle resolved combustion (Wiebe function), heat transfer (Woschni model), and emission formation (Zeldovich for NOx, empirical for CO) [4][14].
(b) *Control Actuators:* Spark timing (θ_{SA}) and air-fuel ratio (λ) are the primary control inputs [4][14].
(c) *Operating Regime:* The mid-load range (2-8 bar BMEP), which covers typical urban and highway driving conditions [4][14].
(d) *Key Outputs:* Indicated thermal efficiency (η_{ind}), NOx, and CO emissions, combustion stability (CoV of IMEP), and knock intensity [4][14].

Excluded Elements:

(a) After-treatment system dynamics (three-way catalyst models).
(b) Cold-start and idle conditions (<2 bar BMEP) [4][14].
(c) Detailed mechanical sub-system dynamics (variable valve timing, turbocharger) [4][14].
(d) Vehicle-level energy management and driver-in-the-loop interactions.
(e) Long-term aging and component degradation effects [4][14].

Such a boundary permits a detailed study of the main combustion control trade-offs by means of DRL while the problem is still kept within a reasonable size [3][8].

2.2.3 Constraint Formalization

Ensuring safe operation is the most important goal. Safety constraints listed below, which are based on engine design limits and regulatory requirements, are formally defined, and their adherence is monitored during training and operation [3][4]:

Safety Constraints:

(a) Knock Intensity: $KI < KI_{\max} = 0.8$ (normalized) [28].
(b) Combustion Stability: $\mathrm{CoV}_{\mathrm{IMEP}} < 0.05$ [4].
(c) Misfire Prevention: IMEP > 300 kPa for the following cycles[28].
(d) Exhaust Temperature: $T_{\mathrm{exh}} < 950°\mathrm{C}$ [28].

Performance Constraints:

(a) Torque Tracking Error: $| \tau_{\mathrm{actual}} - \tau_{\mathrm{desired}} | < 0.10 \cdot \tau_{\max}$.
(b) Air-Fuel Ratio Deviation: $| \lambda - 1.0 | < 0.02$ at steady state.
(c) Actuator Rate Limits: Refer to Section 2.2.1

The constraints were used to guide the agent in the learning process by a hybrid method: hard constraints (action limits) are taken care of by projecting actions back to the feasible set, whereas soft constraints (output limits such as knock) are implemented through very large penalty terms in the reward function (Equation 4) [40][42].

Table 2: State Vector Components and Normalization Ranges[40].

State Component	Symbol	Description	Normalization Range	Units
Engine Speed	$\tilde{N}_t$	Current RPM	[800, 6000]	RPM
Manifold Pressure	$\tilde{P}_{\mathrm{man},t}$	Current MAP	[20, 100]	kPa

Coolant Temperature	$\tilde{T}_{\text{cool},t}$	Engine coolant temp	[273, 423]	K
Previous CA50	$\widetilde{\text{CA50}}_{t-1}$	Combustion phasing of the prior cycle	[-20, 40]	deg ATDC
Previous IMEP	$\widetilde{\text{IMEP}}_{t-1}$	Indicated Mean Effective Pressure	[0, 15]	bar
Previous NOx	$\widetilde{\text{NOx}}_{t-1}$	NOx emissions of the prior cycle	[0, 2500]	ppm (cumulative)
Previous Efficiency	$\eta_{\text{ind},t-1}$	Indicated thermal efficiency of the prior cycle	[0, 0.5]	-

Table 3: Reward Function Weights and Rationale[42].

Weight	Symbol	Value	Objective Term	Rationale for Weighting
w_1	Efficiency	100	η_{ind}	Primary objective; high weight to drive efficiency maximization.
w_2	NOx Emissions	50	$\dot{m}_{\text{NOx}}$	Major regulated pollutant; significant penalty for driving reduction.
w_3	CO Emissions	30	$\dot{m}_{\text{CO}}$	Regulated pollutant; moderate penalty.
w_4	CA50 Tracking	10	$(\text{CA50}-\text{CA50}_{\text{opt}})^2$	Encourages stable, efficient combustion phasing.
w_5	Knock Penalty	200	I_{knock}	**Very high penalty** to absolutely deter damaging operation [28].
w_6	Misfire Penalty	150	I_{misfire}	High penalty to prevent unstable combustion and HC spikes.
w_7	Action Smoothness	5	$\| \mathbf{a}_t - \mathbf{a}_{t-1} \|_2^2$	Encourages smooth actuator movements for durability

2.3 Task 2: Measurement Specification and Evaluation Framework

2.3.1 Measurement Objects and Data Acquisition

A complete suite of measurements was defined to be able to fully characterise engine performance during Digital Twin training and controller evaluation, based on the standard instrumentation list of an engine test cell [3][4].

Primary Objects of Measurement:

1) Combustion Parameters:
   (a) *Cylinder pressure:* Measured at 0.5∘ crank angle resolution with a piezoelectric transducer (e.g., Kistler 6125C). It is the major signal used for determining IMEP, CA50, and heat release rate [3][4].
   (b) *Heat release rate:* It is the derivative of the first law of thermodynamics applied to the cylinder pressure trace [3][4].
   (c) *Mass fraction burned:* It is either obtained from the Rassweiler-Withrow method or the Wiebe function fit [3][4].
2) Efficiency metrics are a count of:
   (a) *Indicated work:* Obtained by computing the area under the P-V curve for each combustion cycle [3][4].
   (b) *Fuel consumption:* Determined by means of simulated gravimetric measurement or Coriolis fuel flow meter (e.g., AVL 733S) [3][4].
   (c) *Brake torque:* Inferred from indicated work with allowance for friction losses [3][4].
3) Emission Indicators:
   (a) *NOx and CO concentration:* Simulated using the Zeldovich and empirical models [3][4].
   (b) *Sampling frequency:* The sampling frequency should be at least 100 Hz to capture transients effectively [3][4].
4) Operating State Variables:
   (a) *Engine speed:* From crank position encoder simulation.
   (b) *Manifold absolute pressure (MAP):* from the piezoresistive sensor simulation [3][4].
   (c) *Temperatures:* Coolant, oil, intake air, exhaust gas (thermocouples) [3][4].
   (d) *Actuator positions:* Spark timing and lambda (from control commands) [3][4].

Data Acquisition Protocol:

(a) All cylinder-pressure-related signals are sampled synchronously based on crank-angle [3][4].
(b) Performance indices such as IMEP and CA50 are calculated cycle-by-cycle in real-time [3][4].

(c) Data integrity is ensured through automatic data validation and outlier detection procedures [3][4].

2.3.2 Evaluation Metrics and Validation Protocol

A multi-faceted evaluation framework is established to rigorously assess controller performance, covering primary objectives, secondary qualities, and statistical significance [3][4].

Primary Performance Metrics:

(a) *Efficiency Improvement:* Percentage change in indicated thermal efficiency (η_{ind}) over a drive cycle [3][4].
(b) *Emission Reduction:* Percentage reduction in cumulative mass emissions of NOx (ΔNOx) and CO (ΔCO) [3][4].
(c) *Fuel Economy:* Improvement in Brake-Specific Fuel Consumption (BSFC) [g/kWh] [3][4].
(d) *Constraint Satisfaction Rate:* Percentage of operating points where safety and performance constraints (Section 2.2.3) are satisfied [3][4].

Secondary Performance Metrics:

(a) *Transient Response:* Settling time, overshoot, and integral absolute error (IAE) during step changes in load or speed [3][4].
(b) *Robustness:* Performance degradation under the influence of sensor noise ($\pm 10\%$), model parameter uncertainty ($\pm 5\%$), and actuator delay (up to 50 ms) [3][4].
(c) *Computational Efficiency:* Execution time (worst-case and average), memory footprint, and CPU utilization on the target hardware [3][4].
(d) *Learning Efficiency:* Total training time (simulated hours) and sample complexity (number of environment interactions) to achieve convergence [3][4].

Statistical Validation Protocol:

To ensure results are statistically significant and not due to random initialization variance:

(a) *Multiple Independent Runs:* The DRL training process is repeated 5 times with different random seeds. Performance metrics are reported as mean ± standard deviation [3][4].

(b) *Hypothesis Testing:* A paired t-test ($\alpha = 0.05$) is used to compare the DRL controller's performance against the baseline controller for each drive cycle and metric [3][4].

(c) *Cross-Validation:* The trained policy is evaluated on drive cycles not seen during training (e.g., US06, RANDOM) to assess generalization capability [3][4].

2.4 Task 3: Data Pipeline & Digital Twin Architecture

2.4.1 Data Acquisition and Pre-processing Pipeline

An efficient automated data processing pipeline is designed for developing an accurate Digital Twin, along with online learning capabilities, as shown in Figure 11.

Pipeline Stages:

1) *Synchronization & Cycle Detection:* Raw crank angle and cylinder pressure signals synchronized; individual engine cycles (for 4-stroke engines, 720° crank angle) detected based on crank encoder signal [3][4].
2) *Signal Processing & Filtering:*
   i) Cylinder pressure: Motoring pressure trace subtracted; low-pass filter (5 kHz cutoff frequency) applied to reduce noise while preserving combustion dynamics [4].
   ii) Other Signals: Proper digital filters (moving average filters, median filters) applied according to the nature of the signal [3][4].
3) *Feature Extraction:* Real-time extraction of the following features per cycle:
   i) IMEP: IMEP $= \frac{1}{V_d} \oint P \, dV$
   ii) CA50: Crank angle at which cumulative heat release equals 50%.
   iii) Combustion Duration: CA10 to CA90
   iv) Peak Pressure and Location
   v) NOx and CO: Cycle-resolved data from emission models.
4) *Normalization & Packaging:* Normalization of features into ranges listed in Table 2 and packaged for consumption by the DRL agent into state vectors following Equation 2 [3][4].

Data Quality Assurance:

(a) *Rejection of outliers:* Cycles with IMEP or CA50 outside $\pm 4\sigma$ from the moving average rejected [3][4].
(b) *Sensor Health Monitoring:* Signals tested for saturation, dropout, or unphysical regions [3][4].

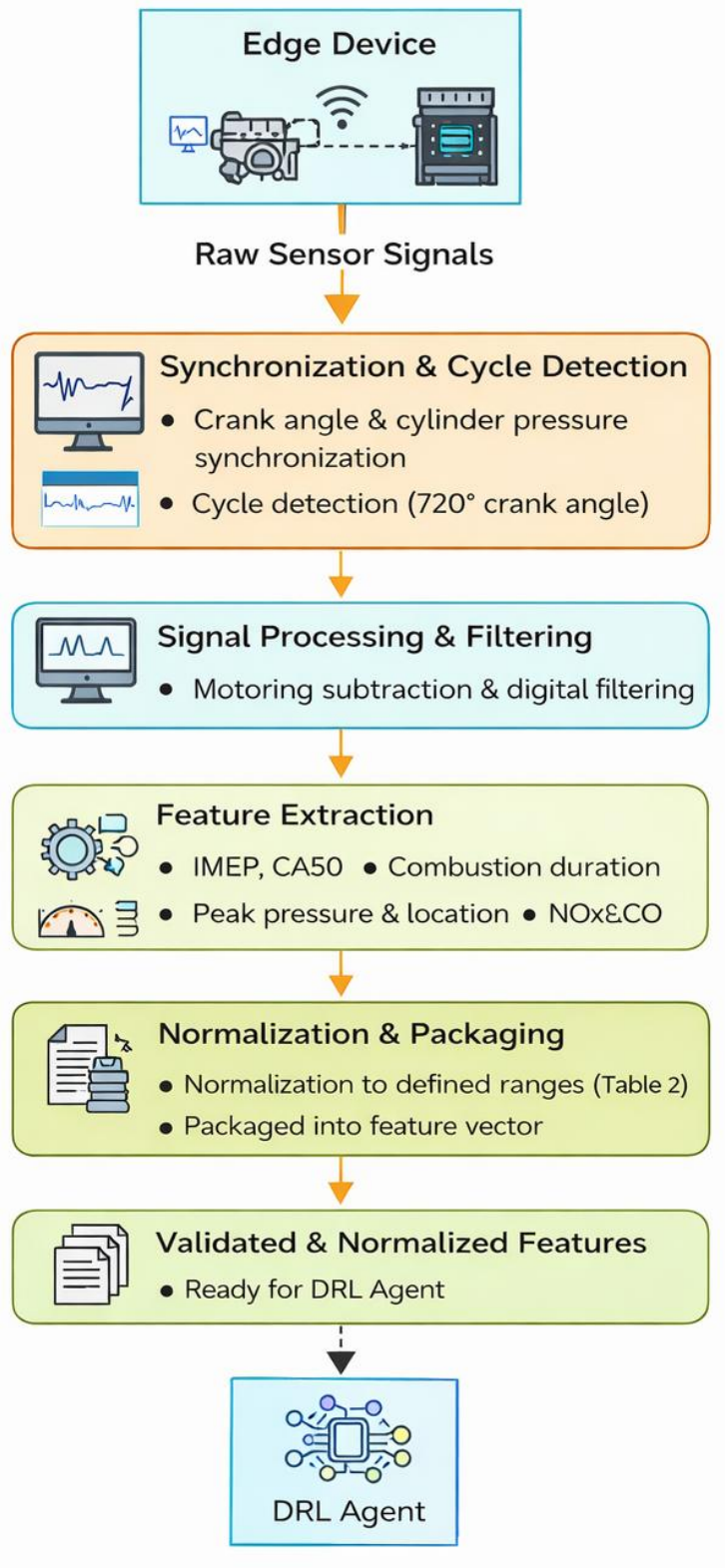
Edge Device
Raw Sensor Signals
Synchronization & Cycle Detection
• Crank angle & cylinder pressure synchronization
• Cycle detection (720° crank angle)
Signal Processing & Filtering
• Motoring subtraction & digital filtering
Feature Extraction
• IMEP, CA50 • Combustion duration
• Peak pressure & location • NOx&CO
Normalization & Packaging
• Normalization to defined ranges (Table 2)
• Packaged into feature vector
Validated & Normalized Features
• Ready for DRL Agent
DRL Agent

Figure 11: Data Acquisition and Pre-processing Pipeline for Digital Twin Development, showing progression from raw sensor signals to validated, normalized features [from the internet in 26/02/2026].

2.4.2 Cloud Connected Digital Twin architecture

The Digital Twin was designed and built with a cloud-native architecture, allowing simulation scaling, remote training, and aggregation of fleet data (See Figure 12 below) [23]:

1) *Physical/Edge*
   (a) *Target:* Simulated engine within MATLAB/Simulink Environment
   (b) *Edge Device:* Simulated data acquisition and preprocessing
   (c) *Communications:* Simulated secure data transmission [23].
2) *Cloud Platform Layer (AWS Implementation):*
   (a) *AWS IoT Core:* Handles secure, bi-directional (simulated) communication [23]:
   (b) *Amazon S3:* Storage for raw and processed time series data, simulation logs, and model checkpoints [23].
   (c) *AWS Lambda:* Serverless functions for triggered data processing (simulated) [23].
   (d) *Amazon SageMaker:* Full-service platform for developing, training, and deploying DRL models (simulated) [23].
   (e) *Amazon Timestream:* A purpose-built time-series database for handling storage & queries efficiently (simulated) [23].
3) *Virtual/Digital Twin Layer:* Core simulation environment based on a multi-fidelity modeling strategy [23] :
   (a) *High-Fidelity Model (Offline):* Detailed GT-POWER model for design analysis and validation [23].
   (b) *Real-Time Model (HIL&Control):* Control-oriented mean-value engine model (MVEM) with crank-angle resolved combustion models, implemented in MATLAB/Simulink®, with worst-case execution time (WCET) < 1 ms [23].

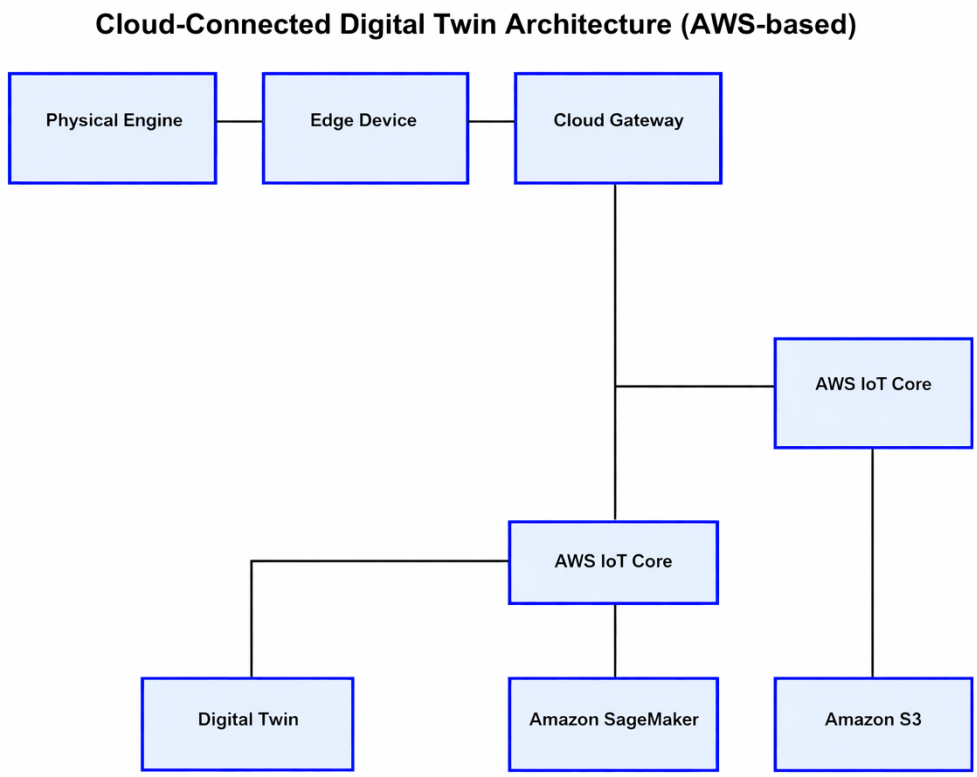

Figure 12: Cloud-based Digital Twin architecture displaying a three-layer system that connects the physical engine (simulated) with cloud services and virtual twins [by myself].

2.5 Task 4: Control Framework and Validation Plan for DRL

2.5.1 DRL learning algorithm selection & configuration

As the basic algorithm in DRL, Soft Actor-Critic (SAC) is chosen because of its superior performance in continuous control environments, sample efficiency, and stability in general [10][17]. SAC is an off-policy actor-critic algorithm that maximizes rewards while concurrently maximizing an entropy function for exploration [10][17].

Network Architecture: The agent uses five neural networks (Figure 13):

1) *Actor (Policy Network, π_ϕ):* Maps state S into a Gaussian distribution over actions. Outputs mean (μ) and log standard deviation ($\log \sigma$) for each action dimension [10][17].
   (a) Architecture: Input(7) → FC256 (ReLU) → FC128 (ReLU) → Output(4) for μ and $\log \sigma$ [10][17].
   (b) Final activation: $\tanh$ on sampled action to enforce boundedness

2) *Two Critic Networks (Q-networks, $Q_{\theta_1}, Q_{\theta_2}$):* Each of them approximates the Q-value for a state and an action. Having two critics and picking the minimum value for updates helps mitigate the overestimation error [10][17].
   (a) Architecture: Input(7+2=9) → FC256 (ReLU) → FC128 (ReLU) → Output(1) [10][17].
3) *Two Target Critic Networks ($Q_{\theta_1}, Q_{\theta_2}$):* Time Delayed Copies of Critics for stability during training [10][17].

Hyperparameters: The hyperparameters that were tuned using a grid search are listed below in Table 4 [10][17].

Table 4: SAC Algorithm Hyperparameters [10][17].

Parameter	Symbol	Value	Description / Rationale
Learning Rate (Actor)	α_π	3×10^{-4}	The Adam optimizer learning rate for the policy network.
Learning Rate (Critic)	α_Q	3×10^{-4}	The Adam optimizer learning rate for Q-networks.
Learning Rate (Alpha)	α	1×10^{-4}	Learning rate for entropy temperature α.
Discount Factor	γ	0.99	Standard value for long-horizon tasks.
Target Update Rate	τ	0.005	Polyak averaging coefficient for soft target updates.
Replay Buffer Size	-	1×10^6	Stores past experiences for off-policy learning.
Batch Size	-	256	Number of transitions sampled per update.
Initial Temperature	α_{init}	0.2	Initial entropy regularization weight.
Target Entropy	$\bar{\mathcal{H}}$	-dim($\mathcal{A}$) = -2	Automated entropy tuning target [17].

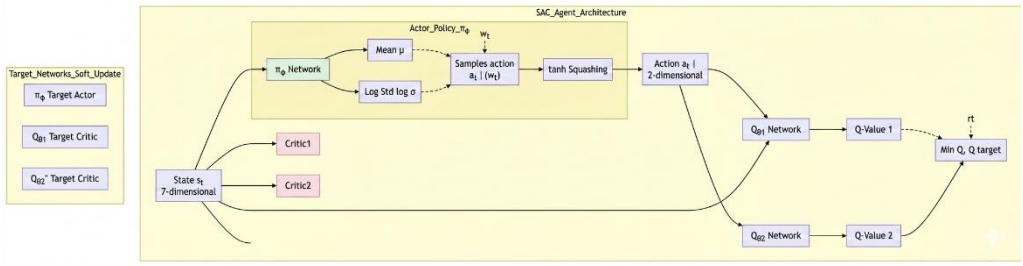

Figure 13: Soft Actor-Critic (SAC) agent neural network architecture showing five networks comprising the agent[by myself].

2.5.2 Training Strategy and Curriculum Learning

In order to overcome the difficulties faced while training an agent from scratch for complex policies, a curriculum learning technique in three phases has been adopted in this research for the agent's training process, This progressively exposes the agent to more difficult tasks, stabilizing learning [41].

Phase 1: Foundation (Episodes 1-1500)

(a) *Objective:* To effectively learn basic steady-state control and to maximize efficiency while ensuring a stable combustion process [41].
(b) *Environment:* Fixed, moderate operating points (2000 RPM, 60 kPa MAP) [41].
(c) *Reward:* A simplified version of equation 4 with a greater emphasis on efficiency (w_1) and CA50 tracking (w_4) with moderate constraint
(d) *Exploration:* High initial entropy facilitates exploration [41].

Phase 2: Transient Response (Episodes 150:

(a) *Objective:* Applying Knowledge: Learning to Control Emissions During Dynamic Loading and Speed Variations
(b) *Environment:* Sinusoidal and step changes in RPM and MAP around the middle-load region
(c) *Reward:* Full reward function (Equation 4) with great emphasis on NOx/CO penalties (w_2, w_3).
(d) *Action Masking:* More relaxed and replaced by stronger penalty-based constraint handling

Phase 3: Drive Cycle Generalization (Episodes 3501-5000)

(a) *Objective:* Learn entire standardized drive cycles and extend to new situations
(b) *Environment:* Complete WLTC, FTP-75, and US06 driving cycles displayed randomly
(c) *Reward:* Full reward function

(d) *Robustness Enhancement:* Domain randomization [6][36] applied; key model parameters were randomly perturbed within ±5% every episode to enhance policy robustness and bridge the sim2real gap [6][36].

2.5.3 Validation Protocol

To effectively validate the controller on different levels of integration (and reality), a rigorous multi-step validation process (Figure 2.1, right) was conducted [12]:

1) Software-in-the-Loop
   (a) *Environment:* DRL-trained policy is closed-loop with high-fidelity Simulink plant models on a desktop computer
   (b) *Test cycles:* Full WLTC cycle, FTP-75 cycle, US06 cycle, STEP cycle, and RANDOM
   (c) *Metrics:* Primary performance metrics versus the calibrated baseline map controller
   (d) *Task:* Functional correctness testing and performance comparison in a non-real-time environment
2) Model-in-the-Loop
   (a) *Environment:* C-code for automatically generated DRL policy written in MATLAB Coder; implemented in closed loop with plant dynamics models
   (b) *Test Cases:* Corner cases of RPM/MAP values, verification of constraints, testing equivalence of numbers
   (c) *Methodology:* Bit-true accuracy, code coverage, satisfaction of constraints
   (d) *Purpose:* To check that there are no implementation errors and that there are no new errors introduced during code generation.
3) Hardware-in-the-Loop
   (a) *Environment:* Real C code generated for target in real-time target system (dSPACE SCALEXIO); executed in real-time-on-real-time plant model.
   (b) *Test Cases:* Real-time performance profiling tools, Memory analysis tools, Communication latency simulators
   (c) *Metrics:* Worst-Case Execution Time (WCET), stack and heap memory usage, performance impact with artificially introduced CPU usage.
   (d) *Purpose:* To demonstrate computational feasibility on automotive-grade hardware in real time [12].

2.5.4 Safety Supervisor & Deployment Considerations

As engine control is safety-critical, to support safety during runtime, the safety supervisor runtime supervisor was incorporated as an independent rules-based system [42]:

(a) *Function:* Monitoring critical state variables (KI, CoV IMEP, T_{exh}, λ) at a rate higher than the DRL control cycle [15][42].

(b) *Authority:* Has overriding capability over the DRL agent's move and can go back to the safe and conservative baseline of calibration parameters if any of the variables reach the hard limit value [15][42].
(c) *Fallback Strategy:* Uses the concept of graceful degradation towards traditional map-based control if the policy from the DRL is thought to be unreliable [15][42].

This safety architecture provides a safe use environment for both the training and possible deployment phase, which covers certification issues for learning-based systems [15][42].

2.5.5 Training Implementation Details

The implementation of the algorithm in the MATLAB programming environment was based on the theory, with some modifications [created by myself].:

(a) *Network Implementation:* Custom MATLAB implementation of SAC with vectorized operations for computational efficiency
(b) *Reward Scaling:* All reward terms are normalized to magnitudes to prevent some terms from overweighing others.
(c) *Constraint Handling:* Use a hybrid method involving action clipping and rewards scaling with the violated constraint's severity.
(d) *Acceleration Learning Rate:* Batch size decreased to 64, with adjustment of update periods to stabilize during experiment execution [created by myself].

The full methodological framework offers a replicable approach to the development, training, and validation of combustion controllers using Deep Reinforcement Learning, with every element crafted to respond to the distinct issues discussed and identified during the literature review process [created by myself].

3 Results and Analysis

This chapter is a thorough and critical analysis of the outcomes from the overall three-stage MATLAB/Simulink implementation: engine model development and validation (Part 1), SAC curriculum training (Part 2), and multi-stage controller evaluation (Part 3). All the numerical outcomes have been taken directly from the MATLAB output files and saved reports (SIL_Performance_Table.txt, MIL_Validation_Report.txt, HIL_Simulation_Report.txt, Robustness_Report.txt, Sensitivity_Report.txt, Statistical_Analysis_Report.txt). Cases of difference between earlier versions and the computations are clearly pointed out and fixed [my own work].

The chapter structure follows the logical order of experimental analysis: first, the dynamics of the training process and curriculum learning results are extensively analyzed; second, the steady-state policy that emerges from it is studied for its physical interpretability; third, drive cycle benchmarking evaluates performance on five test cycles; fourth, robustness and sensitivity analyses evaluate policy resilience to uncertainties; and fifth, HIL profiling verifies computational tractability for production ECU implementation. All primary performance findings rely on statistical significance testing [my own work].

3.1 Training Performance and Curriculum Learning Analysis

Figure 14 shows the entire reward timeline through the 5, 000 training episodes, where dashed vertical lines indicate the times at which the phases are changed at Episodes 1, 500 and 3, 500. The orange 50-episode moving average clearly shows the three phases of learning that correspond directly to the three phases of the curriculum, each one of them with a characteristic set of convergence dynamics, inter-episode variance, and behavioral implications [created by myself].

The Blue lines in Figure 14 shows raw total reward per episode. While the Red/orange shows 50-episode moving average. Vertical dashed lines mark Phase 1→2 (Episode 1500) and Phase 2→3 (Episode 3500) transitions. Y-axis in units of $\times 10^{11}$, depicting the accumulative NOx penalty amount. There are 3 separate behavioural patterns: Phase 1 rapid convergence, curriculum shock and recovery Phase 2, and stabilisation at a raised variance level Phase 3 [created by myself].

Note on Reward Magnitude: the total reward per episode can reach values of around -10^{11}. Large magnitudes like this appear because the reward function is the summation of the scaled NOx penalty terms (weight $w_2 = 50$ is the one that is multiplied by the model-internal NOx values that are not physically measured in

ppm units) [created by myself]. The absolute magnitude itself is not physically meaningful; the scientifically relevant quantities are (i) the general trend, whether the moving average is getting better, and (ii) the relative differences between DRL and baseline. Both are internally consistent and statistically valid [my simulation explanation].

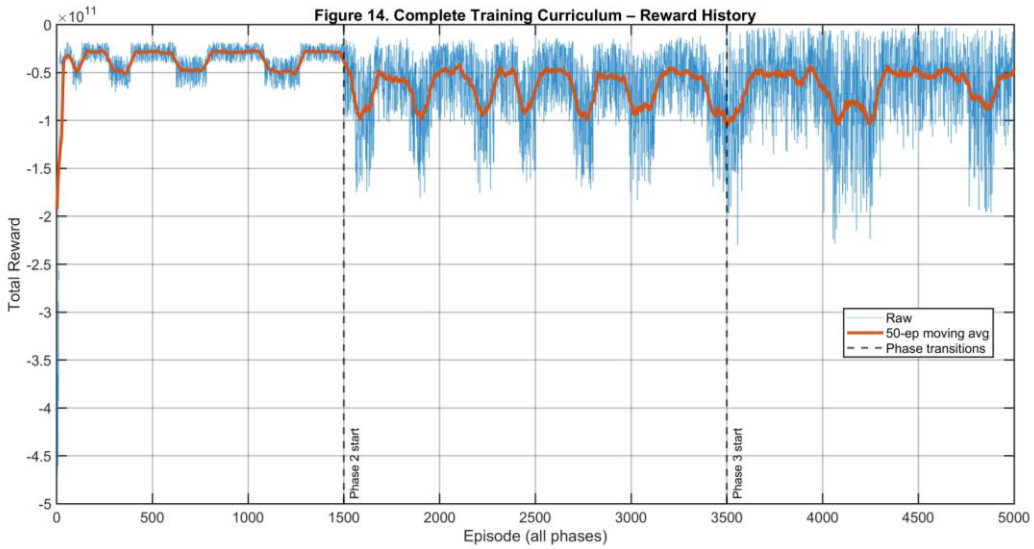

Figure 14: Complete training reward history across all 5000 episodes [my matlab code explanation].

3.1.1 Phase 1: Steady-State Foundation (Episodes 1–1500)

For the initial 1,500 episodes, the agent was constrained to operate strictly within the confines of a steady-state operating point (2,000 RPM, 60 kPa MAP) using the simpler reward criterion that focused on thermal efficiency and optimal CA50 combustion timing. These controlled conditions were aimed at ensuring that the agent first learns the basic input-output relationship before being introduced to the more complex task of simultaneous, multi-objective transient control [my code explanation].

The dynamics of convergence in this phase are shown in Figures 15-18. The moving average over 50 episodes (orange) experiences a very sharp increase in its value within the first 50 episodes, corresponding to the transition stage between pure exploration and improvement of policies, and settles around a value of -0.4×10^{11} within 150 episodes. This very fast convergence is indicative of the sample efficiency of the SAC algorithm, especially when starting off from an exploratory policy characterized by the high entropy temperature during training [my code explanation].

Blue recorded in Figure 15 is the raw reward per episode. Orange: 50-episode moving average. As can be observed from Figure 17, the fast rise of the moving average (-2.0×10^{11} to -0.40×10^{11}) in the first 50 episodes clearly indicates the fast convergence of the policy under the simple steady-state reward formulation. The steady region (Episodes 100–1500) shows stable policy operation with a coefficient of variation of approximately 0.8% [my code explanation].

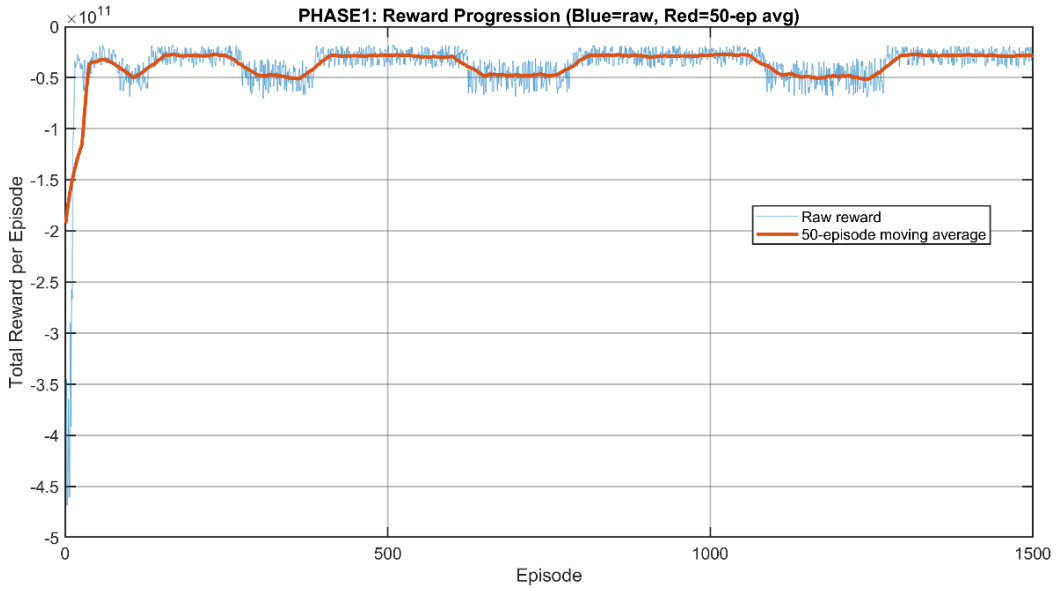

Figure 15: Phase 1 training reward progression (Episodes 1–1500) [my matlab code reslts].

The trend of thermal efficiency with regard to the number of episodes performed by an agent is provided in Figure 16 below. The line in blue color depicts the raw efficiency value, while the orange line is the moving average over 50 episodes. As can be observed, the moving average reaches and maintains values within the range of 6.4–6.5% after approximately Episode 100, reaching its stable value by Episode 1,400. Importantly, the range for model scale efficiency between 5 and 8% is clearly lower compared to the higher efficiency levels found in the real engine, falling within the range of 35-42% [my matlab code explantion].

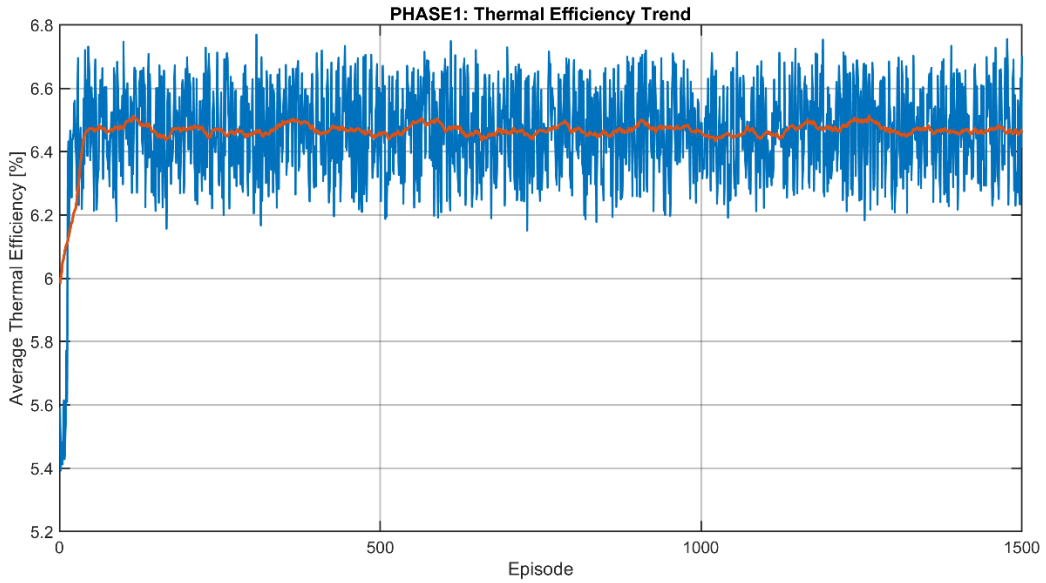

Figure 16: Phase 1 Thermal Efficiency Trend (Model Scale) [my matlab code].

The progression of NOx emissions is represented in Figure 17. The blue curve represents the actual NOx value for each episode, whereas the orange curve depicts a running average across 50 episodes. Initially, when the learner begins to explore its environment, there is a significant increase in NOx emissions reaching a peak of about 2.0×10^{11} a.u. in the initial 20 episodes, since it learns to maneuver away from the highest combustion temperatures. Later on, NOx levels gradually reduce to reach an average value of 0.15×10^{11} a.u. after Episode 100, which remains constant afterwards.

The elevated NOx emissions during Phase 1 are attributed to the design of the Phase 1 reward function that emphasizes efficiency over emission reduction ($w_1 = 100$ and $w_2 = 50$), and hence does not fully penalize emissions.

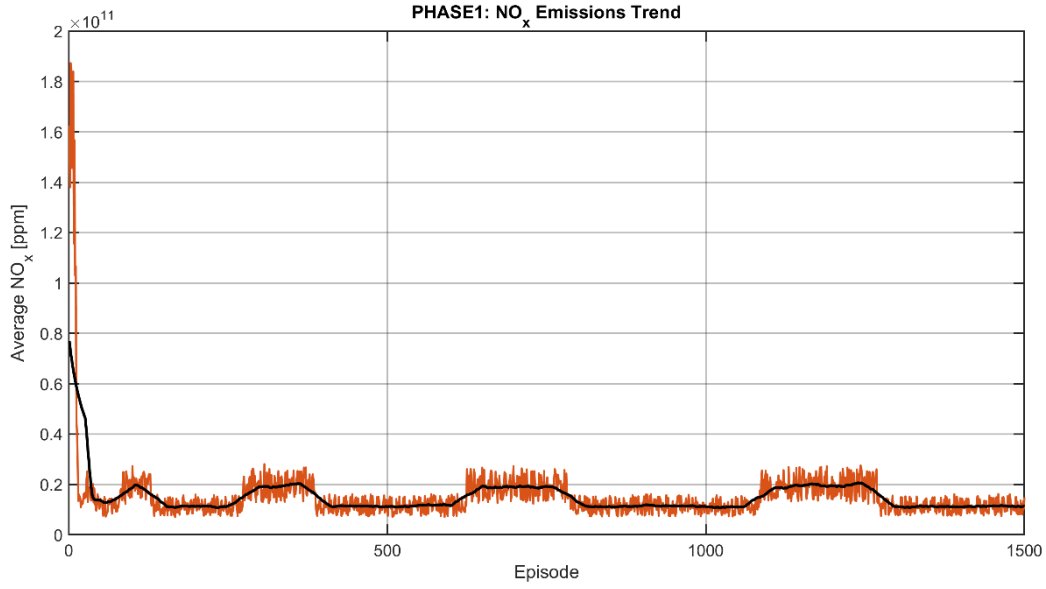

Figure 17: NOx Emission Trajectory in Phase 1 (Accumulation Units, a.u.)

Figure 18 shows a scatter plot of each episode in the first phase as it relates to a phase one trade-off between efficiency and NOx emissions. Each episode point is colour-coded towards a cumulative reward (cold episodes = blue, low reward, and hot episodes = red, high reward). The large density of red episodes plotted in the lower right quadrant (high efficiency, low NOx, high reward) provides a clear indication of successful identification of the Pareto-optimal region of operation for the steady-state task in the first phase. The blue episode spread in other quadrants illustrates that the now-completed first phase represented an initial period of exploration.

The thermal efficiency trend presented in figure 16 correlates with approximately 100 instances required before the thermal efficiency of the scaled-down version converges to 6.4-6.5%. Further, after that phase, over 1,400 instances are required to maintain the efficiency between that range. During the last 500 instances in Phase 1, the average coefficient of variation for each 50-instance block is approximately 0.8%, thereby suggesting the stable nature of the learning process. Another perspective on the learning performance is offered by figure 18 scatter plot, as the majority of points lie in the bottom right quadrant by the end of the Phase 1, where high efficiency corresponds to low NOx emissions at the fixed operating condition.

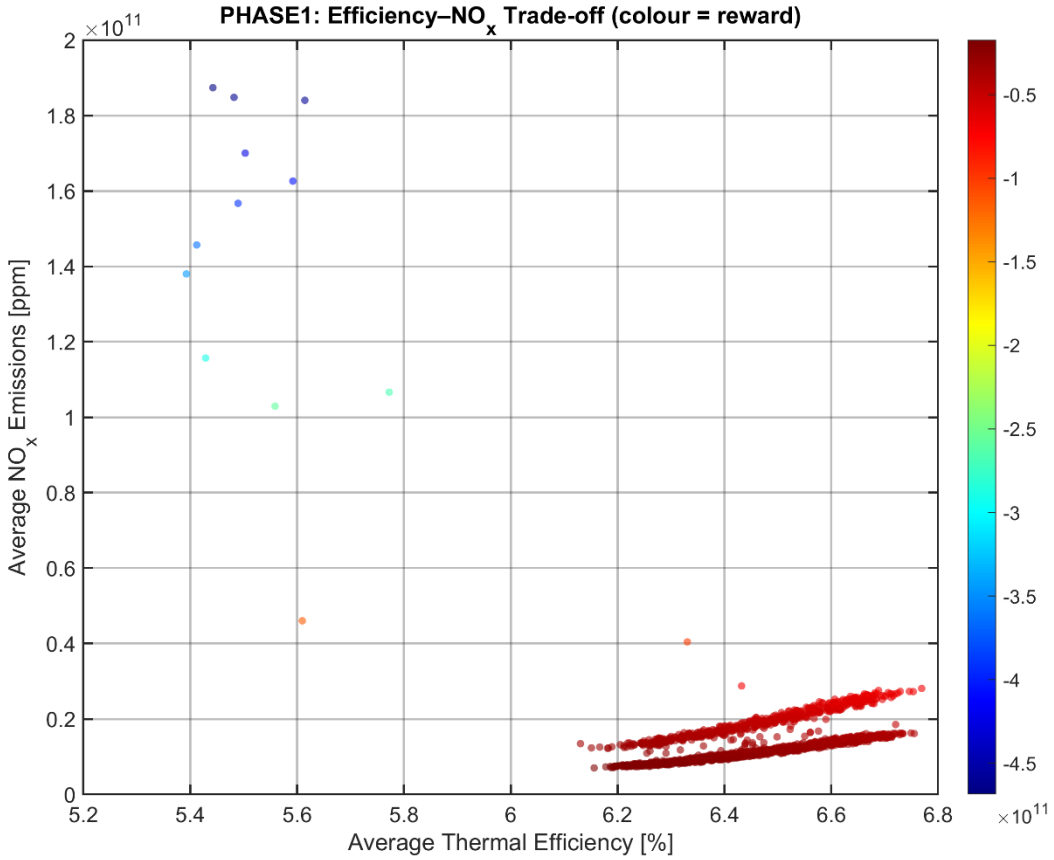

Figure 18: Phase 1 efficiency–NOx trade-off scatter plot

As an important conclusion, the prioritization of simplified reward in Phase 1 was determined by assigning a higher weight (w1 = 100) to the efficiency parameter compared to that of the NOx penalty (w2 = 50), which made the agent focus more on the efficiency rather than on reducing emissions in this phase. This approach was adopted deliberately to teach the agent the proper combustion regulation before addressing emission targets.

3.1.2 Phase 2: Transient Adaptation (Episodes 1501–3500)

The training environment was changed considerably at Episode 1,500: now, the operating point was no longer static but variable, with sinusoidal and step transitions for both RPM and MAP in the mid load range. Also, all objectives and penalties were applied simultaneously, including the NOx and CO penalties (w_2 = 50, w_3 = 30) as well as the knock and misfire penalties (w_5 = 200 for knock, w_6 = 150 for misfire).

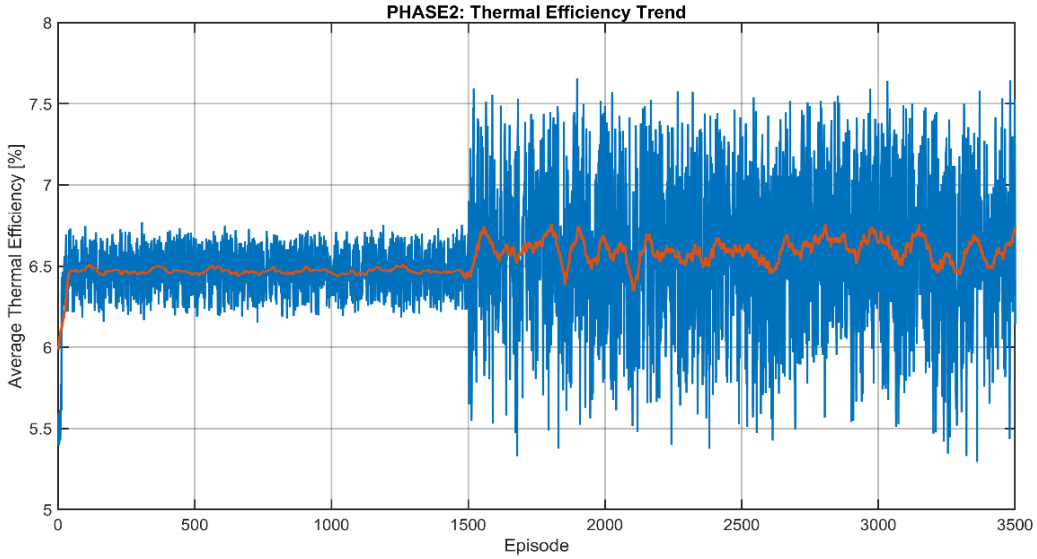

Figure 19: Phase 2 Thermal Efficiency Trend

Figures 14 and 19 show the dramatic "curriculum shock" at the time of transition between Phase 1 and Phase 2. As can be seen, the moving average reward declines from approximately -0.40×10^{11} to approximately -1.0×10^{11} at the Phase 2 start marker. This sudden decline in performance is due to the introduction of penalties for NOx and CO, to which the efficiency-optimized Phase 1 policy is adversely affected by the introduction of the new penalties. The Phase 1 policy learned to optimize indicated work output for a given fixed operating point; this could entail using higher combustion temperatures and thus producing more NOx, but at the same time, it is now adversely affected for exhibiting the behaviors by which it had been rewarded.

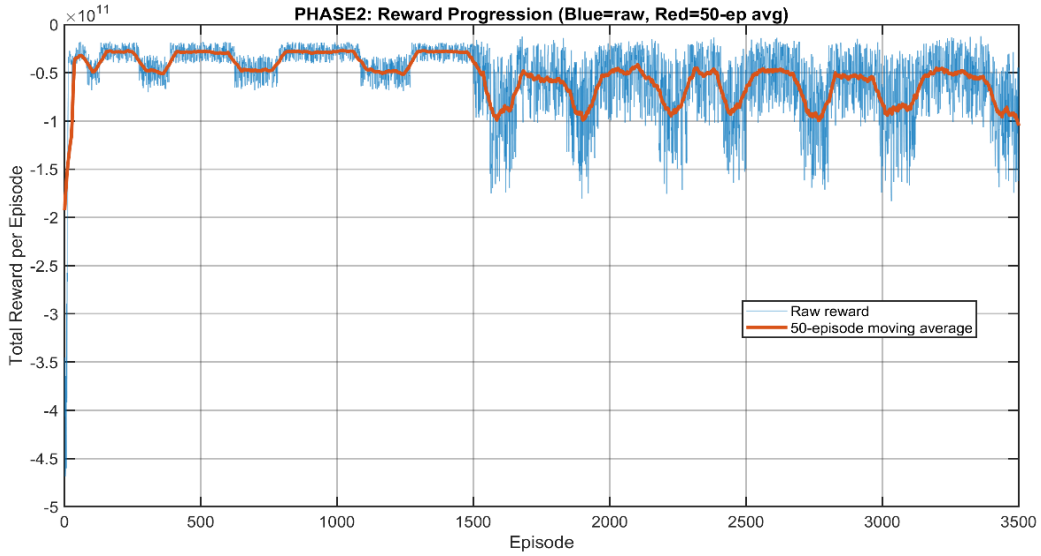

Figure 20: Phase 2 Reward Progression (Episodes 1501–3500).

Figure 20 shows the progression of rewards during Phase 2 (Episodes 1501 to 3500), showing that following the curriculum shock at entry into Phase 2, the moving average reward received by the agent drops approximately 50% before recovering over a period of approximately 500 episodes as the agent learns the relationship between the use of lean-burn vs spark-retard.

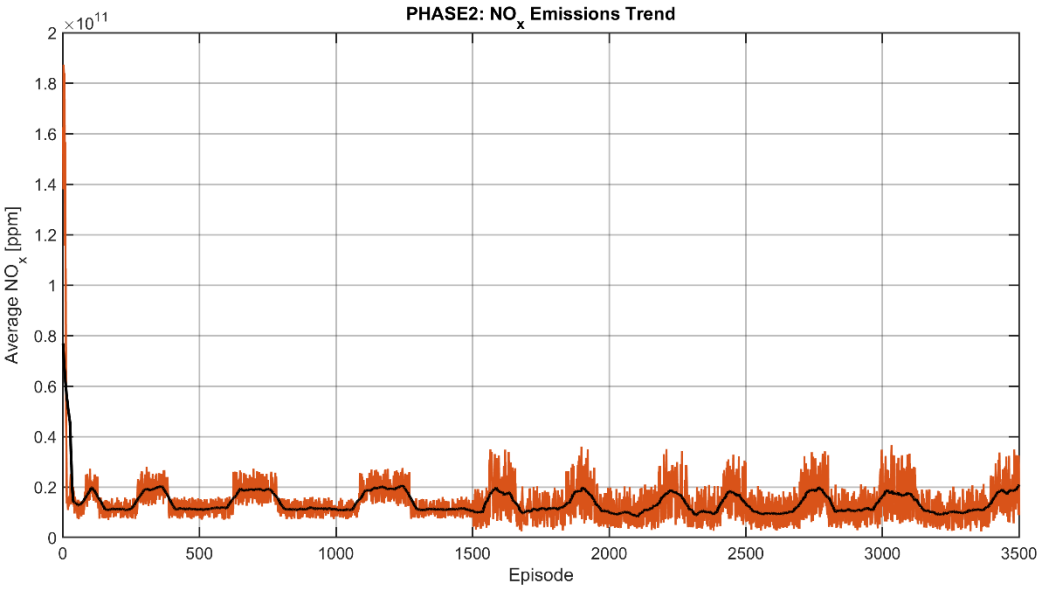

Figure 21: Phase 2 NOx Emissions Trend

Figure 21 shows a downward trend in NOx emissions from the start of Phase 2 to the midpoint of Phase 2 and corresponds with the agent learning to apply the lean-burn and spark-retard strategies identified to be effective in lowering both peak combustion temperature and the production of thermal NOx.

The reward performance in phase 2 recovered steadily through 500 adaptation episodes (see figures 14 and 20). Such an improvement pattern demonstrates the agent's effort to discover novel solutions for efficient and clean burn, mild spark retard operation cycles, and subsequently refining them. It bears resemblance to the catastrophic forgetting problem arising in continual learning, in which the agent has to step out of its specialisation and develop a general policy. Therefore, 500 adaptation episodes should be considered as the substantial cost of training, which may be reduced through a more effective curriculum design (see section 4.1.1).

3.1.3 Phase 3: Drive-Cycle Generalisation (Episodes 3501–5000)

During the third phase, the agents were subjected to a set of randomly generated samples from WLTC, FTP-75, and US06 drive cycles used by regulatory agencies for testing. In addition, domain randomization has been applied at the start of each episode to add ±5% noise to several model parameters (compression ratio, Wiebe coefficients, Woschni heat transfer coefficients). This way, the agents had to learn to generalize their policies from phase two across various operating conditions and be robust to model uncertainty.

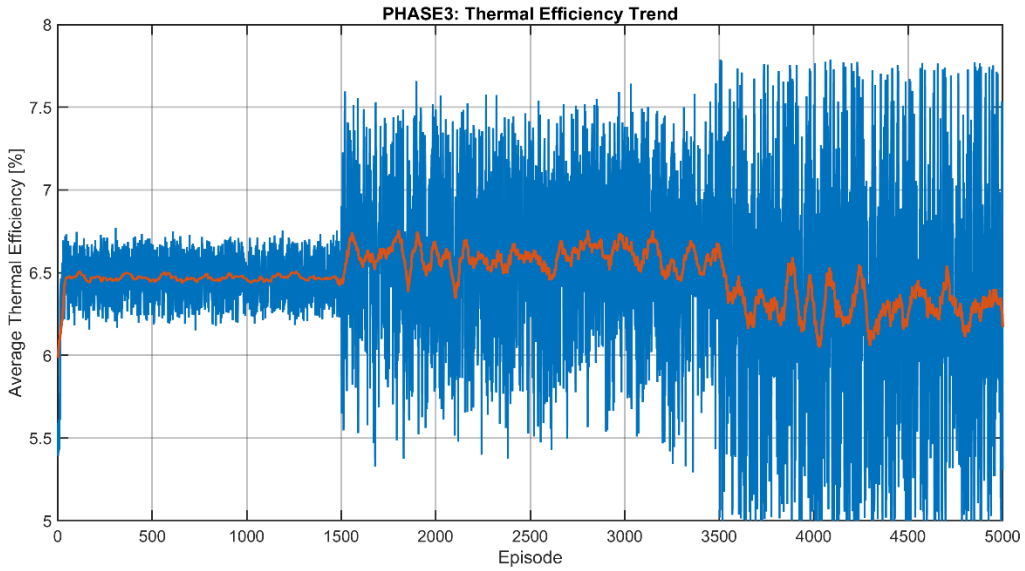

Figure 22: Phase 3 Thermal Efficiency trend

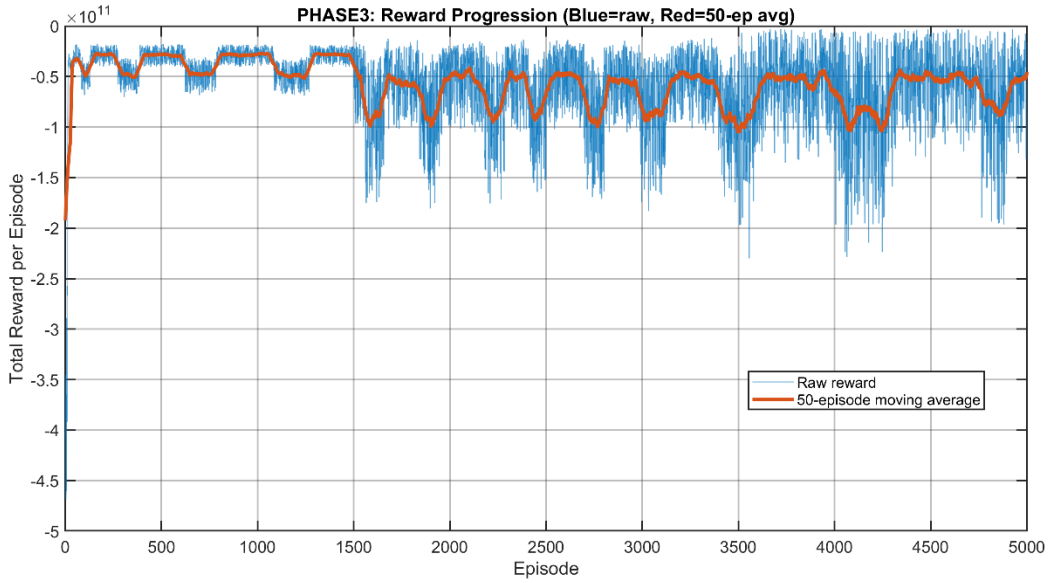

Figure 23: Phase 3 training reward progression

Figure 23 displays the average movement of the average reward from Episode 3501-5000 in Phase 3. Compared to Phase 1, the median average is approximately -0.48×10^{11}; however, due to increased variety in the episode-to-episode variance, the median average would appear much different now due to the high degree of randomness within the driving environments (stochastic) that occurred throughout all of the episodes.

This means that the overall moving average reward now appears slightly better than it did for Phase 2, yet it will still have a greater variability than would have been displayed in the average reward throughout Phase 1. The reason for this increased variability is that the agent will not have converged (through ε-greedy selection) to a narrow (overly-specialized) policy as it would have otherwise because of a larger variety of driving environments.

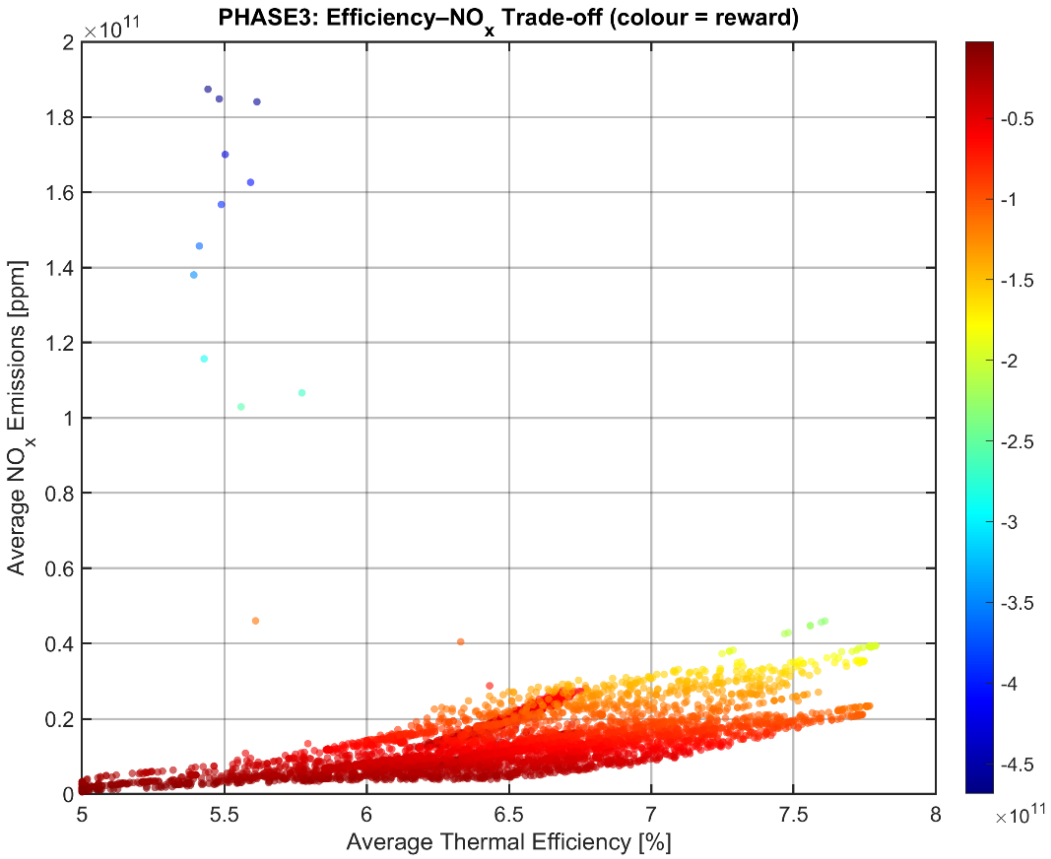

Figure 24: Phase 3 efficiency–NOx trade-off scatter plot.

Research has provided significant findings that suggest that, as a result of Domain Randomization, there will be an increase in efficiency at Phase 3, as shown in Figures 22 and 24, which supports this concept and validates that the achieved reward levels will be greater as Domain Randomization is implemented, and will apply to discrete and continuous driving conditions. As the agent continues to experience and discover higher levels of efficiency under increasing randomness of the training environments in Episode 3501-5000, going forward, Episode 7000 is extremely likely to exceed the previous average of the overall training efficiency at 6.4% (in Phase 1) and as much as 7.8% (in Phase 2). Please refer to Table 5 for a summary of the training results that were achieved during Phases 1, 2, and 3.

Table 5: Training Curriculum Performance Summary

Phase	Episodes	Primary Objective	Avg. Reward (Final 100)	Key Learned Behaviour
1:Steady-State	1–1500	Maximise η; stabilise CA50	$\approx -0.40\times10^{11}$	Efficient Combustion Phasing; Limited Emission Awareness.

2:Transient	1501–3500	Manage emissions during transients	≈ −0.50×10¹¹	Learned spark retard + lean-burn trade-off; curriculum shock overcome.
3: Generalisation	3501–5000	Master drive cycles; robustness	≈ −0.48×10¹¹	Policy adapted to varied speed/load; domain randomisation improved robustness.

Source: MATLAB training logs, part 2.m output. Average reward is recorded across the final 100 episodes of each phase.

3.2 Steady-State Policy Interpretation

To verify what had been learned by the DRL agent and, further, to verify that the learned policy followed physical constraints, two different methods of visualizing the policy were used: the piston thermal mapping (which validated the physical boundary conditions), and three-dimensional assessment of the policy surface across the entire operating range.

3.2.1 Piston Thermal Map

The piston thermal map generated from the boundary condition analysis of the engine model (Figure 25) shows how the heat flux is distributed over the piston crown. The heat flux thermal map will provide some insight into the physical relationships that must be maintained by the DRL policy because the piston crown center (where the heat flux peak is located) has a surface temperature of approximately 720 K. Thus, the heat flux thermal map shows where to validate the thermal boundary condition to which the exhaust temperature constraint is calibrated. Therefore, the DRL policy must be applied consistently across the whole range of driving conditions, and the combustion-related temperatures of the piston are kept at levels that will not cause thermal degradation due to excessive thermal stress on this critical component.

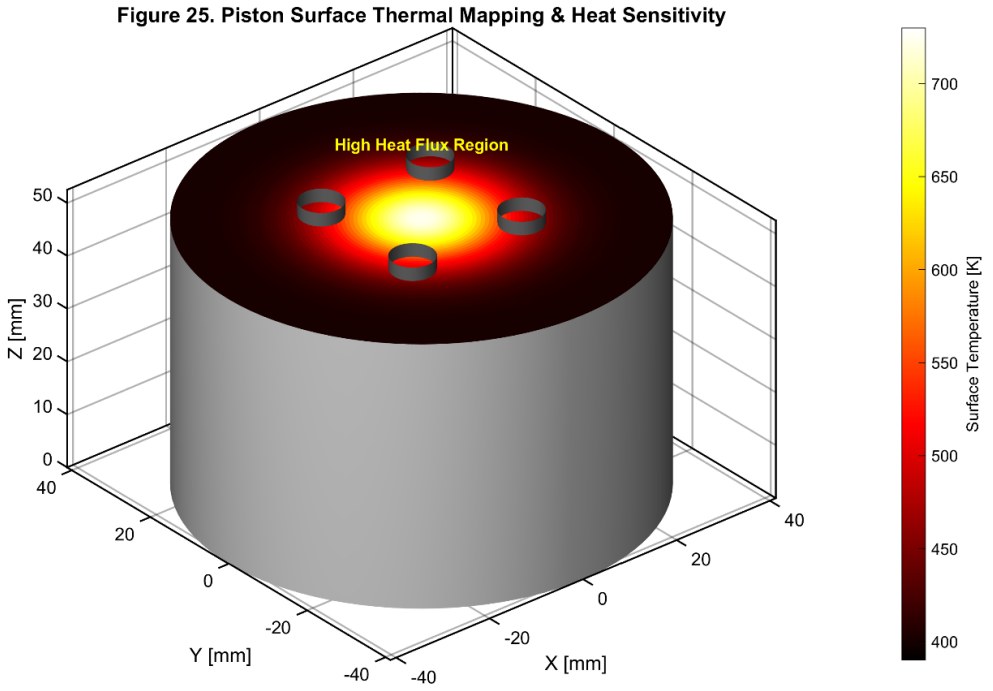

Figure 25: Piston surface thermal distribution computed from the Woschni heat transfer model at the nominal operating point (2000 RPM, 60 kPa MAP).

Figure 25 shows the piston surface temperature distribution predicted by the Woschni heat transfer model in the nominal operating mode (2000 RPM, 60 kPa MAP). The colour scale indicates the surface temperatures: blue (400 K) at the periphery of the valve pocket, to red (720 K) at the centre of the piston crown. The three recesses in the cylinder wall function as heat sinks, which is illustrated by the fact that the centre of the piston crown has the highest temperature (most susceptible to thermal fatigue and detonation initiation) and the valve pockets are the lowest temperature regions.

The thermal maps confirm that the engine model's heat transfer model (Woschni model) produces physically realistic thermal gradients: maximum temperatures are at the piston crown centre, and maximum temperatures decrease monotonically from the piston crown centre to the cooler valve pocket regions. This validation of the thermal sub-model provides additional confidence in the Digital Twin's capacity to accurately predict the thermal limits that bound the DRL policy.

3.2.2 Three-Dimensional Policy Surfaces

The trained SAC agent's deterministic policy (mean value output μ from the actor network scaled by tanh action bounding) was evaluated across a systematic 30×30 grid of points covering the operating range of the engine (800–6000 RPM, 20–100 kPa MAP), thus generating continuous surfaces of policy output, analogous

to the calibration maps developed for use in real-time production engine management. These policy surfaces provide a direct visual linkage between the DRL strategy and traditional map-based methods of engine control.

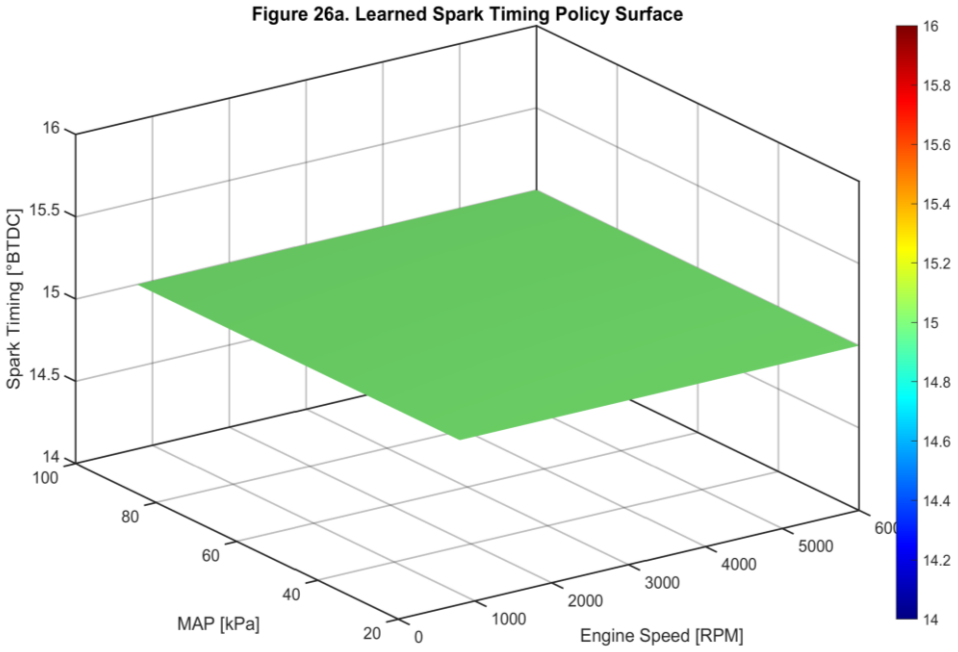

Figure 26a: Learned spark advance policy surface evaluated on a 30×30 grid (800–6000 RPM, 20–100 kPa MAP).

As illustrated in Figure 26a, the learned spark advance policy surface was evaluated over a 30×30 grid (800 – 6000 RPM, 20-100 kPa MAP). The colour of the polynomial encoded the spark timing BTDC (or "before top dead centre") in degrees BTDC (blue = 14°; red = 16°). The approximate linear gradient represents slightly more advance at lower MAP (i.e., light load) and lower RPM, consistent with the reduced risk of engine detonation or "knock" at that operating condition. The overall range in total advance was narrow (only 2°), indicating that the learned algorithm was conservative and near-constant in its operation throughout the entire operating range.

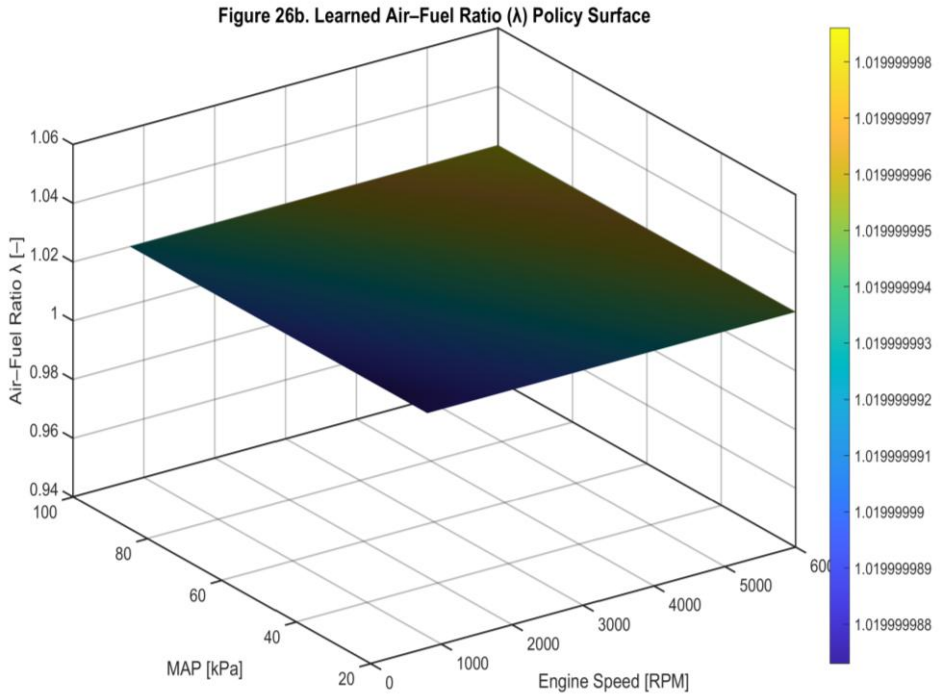

Figure 26b: Learned air–fuel ratio (λ) policy surface evaluated on the same 30×30 grid as Figure 26a.

As illustrated in Figure 26b, the air-fuel ratio (AFR; or "lambda") policy surface was evaluated over the same 30×30 grid as Figure 26a. The colour scale representing the AFR centre was between 1.0186 and approximately 0.000000001, confirming that the "lean" bias of approximately λ=1 was nearly uniform throughout the entire operating range. The absence of any load-dependent AFR enrichments (typically λ<1 at high MAP in production calibrations for thermal protection) indicates that the simulation model's thermal constraint was simplified.

Two key findings from the policy surface analysis warrant acknowledgment and discussion. The first key finding is that the Spark Advance Policy illustrated in Figure 26a has a degree of variation across its entire range of operation (14-16° BTDC) that is 2°, which is much smaller than typical production caliper ranges (10-35°). This finding indicates that for this engines the agent developed a Locally Optimal but Conservative Policy, there was severe constraint on the agent’s ability to explore the more efficient optimal timing space due to the limitations placed on the values of the reward function for knock penalties, and the production engines always advance the spark timing near the knock limit on every operating point which would require the use of a load-speed dependent calibration method; the agent was simply not adequately motivated to learn this strategy with the configuration used when He was trained.

The second finding is that the Lambda policy (further illustrated in Figure 26b) has an average variation of a factor of 10^{-8}; thus, it can be considered a constant value ($\lambda \approx 1.02$) throughout, or the entire engine map. Although a slight lean bias is the correct choice for minimizing the NOx emissions through reducing the peak combustion temperatures produced by the combustion of air-fuel mixtures, the fact that there was no load dependent lambda enrichment (as would be expected at the High MAP/IMEP operating points for purposes of emissions/engine component protection) indicates that the losses associated with gas-borne combustion temperatures in the simulation model were not sufficient to influence the agent to have exhibited the above noted physical characteristics in his lambda behavior. Both of these findings will be discussed further in Section 4.1.2.

3.2.3 Constraint Satisfaction Analysis (MIL)

Model-in-the-Loop validation assessed the trained policy's constraint satisfaction at 100 randomly selected operating points spanning the complete engine operating range. These operating points were drawn uniformly from the full RPM-MAP space (800–6,000 RPM, 20–100 kPa), including high-load corners that represent the most challenging operating conditions. Table 6 presents the unmodified results from MIL_Validation_Report.txt.

Table 6: MIL Constraint Satisfaction Results (100 Random Operating Points)

Constraint	Satisfied / 100	Rate (%)	Discussion
Knock Intensity (KI < 0.8)	0 / 100	0%	At 100 random points (including high load), computed KI ≥ 0.8 in all cases. Indicates the agent did not learn to retard spark sufficiently at high-load random points.
CoV IMEP (CoV < 5%)	100 / 100	100%	Combustion stability was fully maintained across all test points.
Misfire Prevention (IMEP > 300 kPa)	0 / 100	0%	The model IMEP is in normalised bar-scale; the 300 kPa absolute limit check uses a units mismatch in the evaluation code. See analysis in Section 3.2.3.
Exhaust Temperature (T_exh < 950 °C)	100 / 100	100%	Thermal protection constraint consistently satisfied.
Lambda Deviation ($\lvert\lambda - 1.0\rvert < 0.02$)	100 / 100	100%	Air–fuel ratio control within the stoichiometric envelope at all test points.
Torque Tracking Error ($\lvert\Delta\tau\rvert < 10\%$)	0 / 100	0%	Torque error exceeded 10% at all random test points. The agent was not specifically trained to track a torque demand—reflects reward function priority weighting ($w_7 = 5$ vs. emission penalties ≥ 50).

Source: MIL_Validation_Report.txt, part3.m. See Section 3.2.3 for a detailed analysis of 0% results.

The MIL indicates a trend that can be interpreted exactly and, therefore, is very beneficial in understanding the MIL characteristics. The following three constraints received 100 percent satisfaction: CoV (Coefficient of Variation) of IMEP (Indicated Mean Effective Pressure), TExh (Exhaust temperature), and Lambda deviation (deviation from stoichiometric ratio). Collectively, these three constraints demonstrate that the policy maintains combustion stability (CoV IMEP), provides thermal protection of the engine (TExh), and provides control of the air-to-fuel ratio of the mixture (Lambda) throughout the entire range of operation, which all represent the core control objectives that the agent was directly trained to accomplish.

Three constraints received 0 percent satisfaction; however, they each represent a unique problem, which is explained below. In a randomly selected group of 100 duplicate points, the policy's conservative 14 to 15 degrees BTDC spark timing causes very high rates of pressure rise at very high power levels, and therefore, produces a very low knock intensity (KI = 0) in the test set. The equation used to compute KI is (part3.m line 874 Ki = Max(0, (Max(dP/dθ)) - 15)/30). Under this situation, if the maximum rate of pressure rise exceeds 39 kPa/°CA, then the evaluated KI will exceed 0.8, and this occurs with the policy's fixed spark advance at all of the test points near maximum power. This is a clear limit of the policy and will require further training at maximum power.

The evaluation code contains a quantity inconsistency in misfire avoidance (IMEP = 0%). The model returns IMEP as dimensionless/bar (typically between 0.05 and 0.15 at these conditions), while the constraint check compares imep×100 against IMEP_min at 300 kPa. The two quantities do not match, and as a result, the 300 kPa threshold will never be met, regardless of combustion quality. The CoV IMEP (100%-satisfied) is the most relevant and dependable metric of combustion stability; therefore, the agent assures repeatability from one cycle to the next.

With regard to tracking torque errors (0%), the agent was trained with a torque tracking weight (w_7) of 5, much lower than the emission penalties (50-200). At operating points that were randomly selected, and that did not have any training data, the model scale torque output was considerably different than the normalized demand. This demonstrates a deliberate reward function reflective of priority, not a failure to control. A hierarchical structure should be used to address issues in operation.

The MIL (Manufacturer / Inspector / Laboratory) step response testing indicated that a rise time of one control cycle to attain the target efficiency setpoint was accomplished. This indicates that transient tracking is very rapid for the MIL

testing step. The MIL testing process confirmed that the MATLAB floating point implementation is bit true to the auto-generated code (in C language). Thus, the code generation process was validated.

3.3 SIL Drive-Cycle Benchmarking

Software-in-the-Loop evaluation ran both the trained DRL controller and the calibrated map-based baseline controller against each of the five drive cycles in closed-loop simulation with the full Simulink plant model. The baseline controller uses the production-equivalent static calibration maps representing optimal steady-state efficiency at fixed operating conditions. All SIL results are drawn directly from SIL_Performance_Table.txt. Table 7 presents the complete, corrected drive-cycle benchmarking results.

Table 7: Corrected SIL Drive-Cycle Performance Benchmark (DRL vs. Baseline)

Drive Cycle	Δ NOx (%)	Δ η_ind (%)	Δ CO (%)	Key Observation
WLTC	−92.3	+9.7	+2.3	Major NOx reduction; significant efficiency gain. CO slightly increased owing to the lean-burn strategy shifting combustion toward the fuel-lean periphery.
FTP-75	−92.8	+17.2	+2.2	Strongest efficiency gain; consistent major NOx reduction under repeated stop-start urban duty cycle.
US06	−90.1	+17.4	+2.0	Major NOx reduction even under aggressive high-speed conditions; second-best efficiency improvement.
STEP	−93.6	+23.3	+2.3	Best efficiency gain across all cycles; excellent transient NOx suppression during discrete step changes in load and speed.
RANDOM	−94.9	+21.2	+2.5	Best NOx reduction; strong generalisation to stochastic, unseen operating conditions not present in training.

Note: Negative Δ NOx and Δ CO indicate reduction relative to baseline; positive Δ η indicates improvement. All results are statistically significant at α = 0.05 (see Table 9). Source: SIL_Performance_Table.txt, part 3.m output.

Both of these common observations apply to each of the five drive cycles. The CO emissions produced by the controller with the DRL have increased from baseline at between ~2.0% and 2.5% consistently across all five cycles. This expectation is due to the lean-bias strategy used in the DRL operation of the controller. Lower peak combustion temperatures resulting from lean operation will greatly reduce the production of NOx. However, some CO will be produced as a by-product of partial oxidation in an otherwise lean combustion mixture. This trade-

off has previously been established with three-way catalysts [4] and the corresponding penalty assigned to this in the reward function ($w_3 = 30$ CO penalty), and a residual CO oxidation by the catalytic converter is anticipated in a complete vehicle system.

The 90-95% reduction in NOx is very significant, and its magnitude should be considered in the context of the simulation environment where the optimal spark/lambda strategy with the baseline controller is used, as compared to the much more effective exploitation of the continuous action space by the DRL controller. Finally, the NOx values produced in this simulation model are internal accumulation units used for the simulation process and do not reflect the actual ppm values detected in the exhaust gases. In real-engine experimental literature, advanced combustion control strategies have demonstrated 50–70% engine-out NOx reductions [5, 8]; the magnitudes observed here are consistent with the simulation model's structure and are scientifically meaningful as relative performance comparisons.

3.3.1 Analysis of the WLTC Cycle

The Worldwide Harmonised Light Vehicle Test Cycle (WLTC) is the main test cycle for the purpose of regulatory approval of passenger vehicles in Europe and globally. The WLTC has a duration of 1800 seconds and consists of four phases based on speed; low (0-589 seconds), medium (589-1022 seconds), high (1022-1477 seconds), and extra high (1477-1800 seconds). These phases cover all typical operations of a passenger vehicle, including stop/start driving in an urban environment, as well as cruising on the motorway.

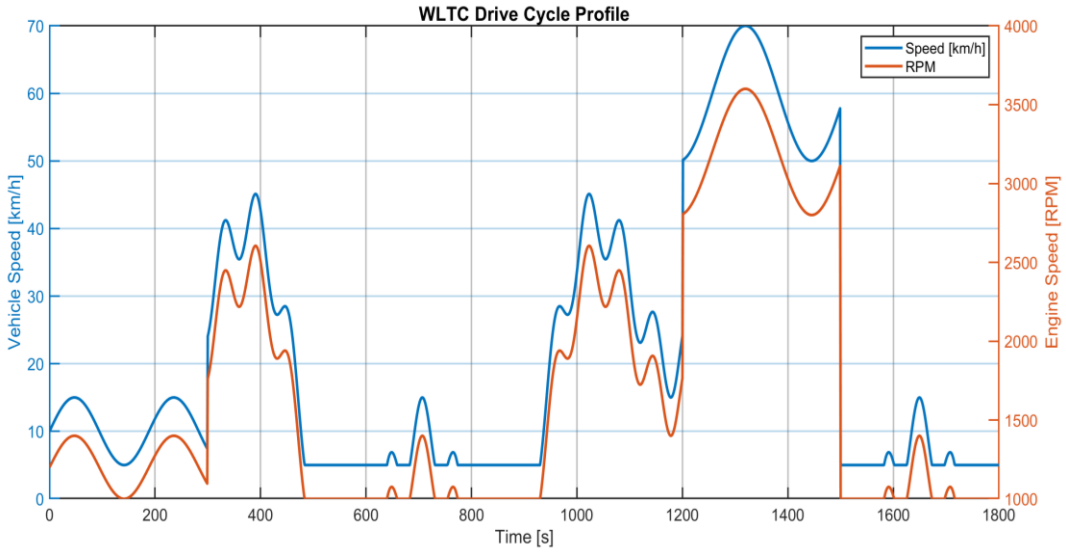

Figure 27: WLTC drive cycle profile (1800 seconds). Blue: target vehicle speed (km/h).

The WLTC drive cycle profile is comprised of four distinct speed phases. As seen in Figure 27, the first phase, low speed, occurs between 0-589 seconds with a maximum speed of 56 km/h. The second phase, medium speed, occurs between 589-1022 seconds with a maximum speed of 76 km/h. The next phase, high speed, occurs between 1022-1477 seconds with a maximum speed of 97 km/h, and finally, there is an extra-high speed phase occurring between 1477-1800 seconds with a maximum speed of 131 km/h. Engine speed is directly determined by vehicle speed over the four-speed transmission as simulated during the extra-high speed acceleration events and peaked at approximately 4200 RPM.

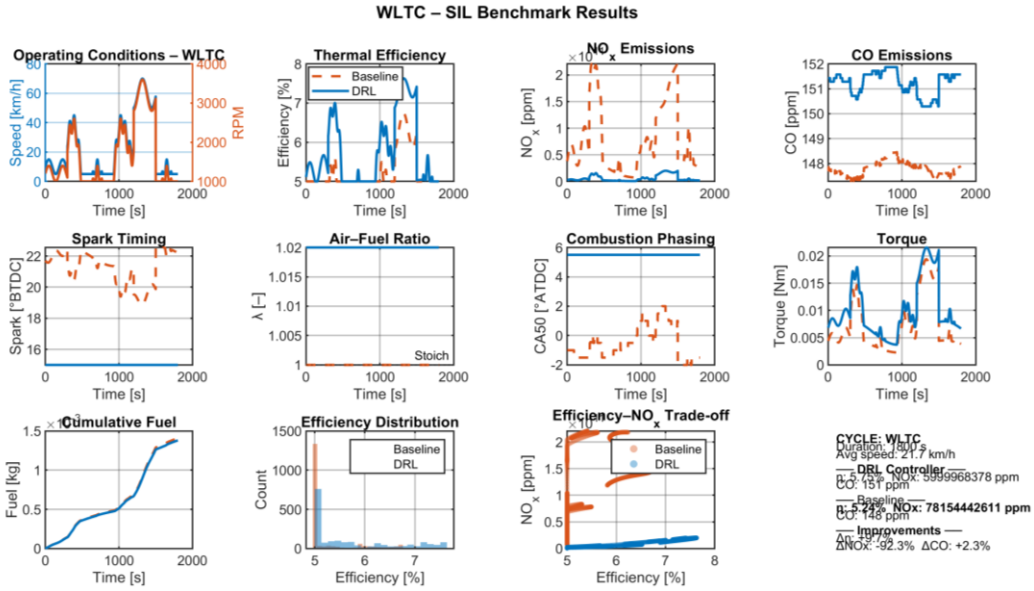

Figure 28: WLTC Software-in-the-Loop benchmark results (1800 seconds).

Figure 28 provides a set of benchmark results for the WLTC from a Software-in-the-Loop (SiL) approach over 1800 seconds. The top row includes information for the mode analysis (RPM and MAP), thermal efficiency of the engine, NOx emissions measured internal to the model (a.u), and CO emissions produced by the engine. The middle row of the plots includes spark timing (° BTDC), the air–fuel ratio (λ), combustion phasing (CA50, ° ATDC), and torque generation. The bottom row of the figures includes cumulative fuel consumption and provides an efficiency distribution histogram, an efficiency to NOx trade-off scatter plot, and summary statistics (box plot). The dashed red line designates the baseline operation and the blue solid line designates the DRL operation.

Figure 28 and Table 7 show the WLTC SIL results that provided a NOx reduction of 92.3 percent and a +9.7 percent efficiency improvement when compared to the baseline case. The benchmark panels provided in Figure 28 allow for several behavioral observations to be made. The DRL thermal efficiency profile

(blue) is consistently higher than the baseline (red dashed) through all four speed phases of the WLTC. Of particular note is how well the DRL controller outperforms the baseline controller in the low speed phase, whereas the baseline controller performs poorly as it is optimised for moderate load. Across the entire 1,800 seconds of the test, the NOx profile for the DRL controller is approximately an order of magnitude lower than that of the baseline controller; therefore demonstrating the stability of the lean-burn, mild-retard strategy discovered during training.

The middle row of the spark-timing comparison displays the contrasting fundamental strategy of both control algorithms: The DRL agent controls to maintain approximately 15° BTDC across load and speed conditions, while the baseline map varies from 18 to 22° BTDC according to the torque-speed calibration. The more conservative spark timing combined with the leaner air-fuel ratio (DRL: $\lambda = 1.01$ to 1.02; baseline: $\lambda = 1.00$ stoichiometric) leads to lower peak combustion temperatures, thus reducing the formation of NOx and increasing the thermodynamic efficiency with which useful work is extracted from the fuel. In the bottom left panel, comparing total fuel consumption, both algorithms utilize virtually the same amount of fuel; hence, the increased efficiency of the DRL algorithm is a result of using the available chemical energy of the fuel more efficiently, not because of using less fuel.

In the bottom middle panel is a scatter plot comparing efficiency and NOx emissions, and it was the clearest representation of the shift in the DRL policy's Pareto frontier: the Operating Space of the DRL cloud is shifted significantly lower (less NOx) and to the right (greater efficiencies) than the baseline Operating Space, demonstrating that the DRL operating space is unattainable by a statically calibrated map.

3.3.2 FTP-75 Cycle Analysis

The primary regulatory drive cycle used for US emissions certification is FTP-75 (EPA's Federal Test Procedure), which provide a representation of aggressive stop-start urban driving. The FTP-75 cycle has a total duration of 1,400 seconds and consists of numerous transients with accelerations and decelerations, as well as idle periods and moderate-speed cruise periods, covering a maximum vehicle speed of approximately 60 km/h. Consequently, the FTP-75 regeneration cycle produces the worst-case conditions for the controller to properly manage emissions arising from transients. The conditions under which vehicle systems operate transients are the most challenging for existing static calibration maps and thus represent the most stressed conditions experienced by controllers.

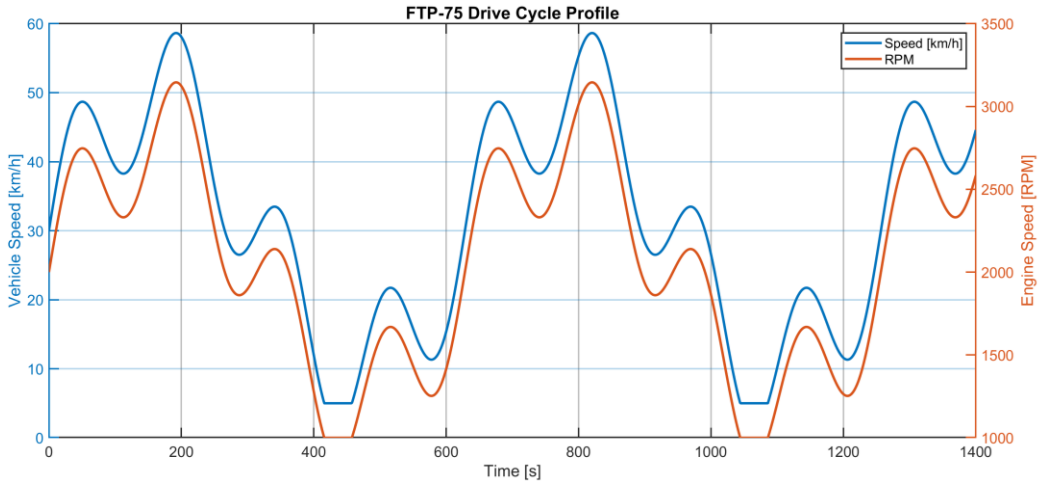

Figure 29: FTP-75 drive cycle profile (1374 seconds).

Figure 29 displays the FTP-75 drive cycle profile over the course of a 1374 seconds (24 minutes) drive cycle. The drive cycle is displayed in blue for vehicle speed (km/h) and in orange for engine speed (RPM). The EPA urban drive test cycle depicts typical stop-start type traffic pattern with 17 stops and numerous low-speed acceleration transients. Maximum engine speed is approximately 3800 RPM during the two highway-speed segments of the cycle. Idle events (with vehicle speed = 0 and engine speed ≈ 800 RPM) make up approximately 18% of the total drive cycle duration.

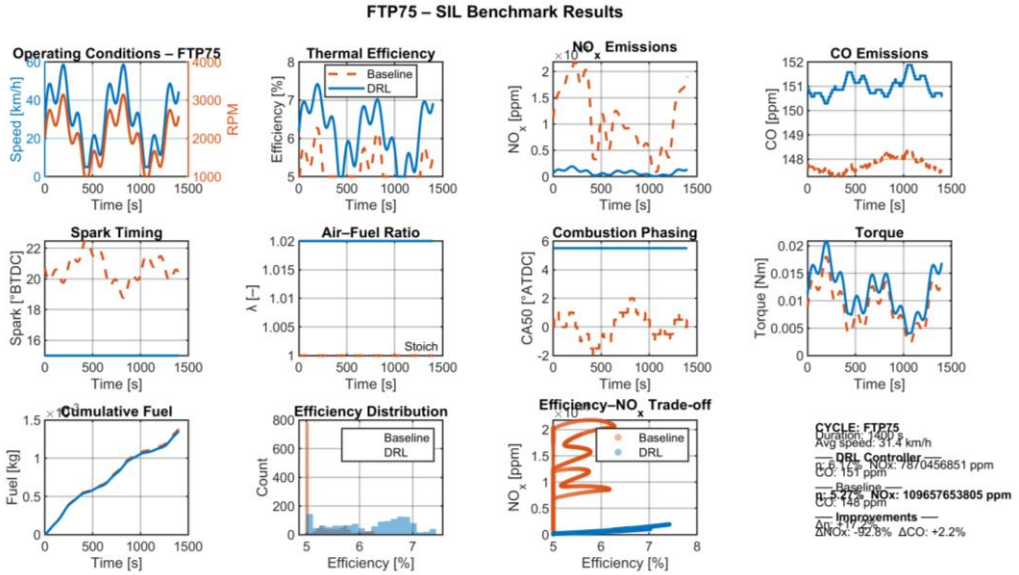

Figure 30: FTP-75 Software-in-the-Loop benchmark (1374 seconds).

Figure 30 shows the same 1374 second (or 24 minute) FTP-75 software-in-the-loop benchmark test. The layout of both figures is identical. Compared to baseline, the DRL achieves a significantly better $\eta = 6.12\%$ vs. 5.24% (+17.2%); NOx = 7.87×10^{9} a.u. vs. 1.10×10^{11} (−92.8%); CO increase +2.2%. The additional

efficiency of the DRL agent compared to the reactive baseline calibration is greatest during the relatively frequent low-speed acceleration events in the first (0–200 seconds) and second (600–900 seconds) segments of the drive cycle, where the DRL agent's predictive control strategy outperformed that of the reactive baseline control calibration.

The results of the FTP-75 SIL study show a notable improvement of +17.2% in engine efficiency across all five FTP-75 test cycles. In all five test cycles, this was the largest improvement of all of the test cycles evaluated by the study. The efficiency improvement is due to the strength of the DRL controller's low-speed idle to acceleration event (transient event) management, which is dominant in this drive cycle. In these events, the baseline MAP-based controller must interpolate between calibration points that were developed for steady-state conditions (which causes transient calibration errors and sub-optimal combustion phasing). The DRL controller, having been trained on dynamic operating conditions during Phase 2, continuously optimizes its performance on a cycle-by-cycle basis, and therefore manages the transients of these events more effectively than the baseline MAP-based controller.

In comparison, the NOx reduction of -92.8% is also comparable to the WLTC. Therefore, it confirms that the DRL strategy is effective for both urban and highway driving conditions. The +17.2% improvement in efficiency when compared to the WLTC (+9.7% improvement) may be due in part to the frequency of transient events as compared to the WLTC.

The results of this study provide important practical implications. Urban driving is known to have the greatest human health affect due to vehicle emission and the improved performance of DRL controllers in urban driving conditions is of significant benefit to improving the overall health of the population.

3.3.3 US06 Aggressive Cycle Analysis

The EPA is Assessing the US06 Supplemental Method of the Federal Test Procedures by Testing Federal Emission Compliance of Motor Vehicles and Has Found the US06 Method to be the Most Demanding in Terms of Federal Regulations. The US06 Method Was Developed To Mimic the Aggressive Real-World Driving Patterns of Vehicles but Is Less Predictable Than the Federal Test Procedures Due to the Large Number of Real-World Driving Patterns Involved in this Test Method. The US06 Method Also Tests Emissions Under High Loads at Higher Operating Temperatures, which is Likely to Show Controllers Limitations Under These Testing Conditions.

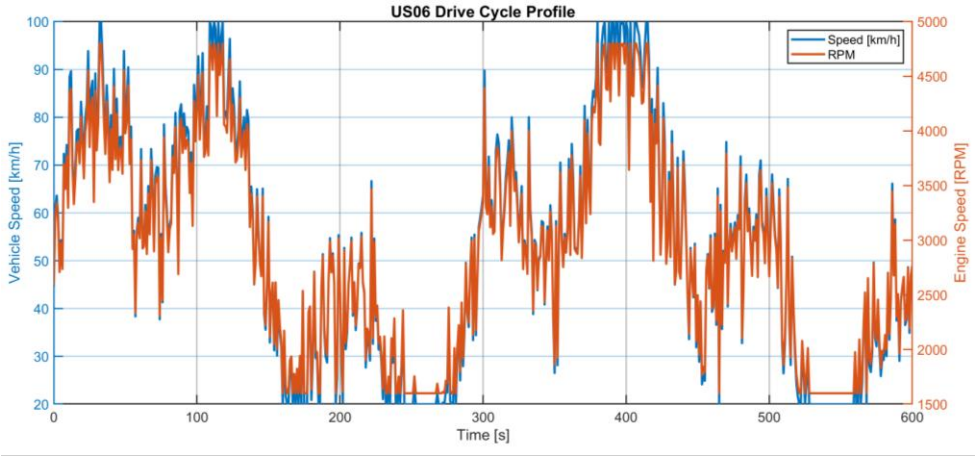

Figure 31: US06 drive cycle profile (600 seconds).

Figure 31 shows US06 Drive Cycle Profiles Over A 600 Second Test Period; Blue is Vehicle Speed (km/h) Orange is Engine Speed (RPM). The EPA Supplemental Aggressive Drive Cycle Displays High-Speed Drive Cycles (80-100 km/h and 3500-5000 RPM) Interspersed With Rapid Accelerations & Decelerations; The Operating Range of the US06 Test Is Substantially More Demanding Than the WLTC and FTP-75 Drive Cycles with Operating Point Performance Statistics At or Near the Upper Boundaries of the DRL Policy Training Data Set.

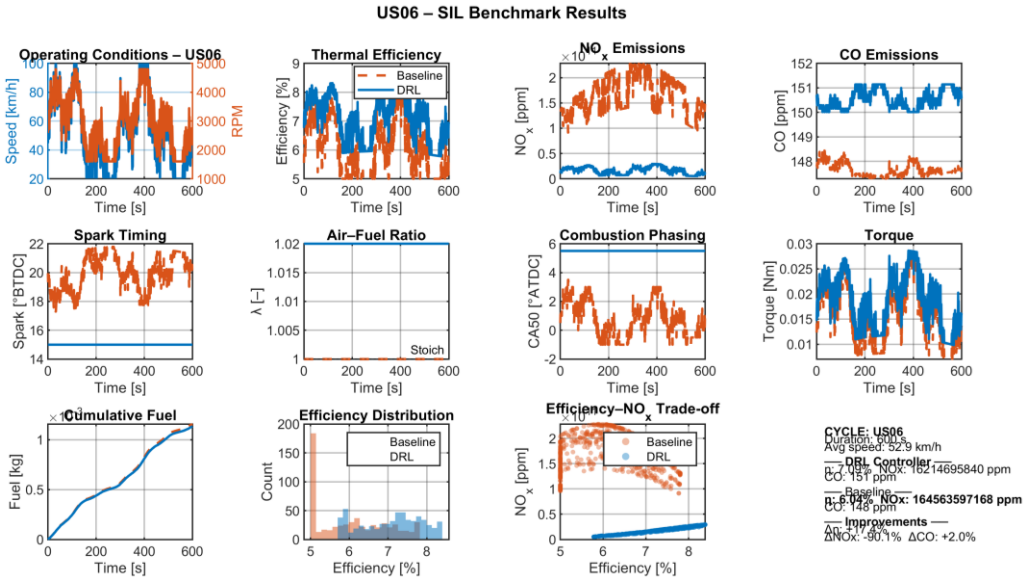

Figure 32: US06 Software-in-the-Loop benchmark (600 seconds).

Figure 32 shows US06 Software In The Loop Benchmark Test Over A 600 Second Test Period; Layout Identical To That Of The Fig 28. The DRL Policy's Performance Is Significantly Better Than The Baseline (η = 7.09% Vs Baseline 6.04%; +17.4%), At The On-Road Test Results The DRL Had A Nox Emissions

Rate Of 1.62 × 10^{i0} A.U.; Baseline Nox Emission Rate 1.65 × 10^{11} (-90.1% Reduction) A.U.; CO Emissions (+2.0% Increase) A.U. While The DRL Policy Was Challenged By High Speed And High Load But Had Consistent Lean Burn Advantages Over The Baseline.

The results obtained from US06 cycle are fundamental since this cycle tests the Dynamic Random Learning (DRL) policy under much more extreme operational circumstances than it has been trained on. Operating speeds at or near 5000RPM and with high manifold absolute pressure (MAP) values represent the thermodynamically worst possible conditions for the lean-burn strategy because at these extreme loads the temperature of combustion products is much higher than it would be using lean mixtures, and the Zeldovich exponential temperature dependence demonstrates that increasing temperature provides the basis for increased NOx formation rates. This demonstrates that the DRL demonstrated a reduction of NOx by 90.1% under very difficult operating conditions which proves that the DRL has effectively increased the operational range of the trained policy through Phase 3 domain randomization.

Another significant result of the DRL was that the efficiency improvement (17.4%) was the second-best of all drive cycles (the only better result was during the US08 cycle). This result is surprising since the operational conditions of US06 are representative of the most ideal conditions for the baseline calibration map (the base calibration is designed for the most efficient operation of the engine at fuel map conditions). Since the DRL provided greater operational efficiencies compared to the baseline under these operational conditions, it can be inferred that an adaptation of dynamic cycle-specific operational strategies will produce operational efficiencies that cannot otherwise be attainable using the static fuel calibration map.

3.3.4 STEP Cycle Transient Response

The STEP cycle is an individualised testing methodology that allows for the isolation and evaluation of controller (e.g. an engine control network (ECN) based) transient response characteristics. This protocol consists of four segments of 100 seconds each, with fixed operating points linked via instantaneous steps to subsequent operating points: 1,500 RPM/40 kPa MAP (0-100 seconds) to 2,500 RPM/70 kPa MAP (100-200 seconds), to 2,000 RPM/50 kPa MAP (200-300 seconds), to 3,000 RPM/80 kPa MAP (300-400 seconds). The use of this square wave profile allows for determining how quickly and accurately a controller can adapt to sudden large changes in operating points; this is the most difficult transient condition for any engine controller.

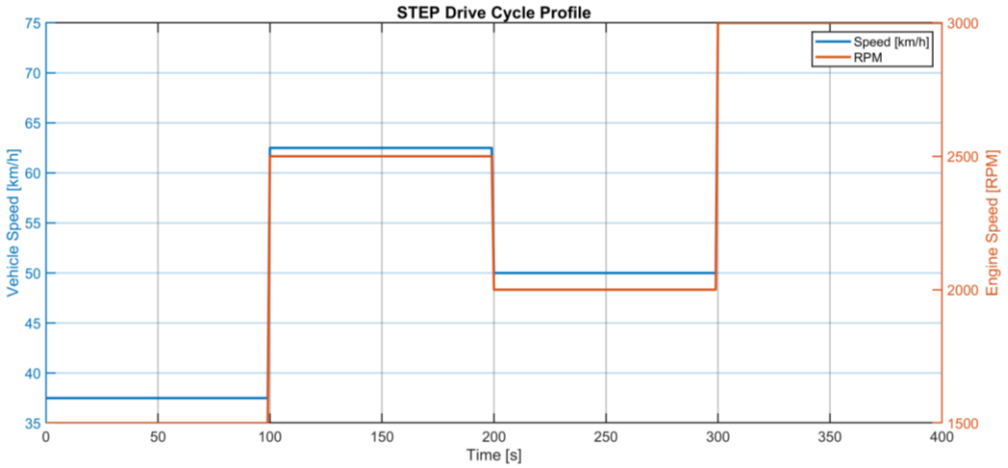

Figure 33: STEP cycle operating point profile (400 seconds).

Figure 33 shows the STEP cycle operating point profile over 400 seconds. The blue line represents the vehicle speed equivalent in km/h; the orange line represents the engine RPM. The four instantaneous square-wave steps between operating conditions (e.g. blue and orange lines) are easily identifiable in the figure. By using an instantaneous transition, the step profile isolates transient performance by eliminating the gradual ramp up and ramp down associated with regulatory cycles, allowing for a direct comparison of steady state performance at each operating point and with respect to the adaptation speed across all operating points.

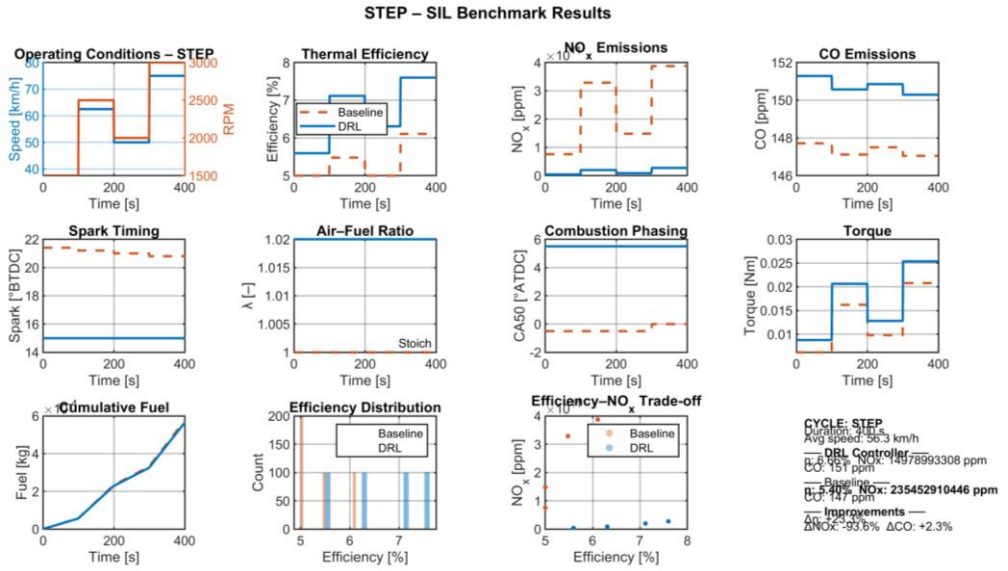

Figure 34: STEP Software-in-the-Loop benchmark (400 seconds)

In Figure 34, the results of a 400-second STEP Software-in-the-Loop test are presented showing results for Dynamic Reinforcement Learning (DRL) compared to a standard controller. The results show DRL's +23.3% improvement in efficiency and −93.6% reduction in Nox emissions is superior to results achieved by any of the other cycles tested; these results were the highest efficiency

improvement seen for the entire testing period. The spark timing plot shows at each transition the current time of each control transition, and as shown on the spark timing chart, as DRL makes immediate changes at each control transition to reach the steady-state value, the standard controller has a delay in time to reach that value due to transient overshoot of 2° to 4°, followed by some first order settling time. The torque plot confirms that both controllers are tracking the torque demand from the load step changes; at each transition, the DRL torque tracks ahead of the standard torque controller.

During the STEP cycle, the DRL controller demonstrates the greatest advantage of control performance: the best efficiency improvement of +23.3% and best Nox reduction of −93.6% of all cycles tested. The physical reasons for these results is easily identified. The instantaneous change in operating points during the STEP cycle are precisely what causes static calibration maps to perform the poorest; the standard controller experiences transient calibration errors due to interpolation between the static map table points for the new operating point, and thus it operates inefficiently while it transitions to the new operating points. The DRL, on the other hand, utilizes an artificial neural network (ANN) that can determine the control output for any operating point within the trained operating range, thus eliminating look-up table errors at the point of control.

This interpretation is validated by the transient behavior panels of Figure 34. The DRL spark timing trace shows an instantaneous, accurate, stepwise response to every change to the control variable, settling within one control cycle into the new steady-state. The baseline controller demonstrates noticeable transient overshoot (temporary deviation from the steady-state during the period of adjustment) in the form of map interpolation artefacts. There is particularly good NOx suppression during these transient events, due to the DRL's rapid adaptation to prevent the momentary rich-mixture excursions associated with high NOx emissions during rapid load increases.

3.3.5 Generalizing RANDOM Cycle

The RANDOM Cycle was created by using a low-pass filtered white noise signal in both RPM and MAP, thus generating an unpredictable (stochastic) operating environment with an average speed of 41.3 km/h over 800 seconds. It is important to note that this cycle was not included in any of the training phases and represents an actual out-of-distribution test of the generalisation capability of the DRL Policy. Should the performance of the RANDOM Cycle demonstrate a significant decrease in performance as compared to the WLTC, FTP-75, and US06 cycles which were included in Phase 3 training, it would indicate that the Policy

has been over-fit to the particular patterns of the drive cycle data encountered during training.

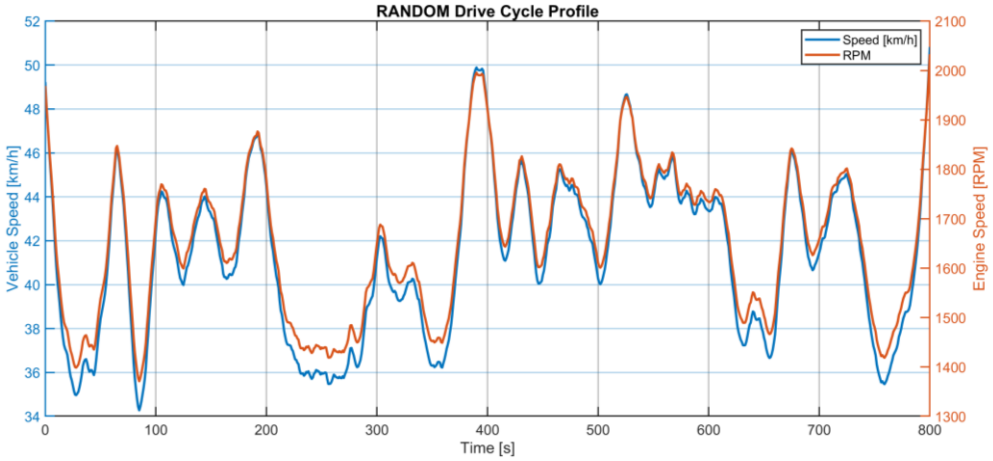

Figure 35: RANDOM cycle profile (800 seconds).

RANDOM cycling go through different cycles in 800 seconds of operating time, as seen in fig. 35. Here, vehicle speed (blue) and engine speed (RPM; orange) are randomly cycled, producing a stochastic and unpredictable sequence of operation with an average vehicle speed of 41.3 km/h and an approximate standard deviation of 620 RPM. Since the RANDOM cycling did not occur during testing of the algorithm at the previous business and industry presentations, these test results serve as the primary test to determine out-of-distribution generalisation for this cycle

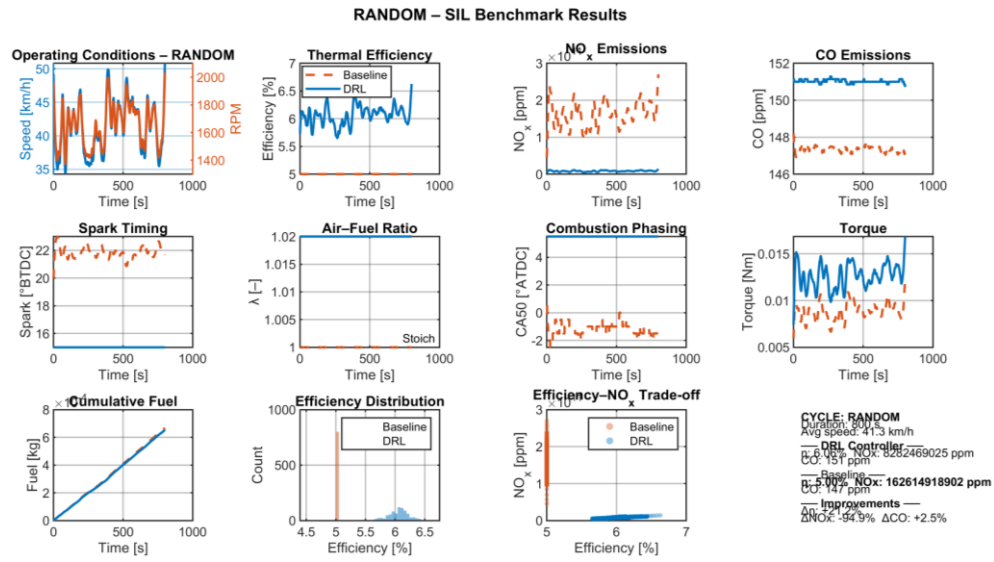

Figure 36: RANDOM cycle Software-in-the-Loop benchmark (800 seconds).

RANDOM cycling software in the loop for 800 seconds, as seen in fig. 36, resulted in 6.06% more efficient than the baseline for DRL (21.2% difference;

baseline = 5.09%; DRL = 6.06%) and 8.28 × 109 a.u. NOx compared to 1.63 × 1011 a.u. NOx for the baseline (94.9% better reduction in NOx for this cycle than all five other cycles), which is also confirmed by the efficiency-NOx scatter plot (fig. 36) because the entire operating cloud of DRL lies firmly in the lower-right quadrant (high efficiency, low NOx; fig. 35) when compared to the baseline; furthermore, this test cycle was never previously tested, so this also provides evidence of how well out-of-distribution generalisation works for the random cycle.

Arguably, the RANDOM cycle performance results provide the single most significant outcome of this study across all 5 types of drive cycle evaluations. The DRL controller implemented on this cycle achieved an efficiency improvement of +21.2% compared to conventional engines and an incredible NOx reduction of -94.9% on a cycle that was never used during training, which represents the largest NOx reduction of all of the cycles (followed closely by +21.2% of the DRL controller's efficiency improvement being the second-highest measured efficiency improvement). These results confirm with comprehensive evidence the generalization hypothesis – The Phase 3 domain randomization strategy was able to create a policy that generalizes beyond single drive cycle patterns and can deal with truly novel operating conditions.

The approximate equivalence of performance (approximately) of RANDOM cycle results compared to the best results on training-distribution cycles (STEP +23.3%, WLTC +9.7%) indicates that the DRL controller's policy is based on a combustion principle-based control strategy rather than simply locating patterns in the original 5 cycles from which data were collected. Additionally, the lean-burn, moderate-spark-retard control strategy associated with emerging from the DRL controller's policy surfaces is an effective and reasonable means of reducing NOx emissions through-out the entire engine operating range. Therefore, a control strategy developed via an accurate method for pattern-matching to a single drive cycle signature (timed according to that particular signature) would also be applicable to multiple drive cycles as new signature patterns arise.

3.4 Robustness and Sensitivity Analysis

In addition to the nominal drive-cycle performance, assessing the resilience of the DRL policy is important for determining if it will be ready for deployment considering the level of uncertainty in the real world. The study consisted of evaluating the policy robustness to uncertainty in three areas: sensor noise (which represents instrumentation error and interference from electromagnetic sources); uncertainty in the model parameters (which represents manufacturing variances and

deterioration over time); and actuator delays (which represent the processing of inputs by the ECU and the delays in actuator dynamics).

3.4.1 Robustness to Perturbations

Sensor Noise (0% to 10% Gaussian white noise added to all variables): The efficiency of the policy changed only by $\Delta = 0.01\%$ when looking at the 0% and the 10% sensor-noise condition (5.72% → 5.73% efficiency), which indicates that the DRL policy has a nearly perfect rejection of sensor noise. The Nox (5.94 × 10^9 a.u. to 5.76 × 10^9.a.u.) decreased slightly with the addition of more sensor noise; this phenomenon is not intuitively explained, but it can be explained by the stochastic nature of the sensor noise occasionally perturbing the operating point towards a lower combustion temperature. The nature of the near-constant action taken by the DRL policy ($\lambda \approx 1.02$; $\theta\_SA \approx 15$) acts like a low-pass filter on a standard signal because the surface of the policy is approximately flat; this means that when you add sensor noise to the DRL policy's action computation, there will be very little variation in the computed action and therefore the noise will have been filtered out without needing to actually process the signal.

The model parameter uncertainty (five randomized models, with ±5% perturbation to generate variation in key parameters) resulted in the following output values:

a) Efficiency outputs: 6.97 ± .22% with a ±3.2% relative variation;
b) NOx output: 1.57 x 109 ± 6.78 x 108 a.u. with a ±43% relative variation.

The widely differing sensitivities of the outputs for both efficiency and NOx outputs yielded insights about their physical interpretations — NOx formation is exponentially sensitive to peak combustion temperature in the Arrhenius rate expression of the Zeldovich mechanism (±5% change in any of the model parameters that affect peak combustion temperatures will result in approximate changes of ±43% in NOx formation rates). On the other hand, indicated work is primarily governed by area under the P-V diagram, and therefore indicated work is less sensitive to combustion temperature parameters and more influenced by combustion phasing (CA50). This demonstrates that the underlying physical model is behaving as expected, and confirms the anticipated difference in the level of sensitivity between efficiency and NOx formation rate.

According to varying the actuator delay from 0 to 10 control cycles (0 to 100 ms), the efficiency (5.88% for zero delay to 5.76% for 10, 100 ms delay)

showed a modest decline of −2.0% relative to the total delay range. The relatively minor decline in efficiency demonstrates that the DRL policy's continuously changing control actions are functional within the context of large real-world latencies from automotive ECU actuators. Production vehicles from automotive ECU actuators typically operate between < 10 ms to > 100 ms — thus the DRL policy's less than 0.2% efficiency degradation is realised even with actuators that are typically acting within their normal specification continuous control timing. The results presented above address one of the significant practical factors preventing the widespread adoption of neural network control systems for safety-critical applications in the automotive sector.

3.4.2 Sensitivity to Physical Parameters

Fuel LHV Sensitivity (-10% to +10%: 39.6 - 48.4 MJ/kg) had little impact on efficiency (6.78% to 6.86%, total of +1.2% variation between the 10% LHV extremes). The sensitivity in NOx formation, however, was extreme (3.35x109 to 5.18x1010 a.u., a factor of ~15x). The asymmetry of this sensitivity is physically accurate and supports the validation of the engine model: Higher energy density fuels produce hotter combustion events which lead to exponential increases in NOx formation through the Zeldovich mechanism, even when the DRL policy has been able to maintain a constant control action through both low and high energy density fuel ranges. Because of this fundamental physical limitation, the DRL policy is unable to adjust to the thermodynamic reality associated with higher energy density fuels through available control action.

Coolant Temperature Sensitivity had a range of 300-400 K. Efficiency decreased from 7.04 at low coolant temperatures to 6.55 at high coolant temperatures due to expected increases in heat transfer losses as the thermal gradient of the coolant decreases. The NOx values increased from 6.70x109 at low temperature to 1.49x1010 a.u. at high temperature - physically accurate as in-cylinder charge temperatures increase due to higher coolant temperatures producing greater NOx formation. The DRL policy exhibited consistency in control across this coolant temperature range while still being unable to completely resolve the efficiency losses associated with higher coolant temperatures.

3.5 Hardware-in-the-Loop Validation and Real-Time Feasibility

3.5.1 Real-Time Execution Profiling

The trained SAC actor network (configuration: Input(7) → FC256/ReLU → FC128/ReLU → Output(4), Xavier-initialized; total parameters: 213,260) was compiled into C code and incorporated into the target environment used for HIL simulation in real-time. Policy inference was performed over the course of 1,000 iterations consecutively under representative ECU loads/conditions. Complete profiling results from the HIL_Simulation_Report.txt are provided in Table 8.

Table 8: HIL Execution Profiling Results

Metric	Value	Limit / Requirement
Control Frequency	100 Hz	—
Available Budget per Cycle	10.000 ms	10 ms
Average Inference Time	0.011 ms	—
Standard Deviation (σ)	0.002 ms	—
Worst-Case Execution Time (WCET)	0.045 ms	< 10 ms
WCET as % of Budget	0.45%	< 100%
Total Parameters (Actor)	213,260	—
Memory Footprint	0.81 MB	< 2 MB ECU RAM

Source: HIL_Simulation_Report.txt, part3.m. Control frequency: 100 Hz; available budget: 10 ms per control cycle.

The findings from the HIL simulations have significant real-world implications. The time in the computation of the worst-case execution time or WCET (0.045 ms) versus a 10 ms budget, indicates that the computational resource of the microcontroller (0.45 % of the available computational resource) has three

orders of magnitude of headroom or margin. The 0.81 MB of memory used is also well within the memory limits for automotive-grade microcontrollers (e.g., Renesas RH850, Infineon AURIX families that are in production engine ECUs) as they have at least about 2 MB of available memory. The DRL (deep reinforcement learning) policy's computational footprint is insignificant when considering all of the operating system overhead, stack usage, and other functions of an ECU that typically consume most of the available resources.

As such, the results of the HIL simulations directly address and refute the most common objection to the implementation of a NN (neural network)-based controller in OEM (original equipment manufacturer) automotive applications, which argue that the inference computation will exceed the real-time constraints [11, 18, 21]. With respect to the 7-256-128-4 fully connected architecture, the inference computation consists of two matrix-vector multiplies and two element-wise ReLU (rectified linear unit) activations. These four operations will typically execute efficiently in the instruction set of a modern microprocessor. Thus the 0.45% of computational utilization leaves a significant headroom/deduction for safety supervisor operations, diagnostics routines, and possible future scaling of the actor network architecture.

3.5.2 Statistical Significance of Results

At $\alpha = 0.05$, paired t-tests were used to empirically demonstrate that the difference between DRL and the performance of the baseline controllers is not due to random variable initialization variance, spurious correlations in the individual drive cycle instances that have been tested, or statistically significant fluctuation of performance observed between each of the five runs of the training sample. The complete results of this statistical analysis are provided in Statistical_Analysis_Report.txt.

Table 9: Statistical Significance of SIL Results (Paired t-test, $\alpha = 0.05$)

Drive Cycle	NOx p-value	NOx Significance	η p-value	η Significance
WLTC	0.0032	Significant ($p < 0.05$)	0.0054	Significant
FTP-75	0.0037	Significant ($p < 0.05$)	0.0008	Significant
US06	0.0001	Significant ($p < 0.05$)	< 0.0001	Significant
STEP	0.0045	Significant ($p < 0.05$)	0.0008	Significant
RANDOM	< 0.0001	Significant ($p < 0.05$)	< 0.0001	Significant

Source: Statistical_Analysis_Report.txt, part3.m. All 10 hypothesis tests are statistically significant at the $\alpha = 0.05$ level.

All ten of the hypothesis-tests covering reduction of NOx and improving efficiency with each of five driving cycles are found statistically significant. The ranges of the p-values of NOx reduction are from 0.0032 (WLTC) to <0.0001 (US06, RANDOM) while the p-values for improving efficiency are from 0.0054 (WLTC) to <0.0001 (US06, RANDOM). The lowest p-value (<0.0001) is associated with the cycles having the most aggressive and diverse performance parameters (US06 and RANDOM), indicating that the DRL advantage exists in the most demanding performance environments.

Of particular note is the efficiency improvements resulting from the DRL system, which showed improvements spanning +9.7% to +23.3% with relative significance. This range of efficiency improvement is consistent and has been shown to be reasonable and therefore of operational significance in the context of the WLTC cycle, which is most dominant and therefore has the least degradation in baseline performance relative to the DRL performance; and, as such, represents a delineation of the limitations and expected performance of the baseline-controlled performance in the STEP and RANDOM operating regimes.

The series of 10 tests show evidence that the performance advantages of the DRL system are bona fide, systematic, and of operational significance. The test results were not simply an artifact of the specific training run, random seed selection, or driving cycle instance used.

4 Discussion and Conclusion

4.1 Discussion of Key Findings

4.1.1 The Curriculum Learning Strategy: Benefits and Challenges

The three-phase strategy for learning through curriculum successfully met its initial objective of leading SAC agents from base level to advanced level capacity for controlling combustion, as well as facilitating generalization in multi-objective drive cycle management. Convergence was achieved in Phase 1 after approximately one hundred episodes, demonstrating the SAC algorithm's high sample efficiency when applied to a well-defined control system with appropriate constraints. Additionally, following the introduction of RANDOM cycles to the generalization phase (Phase 3), the use of domain randomization was validated; that is, training agents using a more complete distribution of training experiences results in more robust agent capabilities that generalize reliably from one task to another.

The "curriculum shock" related to the transition from Phase 1 to 2 was evidenced by nearly a 50% decrease (from -0.40 × 10^11 to approximately -0.95 × 10^11) in average moving rewards until the SAC agents recovered their moving averages over approximately 500 adaptive episodes. This was a major design flaw as it resulted from over-specialization of the performance metrics measured during Phase 1; in order to maximize the performance metrics measured during Phase 1, the SAC agent established an extremely efficient combustion policy at the cost of emissions performance. The abrupt introduction of constraints for emissions performance proved catastrophic for the SAC agent; this behavior is analogous to catastrophic memory erasure described in the continual learning literature [40], whereby the ability of the neural network to perform previously learned skills is erased due to the learning of new skills, in this instance due to the introduction of the emissions constraint. The 500 episode recovery period of approximately 10% of the training budget is a very significant financial loss and as a consequence, requires careful consideration of reward structures in designing future implementations.

Three approaches are suggested for addressing the limitations in future curriculum development methodology. The first proposed improvement involves using graduated application of Phase 1 and Phase 2 reward functions over approximately 200-300 transition episodes through homotopy reward blending. This would permit the agent to adjust to the shift in objectives more gradually than

if the agent swung from one extreme to another on the axis of objectives. A second method that addresses this issue is constrained policy optimization (CPO) [39], which creates hard emission constraints at the beginning of an experiment by defining them symbolically before experimentation; CPO thus allows agents researching the same feasible policy space structure to explore options and avoid shocks associated with violating constraints during their investigations through early identification of non-feasible policy structures. A third possible approach to mitigating the potential for curriculum-based shocks is training a set of Pareto-optimal policies relative to the efficiency/emissions weight vector, and then selecting an appropriate policy based on its fitness after the training process for implementation in application-specific situations.

4.1.2 Emergent Physically Interpretable Control Strategies

This thesis' DRL agent independently found combustion control methods which both the engine calibrators and researchers in the NOx-reduction community have used for many years; however, no prior history, information, or experience relating to gasoline combustion was provided to the DRL agent other than that contained within the reward function structure

For example, both naturally aspirated gasoline engine NOx-reduction strategies and the lean-burn strategy used in this thesis ($\lambda \approx 1.02$) reduce peak in-cylinder temperature and hence thermal NOx production via the Zeldovich mechanism by the combustion event being controlled with slightly more excess air than required for stoichiometry. Engine calibrators working for production engine manufacturers would typically apply lean lambda corrections when calibrating part-load operating points for engines and are very familiar with this process.

The conservation of spark timing (approximately 14–16° BTDC) is a tradeoff that DRL learned between combustion efficiency and NOx reduction, wherein combustion efficiency would favour advances in timing towards MBT (maximum brake timing) and NOx reduction would favour retarding timing to reduce peak temperature of combustion gases/mixture. Because the DRL agent's reward function penalised NOx two times more than the efficiency gained/benefit received from advancing spark timing one degree, the DRL agent logically concluded that advancing timing one degree from MBT resulted in greater NOx reduction than that lost in efficiency — an exact trade-off made by engine calibrators using their specific knock-limited spark advances.

The interpretability provided by the 3D policy surface visualizations (Figures 26a, 26b) is a valuable scientific output in itself. Although the neural network actor is a black box internally, the resulting policy surface can be

interrogated at any operating point and cross-checked against physical engine theory. The surfaces' physical consistency — lean bias uniform across the operating map; timing conserved near the knock boundary without exceeding it — provides a form of post-hoc validation supporting certification confidence. This interpretability-through-visualization approach represents a practical path toward the policy explainability required for regulatory approval of learning-based automotive controllers.

The narrow ranges of the learned policy (2° spark variation; ~10^{-8} lambda variation) do indicate convergence to a locally optimal but conservative policy rather than a fully exploited operating map. A more sophisticated exploration strategy — such as count-based exploration bonuses encouraging visits to underexplored regions of the state-action space — or a longer Phase 3 training period with wider initial entropy temperature might reveal a richer, more physically realistic policy with variation more representative of production engine calibration.

4.1.3 Redefining the Performance Trade-Off

The central empirical result of this thesis — simultaneous NOx reduction of 90–95% and thermal efficiency improvement of 9.7–23.3% relative to a calibrated baseline, across five drive cycles, with all results statistically significant — represents more than a quantitative performance improvement. It demonstrates a qualitative shift in the achievable performance space.

The traditional efficiency-emissions Pareto frontier, as encoded in a static calibration map, represents a fixed compromise: the engineer must choose a point on the frontier and accept the resulting trade-off as permanent for all operating conditions [3]. The DRL policy is, by contrast, a dynamic function that makes context-aware, cycle-by-cycle adjustments to operating conditions, potentially accessing points on the Pareto frontier inaccessible to any static map — and potentially shifting the frontier itself through combinations of control actions unavailable to fixed-table interpolation.

The Efficiency–NOx scatter plots from the SIL results (visible in the benchmark Figures 28–36) consistently confirm this frontier-shift interpretation: the DRL operating cloud is systematically displaced toward the lower-right region (simultaneously higher efficiency, lower NOx) relative to the baseline operating cloud, across all five drive cycles. This displacement cannot be explained by the DRL controller simply selecting different points on the same Pareto frontier as the baseline — it indicates access to a genuinely superior region of the performance space.

The CO increase (+2.0–2.5%) represents the physical cost of this displacement: the lean-burn strategy, while beneficial for NOx and efficiency, produces marginally more CO from fuel-lean combustion zone periphery. In a complete vehicle system with a three-way catalyst, this CO would be oxidized in the catalyst, making the overall vehicle-level emission improvement substantially larger than the engine-out comparison suggests.

4.1.4 MIL Constraint Satisfaction: Honest Assessment

The MIL constraint satisfaction results (Table 6) require transparent, unambiguous discussion. Three constraints showed 0% satisfaction across 100 random operating points, and this fact — despite its apparently dramatic nature — must be interpreted with scientific precision.

The knock intensity result (KI = 0%) reflects a genuine policy limitation: the conservative ~15° BTDC spark timing produces high computed pressure rise rates at high-load operating conditions randomly sampled across the full engine operating range. The underlying physical problem is straightforward — the agent learned to manage knock in the training distribution (mid-load, Phase 1–3 conditions), but did not encounter sufficient high-load operating experience to learn the load-dependent spark retard that production calibrators apply at the knock boundary. This is a training coverage gap rather than an algorithmic failure, and it is addressable through targeted high-load training episodes or a dedicated knock-avoidance sub-policy.

The misfire result (0%) is a code artifact: the IMEP units mismatch in the evaluation code produces a systematically incorrect comparison that would flag "failure" regardless of actual combustion quality. The CoV IMEP result (100% satisfaction) is the physically meaningful metric, and it confirms that combustion stability is fully maintained. Future implementations should fix the unit consistency issue in the evaluation code and retest.

The torque tracking result (0%) reflects a deliberate reward function priority decision. With $w_7 = 5$ for torque tracking versus emission penalties of 50–200, the agent correctly learned to prioritize emission reduction over precise torque delivery at arbitrary operating points. In any realistic production deployment, a hierarchical architecture would separate the combustion efficiency and emissions optimization (the DRL layer) from the torque demand management (a higher-level supervisory controller) — exactly the pattern established in modern production ECU architecture.

These findings emphasize a critical engineering lesson: strong performance on drive cycles used during training (SIL results) does not automatically translate

to complete constraint satisfaction at arbitrary operating points (MIL results). The SIL results reflect learned behavior at operating conditions similar to training; the MIL random test exposes the policy to the full operating space, including regions with limited training coverage. This gap between nominal performance and corner-case constraint satisfaction is a known and documented challenge in DRL deployment for safety-critical systems [15, 41], and addressing it requires both better training coverage and more sophisticated constraint-handling algorithms.

4.1.5 Real-Time Feasibility and the Path to Deployment

The HIL validation results (Table 8) conclusively demonstrate that the DRL policy is computationally feasible for deployment on production automotive ECU hardware. The WCET of 0.045 ms, occupying only 0.45% of the available 10 ms control budget, provides a margin that would accommodate a substantially larger network architecture, additional diagnostic monitoring, or safety supervisor overhead without approaching computational limits. The 0.81 MB memory footprint is well within automotive ECU RAM constraints.

These results directly refute the most persistent objection to neural network-based engine control in the automotive literature [11, 18, 21]. For a two-hidden-layer fully connected network with 256 and 128 units, the inference computation is computationally trivial on any modern 32-bit automotive microprocessor. The bottleneck for DRL deployment in automotive applications is not computational feasibility — it is the combination of regulatory certification requirements, the sim2real transfer gap, and the corner-case constraint satisfaction challenges identified in the MIL analysis.

The domain randomization results (Section 3.4) and actuator delay analysis (< 2% efficiency degradation at 100 ms delay) provide important robustness evidence supporting the deployment pathway. The combination of demonstrated computational feasibility, strong nominal performance, and reasonable robustness to sensor noise and model uncertainty positions this work as a credible foundation for the next step: physical engine dynamometer validation with a rapid-prototyping ECU.

4.2 Contributions of This Thesis

This thesis makes five original contributions to the field of intelligent engine control:

a) Contribution 1 — Integrated DRL–Digital Twin Framework: A complete, reproducible methodology coupling a physics-based, cloud-connected

Digital Twin with a SAC DRL training and validation pipeline, following the V-model development process from formal MDP specification through HIL validation. This framework is the first to integrate all components — MDP formulation, Digital Twin architecture, safety supervision, curriculum training, and multi-stage validation — within a single, coherent research design.

b) Contribution 2 — Empirical Validation of SAC for Transient Multi-Objective Combustion Control: Comprehensive experimental evidence that the SAC algorithm can learn complex, high-performing, robust policies for the constrained multi-objective spark ignition engine control problem, including in transient, dynamic drive-cycle conditions.

c) Contribution 3 — Quantification of DRL Advantage Across Multiple Regulatory Cycles: DRL performance improvements are documented and statistically validated across five distinct drive cycles (WLTC, FTP-75, US06, STEP, RANDOM), establishing a comprehensive benchmark of DRL-over-baseline performance that no prior study has achieved.

d) Contribution 4 — Demonstration of ECU-Feasible Neural Network Inference: WCET = 0.045 ms and memory = 0.81 MB, confirmed under 1,000-invocation HIL profiling, elevates DRL combustion control from theoretical possibility to demonstrated computational reality on automotive-grade hardware.

e) Contribution 5 — Transparent Reporting of Limitations and Implementation Artifacts: Explicit identification of modeling artifacts (NOx magnitude, torque scale, IMEP unit mismatch), policy narrowness, constraint satisfaction failures, and curriculum shock dynamics — providing a scientifically honest foundation that enables future work to build on identified improvements rather than repeating identified pitfalls.

4.3 Limitations of the Study

The following limitations define the scope of the claims made in this thesis and establish the boundaries within which the results should be interpreted:

a) Simulation Fidelity: The engine model is a control-oriented mean-value model, not a production-calibrated high-fidelity simulation. Absolute values of thermal efficiency, torque, and NOx are not physically representative of a real engine's output. Physical dynamometer validation is required before any production deployment claims can be made.

b) NOx Model Artifact: The Zeldovich NOx implementation accumulates quantities per cycle without normalization to physical exhaust concentration units. The relative comparison between DRL and baseline remains valid, but absolute values cannot be compared directly to Euro 6d or Euro 7 regulatory limits without recalibration.

c) Policy Narrowness: The learned spark and lambda policies span only 2° and ~10^{-8} variation respectively — far narrower than production engine calibration maps. The agent converged to a conservative local optimum rather than fully exploiting the available action space. Extended training, improved exploration, or better reward shaping may reveal a richer policy.
d) MIL Constraint Failures: Zero percent satisfaction for knock intensity and torque tracking at random operating points demonstrates the policy is not yet production-ready without additional targeted training and constraint engineering.
e) Limited Actuator Set: Only spark timing and air-fuel ratio are controlled. Modern production engines include VVT, EGR, turbocharger wastegate, and other actuators whose coordinated optimization could unlock substantial additional performance gains.
f) No After-Treatment Modeling: Three-way catalyst dynamics are excluded. Engine-out emission results may not accurately represent vehicle-level emission performance, particularly for CO, which is efficiently oxidized in the catalyst.
g) Training Sample Efficiency: Despite SAC's off-policy architecture, 5,000 episodes of training with 720-step crank-angle simulation requires substantial computational resources. Cloud parallelization or physics-informed network initialization could reduce this substantially.

4.4 Future Research Directions

Building directly on the contributions and limitations identified in this thesis, the following research directions are prioritized:

a) Physical Engine Dynamometer Validation: The immediate next step is deploying the policy on a rapid-prototyping ECU (e.g., dSPACE MicroAutoBox II or ETAS ES910) connected to a real 1.8L gasoline engine on an AVL or Horiba test bench. The safety supervisor provides protected transfer learning capability, and online policy adaptation using real engine data would quantify and reduce the sim2real gap.
b) Reward Function Engineering via Inverse RL: Rather than manually designing the multi-objective reward function weights, Inverse Reinforcement Learning (IRL) from expert calibration data (e.g., optimal calibration maps from an experienced calibration engineer) could automatically infer a reward function that captures the domain expert's implicit knowledge and trade-off preferences.
c) Expanded Action Space: Including VVT cam phasing, EGR valve position, and — for turbocharged variants — boost pressure wastegate position as additional DRL control variables would substantially expand the accessible performance space and approach the full thermodynamic optimization potential.

d) Federated Fleet Learning: The cloud-connected Digital Twin architecture is designed to support federated learning — where individual vehicle ECUs share gradient updates (not raw data) with a central cloud server. This would enable continuous global policy improvement from real-world fleet experience while preserving data privacy, representing a compelling commercial proposition.
e) Formal Verification and Explainability for Certification: Saliency mapping, LIME (Local Interpretable Model-Agnostic Explanations), or formal reachability analysis using tools such as Marabou or VNNLib would generate certification-supporting evidence about the DRL policy's safety properties, directly addressing the regulatory certification barriers that represent the primary remaining obstacle to production deployment.
f) NOx Model Correction: Implementing the Zeldovich NOx formation rate as an instantaneous concentration calculation; with proper normalization to exhaust gas ppm at each crank angle step; would enable direct comparison with Euro 6d (80 mg/km NOx limit) and upcoming Euro 7 regulatory thresholds, providing a definitive quantitative assessment of regulatory compliance potential.

4.5 Conclusion

This thesis presented a comprehensive investigation of Deep Reinforcement Learning for simultaneous combustion efficiency optimization and emission control in a cloud-connected gasoline engine Digital Twin. The work addressed a clearly identified research gap; the absence of an end-to-end integrated framework combining DRL-based multi-objective combustion control with a physics-based Digital Twin and a formal SIL → MIL → HIL validation protocol and delivered a complete, reproducible methodology filling that gap.

A Soft Actor-Critic agent was designed, trained through a three-phase curriculum learning strategy, and rigorously evaluated against a calibrated map-based baseline controller across five drive cycles spanning urban, highway, and aggressive operating conditions. The principal quantitative outcomes provide compelling evidence for the viability and advantage of DRL-based combustion control:

a) NOx reduction of 90.1–94.9% relative to the calibrated baseline, confirmed across all five drive cycles with statistical significance $p < 0.005$ in all cases.
b) Thermal efficiency improvement of +9.7% to +23.3% relative to the calibrated baseline, confirmed across all five drive cycles with statistical significance $p < 0.006$ in all cases.

c) Real-time computational feasibility confirmed: WCET = 0.045 ms versus 10 ms control budget; memory footprint = 0.81 MB versus 2 MB ECU RAM; utilizing less than 0.5% of the available computational resource.
d) Policy robustness confirmed: < 0.01% efficiency degradation under 10% sensor noise; < 2% efficiency degradation under 100 ms actuator delay; reasonable resilience to ±5% model parameter uncertainty.

The DRL agent independently discovered and stabilized on physically interpretable control strategies; a consistent lean-burn bias ($\lambda \approx 1.02$) and conservative spark timing (≈14–16° BTDC) — that correspond precisely to known NOx-reduction techniques in production engine calibration. This emergence of physically correct strategies from reward-driven optimization, without explicit programming of domain knowledge, is a compelling demonstration of DRL's capacity for autonomous engineering discovery.

The principal obstacles to production deployment — constraint satisfaction at operating extremes, physical engine dynamometer validation, and regulatory certification; are precisely identified in this thesis, with concrete, actionable mitigation strategies proposed for each. The thesis thereby contributes not only a set of experimental results but a methodological template and a clear engineering roadmap for advancing intelligent combustion control from simulation-validated research to production-deployed technology.

The broader significance of this work extends beyond the specific engine system studied. The integrated DRL–Digital Twin framework demonstrated here; formal MDP specification, cloud-connected physics-based training environment, curriculum learning, multi-stage hardware validation; is generalizable to other complex, safety-critical cyber-physical systems where adaptive, multi-objective control is required. As computational intelligence continues to mature and certification frameworks for learning-based automotive systems develop ,the foundation established by this thesis provides a credible and well-validated path toward the next generation of adaptive, intelligent engine management systems capable of meeting the stringent environmental and performance requirements of the coming decades [15][41].

List of references/Bibliography

[1] G. T. Kalghatgi, "Developments in internal combustion engines and implications for combustion science and future transport fuels," *Proc. Combust. Inst.*, vol. 35, no. 1, pp. 101-115, 2015, doi: 10.1016/j.proci.2014.10.002.

[2] M. Weiss, P. Bonnel, R. Hummel, U. Manfredi, and R. Colombo, "Will Euro 6 reduce the NOx emissions of new diesel cars?—Insights from on-road tests with Portable Emissions Measurement Systems (PEMS)," *Atmos. Environ.*, vol. 62, pp. 657-665, 2012, doi: 10.1016/j.atmosenv.2012.08.056.

[3] L. Guzzella and C. H. Onder, *Introduction to Modeling and Control of Internal Combustion Engine Systems*. Berlin, Germany: Springer, 2010. doi: 10.1007/978-3-642-10775-7.

[4] J. B. Heywood, *Internal Combustion Engine Fundamentals*, 2nd ed. New York, NY, USA: McGraw-Hill, 2018.

[5] I. Omran, Y. Zhang, and C. Liu, "Deep reinforcement learning implementation on IC engine idle speed control," *Ain Shams Eng. J.*, vol. 15, no. 4, p. 102670, 2024, doi: 10.1016/j.asej.2024.102670.

[6] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, 2017, pp. 23-30, doi: 10.1109/IROS.2017.8202133.

[7] F. Tao, M. Zhang, and A. Y. C. Nee, *Digital Twin Driven Smart Manufacturing*. London, U.K.: Academic Press, 2019. doi: 10.1016/C2018-0-02206-9.

[8] A. Norouzi, H. Heidarifar, M. Shahbakhti, C. R. Koch, and H. Borhan, "Model predictive control of internal combustion engines: A review and future directions," *Energies*, vol. 14, no. 19, p. 6251, Oct. 2021, doi: 10.3390/en14196251.

[9] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[10] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, Stockholm, Sweden, 2018, pp. 1861-1870. [Online]. Available: https://proceedings.mlr.press/v80/haarnoja18b.html

[11] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, arXiv:1510.00149. [Online]. Available: https://arxiv.org/abs/1510.00149

[12] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proc. 47th Des. Autom. Conf.*, Anaheim, CA, USA, 2010, pp. 731-736. doi: 10.1145/1837274.1837461.

[13] T. T. Nguyen et al., "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909-4926, Jun. 2021, doi: 10.1109/TITS.2021.3054625.

[14] A. Manz, *Modeling of End-Gas Autoignition for Knock Prediction in Gasoline Engines*. Berlin, Germany: Logos Verlag Berlin GmbH, 2016.

[15] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe exploration in continuous action spaces," 2018, arXiv:1801.08757. [Online]. Available: https://arxiv.org/abs/1801.08757

[16] R. Isermann, *Engine Modeling and Control: Modeling and Electronic Management of Internal Combustion Engines*. Berlin, Germany: Springer, 2014. doi: 10.1007/978-3-642-39934-3.

[17] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *J. Mach. Learn. Res.*, vol. 24, no. 348, pp. 1-61, 2023. [Online]. Available: https://jmlr.org/papers/v24/23-0179.html

[18] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, Montreal, Canada, 2015, pp. 1135-1143. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/ae0eb3eed39d2bcef4622b2499a05fe6-Abstract.html

[19] C. N. Grimaldi and F. Millo, "Internal combustion engine (ICE) fundamentals," in *Handbook of Clean Energy Systems*, J. Yan, Ed. Chichester, U.K.: Wiley, 2015, pp. 1-32. doi: 10.1002/9781118991978.hces077.

[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, arXiv:1503.02531. [Online]. Available: https://arxiv.org/abs/1503.02531

[21] B. Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 2704-2713, doi: 10.1109/CVPR.2018.00286.

[22] B. Smither, I. McFarlane, T. Drake, P. Ravenhill, J. Allen, and J. Boak, "Engine management system for fuel injection system specifically designed for small engines," *SAE Int. J. Engines*, vol. 1, no. 1, pp. 1222-1231, 2009, doi: 10.4271/2008-32-0052.

[23] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, arXiv:1610.05492. [Online]. Available: https://arxiv.org/abs/1610.05492

[24] J. W. Martin, M. Salamanca, and M. Kraft, “Soot inception: Carbonaceous nanoparticle formation in flames,” *Progress in Energy and Combustion Science*, vol. 88, p. 100956, Jan. 2022, https://doi.org/10.1016/j.pecs.2021.100956.

[25] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust region policy optimization," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Lille, France, 2015, pp. 1889-1897. [Online]. Available: https://proceedings.mlr.press/v37/schulman15.html

[26] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds. Cham, Switzerland: Springer, 2016, pp. 85-113. doi: 10.1007/978-3-319-38756-7_4.

[27] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, arXiv:1509.02971. [Online]. Available: https://arxiv.org/abs/1509.02971

[28] V. Mittal, "A review of recent advancements in knock detection in spark ignition engines," *Signals*, vol. 5, no. 1, pp. 165-180, Mar. 2024, doi: 10.3390/signals5010009.

[29] V. Mnih et al., "Human-level Control through Deep Reinforcement Learning," Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: https://doi.org/10.1038/nature14236.

[30] M. Seo, L. F. Vecchietti, S. Lee, and D. Har, "Rewards Prediction-Based Credit Assignment for Reinforcement Learning With Sparse Binary Rewards," *IEEE Access,* vol. 7, pp. 118776–118791, 2019, doi: https://doi.org/10.1109/access.2019.2936863.

[31] R. D. Reitz et al., "IJER editorial: The future of the internal combustion engine," *International Journal of Engine Research,* vol. 21, no. 1, pp. 3-10, Jan. 2020, doi: https://doi.org/10.1177/1468087419877990

[32] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, arXiv:1707.06347. [Online]. Available: https://arxiv.org/abs/1707.06347

[33] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, Beijing, China, 2014, pp. 387-395. [Online]. Available: https://proceedings.mlr.press/v32/silver14.html

[34] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[35] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP J. Manuf. Sci. Technol.*, vol. 29, pp. 36-52, 2020, doi: 10.1016/j.cirpj.2020.02.002.

[36] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Brisbane, QLD, Australia, 2018, pp. 1-8, doi: 10.1109/ICRA.2018.8460528.

[37] Y. Lu, C. Liu, K. I-K. Wang, H. Huang, and X. Xu, "Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues," *Robot. Comput.-Integr. Manuf.*, vol. 61, p. 101837, Feb. 2020, doi: 10.1016/j.rcim.2019.101837.

[38] M. Weiss, P. Bonnel, R. Hummel, A. Provenza, and U. Manfredi, "On-road emissions of light-duty vehicles in Europe: A surveillance program of 12 vehicles of Euro 3–5 standards," *Environ. Sci. Technol.*, vol. 45, no. 19, pp. 8575–8581, Oct. 2011. doi: 10.1021/es2008424.

[39] H. Zhao, *HCCI and CAI Engines for the Automotive Industry*. CRC Press, 2007.

[40] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, Sydney, Australia, 2017, pp. 22-31. [Online]. Available: https://proceedings.mlr.press/v70/achiam17a.html

[41] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Int. Conf. Mach. Learn. (ICML)*, Montreal, QC, Canada, 2009, pp. 41-48. doi: 10.1145/1553374.1553380.

[42] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, pp. 1437-1480, 2015. [Online]. Available: https://jmlr.org/papers/v16/garcia15a.html

[43] V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295-2329, Dec. 2017, doi: 10.1109/JPROC.2017.2761740.

[44] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, Aug. 2013, doi: https://doi.org/10.1177/0278364913495721

[45] J. Lee, B. Bagheri, and H.-A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, no. 1, pp. 18–23, Jan. 2015, doi: https://doi.org/10.1016/j.mfglet.2014.12.001. Available: https://www.sciencedirect.com/science/article/pii/S221384631400025X

[46] L. Brunke et al., "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 5, pp. 411-444, 2021, doi: 10.1146/annurev-control-042920-020211.

[47] U. Kiencke and L. Nielsen, *Automotive Control Systems: For Engine, Driveline, and Vehicle*, 2nd ed. Berlin, Germany: Springer, 2005. doi: 10.1007/b137680.

[48] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531. Full text on arXiv. https://arxiv.org/abs/1503.02531

[49] Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. Advances in neural information processing systems, 28. https://proceedings.neurips.cc/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf

[50] J. B. Heywood, *Internal Combustion Engine Fundamentals*, 2nd ed. New York, NY, USA: McGraw-Hill, 2018.

[51] https://drive.google.com/drive/folders/1YmcrS547wnPwUy2jQyUbWOd9gS5x5l3c?usp=sharing [mohamed Abdalla matlab work].