Proceedings of Combustion Institute – Canadian Section
Spring Technical Meeting
University of Calgary
May 12-15, 2025

arXiv:2504.16875v2 [cs.LG] 6 May 2025

# Hybrid Reinforcement Learning and Model Predictive Control for Adaptive Control of Hydrogen-Diesel Dual-Fuel Combustion

J. Bedei, A. Winkler, J. Andert

*Teaching and Research Area Mechatronics in Mobile Propulsion, RWTH Aachen University, Aachen, Germany*

M. McBain, C.R. Koch and D. Gordon*

*Department of Mechanical Engineering, University of Alberta, 116 St and 85 Ave, Edmonton, AB, Canada, T6G 2R3*

**Abstract**

Reinforcement Learning (RL) and Machine Learning Integrated - Model Predictive Control (ML-MPC) are promising approaches for optimizing hydrogen-diesel dual-fuel (H2DF) engine control, as they can effectively manage multiple-input multiple-output systems and control nonlinear processes. ML-MPC is advantageous for providing safe and optimal control inputs, ensuring the engine operates within predefined safety limits. In contrast, RL is distinguished by its adaptability to changing conditions through its learning-based approach. However, the practical implementation of either method alone poses significant challenges. RL requires high variance in control inputs during early learning phases, which can pose risks to the system by potentially executing unsafe actions, leading to mechanical damage. Conversely, ML-MPC relies on an accurate system model to generate optimal control inputs and has limited adaptability to system drifts, such as injector aging, which naturally occur in engine applications.

To address these limitations, this study proposes a hybrid RL and ML-MPC approach that uses a well-established ML-MPC framework while incorporating an RL agent to dynamically adjust the ML-MPC load tracking reference in response to changes in the environment. At the same time, the ML-MPC ensures that actions stay safe throughout the RL agent's exploration. To evaluate the effectiveness of this approach, fuel pressure is deliberately varied to introduce a model-plant mismatch between the ML-MPC and the engine test bench. The result of this mismatch is a root mean square error (RMSE) in indicated mean effective pressure (IMEP) of 0.57 bar when running the ML-MPC standalone. The experimental results demonstrate that RL successfully adapts to changing boundary conditions by altering the tracking reference while ML-MPC ensures safe control inputs, keeping pressure rise rates constrained. The quantitative improvement in load tracking by implementing RL is an RSME in IMEP of 0.44 bar. These findings highlight the feasibility of combining RL and ML-MPC for engine control and encourage further research on learning-based and adaptive control strategies in safety-critical real-world environments, contributing to the advancement of sustainable transportation technologies.

## 1 Introduction

In 2018, 12% of all vehicles sold annually in Alberta were Class 8 heavy-duty vehicles. This small portion of vehicles contributed to 54% of the fuel burned in the province. Therefore, focusing efforts on developing solutions that reduce emissions by using alternative fuels can impact the reduction of carbon emissions. Hydrogen can be used in diesel engines by incorporating dual-fuel combustion processes, referred to as hydrogen–diesel dual-fuel (H2DF). By substituting a portion of the diesel fuel with port-injected hydrogen, these engines can substantially reduce greenhouse gas emissions while utilizing existing production diesel engines. However, when burned in a combustion engine, hydrogen exhibits nonlinear and complex combustion characteristics. The potential for abnormal combustion, particularly at higher loads, can limit the amount of hydrogen that can be safely added. These factors introduce new challenges for engine control, necessitating advanced control strategies [1]

*Corresponding author: dgordon@ualberta.ca

to ensure efficiency, safety, and accurate load tracking. Numerous control strategies have been proposed in the literature. One of these methods is Machine Learning Integrated - Model Predictive Control (ML-MPC), which has demonstrated encouraging results in Compression Ignition (CI) engines [2, 3] and Homogeneous Charge Compression Ignition (HCCI) engines [4, 5]. The four key advantages to using ML-MPC for H2DF are [2]: 1) constraints for state, input, and output variables. 2) uses a future horizon to improve optimal output for the set control law. 3) capable of managing uncertainties in the system's delays, parameters, and nonlinearity. 4) requires no complex physical plant model compared to traditional MPC.

However, ML-MPC faces a key challenge experienced by most advanced control methods: all potential system states cannot be considered. This can prevent the control system from reaching the desired output, as there could be key elements of the system causing disturbances that the controller cannot account for, resulting in a model-plant mismatch. An example of this is the aging of the system. A promising approach to address this challenge is the integration of Reinforcement Learning (RL) with ML-MPC, where RL enables the ML-MPC to maintain robust control performance despite such discrepancies. The combination of RL and ML-MPC has already shown great potential, for instance, in nonlinear quadcopter trajectory tracking [6] or building energy management [7].

RL is an ML algorithm that has achieved widespread success for engine control in various simulation environments [8, 9] by learning optimal actions through exploration of system states and maximizing the long-term reward. However, the application of RL in real-world environments remains limited due to safety concerns, particularly the risks associated with high-variance exploratory control inputs in early learning phases. In [10], RL was used to optimize fuel injection in spark-ignition engines, reducing cycle-to-cycle variability and improving fuel efficiency. In [11], RL-based HCCI control with real-time safety monitoring to ensure safe operation is presented. However, standalone RL in real-world applications remains challenging due to limited stability guarantees and high data demands in safety-critical systems. Integrating RL with ML-MPC enables adaptation to system drift, while using ML-MPC's stability and existing process model to ensure safe exploration.

In this work, the integration of RL and ML-MPC is presented in detail. The main goal is to adapt the ML-MPC to a changing plant, represented by changing the injection pressure to simulate injector drift (injector aging) and evaluate the controller's performance. The potential of RL used to enhance the performance of the ML-MPC for an internal combustion engine is experimentally shown.

## 2 Methodology

### *2.1 Experimental setup*

The hybrid RL and ML-MPC approach, schematically shown in Figure 1, is experimentally tested on a single cylinder of a Cummins QSB 4.5L 4-cylinder engine. A dSPACE MicroAutoBox II (MABX) rapid control prototyping (RCP) engine control unit (ECU) is used to control the engine and determine critical combustion metrics in real-time [5]. The combustion metrics considered in this study consist of indicated mean effective pressure (IMEP), maximum pressure rise rate (MPRR) and nitrogen oxide emissions ($NO_x$) while the controls set by RL-MPC are the start of injection of diesel (SOI diesel), duration of injection of hydrogen gas (DOI hydrogen), and duration of injection of diesel (DOI Diesel). The combustion metrics are transmitted to a Raspberry Pi 400 (RPI), where the ML-MPC calculation is executed on an ARM Cortex A72 processor. In addition, the RPI is provided with a reference load profile. The reference load profile consists solely of a target IMEP.

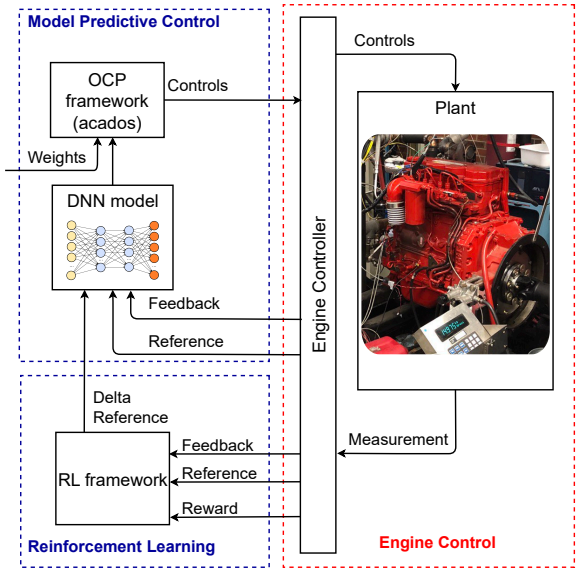



Fig. 1: Hybrid Reinforcement Learning and Model Predictive Control Approach.

An RL agent is established on a standalone PC with a Linux operating system. The RL agent receives the same information as the RPI, along with a reward calculated on the MABX for each cycle. The agent calculates

a reference load profile change, which is sent to the RPI to modify the ML-MPC reference trajectory, thus affecting the ML-MPC optimization and the resulting ECU control inputs for the next cycle.

### *2.2 Reinforcement Learning*

RL is a machine learning approach in which an agent learns to make decisions through interaction with an environment - in this case, comprising both the ML-MPC and the real engine (Figure 1). By receiving rewards $r$ from the environment, the agent gradually improves its performance by training an actor, which defines the agent's policy $\mu$ that aims to maximize the discounted cumulative reward $\sum_{i=0}^{\infty} \gamma^i \cdot r_i$. This so-called return is estimated by a critic, which is trained on the data generated during the agent's exploration. In this study, both the actor and the critic are approximated by neural networks. To enable the application of RL in real time, the Open Source Learning and Experiencing Cycle Interface (LExCI) [12] is used, which facilitates the deployment of RL algorithms on embedded hardware.

The Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm is used in this study. Details on its internal mechanisms can be found in [13]. The presented approach demonstrates how the flexibility and adaptability of RL can be harnessed to compensate for model–plant mismatches, thereby maintaining high ML-MPC performance even in the presence of system drifts. To support the ML-MPC algorithm, the action $a$ of the RL agent provides an offset on the load reference $\Delta y_{\mathrm{ref,IMEP},i}$ through its policy $\mu$ based on the state $s_{i-1}$ that includes the combustion metrics as well as the desired IMEP of the current and last cycle.

$$\Delta y_{\mathrm{ref,IMEP},i} = a_i = \mu\left(s_{i-1}\right), \quad s_{i-1} = \begin{bmatrix} y_{\mathrm{IMEP},i-1} \\ y_{\mathrm{NO_x},i-1} \\ y_{\mathrm{MPRR},i-1} \\ y_{\mathrm{ref,IMEP},i} \\ y_{\mathrm{ref,IMEP},i-1} \end{bmatrix} \in \mathbb{R}^5 \tag{1}$$

The primary objective of the RL agent is to enhance the load tracking performance under system drift, while the ML-MPC ensures safety and addresses additional control objectives such as emission reduction and fuel efficiency. Consequently, the reward function focuses exclusively on the load tracking error. The reward for each combustion cycle $i$ is defined as:

$$r_i = -0.05 \cdot \left(y_{\mathrm{ref,IMEP},i} - y_{\mathrm{IMEP},i}\right)^2 - 0.45 \cdot \tanh\left(\left(y_{\mathrm{ref,IMEP},i} - y_{\mathrm{IMEP},i}\right)^2\right) \tag{2}$$

The tanh term provides strong gradients near the target and saturates for large deviations, preventing exploding gradients and contributing to the numerical stability of the training using the gradient descent algorithm. Meanwhile, the small quadratic term ensures a non-zero gradient far from the target, maintaining motivation for the agent to adjust its policy [11].

### *2.3 Machine learning integrated model predictive control*

To model the H2DF engine performance and emissions, a deep neural network (DNN) with seven hidden layers (six Fully Connected (FC) layers and one GRU layer) similar to the model shown in [2]. This model is then formulated using a nonlinear state-space representation to allow for integration into the MPC framework provided by `acados` as described in [3]. Thus, the discrete Optimal Control Problem (OCP) is defined based on the standard `acados` formulation [14] as follows:

$$\begin{aligned} \min_{\substack{\Delta u_0,\ldots,\Delta u_{N-1} \\ x_0,\ldots,x_N \\ y_0,\ldots,y_N}} \quad & \sum_{i=0}^{N} \|y_{ref,i} - y_i\|_Q^2 + \|\Delta u_i\|_R^2 \\ \text{s.t.} \quad & x_0 = \left[x_i,\ u_{i-1}\right]^\top, \\ & x_{i+1} = f(x_i, \Delta u_i) \qquad \forall i \in \mathbb{H} \setminus N, \\ & y_i = g(x_i, \Delta u_i) \qquad \forall i \in \mathbb{H}, \\ & u_{\min} \le F_u \cdot u_i \le u_{\max} \quad \forall i \in \mathbb{H}, \\ & y_{\min} \le F_y \cdot y_i \le y_{\max} \quad \forall i \in \mathbb{H} \end{aligned} \tag{3}$$

where $\mathbb{H} = \{0, 1, \ldots, N\}$. Here $x_i$ are the internal model states (corresponding to the GRU: eight hidden states, $h_i$), $y_i$ the model outputs, and $u_i$ the model inputs. These are:

$$x_i = \begin{bmatrix} h_{i-1} \end{bmatrix} \in \mathbb{R}^8, \quad y_i = \begin{bmatrix} y_{\mathrm{IMEP},i} \\ y_{\mathrm{NO_x},i} \\ y_{\mathrm{MPRR},i} \end{bmatrix} \in \mathbb{R}^3, \quad u_i = \begin{bmatrix} y_{\mathrm{IMEP},i-1} \\ u_{\mathrm{SOI,diesel},i} \\ u_{\mathrm{DOI,hydrogen},i} \\ u_{\mathrm{DOI,diesel},i} \end{bmatrix} \in \mathbb{R}^4. \tag{4}$$

The reference $y_{ref}$ and the weighting matrix $Q$ are selected such that deviations from the requested IMEP are penalized while minimizing $\mathrm{NO_x}$ emissions, duration of injected diesel fuel DOI,diesel($i$) and hydrogen DOI,hydrogen($i$) and change in control input $\Delta u$. Therefore, the specific cost function $J$ is specified as:

$$\begin{aligned} J = \sum_{i=0}^{N} & \underbrace{\|y_{ref,\mathrm{IMEP},i} + \Delta y_{\mathrm{ref,IMEP},i} - y_{\mathrm{IMEP},i}\|^2_{q_{\mathrm{IMEP}}}}_{\text{Reference Tracking}} + \underbrace{\|y_{\mathrm{MPRR},i}\|^2_{q_{\mathrm{MPRR}}}}_{\text{Combustion Noise Reduction}} + \underbrace{\|y_{\mathrm{NO}_x,i}\|^2_{q_{\mathrm{NO}_x}}}_{\text{Emission Reduction}} \\ & + \underbrace{\|u_{\mathrm{DOI,diesel},i}\|^2_{r_{\mathrm{DOI,fuel}}} + \|u_{\mathrm{DOI,hydrogen},i}\|^2_{r_{\mathrm{DOI,hydrogen}}}}_{\text{Fuel / Hydrogen consumption reduction}} + \underbrace{\|\Delta u_i\|^2_R}_{\text{Oscillation Reduction}} \end{aligned} \tag{5}$$

One significant advantage of using ML-MPC for combustion control is the ability to impose constraints on inputs and outputs to ensure safe engine operation. $F_u$ and $F_y$ in Eq. 3 are diagonal matrices that map the bounded outputs and inputs. The control outputs are limited to match the hardware used ($u_{\mathrm{min,max}}$) while constraints imposed on the outputs ($y_{\mathrm{min,max}}$) are used to guarantee safe engine operation.

For the baseline ML-MPC testing, $\Delta y_{\mathrm{ref,IMEP},i} = 0$, and when integrated with RL, the reference is modified by the RL agent as described in section 2.2.

## 3 Results

Initially, the ML-MPC's DNN is trained on data generated using a pseudorandom binary sequence on the controls while operating with a diesel fuel rail pressure of 1000 bar [2]. To evaluate the hybrid RL/ML-MPC approach, the fuel rail pressure is reduced to 800 bar, introducing a model-plant mismatch. Figure 2 illustrates the performance of the ML-MPC under this mismatch and the improvements achieved through the hybrid RL/ML-MPC approach.

A reference load profile is used to compare the controllers' performance. This reference load profile ranges from 4.5 bar to 9 bar IMEP with several step changes and linear changes in load. The reference load profile is illustrated in red in Figure 2.

The actual IMEP of the ML-MPC with model-plant mismatch is shown in light blue. The deviations from the IMEP reference demonstrate that the ML-MPC struggles to track the profile and consistently underestimates the required fuel energy. This results from the rail pressure not being provided as an input to the ML-MPC, leaving the controller unaware of the resulting change in the environment. The key performance indicator (KPI) used to evaluate load tracking performance is the root mean square error (RMSE) between the actual IMEP and the reference IMEP. Under this model-plant mismatch, the ML-MPC achieves an RMSE of 0.57 bar.

With a KPI baseline determined, the RL agent is deployed to reduce the RMSE in engine load. The RL agent is trained for a total of 56 episodes with 2000 combustion cycles each until it converges. Starting from randomly initialized actor and critic networks, the RL agent performs repeated runs of four training episodes with noise added to the policy output, followed by one validation episode without added noise. Each training episode uses random step changes for the IMEP reference between 4.5 and 9 bar, and the agent offsets its policy $\mu$ by Gaussian noise to explore the state-action space. The resulting data is used for network training to improve the agent's policy to maximize the cumulative reward.

After completing the 56 episodes, the hybrid RL/ML-MPC approach improved load tracking compared to the ML-MPC, achieving an RMSE of 0.44 bar. The adapted reference profile provided by the RL agent and the resulting IMEP trace are expressed in yellow and dark blue in Figure 2. The agent introduces a consistent offset

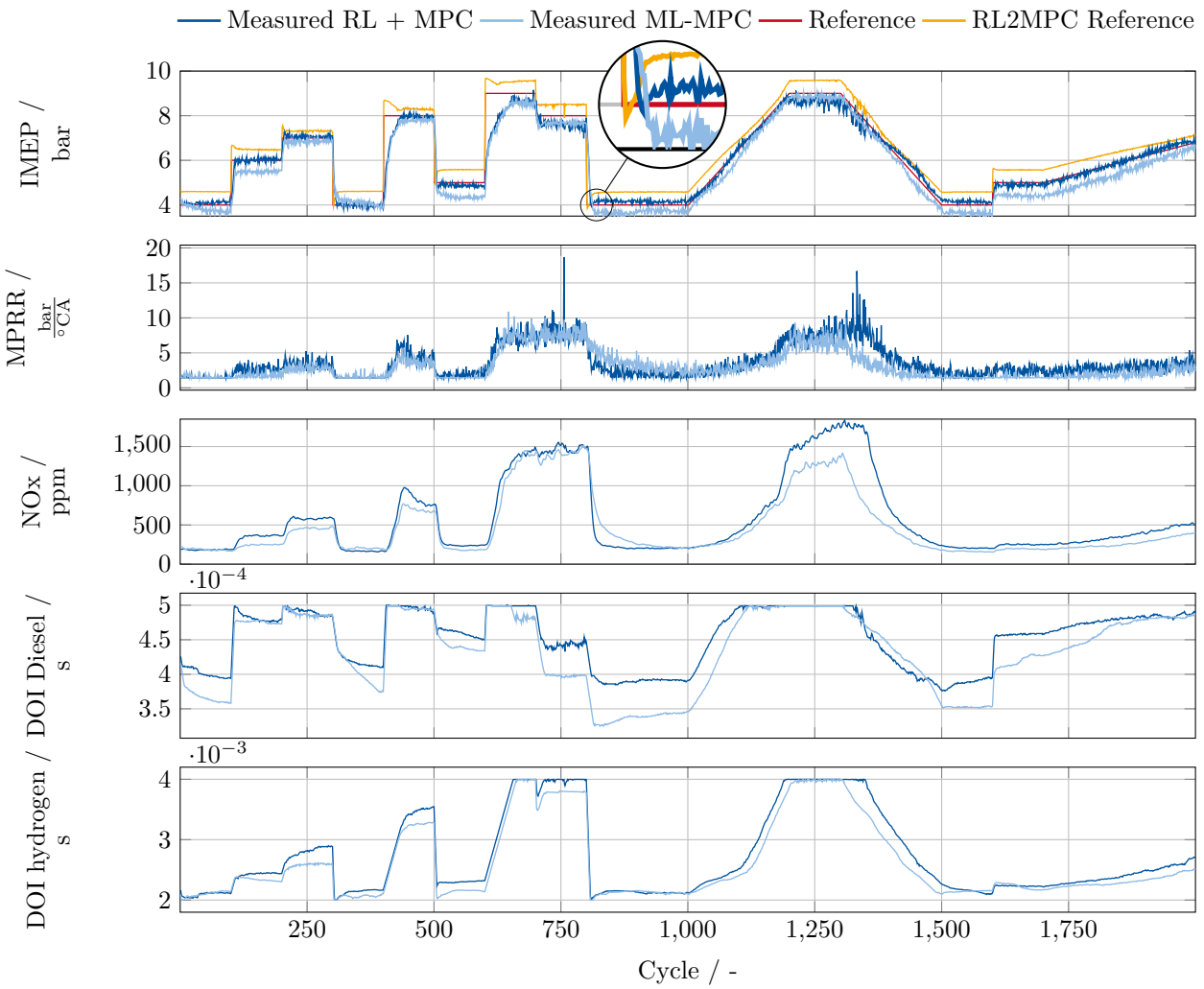



Fig. 2: Comparison of Control Results of MPC and Hybrid RL/ML-MPC approach with model-plant mismatch.

compared to the IMEP reference. This offset correlates directly with the underestimated IMEP in the initial model-plant mismatch in most instances. Two instances, from cycle 657 to 800 and from cycle 1190 to 1350, occur where the agent increases its reference profile offset in an attempt to improve load tracking. However, the load tracking performance cannot improve because both DOIs are constrained by the ML-MPC for safety reasons.

Moreover, the RL learned an overshooting feature at each step change in the load profile. An example of this overshoot feature can be observed in the zoomed-in bubble in Figure 2. This has the potential to improve the load tracking performance; however, due to the DOIs rate constraints set by the ML-MPC for safety, the performance improvement is limited. Future research will exploit this potential to improve load tracking.

## 4 Conclusion

In this study, RL is integrated with ML-MPC to control a H2DF combustion process within a single cylinder of a Cummins QSB 4.5L 4-cylinder engine. A model-plant mismatch is introduced by reducing the fuel rail pressure to simulate injector wear and aging. The ML-MPC without RL underestimates the required DOIs to meet the reference load profile. The RMSE between the actual IMEP of the ML-MPC and the target IMEP is 0.57 bar. An RL agent is then introduced to adjust the reference load profile to improve load tracking. The agent successfully develops an adapted reference with an offset from the initial reference corresponding to the absolute difference between the original model-mismatch ML-MPC IMEP and the target IMEP. Additionally,

an overshoot feature is added to the reference, causing the ML-MPC to reach the target IMEP in fewer cycles. The resulting load tracking error with RL-MPC improves to an RMSE of 0.44 bar. Therefore, when introducing a plant-model mismatch, the RL and ML-MPC hybrid approach improves load tracking. However, there are three key limitations to implementing RL with ML-MPC: 1) the RL/ML-MPC hybrid approach may fail if the model and plant are substantially different. 2) potential improved solutions can be limited by constraints set by the ML-MPC. These constraints within the ML-MPC may no longer be accurate. 3) limited potential to alter controller behavior due to the agent only adapting the reference load profile. Though these limitations are present in this study, the results provide a foundation for future research into this hybrid approach to engine control. Future research will expand the RL agent's ability to alter the ML-MPC behavior by adjusting the ML-MPC's weights to improve load tracking.

## Acknowledgments

The authors would like to support the previous work by Alexander Winkler has formed the foundations for this research. Funding for this work has been provided by the RWTH Aachen – University of Alberta Junior Research Fellowship and NSERC Discovery Grant RGPIN-2024-04990.

## References

[1] K. J. Astrom, B. Wittenmark, Adaptive Control, 2nd Edition, Addison-Wesley Longman Publishing Co., Inc., USA, 1994.

[2] D. C. Gordon, A. Norouzi, A. Winkler, J. McNally, E. Nuss, D. Abel, M. Shahbakhti, J. Andert, C. R. Koch, End-to-end deep neural network based nonlinear model predictive control: Experimental implementation on diesel engine emission control, Energies 15 (24) (2022) 9335. doi:10.3390/en15249335.

[3] A. Norouzi, S. Shahpouri, D. Gordon, A. Winkler, E. Nuss, D. Abel, J. Andert, M. Shahbakhti, C. R. Koch, Deep learning based model predictive control for compression ignition engines, Control Engineering Practice 127 (2022) 105299. doi:10.1016/j.conengprac.2022.105299.

[4] D. C. Gordon, Realtime Machine Learning based In-Cycle Control of Homogeneous Charge Compression Ignition, Ph.D. thesis, University of Alberta (2023).

[5] D. C. Gordon, A. Winkler, J. Bedei, P. Schaber, S. Pischinger, J. Andert, C. R. Koch, Introducing a deep neural network-based model predictive control framework for rapid controller implementation, in: 2024 American Control Conference (ACC), 2024, pp. 5232–5237. doi:10.23919/ACC60939.2024.10644830.

[6] A. Ghezzi, R. Reiter, K. BaumgÃĪrtner, A. Bemporad, M. Diehl, A numerically efficient method to enhance model predictive control performance with a reinforcement learning policy, arXiv preprint arXiv:2504.02710 (2025).

[7] J. Arroyo, C. Manna, F. Spiessens, L. Helsen, Reinforced model predictive control (rl-mpc) for building energy management, Applied Energy 309 (2022) 118346. doi:https://doi.org/10.1016/j.apenergy.2021.118346.
URL https://www.sciencedirect.com/science/article/pii/S0306261921015932

[8] A. Norouzi, S. Shahpouri, D. Gordon, M. Shahbakhti, C. R. Koch, Safe deep reinforcement learning in diesel engine emission control, Proceedings of the Institution of Mechanical Engineers. Part I, Journal of systems and control engineering 237 (8) (2023) 1440–1453. doi:10.1177/09596518231153445.

[9] V. Sharma, A. Winkler, A. Norouzi, J. Andert, D. Gordon, H. Guo, Safe reinforcement learning-based control for hydrogen diesel dual-fuel engines (2025). doi:10.48550/arXiv.2502.09826.

[10] B. P. Maldonado, B. C. Kaul, C. D. Schuman, S. R. Young, Reinforcement learning applied to dilute combustion control for increased fuel efficiency, International Journal of Engine Research (2024). doi:10.1177/14680874241226580.

[11] J. Bedei, L. Koch, K. Badalian, A. Winkler, P. Schaber, J. Andert, Safe reinforcement learning for real-world engine control (2025). doi:10.48550/arXiv.2501.16613.

[12] K. Badalian, L. Koch, T. Brinkmann, M. Picerno, M. Wegener, S.-Y. Lee, J. Andert, LExCI: A framework for reinforcement learning with embedded systems, Applied Intelligence 54 (8) (2024) 8384–8398. doi:10.1007/s10489-024-05573-0.
URL https://doi.org/10.1007/s10489-024-05573-0

[13] S. Fujimoto, H. van Hoof, D. Meger, Addressing function approximation error in actor-critic methods (2018). doi:10.48550/arXiv.1802.09477.

[14] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, M. Diehl, acados – a modular open-source framework for fast embedded optimal control, Mathematical Programming Computation (2021). doi:10.1007/s12532-021-00208-8.