2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Air Compressor Control Optimization in Commercial Vehicles Using Reinforcement Learning

Fabian Schuppert[1,2,3*], Raul Gorek[1,2], Bodo Rosenhahn[1] and Timo von Marcard[3]

***Abstract*— Air compressors are critical for braking, suspension, and auxiliary systems in heavy-duty commercial vehicles, but their operation increases fuel consumption and contributes to mechanical wear. Existing rule-based controllers cannot anticipate future driving conditions or leverage correlations across vehicle states, limiting efficiency. We propose a Reinforcement Learning (RL) approach for predictive compressor control, supported by a tunable Hidden Semi-Markov Model (HSMM) generator that produces realistic Controller Area Network (CAN) driving profiles. The generator reproduces braking dynamics that govern air consumption, enabling scalable RL training without reliance on extensive real-world data. Using Proximal Policy Optimization (PPO), the RL agent outperforms rule-based baselines by balancing two objectives: maximizing compressor operation during free-air phases—driving states such as downhill or deceleration where kinetic energy can power the compressor without fuel—and limiting switch frequency to ensure mechanical longevity. These results demonstrate the feasibility of RL for subsystem-level control and its potential as a foundation for future efficiency-oriented vehicle strategies.**

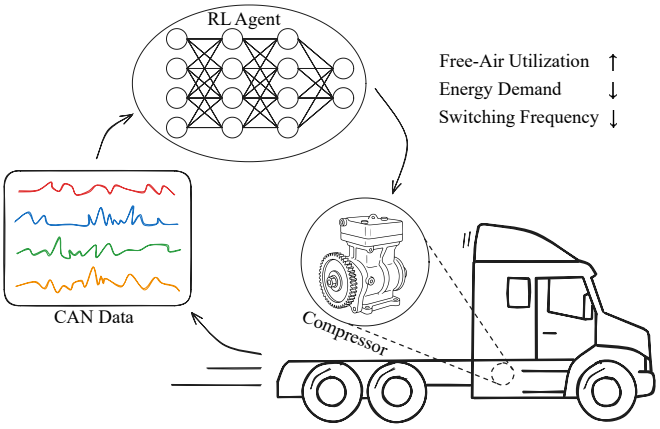


Fig. 1. Schematic overview of the proposed RL framework for compressor control in heavy-duty trucks. CAN bus driving data is used by an RL agent to learn a control policy that directly controls the air compressor. The learned policy improves key objectives: higher Free-Air Utilization (↑), lower Energy Demand (↓), and lower Switch Frequency (↓)

I. INTRODUCTION

Artificial Intelligence (AI) has demonstrated impressive success in domains involving complex dynamic systems, such as robotics [1], autonomous driving [2], and energy management [3]. Within the transportation sector, learning-based methods are increasingly explored not only for end-to-end vehicle automation [4] but also for improving the efficiency of individual onboard systems [5]. Modern commercial vehicles contain numerous subsystems—ranging from powertrain and thermal management to pneumatic supply—whose operation significantly impacts overall energy demand and component lifetime. Among these, the pneumatic system plays a central role, as it provides compressed air for braking, suspension, and auxiliary functions.

Unlike passenger cars that use hydraulic brakes, heavier vehicles rely on compressed air as an energy reservoir to deliver the high braking forces required for safe operation. This system is characterized by a steady air consumption pattern, overlaid by short, high-demand events such as braking and suspension adjustments. Air compressors, typically mechanically driven by the engine or a dedicated electric motor, replenish the pressure reservoirs as needed. However, conventional compressors and control strategies are energetically inefficient, especially during offload phases or low air demand, leading to excessive fuel consumption.

[1]Inst. for Information Processing, Leibniz University Hannover, Germany
[2]ZF Commercial Vehicle Systems Hannover GmbH, Germany
[3]Hochschule Hannover, Germany
*corresponding author fabian.schuppert@zf.com

Modern systems attempt to mitigate this through power reduction mechanisms or clutches [6], and electric vehicles offer further control flexibility via independent motors [7]. Yet, existing control remains primarily rule-based, relying on predefined thresholds or simple state machines. Such approaches cannot anticipate future driving states or leverage correlations across vehicle signals, and therefore miss opportunities to operate the compressor during energetically favorable free-air phases—vehicle states such as downhill or deceleration—where kinetic energy can drive the compressor without additional fuel consumption.

In this work, we propose the use of Reinforcement Learning (RL) to enable predictive, energy-aware control of air compressors in commercial vehicles. A schematic overview of the approach is shown in Fig. 1. By learning from data, an RL agent can anticipate favorable free-air phases, exploit multivariate correlations across vehicle states (e.g., speed, load, road slope, traffic), and optimize control decisions with respect to both short-term dynamics and long-term efficiency. We evaluate the approach against three performance targets:

- **Free-Air Utilization [%]**—proportion of required air supplied during free-air phases, i.e., driving states where the compressor can be operated without additional fuel consumption,
- **Energy Demand [kJ/h]**—total compressor energy use, reflecting both free-air utilization and the varying efficiency of compressor operation across different driving states,

- **Switch Frequency [1/h]**—normalized rate of compressor state changes, which directly impacts mechanical wear, maintenance intervals, and overall system reliability.

Training such agents directly on real vehicles is infeasible due to safety, cost, and data scarcity. To address this, we develop a configurable profile generator based on a Hidden Semi-Markov Model (HSMM) that synthesizes Controller Area Network (CAN) driving sequences. Unlike full vehicle simulators, the generator is lightweight, governed by interpretable parameters, and computationally efficient, making it suitable for large-scale scenario generation. Profiles can be generated offline, and the RL training environment reduces to a compact dynamic model of the compressor, further improving simulation efficiency.

Our main contributions are:

- **A configurable and efficient HSMM-based generator** that produces realistic CAN driving profiles governed by interpretable distributions and parameters, enabling scalable training without reliance on full vehicle simulation models;
- **A learning-based compressor controller** trained on HSMM-generated synthetic CAN profiles that outperforms two rule-based baselines across free-air utilization, energy demand, and switch frequency, demonstrating feasibility and enabling future extensions with contextual signals (e.g., GPS-based slope information).

II. RELATED WORK AND TECHNICAL BACKGROUND

Conventional air compressor control in commercial vehicles has long relied on on-off (two-point) controllers. In this scheme, the compressor is engaged once reservoir pressure drops below a lower threshold and disengaged when an upper threshold is reached, without consideration of vehicle dynamics or upcoming demand. More advanced mechatronic systems integrate real-time vehicle signals (e.g., velocity, engine load) via the CAN bus to better utilize rolling phases or excess kinetic energy for compression. Despite these enhancements, modern implementations remain predominantly rule-based, typically realized as threshold logic or finite state machines.

Directly related research on learning-based compressor control is scarce. Nevertheless, the underlying challenge——efficiently managing a finite reservoir that supplies a critical subsystem——has been widely studied in other domains. Examples include predictive battery management [8], thermal regulation in vehicle cabins [3], and hybrid energy system optimization [5], where the objective is to replenish or regulate a limited resource while minimizing energy costs. In the energy sector, reinforcement learning has also been applied to improve scheduling, resource allocation, and renewable integration [8]–[11]. These works demonstrate that RL can capture long-term trade-offs and exploit predictive information to achieve efficient resource management—properties directly relevant to compressor control.

Within the automotive domain, RL research is largely driven by autonomous driving tasks, including motion planning, driving policy, and safe decision-making under uncertainty [12]–[17]. Kiran et al. provide a comprehensive survey of RL methods for these applications, emphasizing challenges of sample efficiency and safe exploration [18]. Beyond autonomy, RL has also been explored for subsystem optimization, for example in modular vehicle control architectures [19] and powertrain management [20]. These works highlight the growing interest in applying RL beyond driving tasks, but none have addressed compressor control.

To the best of our knowledge, no prior work has investigated the application of RL to air compressor control in commercial vehicles. This gap is significant, as compressor control shares many of the characteristics of domains where RL has shown promise: high-dimensional state spaces, dynamically changing operating conditions, sparse and delayed reward signals, and the need to balance efficiency with component longevity.

A key enabler for RL is access to sufficient training and evaluation data. In the automotive domain, Virtual Measurement Campaigns (VMCs) are widely used to synthesize reproducible test scenarios for powertrain optimization and ECU validation [21], [22]. Traffic-level simulators such as SUMO [23] and VTD [24] provide large sets of diverse driving profiles, while standards like OSI [25] facilitate modular data exchange in virtual testing pipelines. In parallel, several real-world CAN-bus datasets have been released [26], [27], primarily for applications such as intrusion detection. These resources have advanced data-driven experimentation, but they are not ideally suited for compressor-control RL: standardized cycles are too rigid to capture stochastic real-world variability, recorded CAN traces are scarce, non-reproducible, and not subsystem-specific, and traffic-level simulators, though powerful, require heavy scenario setup and computational cost that is disproportionate for training policies at the subsystem level.

To address this gap, we introduce a configurable HSMM-based CAN data generator. It produces diverse and realistic driving profiles governed by interpretable distributions and parameters, enabling targeted variation across urban, rural, and highway regimes without reliance on full vehicle simulation. Compared to existing solutions, our generator is computationally efficient, as vehicle dynamics are precomputed offline and only the air reservoir evolves online. It is easy to configure to a broad range of driving scenarios through high-level parameters such as segment composition and state sojourn times. Finally, its modular design allows straightforward integration of additional contextual signals such as GPS-based slope or route plans. These properties establish the generator as a scalable and reproducible foundation for systematic RL training and benchmarking of compressor control.

III. CAN PROFILE GENERATOR

A. Motivation

In modern vehicles, the CAN bus continuously transmits time-stamped messages containing sensor readings and control signals, providing a detailed representation of the vehicle's operational state. These signals are essential for data-driven subsystem control. However, relying solely on real-world CAN traces to train RL agents is impractical: data collection is costly, labels are scarce, and the diversity of driving situations is limited to those observed. Moreover, RL benefits from controlled exposure to rare or extreme conditions that are unlikely to occur in practice. Our objective, therefore, is to synthesize realistic multivariate CAN time series that preserve the statistical structure and temporal dynamics of real driving.

To this end, we require a generative model that captures both the composition of route segments and the variability of driving states over time. A conventional Hidden Markov Model (HMM) [28] can represent transitions between contexts, but its implicit geometric assumptions on state durations often result in unrealistic dynamics, producing either overly frequent or excessively sparse transitions. We therefore adopt a Hidden Semi-Markov Model (HSMM) [29], which explicitly models state sojourn times. In our hierarchical design, the upper level governs route composition (highway, rural, urban, break), while the lower level specifies distributions over driving states associated with each segment (e.g., cruising, reduced speed, stop-and-go, idle). This dependency ensures that driving dynamics remain consistent with the current road context.

To formalize this, we represent a synthetic CAN driving sequence as a multivariate time series

$$\mathbf{y}_{1:T} = \left\{ \mathbf{y}_t \in \mathbb{R}^d \right\}_{t=1}^{T}, \tag{1}$$

where T denotes the sequence length and each observation vector $\mathbf{y}_t$ contains d CAN-relevant signals at time t. These signals comprise both kinematic variables and subsystem-related quantities relevant for pneumatic control, including:

- v_t: vehicle speed (m/s),
- a_t: longitudinal acceleration (m/s^2),
- n_t: engine speed (min^{-1}),
- g_t: current gear (categorical),
- P_t: engine power (W),
- b_t: brake pedal position (normalized),
- θ_t: road gradient (radians),
- $\dot{V}$: air flow rate (L/min),
- f_t: free-air availability (binary proxy).

The signals within a CAN profile are not independent—they are closely linked through physical laws, control strategies, and driver behavior. To generate realistic sequences $\mathbf{y}_{1:T}$, the generator must account for a range of interacting factors, including:

- Vehicle dynamics, such as aerodynamic drag, rolling resistance, slope-induced forces, and the available tractive effort

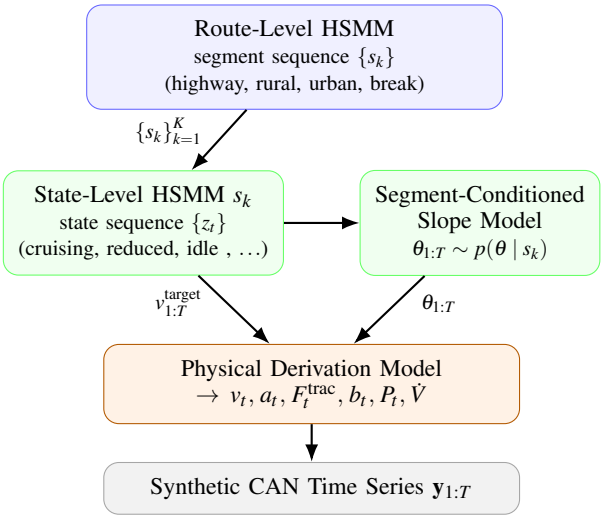


Fig. 2. Overview of the CAN data generator. A route-level HSMM produces a segment sequence $\{s_k\}$. For each segment, a state-level HSMM generates a target velocity profile $v_{1:T}^{\text{target}}$, while a segment-conditioned gradient model generates gradient profiles $\theta_{1:T}$. Both profiles feed a physical derivation model that enforces vehicle dynamics, resulting in a realistic multivariate CAN time series $\mathbf{y}_{1:T}$.

- Powertrain behavior, including how gears are selected, how the engine responds to throttle input, and how braking is applied
- Auxiliary system loads, particularly the air system's consumption patterns, which are influenced by brake actuation, leakages, and other pneumatic components

These interdependencies are essential to produce physically plausible and behaviorally realistic driving profiles.

B. Generator Architecture

Our CAN-profile generator is structured as a modular, hierarchical system that mimics the layered nature of real-world driving behavior. The pipeline consists of five main components, illustrated in Fig. 2.

1) Route-level Model: We model the overall driving sequence as a concatenation of route segments

$$\mathcal{S} = \{s_k\}_{k=1}^{K}, \quad s_k \in \{\text{Highway}, \text{Rural}, \text{Urban}, \text{Break}\} \tag{2}$$

Each segment type is associated with a duration distribution

$$D_s \sim \text{Triangular}(a_s, b_s, c_s) \tag{3}$$

where a_s, b_s and c_s denote the minimum, maximum and mode of the segment duration, respectively, or as an empirical histogram estimated from real driving data. This layer defines the overall structure of the route (e.g., long highway phases interspersed with urban traffic).

2) State-level Model: Each route segment s_k is modeled by a Hidden Semi-Markov Model (HSMM), which generates target velocity profiles by capturing the stochastic structure of short-term driving dynamics. An HSMM is defined by:

- A finite set of latent driving states $z_t \in \mathcal{Z}$, e.g., cruising, reduced, stop-and-go and idle

- Explicit sojourn time distributions $d_i(\tau)$, typically modeled as truncated Gaussian, Triangular or Exponential
- A transition matrix $A = [a_{ij}]$, where $a_{ij} = P(z_{t+1} = j | z_t = i)$
- Emission functions $e_i(v_t) = P(v_t | z_t = i)$, with v_t modeled via Gaussian or GMM distributions

Target velocity sequences $v_{1:T}^{\text{target}}$ are sampled by first generating latent state trajectories $z_{1:T}$ according to A and $d_i(\tau)$, and then drawing emissions from the corresponding distributions e_i. The resulting sequence represents a behavioral intention that will later be transformed into physically feasible motion by the longitudinal dynamics model.

3) Slope Model: Road slope is a key external factor influencing vehicle dynamics and, consequently, compressor operation. To capture this effect, we generate the slope profile $\theta_{1:T}$ separately from the HSMM-based speed profile. This separation enables flexible combinations of driving behavior and road conditions, while avoiding conflation of effects that are distinct in real-world driving.

Slope, however, cannot be modeled in isolation: it depends on the route segment type (e.g., highways are systematically flatter than urban roads), and it can in turn constrain feasible speeds through engine power limits. To preserve these dependencies while retaining modularity, slope generation is conditioned on the route-level model. In practice, slopes are sampled with variability that reflects the segment: short, steep fluctuations in cities, smoother transitions on rural roads, and gradual gradients on highways.

This design yields trajectories that are both realistic and computationally efficient. Slopes are sampled once per segment type, producing a piecewise-constant profile θ_t that simplifies the physical derivation while preserving the essential variability across route contexts. Although this abstraction omits fine-grained continuous fluctuations, the forward-simulation model remains physically consistent and sufficiently detailed for subsystem-level training.

4) Physical Derivation Model: We treat the target velocity $v_{1:T}^{\text{target}}$ from the HSMM as a behavioral intention. The actual v_t and a_t are computed via forward simulation under physical constraints.

At each timestep t, we first compute the **required acceleration** to reach the next target velocity:

$$a_t^{\text{req}} = \frac{v_{t+1}^{\text{target}} - v_t}{\Delta t}. \tag{4}$$

Reaching this target acceleration in practice requires counteracting the longitudinal resistances of the vehicle. We represent these resistive loads—aerodynamic drag, rolling resistance, and the component due to road grade—as

$$F_t^{\text{drag}} = \tfrac{1}{2}\rho C_d A v_t^2, \tag{5}$$

$$F_t^{\text{roll}} = C_r mg\cos(\theta_t), \tag{6}$$

$$F_t^{\text{slope}} = mg\sin(\theta_t), \tag{7}$$

where ρ is air density (kg/m^3), C_d the aerodynamic drag coefficient ($-$), A the frontal area (m^2), C_r the rolling resistance coefficient ($-$), m the vehicle mass (kg), g gravitational acceleration ($9.81\,\text{m}/\text{s}^2$), and θ_t the road grade angle (rad; positive uphill).

The **total tractive force required** is then given by the sum of inertial and resistive terms:

$$F_t^{\text{req}} = ma_t^{\text{req}} + F_t^{\text{drag}} + F_t^{\text{roll}} + F_t^{\text{slope}}. \tag{8}$$

The tractive effort that can be delivered is physically constrained by the engine's maximum power and the braking system's maximum deceleration. We therefore bound the available force between these two limits. The maximum engine force at a given speed v_t is derived from the speed-dependent power curve $P^{\max}(v_t)$ as

$$F^{\text{eng,max}}(v_t) = \frac{P^{\max}(v_t)}{v_t + \varepsilon}, \tag{9}$$

where ε is a small constant to avoid division by zero. The actual tractive force is then obtained by clipping the required force to the admissible range:

$$F_t^{\text{trac}} = \text{clip}\Big(F_t^{\text{req}}, -F^{\text{brk,max}}, F^{\text{eng,max}}(v_t)\Big), \tag{10}$$

with $F^{\text{brk,max}}$ denoting the maximum braking capacity.

The **vehicle dynamics** are then updated according to Newton's second law:

$$a_t = \frac{F_t^{\text{trac}} - F_t^{\text{drag}} - F_t^{\text{roll}} - F_t^{\text{slope}}}{m}, \tag{11}$$

$$v_{t+1} = v_t + a_t \Delta t. \tag{12}$$

Finally, additional CAN-relevant signals are derived from these quantities: engine speed n_t and gear g_t from (v_t, a_t), engine power $P_t = F_t^{\text{trac}} \cdot v_t$, a brake signal b_t from deceleration demand, and the air flow $\dot{V}$ based on braking events and leakage dynamics.

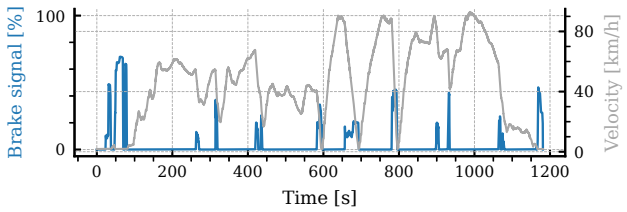
This process produces a corrected velocity and acceleration trajectory that respects real-world mechanical and physical constraints. It bridges the gap between high-level behavior modeling and low-level physical feasibility, ensuring that synthetic CAN profiles remain plausible and useful for downstream control policy training.

5) CAN Data Synthesis: The final vector $\mathbf{y}_t$ is assembled from the derived quantities, augmented by a binary "free-air" signal f_t defined via heuristic rules: $f_t = \mathbb{I}[a_t < 0 \wedge P_t < \varepsilon]$ indicating deceleration or coasting with low engine load—favorable for compressor operation.

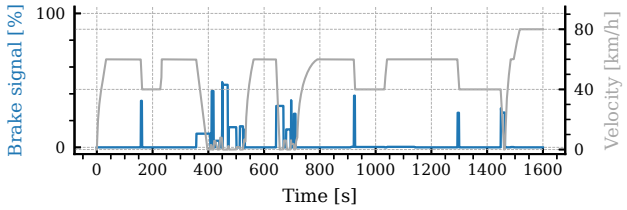
C. Evaluation Against Real-World Data

Evaluating the CAN data generator is challenging, as direct one-to-one comparisons with real driving sequences are infeasible due to the generator's stochastic nature. Instead, we assess its validity through a qualitative analysis, focusing on features most relevant to compressor control: braking patterns and their associated deceleration phases.

Fig. 3 illustrates that the generator reproduces these key aspects with high fidelity. Both the frequency and intensity of braking events are well captured, including the alternation between sharp and moderate actuation. The generated profiles also preserve realistic transitions between acceleration,



(a) exemplary real driving profile



(b) exemplary generated profile

Fig. 3. Comparison of real and synthetically generated driving profiles. The frequency and intensity of braking events—the primary contributors to air consumption—are well aligned across both datasets, which is critical for our application. Major acceleration and deceleration patterns also show close agreement. In contrast, phases of steady driving appear smoother in the generated sequences, whereas real-world data exhibit additional variability due to driver behavior and road conditions. Since these fluctuations have little impact on air pressure demand, the abstraction is acceptable for compressor-control training and can be refined in future work.

steady driving, and deceleration phases, ensuring that the air-consumption dynamics induced by braking are represented accurately.

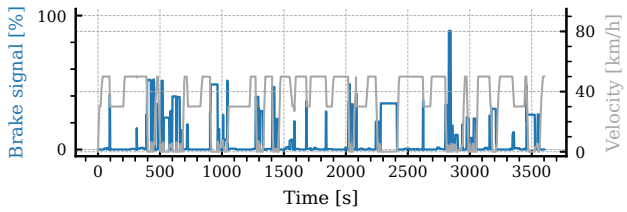
A noticeable deviation arises in constant-speed phases, where generated profiles exhibit smoother plateaus than real-world recordings. This reflects both reduced noise in the synthetic sequences and the absence of small fluctuations caused by driver behavior and road conditions. Since these variations have little influence on air pressure demand, the abstraction is acceptable for compressor-control training and can be refined in future extensions.

In summary, the generator successfully reproduces braking dynamics and major driving transitions—the dominant factors of compressor demand—thereby providing a reliable training basis for reinforcement learning strategies.

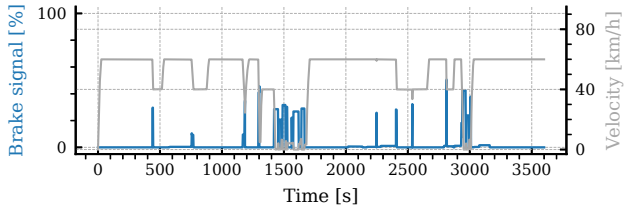
Beyond replicating a single reference drive, the generator adapts to diverse traffic regimes. Fig. 4 shows representative profiles for (a) city, (b) country road, and (c) highway driving. In each case, braking patterns and speed plateaus are preserved, enabling the synthesis of targeted datasets for task-specific training. The framework thus supports efficient development and evaluation of RL-based control strategies while maintaining fidelity to real-world driving behavior. Its modular structure further facilitates extension to different vehicle configurations, control units, or operating conditions.

IV. COMPRESSOR CONTROL VIA REINFORCEMENT LEARNING

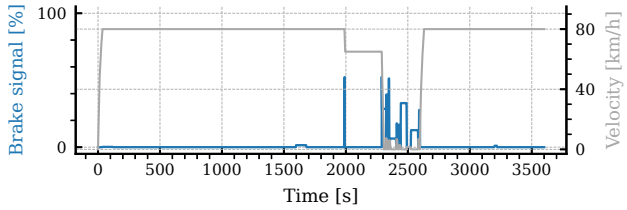
Building on the synthetic data foundation, we cast compressor control as a Markov Decision Process (MDP). In this formulation, the RL agent observes relevant vehicle



(a) City



(b) Country road



(c) Highway

Fig. 4. Generated driving profiles for three distinct road scenarios. Each panel shows brake actuation (blue) and vehicle velocity (grey), illustrating the generator's ability to reproduce characteristic dynamics across traffic regimes.

signals and selects compressor actions with the objective of maximizing long-term efficiency. Specifically, the policy seeks to increase energy savings by exploiting favorable driving conditions while limiting switch frequency to ensure mechanical longevity. This framework provides a principled basis for learning predictive, context-aware control strategies beyond conventional rule-based logic.

A. Simulation Environment

Compressor control is modeled as a discrete-time RL task within a parameterized simulation environment conforming to the OpenAI Gymnasium API [30]. The overall structure and the interaction between agent, compressor, and air reservoir are illustrated in Fig. 5.

Driving scenarios are provided by the CAN generator (Section III), which outputs a subset of preprocessed signals

$$\mathbf{y}_t = \{v_t, a_t, n_t, g_t, P_t, b_t, \theta_t, \dot{V}, f_t\},$$

capturing vehicle dynamics, braking events, etc. These signals are fixed during training and independent of compressor control. The agent observes an augmented state vector

$$\tilde{\mathbf{y}}_t = \mathbf{y}_t \cup \{p_t\},$$

where p_t denotes the air reservoir pressure. Unlike $\mathbf{y}_t$, this state evolves online in response to the agent's actions. This design reflects the real-world setting: vehicle dynamics

are exogenous to compressor control, while the reservoir pressure is directly influenced by the policy. It also improves computational efficiency, since complex vehicle dynamics are simulated once during CAN profile generation and only the pressure dynamics are updated online. The action space is binary, $u_t \in \{switch, idle\}$, corresponding to compressor activation or inactivity.

Reservoir pressure evolves according to the net balance of air inflow and outflow. When activated, the compressor contributes an inflow $\dot{V}_{t,s}$, obtained from a parameterized performance map of a commercial vehicle unit as a function of rotational speed n_t and back-pressure p_{t-1}. Outflow $\dot{V}_{t,d}$ comprises discrete consumptions from braking and gearbox actuation, along with continuous losses from leakage and auxiliary systems such as air suspension. Assuming a fixed reservoir volume and approximately constant temperature, volumetric flows are evaluated at reservoir conditions, so that a positive net flow increases pressure and a negative net flow reduces it. Safety-critical constraints—most importantly maintaining sufficient braking pressure—are enforced throughout training and evaluation in accordance with UN/ECE Regulation No.13 [31].

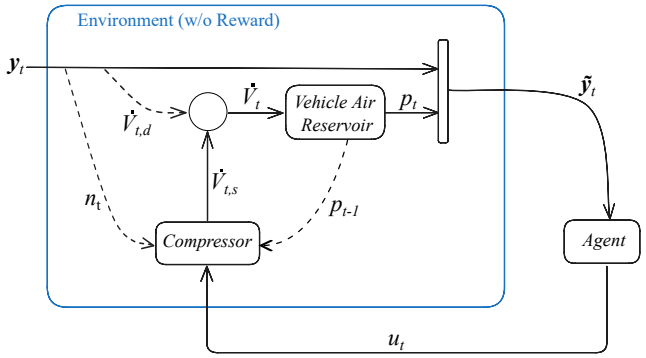


Fig. 5. Architecture of the compressor-control RL environment. Preprocessed CAN signals $\mathbf{y}_t$ provide fixed driving dynamics, while the reservoir pressure p_t evolves online under the influence of compressor actions u_t. The agent receives the full observation vector $\tilde{\mathbf{y}}_t = \mathbf{y}_t \cup \{p_t\}$ and learns to schedule compressor operation accordingly.

B. Reward Design

To guide the agent toward policies that balance efficiency, durability, and safety, we define a structured reward function aligned with the subsystem objectives. Building on the environment dynamics described above, the reward provides per-step shaping terms for survival, free-air exploitation, and switching behavior, complemented by a strong penalty for safety violations:

$$R_{t+1} = R_t + \begin{cases} -w_4 \cdot r_{\text{terminal}} & \text{if terminated} \\ w_1 \cdot r_{\text{live}} + w_2 \cdot r_{\text{free}} + w_3 \cdot r_{\text{few}} & \text{otherwise.} \end{cases} \tag{13}$$

The reward components are defined as follows:

- **Survival reward** (r_{live}, $w_1 = 1$): a small per-step reward that encourages continuous valid operation,
- **Free-air bonus** (r_{free}, $w_2 = 1$): a positive reward when the compressor is operated during energy-favorable free-air phases,
- **Switching penalty** (r_{few}, $w_3 = -10$): a cost assigned to each compressor toggling event, discouraging excessive wear,
- **Terminal penalty** (r_{terminal}, $w_4 = 300$): a strong penalty if system pressure falls below safe limits, enforcing compliance with braking requirements.

C. RL Training Setup

The RL agent is trained using Proximal Policy Optimization (PPO) within the Stable Baselines3 (SB3) framework [32]. We customized the SB3 PPO implementation with dynamic entropy scheduling to facilitate broad exploration early in training and policy refinement as convergence progresses. All hyperparameters were tuned empirically to ensure stable learning progression and are shown in Table I.

TABLE I
PPO HYPERPARAMETERS USED IN TRAINING.

Hyperparameter	Value
Network structure	3 hidden layers (8, 8, 4 units)
Activation function	Tanh
PPO clip factor ε	0.3
Entropy coefficient	$0.03 \rightarrow 0$ (linear over 90% training)
n_{steps}	4096
Batch size	2048
Vectorized envs.	32
Training steps	3×10^8

Training required sufficiently long driving profiles to capture temporal dependencies and avoid artifacts from frequent resets. We therefore used 500 one-hour profiles covering urban, rural, and highway scenarios with diverse gradients and air system demands.

D. Baselines and Evaluation Metrics

To assess the effectiveness of the proposed RL policy, we compare it against two representative baselines:

- **Two-Point Controller**: a conventional strategy that activates the compressor whenever the reservoir pressure falls below a fixed lower threshold and deactivates it once an upper threshold is reached,
- **Free-Air Baseline**: an opportunistic strategy that additionally activates the compressor whenever energy-favorable "free-air" phases (e.g., downhill driving or deceleration) are present.

In practice, state-of-the-art industrial controllers implement more elaborate rule sets that combine these principles with additional logic. However, such implementations are proprietary and not publicly available, which precludes direct benchmarking. The chosen baselines therefore serve as transparent and reproducible proxies that capture the essential control paradigms: threshold-based activation and opportunistic exploitation of free air phases.

Performance is evaluated using the three domain-specific metrics introduced in Section I: free-air utilization, compressor energy demand, and switch frequency.

All methods were additionally monitored for violations of pressure constraints. In practice, survival rates consistently exceeded 99%, confirming both the physical plausibility of the simulation and the regulatory compliance of the learned policies. Since safety margins are typically enforced by hard-coded vehicle-level safeguards, this metric is treated as a secondary indicator rather than a primary comparison criterion.

V. RESULTS

A. Quantitative Results

The RL policy was evaluated against two baseline controllers on 500 held-out profiles generated by the HSMM framework (Section III) that were not used during training. Across all metrics, it achieves a favorable balance between efficiency and durability (Fig. 6). Compared to the two-point controller, the RL agent substantially increases free-air utilization, approaching the dedicated free-air baseline, while lowering overall energy demand relative to it, demonstrating efficiency gains beyond utilization alone. The RL policy operates under the same observation horizon and information constraints as the baselines, so improvements arise from better control decisions rather than privileged foresight. Finally, it maintains a switch frequency only slightly above the two-point controller and far below the free-air baseline, indicating that improved free-air use is not achieved at the cost of excessive wear.

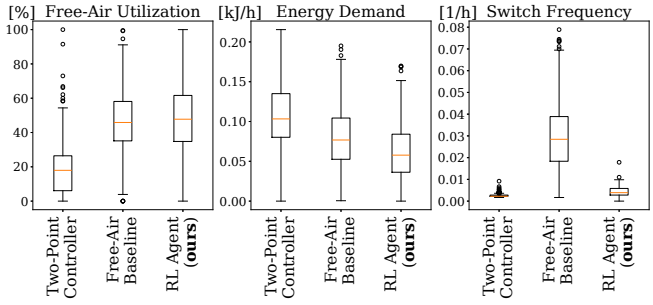


Fig. 6. Performance comparison of the RL Agent against both baselines over 500 synthetic one-hour driving profiles. The RL Agent achieves substantially higher Free-Air Utilization [%] (↑) and lower Energy Demand [kJ/h] (↓) than both baselines, while keeping Switch Frequency [1/h] (↓) close to the Two-Point Controller and far below the Free-Air Baseline.

B. Qualitative Results

A representative profile is shown in Fig. 7. The two-point controller switches sparsely with long dwell times, reacting only at threshold crossings and missing low-cost opportunities. In contrast, the free-air baseline switches very frequently—often in short bursts tied to free-air windows—leading to excessive and sometimes mistimed activations.

The RL agent strikes a middle ground: it clusters activations within free-air windows while keeping switching low. Relative to the two-point controller, it advances some activations to align with favorable conditions; relative to the free-air baseline, it suppresses rapid toggling and avoids late off-window activations. The resulting schedule exploits free-air efficiently while maintaining pressure availability and limiting wear.

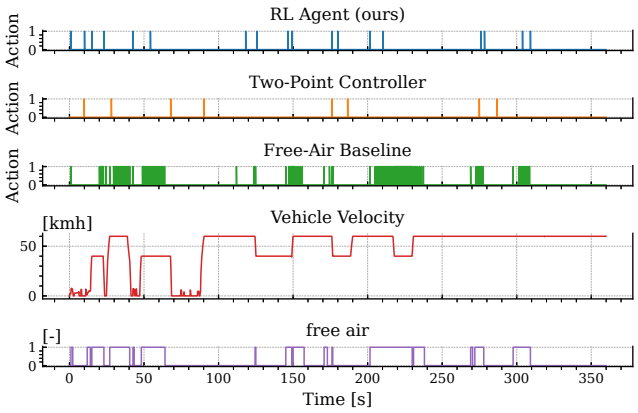


Fig. 7. Qualitative comparison of action sequences for the three policies—RL Agent (ours), Two-Point Controller, Free-Air Baseline—on a representative CAN profile. The bottom panels show the vehicle velocity and a binary *free-air* indicator (periods of excess kinetic energy). Differences in action frequency and placement are clearly observable: our Agent clusters activations within free-air windows while avoiding both the delayed, sparse behavior of the Two-Point Controller and the high-frequency chatter of the Free-Air Baseline.

C. Reward Function Ablation Study

We evaluate the contribution of individual reward components introduced in Section IV by performing an ablation study on the switching penalty and the free-air reward.

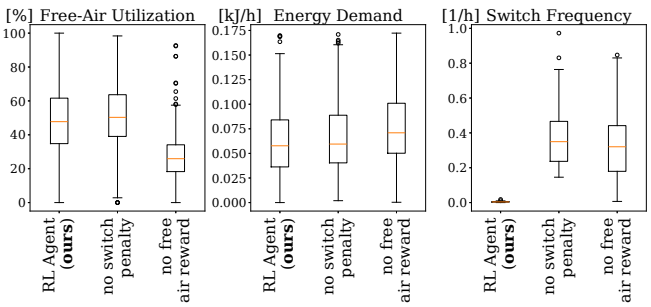


Fig. 8. Reward ablation study. We compare the full reward to (i) removing the switching penalty ($w_3 = 0$) and (ii) removing both the switching penalty and the free-air reward ($w_3 = w_2 = 0$). The free-air reward directly improves energy efficiency, while the switching penalty is essential to keep switching at practically viable levels.

When the switching penalty is removed ($w_3 = 0$), energy efficiency and free-air utilization remain similar, but switching rises to impractically high levels (Fig. 8), corresponding to excessive toggling and accelerated wear in practice.

When the free-air reward is also removed ($w_3 = w_2 = 0$), performance deteriorates across all objectives: switching remains high, free-air utilization drops, and energy demand increases further (Fig. 8). Together, these results confirm that the switching penalty ensures practical operation by limiting wear, while the free-air reward directly drives efficiency.

D. Additional Observations

Not all investigated training strategies yielded reliable outcomes. A curriculum-learning approach based on behavior-cloning pretraining from a conventional two-point controller followed by RL fine-tuning proved highly sensitive to initialization and often destabilized policy updates, limiting its practical utility.

In contrast, direct RL training from scratch, combined with tailored reward shaping and exploration scheduling, consistently produced stable and effective control policies.

E. Discussion and Key Insights

The study demonstrates two central insights. First, the proposed CAN data generator provides a scalable and configurable means of synthesizing realistic driving scenarios. By reproducing braking dynamics and major driving transitions with high fidelity, it enables efficient and reproducible RL training without the need for costly vehicle-level simulations or large volumes of proprietary real-world data.

Second, the learned RL policy demonstrates the feasibility of predictive compressor control. It exploits energy-favorable free-air phases to reduce compressor energy use, while keeping switch frequency at practically viable levels. Crucially, the switching penalty in the reward function was essential to avoid excessive actuator wear, highlighting the role of domain knowledge in shaping effective reward design.

Taken together, these findings indicate that combining synthetic CAN profile generation with RL-based control offers a promising pathway toward context-aware, energy-efficient subsystem optimization in commercial vehicles.

F. Future Work

Future work will extend both the data generation and control aspects. On the data side, the HSMM-based generator can be enriched with additional contextual signals, such as driver behavior models, GPS-based slope profiles, or route planning information. On the control side, validation on real-world measurement data and ultimately in-vehicle experiments will be critical to assess robustness under operational variability. These steps will be essential for transferring the concept from simulation to deployment in commercial vehicle fleets.

REFERENCES

[1] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, pp. 1238–1274, 2013.

[2] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of field robotics*, vol. 37, pp. 362–386, 2020.

[3] J. Brusey, D. Hintea, E. Gaura, and N. Beloe, "Reinforcement learning-based thermal comfort control for vehicle cabins," 2017.

[4] P. S. Chib and P. Singh, "Recent advancements in end-to-end autonomous driving using deep learning: A survey," 2023.

[5] M. Fayyazi, P. Sardar, S. I. Thomas, R. Daghigh, A. Jamali, T. Esch, H. Kemper, R. Langari, and H. Khayyam, "Artificial intelligence/machine learning in energy management systems, control, and optimization of hydrogen fuel cell vehicles," *Sustainability*, vol. 15, p. 5249, 2023.

[6] R. Fambach, "Compressor coupling for torque transfer connection between compressor input and drive member," Patent DE102 008 002 252 A1, 2009.

[7] R. Bremeier, "Electric motor for driving a compressor," Patent EP2 340 607 A1, 2011.

[8] J. Lydia, R. Karpagam, S. L. S. Vimalraj, P. R. Kamala, and I. Sailaja, "Intelligent battery management system for solar photovoltaic systems with iot integration," Patent IN202 441 042 772, 2024.

[9] G. Henri, C. Carrejo, and N. Lu, "Mode-based energy storage management using machine learning," Patent WO2 019 117 957 A1, 2019.

[10] G. Gokhale, B. Claessens, and C. Develder, "Explainable reinforcement learning-based home energy management systems using differentiable decision trees," 2024.

[11] J. Bakakeu, D. Kisskalt, J. Franke, S. Baer, H.-H. Klos, and J. Peschke, "Multi-agent reinforcement learning for the energy optimization of cyber-physical production systems," in *Proc. IEEE CCECE*, 2020, pp. 1–8.

[12] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 8248–8254.

[13] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 740–759, 2020.

[14] Z. Huang, X. Weng, M. Igl, Y. Chen, Y. Cao, B. Ivanovic, M. Pavone, and C. Lv, "Gen-drive: Enhancing diffusion generative driving policies with reward modeling and reinforcement learning fine-tuning," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 3445–3451.

[15] F. Christianos, P. Karkus, B. Ivanovic, S. V. Albrecht, and M. Pavone, "Planning with occluded traffic agents using bi-level variational occlusion models," *arXiv preprint arXiv:2210.14584*, 2022.

[16] L. Wen, J. Duan, S. E. Li, S. Xu, and H. Peng, "Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–7.

[17] E. Ghignone, N. Baumann, and M. Magno, "Tc-driver: A trajectory conditioned reinforcement learning approach to zero-shot autonomous racing," *IEEE Transactions on Field Robotics*, 2024.

[18] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," 2020.

[19] A. Abouelazm, J. Michel, and J. M. Zoellner, "A review of reward functions for reinforcement learning in autonomous driving," 2024.

[20] D. Egan, Q. Zhu, and R. Prucka, "A review of reinforcement learning-based powertrain controllers: Effects of agent selection for mixed-continuity control and reward formulation," *Energies*, vol. 16, p. 3450, 2023.

[21] M. Speckert, K. Dreßler, M. Lübke, *et al.*, "Virtual measurement campaign (vmc): Geo-referenced environmental conditions for vehicle loads and energy efficiency," in *Proc. CVT*, 2018, pp. 88–98.

[22] S. Ebbesen, M. Salazar, S. Delprat, and L. Guzzella, "Optimization-based energy management for hybrid electric vehicles using virtual driving cycles," *IEEE Trans. Veh. Technol.*, vol. 61, pp. 983–994, 2012.

[23] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo: Simulation of urban mobility—an overview," in *Proc. SIMUL*, 2011, pp. 63–68.

[24] W. Wachenfeld, P. Junietz, T. Wenzel, and H. Winner, "Virtual test drive: Provision of a virtual environment for safety validation of automated driving," *ATZelektronik*, vol. 11, pp. 42–47, 2016.

[25] ASAM e.V., "Asam open simulation interface (osi) specification," 2023.

[26] A. Narayanan, S. Mittal, and A. Joshi, "A survey of datasets for automotive intrusion detection systems based on can bus," *IEEE Access*, vol. 9, pp. 113 439–113 457, 2021.

[27] K. Müller, G. Dittmann, J. Machowinski, *et al.*, "Automotive can bus data: A comprehensive open dataset for intrusion detection," in *Proc. IEEE DSN-W*, 2022, pp. 51–58.

[28] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *Ann. Math. Stat.*, vol. 41, pp. 164–171, 1970.

[29] S.-Z. Yu, "Hidden semi-markov models," *Artif. Intell.*, vol. 174, pp. 215–243, 2010.

[30] Farama Foundation, "Gymnasium documentation," 2025.

[31] European Commission, "Un/ece regulation no. 13: Uniform provisions concerning the approval of vehicles (braking)," 2016.

[32] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *J. Mach. Learn. Res.*, vol. 22, pp. 1–8, 2021.