Fuel-efficient switching control for platooning systems with deep reinforcement learning

Tiago Rocha Gonçalves, Rafael Fernandes Cunha, Vineeth Satheeskumar Varma, Salah Eddine Elayoubi

Fuel-Efficient Switching Control for Platooning Systems With Deep Reinforcement Learning

Tiago R. Gonçalves[1], Rafael F. Cunha[2], Vineeth S. Varma[3], and Salah E. Elayoubi[1]

***Abstract*—The wide appeal of fuel-efficient transport solutions is constantly increasing due to the major impact of the transportation industry on the environment. Platooning systems represent a relatively simple approach in terms of deployment toward fuel-efficient solutions. This paper addresses the reduction of the fuel consumption attainable by dynamically switching between two control policies: Adaptive Cruise Control (ACC) and Cooperative Adaptive Cruise Control (CACC), in platooning systems. The switching rule is dictated by Deep Reinforcement Learning (DRL) technique to overcome unpredictable platoon disturbances and to learn appropriate transient shift times while maximizing fuel efficiency. However, due to safety and convergence issues of DRL, our algorithm establishes transition times and minimum periods of operation of ACC and CACC controllers instead of directly controlling vehicles. Numerical experiments show that the DRL agent outperforms both static ACC and CACC versions and the threshold logic control in terms of fuel efficiency while also being robust to perturbations and satisfying safety requirements.**

***Index Terms*—Vehicle platoons, deep reinforcement learning, cooperative adaptive cruise control (CACC).**

I. Introduction

The efficient operation of platooning systems is meaningful due to their substantial economic and environmental impact. This paper's focus is on the suitability of switching classical controllers in order to improve fuel efficiency in platooning systems. In this context, a switching control architecture can be viewed as a dynamical control constituted by a family of controllers and a rule that coordinates the switching among them. In other words, a logical strategy that decides the activation of a specific controller at each instant of time. In this work, we adopt a Deep Reinforcement Learning (DRL) algorithm to dictate the switching rule. We identify that based on the disturbances caused by the vehicle that precedes the platoon, namely the jammer, a specific controller might be more appropriate than others, in terms of fuel efficiency. In particular, we evaluate such disturbances in a platoon system under two well-known controllers: Adaptive Cruise Control (ACC) and Cooperative Adaptive Cruise Control (CACC). The former is pertinent due to the relatively low complexity of the controller, which does not rely on the communication system, and, therefore, might boost the deployment of platooning systems in the near future. Moreover, it is generally adopted as a backup strategy in case of losing the communication system link [1], [2]. Whereas the second controller allows shorter inter-vehicle distances, which translates to substantial improvements in fuel performance due to the air-drag reduction. However, the control effort for each alternative plays an important role in the fuel efficiency [3], [4], and must be carefully evaluated. As an additional remark, we would like to point out that the switch between both controllers is motivated by possible problematic scenarios, for instance, when a long burst of losses in the communication network is observed or by the requirement of extra inter-vehicle distances imposed by merging and splitting maneuvers, and when aggressive jammer behavior due to poor road traffic conditions is detected for some period. Note that object detection and behavior prediction in the road environment are the input and precondition for DRL control which recent works have covered [5], [6]. Recently, there has been an increase in the number of DRL algorithms that surpass human performance across various fields. For a comprehensive overview, please refer to François-Lavet et al. [7].

The main contribution of this paper is to demonstrate the feasibility of a DRL approach to dictate the switching control rule in order to improve the fuel efficiency of platooning systems. Firstly, we identify the burden caused by abrupt switching controllers under deterministic disturbances, and thus, we propose an enhanced controller to mitigate such transition losses. Secondly, we model such disturbances as a random process and reformulate the vehicle platoon fuel efficiency problem in a DRL framework. To the best of our knowledge, the present study is the first to propose a DRL approach to command the switching process in order to increase fuel efficiency in a platoon while accounting for stochastic traffic conditions.

II. Related work

In the literature, many works have addressed external forces such as aerodynamic drag, rolling resistance, and gravitation forces, which indeed are imperative to investigate the fuel efficiency problem of platooning systems [4], [8], [9]. Turri *et al.* [4] exploits the road topography information to predict the behavior of vehicles to improve fuel efficiency in the platoon. Unlike this work, the authors in [4] assume that external disturbances, such as traffic ahead, are handled manually by the drivers. Alam *et al.* [8] conducted an experimental study on the fuel reduction potential for platooning systems under CC and ACC control with different time-gap parameters. Unlike

[1]T. R. Gonçalves, and S. E. Elayoubi are with Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France. E-mail: {tiago.rochagoncalves; salaheddine.elayoubi}@centralesupelec.fr.

[2]R. F. Cunha is with University of Groningen, Netherlands. E-mail: r.f.cunha@rug.nl.

[3]V. S. Varma is with Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France. E-mail: vineeth.satheeskumar-varma@univ-lorraine.fr.

this work, the authors in [8] adopt fixed control parameters over the corresponding scenario, while we address the fuel efficiency problem by constantly adapting the controllers based on the platoon's disturbance behavior. Liang *et al.* [9] proposed a parameter, called the platooning incentive factor, which is responsible for indicating whenever is beneficial to form a platoon. However, their approach is only valid under no traffic conditions, which has limited practical purposes.

Another meaningful contribution of our study is related to the DRL approach, adopted to learn from trial and error the most suitable action, in order to increase the fuel efficiency of the platoon when under stochastic disturbances. In this context, researchers try to find solutions using such machine learning techniques [10]–[12]. In a platoon framework, Ng *et al.* [10] proposed a gain schedule control to be learned by an RL technique. The authors show that when under platooning, their approach performs better than a simple linearization of the longitudinal model. The first attempt to use RL for controlling CACC was made by Desjardins and Chaib-draa [11]. Different from this work, the authors in [11] adopt a policy-based algorithm, which allows them to handle continuous-state variables and directly attempt to achieve longitudinal control, as the output of the neural network. However, they faced oscillatory behavior of the RL approach, which is avoided in our work, as we do not attempt to control the platoon directly. Ling *et al.* [12] considered a platoon in which the future velocity profile of the preceding vehicle is predicted by Artificial Neural Networks (ANNs) techniques that use a topographic map of the road as input. Such a velocity prediction system is used together with a Model Predictive Control (MPC) that controls the platoon. Similarly, [13] addresses the energy-consumption problem via an eco-driving control architecture based on an MPC strategy. The authors in [14] also adopt an MPC approach for fuel-saving but focus on the signalized intersection problem where platoons are capable to split and merge. However, the aforementioned works assume perfect communication between all vehicles in the platoon, while we only consider communication between consecutive vehicles.

Lately, Deep Neural Networks (DNNs) have been successfully applied to improve the learning ability of RL techniques, which has led to the development of the DRL framework. In this context, Chu and Kalabić [15] proposed a model-based DRL approach that learns the best headway signals for CACC in a platoon. They simply investigate a catch-up maneuver to the leader vehicle, which does not justify the DRL framework. Whereas in this work, our platooning system is under uncertain and severe traffic conditions modeled by stochastic disturbances that pose an enormous challenge to maintain the platoon within the system's constraints, which clearly motivates the adopted framework. Chen *et al.* [16] focuses on a path planning point of view that attempts to determine the best path strategy for the platoon through the employment of DRL techniques. The authors make a restrictive assumption by selecting a mild road selected area for the path alternatives that the vehicles are able to choose.

Therefore, in the literature, there is a lack of work that investigates the feasibility of platooning under non-ideal traffic conditions. Thus, the impact of time-varying traffic conditions on the fuel efficiency of the platoon, and a DRL approach that improves the performance by properly guiding switching classical controls is our main contribution.

The notation used throughout is standard. For real vectors or matrices, $(')$ refers to their transpose. The symbols $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}_+$, $\mathbb{N}$, $\mathbb{K}$, Γ, denote the sets of real, real non-negative, integer non-negative, natural numbers, $\mathbb{K} = \{1, 2, \cdots, N\}$ for a natural integer N, $\Gamma = \{0, 1, 2, \cdots, r\}$, where r is a fixed positive integer, respectively. Finally, we denote $\otimes$ the Kronecker product. For the sake of compactness, whenever possible, the time dependence of vector or matrix variables is omitted.

III. System Model and Problem Statement

The objective of this section is to present the platoon model, the fuel consumption model, the control schemes adopted, and finally, the problem statement.

A. Platoon modeling with external forces

We aim to introduce external forces, particularly air-drag resistance, which is one of the main parameters that alter the fuel consumption of the platoon. We adopt the constant spacing policy to exploit the gains of platoon formation, where the inter-vehicle spacing of the ith vehicle in the discrete-time is given by

$$e_i(k) = p_i(k) - p_{i-1}(k) + l_{i-1} \tag{1}$$

and its difference as

$$\epsilon_i(k) = v_i(k) - v_{i-1}(k) \tag{2}$$

where $k \in \mathbb{Z}_+$, i denotes the vehicle index and $i \in \{0, 1, \ldots, N-1\}$, the leader vehicle being 0. l_{i-1} is the length of the vehicle $i-1$, and $p_i(t)$ and $v_i(t)$ are the front position and velocity of the vehicle i, respectively. Note that we adopted a coordinate system where $p_{i-1} > p_i$. We consider a longitudinal vehicle model with additional external forces as follows

$$a_i(k) = \frac{F_{eng_i}(k)}{m_i} - \frac{c_{D_i}\psi_i(e_i)A_{f_i}\rho_{air}}{2m_i}v_i(k)^2 - g(c_{r_i}\cos\theta(k) + \sin\theta(k)) \tag{3}$$

where the engine force is denoted F_{eng_i}, the second term is the air-drag force, the third and last terms are the roll resistance and gravitational force, respectively. Furthermore, concerning the vehicle i, m_i designates the vehicle mass, v_i the vehicle speed, c_{D_i} is the air-drag coefficient and $\psi_i(e_i) \in [0, 1]$ is air-drag ratio which depends on the inter-vehicle spacing e_i as in (1) when platooning, c_{r_i} the roll resistance coefficient, A_{f_i} is the front area of the vehicle, ρ_{air} is the air density, $\theta(k)$ denotes the road slope, and g the gravitational constant. Note that the function $\psi_i(e_i)$ is responsible for taking into account the influence of the inter-vehicular distance on the aerodynamic force that plays an essential role in platooning. We have adopted the air-drag ratio model by [17]. Note that (3)

as it is, presents very complex dynamics, so in order to cope with the non-linearity and simplify its dynamics, we adopt the following control law:

$$F_{eng_i}(k) = u_i(k)m_i + \frac{c_{D_i}\psi_i(e_i)A_{f_i}\rho_{air}}{2}v_i(k)^2 + gm_i(c_{r_i}\cos\theta(k) + \sin\theta(k)) \quad (4)$$

where $u_i(k)$ is the new input signal to be designed. Note that we assume perfect knowledge about the system's parameters, but the interest in such a feedback linearization controller is to linearize the vehicle dynamics and to eliminate nonlinear terms. After linearization, we adopt a reasonable model for the vehicle dynamics widely used in the literature in the discrete-time form [18]–[21]:

$$\begin{bmatrix} p_i(k+1) \\ v_i(k+1) \\ a_i(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & T_s \\ 0 & 0 & 1\text{-}\frac{T_s}{\tau_i} \end{bmatrix}}_{\tilde{A}} \begin{bmatrix} p_i(k) \\ v_i(k) \\ a_i(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \frac{T_s}{\tau_i} \end{bmatrix}}_{\tilde{B}} u_i(k) \quad (5)$$

where $\{p_i, v_i, a_i\}$ are the position, velocity and acceleration of the vehicle i, respectively. The subscript i is the vehicle platoon member index where $i \in \{0, i, \cdots, N-1\}$ and 0 the platoon leader's index. u_i is the control input of vehicle i after linearization, i.e., its desired acceleration. τ_i is the time constant of the first-order low pass filter for each vehicle i, and T_s is the sample time. The idea is to approximate the dynamics of the throttle body and vehicle inertia in order to avoid instantaneous response. Furthermore, control input constraints are applied to avoid unpractical acceleration signals as

$$u_{min} \leq u_i(k) \leq u_{max} \quad (6)$$

where u_{min} and u_{max} are the minimum, and maximum acceleration signals admitted that compass the control signal. In this paper, we assume an actuator lag of $\tau_i = 0.2$ s $\forall i \in \{0, i, \cdots, N-1\}$ as in [22], and we adopt a sample time of $T_s = 100$ ms and a zero-order hold for the control input. So, the general notation of the open-loop model of the system in the discrete-time can be written as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Dw(k) \end{aligned} \quad (7)$$

where $x(k) \coloneqq [p_0\ v_0\ a_0\ e_1\ \epsilon_1\ a_1 \cdots\ e_{N-1}\ \epsilon_{N-1}\ a_{N-1}]'$, indicates the state space vector of the system, $u(k) \coloneqq [u_0\ \cdots\ u_{N-1}]'$, is the vector of all control inputs. The output vector available for feedback is defined as $y(k) \coloneqq [e_0\ \epsilon_0\ v_0\ 0\ e_1\ \epsilon_1\ v_1\ a_1\ \cdots\ e_{N-1}\ \epsilon_{N-1}\ v_{N-1}\ a_{N-1}]'$, and $w(k) \coloneqq [p_j\ v_j]'$ is the exogenous input, i.e. the jammer's rear position and velocity. Define $R = (r_{nm}) \in \mathcal{R}^{3N\times 3N}$, where r_{nm} = -T_s for $n = 3i+5$ and $m = 3i+3$, $i = \{0, \cdots, N-1\}$ and 0, otherwise. Thus, $A = I_N \otimes \tilde{A} + R$, $B = I_N \otimes \tilde{B}$, where $\tilde{A}$ and $\tilde{B}$ are defined in (5). Finally, note that

$$D = \begin{bmatrix} -I_{2\times 2} \\ 0_{3N-2\times 2} \end{bmatrix}$$

whereas C can be easily identified since the state space $x(k)$ and the output $y(k)$ are defined. Next, we aim to introduce the fuel consumption model, which will be used to estimate the efficiency of the proposed techniques against classical approaches.

B. Fuel consumption model

In this section, we derive a simple model that captures the intrinsic relation between consumed fuel and generated longitudinal force. In other words, we are mainly interested in understanding how the system dynamics and external forces affect the fuel consumption of each vehicle when platooning under different inter-vehicle distances. In order to find such influence, we start by modeling the energy loss (W) of the system model in discrete-time over time T_f as Oguchi *et al.* [23]:

$$W(k) = \sum_{k=0}^{T_f} \zeta(k) \cdot F_{eng_i}(k) \cdot v_i(k) \cdot T_s \quad (8)$$

with

$$\zeta(k) = \begin{cases} 1 \textit{ if } F_{eng_i}(k) > 0 \\ 0 \textit{ otherwise} \end{cases} \quad (9)$$

where $F_{eng_i}(k) > 0$ indicates that propellant is used to power the vehicle i as in (4), thus, resulting in losses to be computed. In order to represent such energy losses in terms of fuel consumed, we adopted the following converting method defined by the following cost

$$J(u) = \frac{1}{\rho_{prop} \cdot \eta_{eng}} W \quad (10)$$

where ρ_{prop} and η_{eng} are the energy density of the propellant in $[J/L]$ and the constant efficiency of the engine, respectively. Note that we consider those parameters as $\rho_{pro} = 34.9MJ/L$ and $\eta_{eng} = 30\%$ which correspond to average gasoline density energy and efficiency [24]. From (10), we can see that fuel consumption depends mainly on the relative distance between vehicles in the platoon, the speed of the vehicle, and the control effort. However, there is a consensus about the speed adjustment that even though the air-drag can be significantly reduced when the velocity v is decreased, this is generally not economically viable due to tight delivery schedules. Therefore, when seeking fuel consumption efficiency, we must optimize the inter-vehicle distance in the platoon to improve the air-drag ratio function ($\psi_i(e_i)$) and the control effort $u_i(k)$. An illustration of the achievable air-drag reduction in terms of the inter-vehicle distance between vehicles when platooning is given in Fig. 1.

C. Classical control schemes for platooning

1) Adaptive Cruise Control: We make use of the ACC controller due to its relevance for the deployment application of platooning systems in a decentralized design that does not require any type of communication. Additionally, such a controller is string stable with a constant time-gap, which translates to robustness and safety under perturbations. Therefore, consider the following output feedback control law

$$u(k) = -K_0 y(k) \quad (11)$$

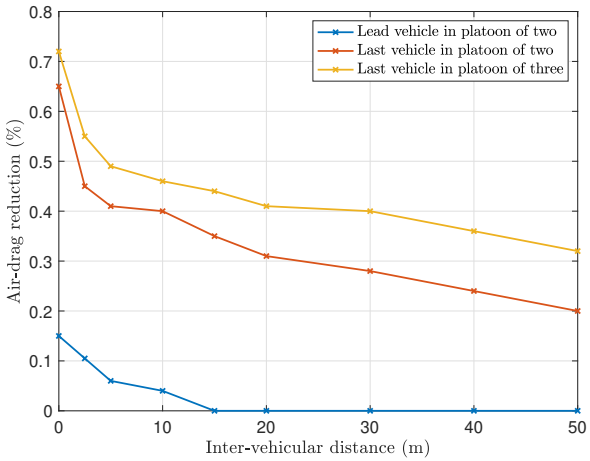

Fig. 1: Air-drag reduction for trucks in a platoon at $80km/h$ empirically obtained. The figure is adapted from [17].

where $K_0 \in \mathbb{R}^{N\times 4N}$ is the controller ACC gain defined by

$$K_0 = \begin{bmatrix} \chi_i & 0 & \cdots & 0 \\ 0 & \chi_{i+1} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \chi_{N-1} \end{bmatrix} \tag{12}$$

where

$$\chi_i = \begin{bmatrix} \frac{\lambda_i}{h_i} & \frac{1}{h_i} & \lambda_i & 0 \end{bmatrix}, \quad i = \{0, 1, \cdots, N-1\} \tag{13}$$

are the ACC controller gains with a constant time-gap spacing policy proposed by [25], and h and λ are the time-gap and design gain parameters, respectively. Note that this generic notation allows us to consider a centralized ($\lambda_i = \lambda \ \wedge \ h_i = h, \quad \forall i$) or a decentralized ($\lambda_i \neq \lambda \ \wedge \ h_i \neq h, \quad \forall i$) ACC controller.

2) Cooperative Adaptive Cruise Control: Besides the ACC controller, we adopt the CACC controller to exploit the communication features of forwarding the acceleration signal. Therefore, this controller allows a constant spacing policy that aims to keep a certain desired distance (d_{des}) between successive vehicles. However, in this paper, we adopt the weight of the leader parameter as zero, $c = 0$, which corresponds to the semi-autonomous control. The motivation behind this is to eliminate the impact of leader packet delay in the platoon. Thus, the communication analysis is substantially simplified since we only assume perfect communication for vehicle-to-neighbor links. This is reasonable as a result of the very low probability of packet error for consecutive links thanks to line-of-sight propagation [26].

Therefore, consider the following output feedback control

$$u(k) = -K_1 y(k) \tag{14}$$

where $K_1 \in \mathbb{R}^{N\times 4N}$ is the controller gain defined by

$$K_1 = \begin{bmatrix} \chi_0 & 0 & \cdots & 0 \\ 0 & \varphi_i & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \varphi_{n_u} \end{bmatrix} \tag{15}$$

where the first term is the ACC controller previously implemented in the leader vehicle to be in conformity with spacing policies imposed by public entities. The next term is then,

$$\varphi_i = \begin{bmatrix} k_p & k_d & k_c & 1 - c_i \end{bmatrix}, \ i = \{1, \cdots, n_u - 1\} \tag{16}$$

which are the CACC controller gains defined by

$$k_c = (\xi + \sqrt{\xi^2 - 1})\omega_n c_i \tag{17}$$

$$k_d = (2\xi - c_i(\xi + \sqrt{\xi^2 - 1}))\omega_n \tag{18}$$

$$k_p = \omega_n^2 \tag{19}$$

which are the different control gains that depend on the following parameters c_i, ξ, and ω_n, which correspond to the weight of the leader information, the controller damping ratio, and bandwidth, respectively.

D. Enhanced proposed controller

We propose to design a switching controller that alternates between CACC and ACC based on the jammer behavior. Unfortunately, an abrupt switching produces undesired transient responses in the control signal, which translates to misuse of fuel. In order to cope with that, we propose an enhanced controller responsible for mitigating such misuse. So, having presented the classical control schemes, we introduce the proposed enhanced switching control scheme, and we perform a case comparison over previous control strategies in terms of fuel efficiency. The particular reason is that in such a transient stage, the new controller is taking place, which is significantly different from the previous one (ACC to CACC and vice-versa). In other words, the initial conditions are far away from the steady-state conditions, which causes very sharp transient responses leading to non-optimal switching logic in terms of fuel efficiency. In order to smooth such unsuitable transient responses, and to improve fuel efficiency, we propose the following enhanced control

$$u(k) = -(\beta(k)K_1 + (1 - \beta(k))K_0)y(k) \tag{20}$$

where $\beta(k) \in \{0, 1/\delta, 2/\delta, \cdots, 1\}$ is a dynamic coefficient, which will be detailed in the following, and δ is a control design parameter that corresponds to the minimum subinterval considered for a control action, in the order of seconds. Note that $\beta(k)$ is responsible to weight the influence of each state-feedback gain for the ACC and CACC controller, given by K_0 and K_1, respectively. In other words, it corresponds to the parameter used to smooth the switching transition control. Another important parameter is the set of transitions times defined by

$$\mathcal{K} = \{k_1, k_2, \cdots, k_W\} \tag{21}$$

where the following holds

$$k_i - k_{i+1} \geq \delta \quad \forall i \in \{1, \cdots, W-1\} \tag{22}$$

$$0 < k_1 < k_2 < \cdots < k_W < T \tag{23}$$

where T is the maximum simulation time adopted. Furthermore, the dynamics of the smooth switch parameter follows:

$$\beta(k+1) = \beta(k) + \varrho(k) \cdot (-1)^{\beta(k_i)} \cdot \frac{1}{\delta} \tag{24}$$

where $k_i \in \mathcal{K}$ such that $k_i \leq k < k_i + \delta$, and

$$\varrho(k) = \begin{cases} 1 \textit{ if } k \in [k_i, k_i + \delta] \\ 0 \textit{ otherwise} \end{cases} \tag{25}$$

$\forall k_i \in \mathcal{K}$, where we initialize with $\beta(0) = 0$, which corresponds to ACC controller. We expect substantial improvements with the enhanced proposed controller. The intuition behind such gains is related to the incorporation of the control parameter $\beta(k)$ in the final control law, which leads to smoother switching of the combined control actions with respect to time. Whereas without such a parameter, the non-enhanced control abruptly changes the control law in just one time step, which causes a much larger transient response that translates to a waste of fuel.

E. Objective

Once the system dynamics, the fuel consumption model, and the controllers are defined, we are able to state the main objectives of this paper. Considering the system (7), we want to minimize the fuel consumption cost (10) for an established set of control as in (20) and its transition times (21) chosen by the policy learned by the learning algorithm subject to

$$\begin{aligned} &p_{i-1} - p_i - l_{i-1} \geq d_{min} \\ &p_{i-1} - p_i - l_{i-1} \leq d_{max} \\ &v_{min} \leq v_i \leq v_{max} \\ &u_{min} \leq u_i \leq u_{max}. \end{aligned} \tag{26}$$

Therefore, the objective function can be written as

$$\min_{\mathcal{K} \text{ as } (21)-(23)} \{J(\mathcal{K}) : \text{constrained by } (7), (20), (24), (26)\} \tag{27}$$

Additionally, note that $J(\mathcal{K})$ is given by (10) where $u(k)$ is of the type given in (20), that depends of k and β. The latter is related to $\mathcal{K}$ via its k_i dependence, as in (24), which fully describes the controller under consideration in this work. While under no jammer disturbance, CACC is better than ACC in terms of fuel efficiency as it accounts for the air-drag reduction, $0 \leq \psi_i(e_i) < 1$, the same is not true for random/stochastic jammers or disturbances. Indeed, we have noticed that ACC expends less fuel due to its moderate behavior when compared to CACC stringent constant distance gap policy. These two features motivate the use of DRL techniques to find fuel-efficient control policies to cope with the uncertainty of the jammer profile.

IV. Optimal policy for constant jammers

In this section, we introduce an optimal switching control policy, in terms of fuel efficiency, for a platoon environment with constant jammers. We aim to isolate the contribution to fuel consumption by two main factors: air-drag and control effort. The air-drag force F_{air} is modeled as in (3), and its reduction amount fluctuates based on the inter-vehicle spacing between vehicles, as presented Fig. 1. The aim is to investigate the specifics of both ACC and CACC control in terms of fuel efficiency. It is clear that both controllers are very distinct and require different information in order to adjust their parameters accordingly. Therefore, in the light of the idea of Liang *et al.* [9] that focuses on a method where a platoon member drives faster and catches up with the lead vehicle, we present a theorem that confirms the burden caused by the switching control policies under constant jammer profile.

Theorem 1. *Consider the fuel consumption given by (10), the engine force as in (4), and the controllers ACC and CACC by (11) and (14), respectively. Assume that both controllers are tracking the same speed. Define the transition times between controllers as the set $\mathcal{K} = \{k_1, k_2, \cdots, k_W\}$ subject to (22)-(23). Let J_{switch} be the fuel consumed for at least one transition time from one controller to the other, i.e. $\mathcal{K} \neq \emptyset$, and J_{hold} the fuel consumed when the platoon keeps the CACC control the whole time, i.e. no possible transition $\mathcal{K} = \emptyset$. Then, the following holds true for a constant jammer profile when considering the same travel distance.*

$$J_{switch} > J_{hold} \tag{28}$$

Proof. In order to evaluate the impact of switching control in terms of fuel burden, we must keep all the other factors constant. Thus, all vehicle parameters are set equal, and a constant jammer profile is considered, i.e. no disturbance. Note that the following holds

$$\bar{v}_\beta > v_{acc} = v_{cacc} \tag{29}$$

$$1 \geq \psi(e_{acc}) > \psi(e_\beta) > \psi(e_{cacc}) > 0 \tag{30}$$

$$v_{acc} T_{acc} + \sum_{t_{acc}}^{t_{acc}+t_\beta} v_\beta T_s + v_{cacc} T_{cacc} = v_{cacc} T_f \tag{31}$$

$$v_{acc}, v_{cacc}, \psi(e_{acc}), \psi(e_{cacc}) \quad \text{constants} \tag{32}$$

where the first inequality means that the mean of the transient velocity is greater than ACC and CACC velocity, this is a consequence of the fact that both controllers track the same speed. The second inequality ensures that the possible reduction air-drag is monotonic due to inter-vehicle spacing fluctuation. The third equation guarantees that the travel distance is the same. The fourth represents that, in the steady state, both controllers keep a constant speed and distance. To simplify the notation, consider $v_{acc} = v_a$, $v_{cacc} = v_c$, $\psi(e_{acc}) = \psi_{e_a}$, $\psi(e_{cacc}) = \psi_{e_c}$, $\psi(e_\beta) = \psi_{e_\beta}$, and the subscripts acc = a, and cacc = c.

Describe the total fuel burden components due to switching logic control components as the following phases:

$$J_{switch} = J_{ACC} + J_{trans} + J_{CACC} \tag{33}$$

where the three terms stand for the total fuel spent by the ACC, transient and CACC control, respectively. Consider that $switch$ performs a single transition $\mathcal{K} = \{k_1\}$, here namely $k_1 = t_{acc}$, which is the time spent over the first controller that corresponds to ACC controller as previously defined with $\beta(0) = 0$. Thus, inserting (10) in (33) and considering $\phi(\cdot)$ a transformation function that accomplishes for the remaining terms in (10), leads to:

$$\begin{aligned}
&\phi(J_{ACC} + J_{trans} + J_{CACC}) = \\
&= \sum_{k=0}^{t_a} v_a^3 \psi_{e_a} T_s + \sum_{k=t_a}^{t_a+t_\beta} v_\beta^3 \psi_{e_\beta} T_s + \sum_{k=t_a+t_\beta}^{t_a+t_\beta+t_c} v_c^3 \psi_{e_c} T_s \\
&> v_a^3 \psi_{e_c} T_a + \psi_{e_c} \sum_{k=t_a}^{t_a+t_\beta} v_\beta^3 T_s + v_c^3 \psi_{e_c} T_c && (34) \\
&> v_c^2 v_a \psi_{e_c} T_a + \psi_{e_c} v_c^2 \sum_{k=t_a}^{t_a+t_\beta} v_\beta T_s + v_c^3 \psi_{e_c} T_c && (35) \\
&= v_c^2 \psi_{e_c} \big(v_a T_a + \sum_{k=t_a}^{t_a+t_\beta} v_\beta T_s + v_c T_c \big) && (36) \\
&= v_c^2 \psi_{e_c} (v_c T_f) && (37) \\
&= \phi(J_{hold}) && (38)
\end{aligned}$$

Where inequality (34) holds by (30), and we also used the fact that v_{acc}, v_{cacc}, $\psi(e_{acc})$, $\psi(e_{cacc})$ are constants, according to (32). Inequality (35) holds by (29). Equality (37) holds by (31), that is, we are comparing fuel consumption over the same distances, and (38) is obtained by noting that (37) is the definition of $\phi(J_{hold})$. As a consequence, the result provides the exact inequality as (28), which thus concludes the proof. □

Therefore, under a constant jammer disturbance, no switching logic is required as it will add a costly transient term that will never be beneficial due to steady conditions after switching. Moreover, the optimal choice is to keep the CACC control which benefits from the air-drag reduction (see *Remark 1*).

Remark 1. *The superiority of CACC over ACC in terms of fuel efficiency is straightforward under constant speed, i.e. no jammer. In order to verify it, the following must hold:*

$$J_{CACC} < J_{ACC} \tag{39}$$

Then, by inserting (10) in (39) and considering (30) *due to larger distances of the constant time-gap spacing policy of ACC controller when compared to CACC, negligible (des)acceleration phases, and a flat road ($\theta = 0$), it yields to*

$$\psi(e_{cacc}) v_{cacc}^3 T_{cacc} < \psi(e_{acc}) v_{acc}^3 T_{acc} \tag{40}$$

which holds when (30)-(32) are true under no switching assumptions, i.e. $J_{transient} = 0$.

V. Stochastic disturbances

So far, we have considered deterministic jammer profiles, which allow us to conveniently address the fuel efficiency platooning problem. However, in practical terms, such an analysis is very limited since the jammer vehicle presents stochastic characteristics due to its unpredictable behavior. In this work, we model only one external vehicle on the highway, which is the jammer vehicle, and the platoon is assumed to travel behind it. Therefore, first of all, we aim to present how we model the uncertainty of the jammer's dynamics. In the sequel, we introduce DRL techniques and a threshold switching rule as solutions for the problem.

A. Jammer profile modeled with Markov chains

In particular, we have adopted a discrete-time Markov process to model the jammer velocity profile. In order to be conservative, we have adopted essentially two different modes. The first one is a constant profile which indicates that the jammer is driving mainly at a constant speed. The second is an aggressive velocity profile commonly used in the literature to evaluate the robustness of platoon systems as in [26], [27]. In particular, we have adjusted the jammer acceleration bounds to $-2 \leq a_j(k) \leq +2$ in order to produce a zero average. More formally, we have the jammer's dynamics given by:

$$w_{\sigma(k)}(k+1) = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_j(k) \\ v_j(k) \end{bmatrix} + \begin{bmatrix} 0 \\ T_s \end{bmatrix} a_j^{\sigma(k)}(k) \tag{41}$$

where v_j is the velocity of the jammer for a certain discretization time T_s, and $a_j^{\sigma(k)}$ is the acceleration of the jammer profile dictated by $\sigma(k)$ that is a random variable governed by a discrete-time Markov process. Therefore, we are able to model the jammer dynamics by adjusting its acceleration with the $\sigma(k)$ parameter as introduced next.

Assumption 1 (Markov switching signal for the jammer). *We adopt a discrete-time Markov chain to model the possible modes of the jammer profile. The random switching process $\{\sigma_k\}$ is said to be a finite and homogeneous Markov chain if for $\forall k \in \Gamma$,*

$$\begin{aligned}
&P(\sigma_{k+1} = j | \sigma_k = i, \sigma_{k-1} = i_{k-1}, \cdots, \sigma_0 = i_o) \\
&\qquad = P(\sigma_{k+1} = j | \sigma_k = i) = p_{ij}. \qquad (42)
\end{aligned}$$

where p_{ij} is the transition probability that does not depend on time k. Note that the rows of any state transition matrix must sum up to one, more formally

$$\sum_{k=1}^{r} p_{ik} = \sum_{k=1}^{r} P(\sigma_{k+1} = k | \sigma_k = i) = 1. \tag{43}$$

Based on the Champman Kolmogorov equations, one might define the k-step transition probabilities for a homogeneous Markov chain based on the initial probability distribution $\pi_0 = [\pi_1, \pi_2, \ldots, \pi_r]$ that yields to

$$\pi_k = \pi_0 P^k. \tag{44}$$

If the chain is irreducible and aperiodic, then the set of equations (43)-(44) has a unique solution known as the limiting distribution of the Markov chain, i.e.,

$$\pi_j = \lim_{k\to\infty} P(\sigma_k = j|\sigma_0 = i) \tag{45}$$

for all $i, j \in \Gamma$. *See [28].*

Therefore, we consider the case where $\sigma(k) \in \Gamma = \{0, 1\}$, which here denote steady and aggressive modes, respectively. Both are characterized by the following

$$a_j^{\sigma(k)}(k) = \begin{cases} \vartheta * \mathcal{U}[-\varsigma, \varsigma] & \text{if } \sigma(k) = 0 \\ h(k) & \text{if } \sigma(k) = 1 \end{cases} \tag{46}$$

where $\mathcal{U}[-\varsigma, \varsigma]$ follows an uniform distribution from $-\varsigma$ to $+\varsigma$ where $\varsigma \in \mathbb{R}_+$, and ϑ is a scalar variable responsible to adjust the acceleration's amplitude of the steady mode, and $h(k)$ corresponds the following function

$$h(k) = \begin{cases} -\varsigma & \text{if } k \leq \delta/2 \\ +\varsigma & \text{otherwise} \end{cases} \tag{47}$$

where δ is the subinterval considered between transitions times defined in (22). Note that a triangular shape function is obtained as the jammer's speed. Finally, we can rewrite the output of the system (7) as a function of the stochastic disturbance

$$y(k) = Cx(k) + Dw_{\sigma(k)}(k) \tag{48}$$

where $\sigma(k) = \{0, 1\}$ is the random variable governed by a discrete-time Markov process.

B. Troublesome conditions

We have successfully treated the jammer behavior as a random process by adopting a discrete-time Markov chain. However, despite that, we are interested in modeling unlikely scenarios such as the case when suddenly the jammer behavior changes to the opposite mode only for some short fixed interval. In this framework, we have considered for each subinterval time δ a probability of 5% and 10% for such a condition to happen, namely troublesome condition. Therefore, based on the actual state space $\Gamma = \{0, 1\}$ selected, the jammer profile shifts to the opposite state with the aforementioned probability. The idea behind this additional obstacle is to stress even more the system in order to achieve robust outcomes. More precisely, our goal is to avoid a very clear distinction between profiles, for which a simpler threshold switching logic solution would be enough, and to model the real-life behavior of drivers who are not always rational/predictable. Furthermore, we want to evaluate the burden of changing controllers to address such troublesome conditions.

VI. Proposed switch controllers

In this section, we aim to propose two switching controller approaches to handle the stochastic disturbances previously introduced.

A. Threshold logic for switching platoon controller

First, we adopt a simpler approach that requires little computational burden. We aim to mitigate large oscillation amplitudes that might occur during switching controllers. Our goal is to provide a threshold logic that triggers a particular control strategy. Therefore, we can design a threshold rule based on the parameters of the state space of the system $x_i(k)$ responsible for specifying the controller set $(\mathbf{u})$ that is a combination of ACC and CACC as (20). In order words, such a controller generates a set of transition times $\mathcal{K} = \{k_1, k_2, \cdots, k_W\}$ based on the jammer behavior. In order to smooth out short-term fluctuations and highlight longer-term tendencies, we apply a moving average where the mean is calculated over a sliding window of length sw across neighboring elements of the state space parameter $x_i(k)$. More formally, the set of transitions time $\mathcal{K}$ uses a threshold logic and is defined as

$$\begin{aligned} \mathcal{K} = \big\{k \in \mathbb{Z}_+ | (\beta(k) = 0 \text{ and } \bar{a}_0 > \varepsilon_{th}) \text{ or} \\ (\beta(k) = 1 \text{ and } \bar{a}_0 \leq \varepsilon_{th})\big\} \end{aligned} \tag{49}$$

where $\bar{a}_0 = \sqrt{\frac{1}{sw}\sum_{k-sw}^{k} a_0(k)^2}$ is the moving average of the acceleration of the leader over a sliding window of length sw. The variable ε_{th} is the threshold value, which will be determined in the next section.

B. Deep reinforcement learning for switching platoon controller

Although simple to implement, the previous controller has its limitation since the threshold parameter ε_{th} is adjusted empirically based on observations. Thus, when the traffic conditions are time-varying and unpredictable (as is often the case in practice), tuning this parameter offline becomes infeasible. Another option is to adopt an online model-free learning approach since we are dealing with stochastic disturbances. In this work, we adopt a DRL framework to determine the most appropriate action in terms of fuel efficiency and safety. However, due to safety and convergence issues of DRL, our algorithm establishes transition times and periods of operation of both ACC and CACC controllers instead of directly controlling the vehicles. In other words, the learning dictates the switching rule by defining the set of transitions times $\mathcal{K}$ where the switching of controllers takes place. The appropriate choice is unknown due to the unpredictable behavior of the jammer vehicle. Such a challenge motivates the use of DRL algorithms that are able to learn the preceding vehicle dynamics from iterative experiences with the environment.

The problem of longitudinal vehicle platoon control is formulated as a Markovian Decision Process (MDP, [29]). An MDP is defined as a tuple $M \equiv (\mathcal{S}, \mathcal{A}, p, r, \gamma)$. The set $\mathcal{S}$ is the DRL state space which in our case is continuous. $\mathcal{A}$ is a finite action space. For each $s \in \mathcal{S}$ and $\tilde{a}$[1]$\in \mathcal{A}$, the transition function $p(\cdot|s, \tilde{a})$ gives the next-state distribution upon taking

[1]In the literature, the action is usually denoted by a. However, in order to differentiate from the acceleration of the vehicles, we adopted $\tilde{a}$ to symbolize the action from the agent.

action $\tilde{a}$ in state s. Observe that p is related to (7, 20, 48), and it is stochastic due to the presence of $\sigma(k)$. $r : \mathcal{S} \times A \times \mathcal{S} \to \mathbb{R}$ is a reward function more detailed in the sequel. Let S_t, $\tilde{A}_t$ and $R_t = r(S_t, \tilde{A}_t, S_{t+1})$ be random variables corresponding to the state, action, and reward, respectively, at time step t. $\gamma \in [0, 1)$ is a discount factor that gives smaller weights to future rewards. The goal of the agent is to find a policy $\tilde{\pi} : \mathcal{S} \to \mathcal{A}$ that maximizes the expected discounted sum of reward *(return)* $G_t = \sum_{i=0}^{\infty} \gamma^i R_{t+i}$.

Therefore, since the state set $\mathcal{S}$ is continuous and the action set $\mathcal{A}$ is discrete and taking into consideration its ease of implementation and performance properties [7], we have adopted the value-based Double Deep Q-Networks (DDQN) algorithm proposed by van Hasselt *et al.* [30]. In order to learn the best actions to be chosen, this network approximates the Q-function while addressing the overestimation problem and improving the stabilization of the training. The experience replay buffer is used to store past experiences, and randomly use subsets of them to update the Q-network improving the sample efficiency of the training. The DDQN algorithm adopts the epsilon-greedy method as its exploration strategy as defined by

$$g(t) = 0.05 + 0.85e^{-\frac{t}{7}} \tag{50}$$

which means that if a random number generated by the model at a certain step t is lower than (50), the model selects a random action (exploration), but if it is higher than (50) the model chooses an action based on what it has learned so far (exploitation). The main components of our formulated MDP are explained next.

1) State space ($\mathcal{S}$): Characterized by the relative position, velocity, absolute acceleration of each vehicle of the platoon, and the platoon's accumulated fuel consumption as defined in (10). In summary, the DRL's state space is defined as $\mathcal{S} = [e_1\ \epsilon_1\ a_1 \cdots\ e_{N-1}\ \epsilon_{N-1}\ a_{N-1}\ J_1\ \cdots J_{N-1}]'$ where N is the platoon size. Note that the DRL state space is different from the dynamical system's state space, as defined in (7).

2) Action space ($\mathcal{A}$): The discrete set $A \equiv \{0, 1\}$ is the action space. It corresponds to two discrete actions (0 or 1) in which the ACC or CACC controller is selected, respectively, as previously defined in Section III.

3) Reward function (r): A proper design of the reward function is crucial for the convergence of the DRL algorithm. In this study, the interest is to improve fuel efficiency while maintaining safety, so we considered the following reward function evaluated at each subinterval δ:

$$r = r_{step} + r_{collision} \tag{51}$$

where r_{step} represents the time-step cost, which can be measured by the number of running cycles, and it is defined as:

$$r_{step} = \begin{cases} 1 & \textit{if } k \cdot T_s \leq V \cdot \delta \\ \kappa/\delta & \textit{otherwise} \end{cases} \tag{52}$$

where $k \cdot T_s$ is the discrete-time of the system with a sample time of T_s, κ is the total number of time-steps in which the limit of fuel supply is attained, δ is the MDP sampling time, and V is the maximum positive integer multiple of δ defined by $V = \frac{T_f - \kappa}{\delta}$, such that $V\delta < T_f$. Note that $0 \leq \kappa/\delta \leq 1$ is a fraction of a unitary reward which is proportional to the remaining time. Finally, $k_p = p\delta$ where $p = 0, \cdots, V - 1$. As a result, the agent receives a positive reward for each subinterval proportional to the sampling time, and in the case of reaching out of fuel condition sooner than the sampling time, only its fraction is considered. Note that one of the termination conditions of the simulation is when the platoon fuel is completely used up. The r_{step} term of the reward will make the *return* increase when the simulation runs for more steps, meaning that the platoon used its fuel more efficiently since it lasted for a longer time. Finally, in order to raise safety performance, collisions were treated as penalties with the following reward policy:

$$r_{collision} = \begin{cases} -k_{col} & \textit{if } e(k) < d_{min} \\ 0 & \textit{otherwise} \end{cases} \tag{53}$$

where k_{col} is a positive constant that can be adjusted, and $e(t)$ is the inter-vehicle spacing defined in (1) which, in this case, is lower bounded by the minimum distance d_{min} in meters. The other termination condition of the simulation is when a collision happens. Prematurely terminating the simulation and giving a negative reward at the final step due to collision will minimize the *return*, making the learning algorithm avoid triggering collision events.

Experimental settings: We adopt two hidden layers of rectified non-linearity with 64 units each. The final layer of the DDQN is linear with a scalar output of the Q-value for the possible actions that could be taken. Default hyper-parameters are used for training DNN weights as follows: learning rate $\alpha = 10^{-3}$, discount factor $\gamma = 0.99$, and batch size of 64. The reward constants are set to be $k_{col} = 1$, and minimum and maximum distance bounds for reward penalty as $d_{min} = 1$ and $d_{max} = 70$, respectively. The MDP problem is set with a time-step of $T_s = 20$ s while the system dynamics time-step is $T_s = 0.1$ s. State normalization was demonstrated to be of utmost importance for algorithm convergence. Because in DNN training, the scale of the input signal is maintained when it is passed through the DNN.

VII. Performance evaluation

In this section, we present the performance evaluation of the system according to different control approaches.

A. Numerical stochastic profiles of the jammer

Before introducing the simulation environment, we highlight the particular class of stochastic disturbances modeled by a two-state Markov chain. The first mode represents the constant profile, and the second is the aggressive one. Furthermore, a Markov process is completely determined by the well-known transition matrix P, which for the adopted scenario is defined by

$$P = \begin{bmatrix} 0.9975 & 0.0025 \\ 0.0165 & 0.9835 \end{bmatrix}. \tag{54}$$

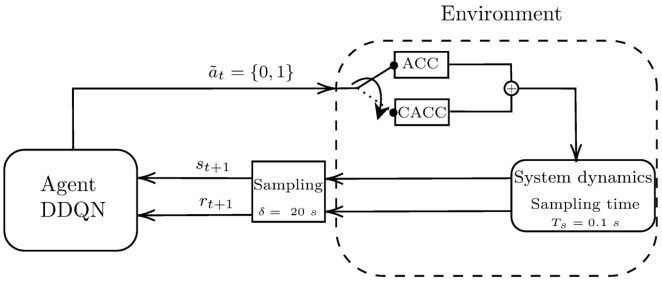

Fig. 2: Overview of the DRL framework for our platoon system.

This leads to a higher recurrence of the constant mode when compared to the aggressive mode. We aim to simulate a highway scenario where traffic jams occur less often. Consequently, we adopt a uniform distribution of the acceleration of the jammer as $\varsigma = 2$, so $\mathcal{U}[-2, 2]$ weighted by $\vartheta = 0.01$ constant in order to produce such nearly steady behavior.

B. Simulation environment

The entire simulation was built with Python to allow straightforward analysis since the DDQN algorithm is smoothly attainable on it due to the support of external libraries. Likewise, we built our own system environment, which includes the longitudinal platoon, the fuel consumption model, and the stochastic behavior of the jammer. In the learning framework, we adopt a considerably different sampling time when compared to the discretization interval $T_s = 100$ ms of the system dynamics, as shown in Figure 2. In fact, we exploit an MDP system with subintervals of $\delta = 20$ s, which translates to updating the learning algorithm less frequently. Note that as the environment is under a sampling time of $T_s = 100$ ms, the fuel consumption, the traveled distance, and possible collisions are still evaluated in a refined manner. Such difference is fundamental to mitigate the computation complexity for the DRL approach. This approach implies that the chosen control does not change, for at least $\delta = 20$ s, which is pertinent to real traffic situations, but the proper value is beyond the scope of this work and is left for future evaluation. Finally, all the simulation parameters adopted for both the control and DRL framework (hyperparameters) are depicted in Table I. The values of the parameters for the energy consumption and the vehicle model were borrowed from Table 3 in Kulava *et al.* [31].

In the considered scenario, we adopted a homogeneous platoon with $N = 3$ vehicles and with actuator lag of $\tau_i = 0.2\ s\ \ \forall i$. More precisely, the environment is initialized with the leader and two platoon members and with a random jammer profile that follows a Markov chain, as previously stated. In each step of an episode, the agent examines the most updated state and the reward feedback before deciding which actions to take. Therefore, based on the environment, i.e., the disturbances caused by the jammer, the agent gets a reward and calculates the most appropriate action (ACC or CACC control) that leads to the most efficient fuel consumption policy. Note

TABLE I: Neural network, control and traffic simulation parameters

Control and Traffic		**Neural Network**	
Parameter	**Value**	**Parameter**	**Value**
Simulation duration	1000 s	Learning rate (α)	10^{-3}
Jammer profile	Stochastic	Discount factor (γ)	0.99
Platoon size (N)	3	Batch size	64
ACC		Reward $\{k_{col}\}$	$\{1\}$
Time-gap (h)	1.4	Bounds $\{d_{min}, d_{max}\}$	$\{1, 70\}$
Gain (λ)	0.5	Hidden layers	2
Standstill dist. (d_{ss})	7 m	Buffer size	10000
CACC		MDP sampling time (δ)	20 s
Leader factor (C)	0	Steps update target NN	500
Desired dist. (d_{des})	7 m	Epsilon-greedy	Eq. (50)
Damping ratio (ξ)	2		
Bandwidth (ω_n)	0.5 Hz		

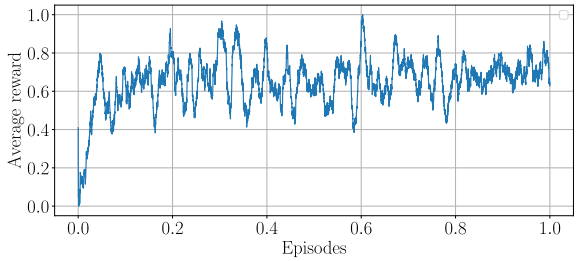

Fig. 3: Performance results for the training phase of the DDQN agent. We can see an empirical convergence of the average reward function over the respective epochs.

that due to the modification of the sampling time, the agent is only allowed to make decisions at each $\delta = 20$ s. Figure 3 shows the normalized average reward during the training of the DDQN agent, and illustrates that empirical convergence is achieved. Note that after a certain number of episodes, the average reward in the normalized graph increases from 0.0 to approximately 0.7 ± 0.1.

C. Performance over baseline

In order to evaluate the performance of the proposed controllers, we considered the static ACC controller as the baseline approach. It performs the safest outcomes due to requiring larger inter-vehicle distance and does not rely on any type of V2X communication. Moreover, such a controller is generally established as a backup system in case of losing the wireless link for an extended period of time [1], [2]. Therefore, to validate the performance of both threshold and DRL approaches over static control strategies, we compare them under the same environment settings per episode, we generate several episodes to obtain a reliable amount of samples. To validate our model performance against all approaches, we generated a test data set consisting of a thousand unseen jammer profiles modeled with Markov chains. Table II encompasses the average performance comparison of different approaches against baseline (ACC) for the test data set. It includes naive and optimized threshold policies and the DRL approach, as explained next. We can conclude that the DRL approach achieves the most suitable behavior among all the evaluated alternatives with an average $+6.83\%$ of superiority

TABLE II: Average fuel-efficient performance comparison against baseline (ACC) for 1k episodes for transition matrix P as in (54).

	Threshold naive	**Threshold optimized**	**DRL**
5% troublesome	4.68 %	6.13 %	6.83 %
10% troublesome	3.16 %	5.03 %	5.74 %

in terms of fuel consumption efficiency for 5% of troublesome conditions. It should also be noted that fuel reduction is obtained without reducing the average velocity.

Both threshold policies are defined by the same sliding window of length $sw = 50$ s as in (49), but different threshold parameter values ε_{th} are adopted. For instance, the optimized threshold version, namely $\varepsilon_{th_o} = 1.23$, is obtained after careful heuristic optimization by observing several simulations. Whereas the naive threshold, given by $\varepsilon_{th_n} = 0.1$, represents a simpler method acting for small acceleration changes. The optimized threshold value is set up experimentally based on the effect of the jammer disturbance and the suitable control signal to react to it.

We next describe the performance in terms of fuel efficiency for a particular jammer profile as shown in Figure 4a. This particular sample displays a robust profile with reasonable troublesome conditions, in addition to aggressive and steady behaviors. Furthermore, notice that the steady mode prevails as we attempt to model the stochastic behavior of vehicles on the highway where traffic jams occur sporadically. Figure 4b exhibits the performance over the baseline for a particular disturbance sample, i.e., it is the percentage error of the evaluated controllers compared against the baseline under the disturbance shown in Figure 4a. As can be seen, the DRL approach, in solid blue, detains the highest fuel efficiency among all cases compared to the baseline. Note that in Figure 4b, we used the trained DDQN agent and not its performance under exploration. Next, Figure 4c illustrates the relative distance for the platoon members for different approaches for the same experiment. Unlike the speed profile, we can see a substantial difference between the approaches. In particular, due to fewer switching control modes, the DRL approach can keep short distances (7 m) for around 400 s to 1000 s despite the disturbances. Due to many switching controls, the threshold displays considerable fluctuations between 30 to 7 m, producing uncomfortable behavior for the passengers and undesired fuel.

Finally, Figure 4d displays the smooth control design parameter $\beta(k)$ proposed to mitigate the losses caused by the switching control. We can observe the DRL approach properly adopts the ACC control ($\beta(k) = 0$) during the aggressive mode of the jammer (as seen in the time-scale between 200 to 400 s) and refuses to switch during troublesome conditions that do not remain for long. Moreover, in dotted green, the naive threshold logic approach performs almost three times more switching behavior than the DRL approach, which translates to unnecessary actions and undesired losses. On the other hand, the optimized threshold logic performs better than the naive approach, with slightly more switching behavior than the DRL, highlighting the importance of properly tuning the threshold parameter.

VIII. Conclusions

In this paper, we have precisely addressed the fuel efficiency in a longitudinal platoon by means of switching classical control policies such as ACC and CACC through a DRL approach. We contemplate stochastic disturbances, which are modeled by Markov chains. To cope with such a stochastic framework, we proposed two different switching control strategies: a threshold switching rule and a DRL approach. Our simulation results show that the DRL approach is the most fuel-efficient when compared to all evaluated controllers. Despite the relatively small advantage obtained, we expect a substantial improvement when the hyper-parameters of the neural network are properly configured. Also, note that we assumed a simple model for the fuel consumption of the vehicles that is a function of velocity and engine force. In actual vehicles, due to gear shifts, the actual fuel consumption may be even higher, which we expect to boost the platoon gains. For future work, we aim to adopt selected datasets with realistic velocities profiles of vehicles to improve the jammer model and to extend the theorem analysis for the case of not constant jammer disturbances. Furthermore, we aim to consider a decentralized setting, treating each platoon's vehicle as an individual DRL agent. Finally, we can focus on our own reinforcement learning algorithm and compare its performance over traditional approaches.

References

[1] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful degradation of cooperative adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 488–497, 2014.

[2] J. Ploeg, B. T. Scheepers, E. Van Nunen, N. Van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 260–265, IEEE, 2011.

[3] V. Turri, B. Besselink, and K. H. Johansson, "Gear management for fuel-efficient heavy-duty vehicle platooning," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 1687–1694, IEEE, 2016.

[4] V. Turri, B. Besselink, and K. H. Johansson, "Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 12–28, 2016.

[5] D. Tian, C. Lin, J. Zhou, X. Duan, Y. Cao, D. Zhao, and D. Cao, "Sa-yolov3: An efficient and accurate object detector using self-attention mechanism for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[6] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 33–47, 2020.

[7] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, J. Pineau, *et al.*, "An introduction to deep reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[8] A. Al Alam, A. Gattami, and K. H. Johansson, "An experimental study on the fuel reduction potential of heavy duty vehicle platooning," in *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 306–311, IEEE, 2010.

[9] K.-Y. Liang, J. Mårtensson, and K. H. Johansson, "When is it fuel efficient for a heavy duty vehicle to catch up with a platoon?," *IFAC Proceedings Volumes*, vol. 46, no. 21, pp. 738–743, 2013.

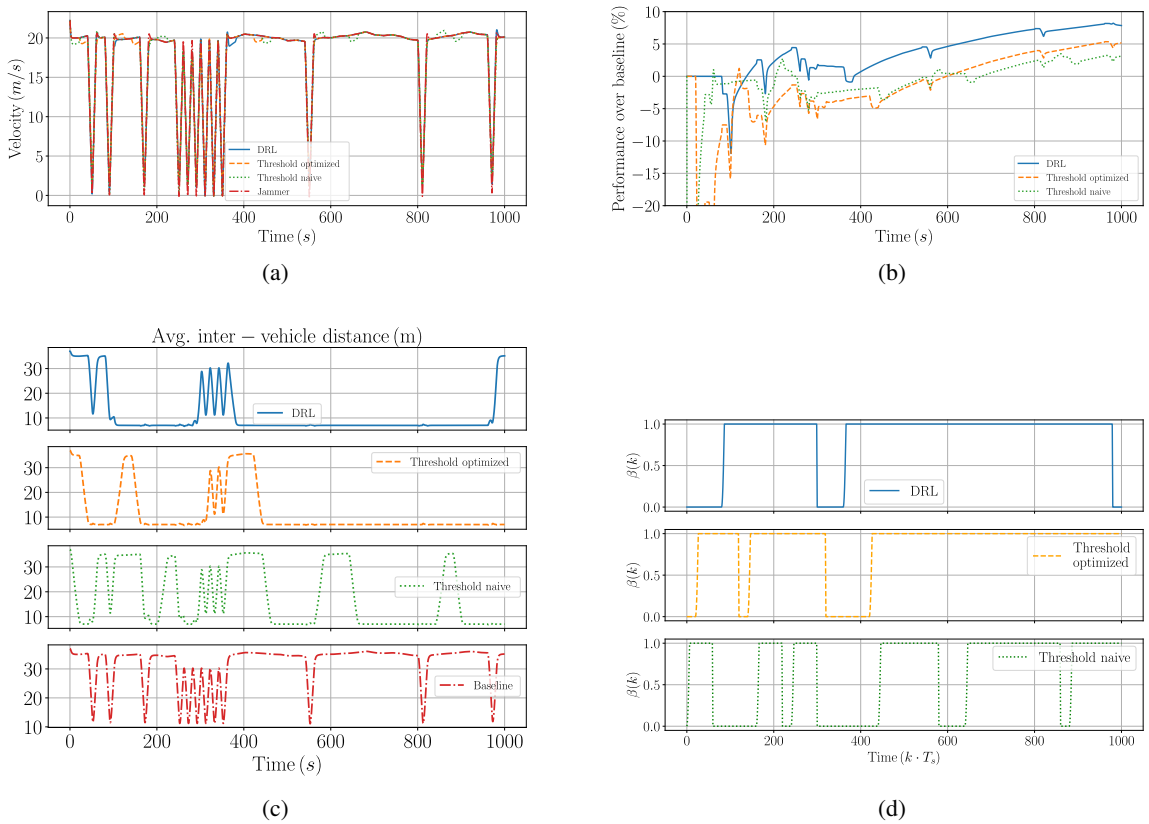

Fig. 4: Illustration of a particular jammer profile investigated in (a), the corresponding fuel platoon performance relative to the baseline approach in (b), the average inter-vehicle distance of the platoon members in (c), and the smooth control design parameter $\beta(k)$ for the DRL and both threshold switch control approaches in (d).

[10] L. Ng, C. M. Clark, and J. P. Huissoon, "Reinforcement learning of adaptive longitudinal vehicle control for dynamic collaborative driving," in *2008 IEEE Intelligent Vehicles Symposium*, pp. 907–912, IEEE, 2008.

[11] C. Desjardins and B. Chaib-Draa, "Cooperative adaptive cruise control: A reinforcement learning approach," *IEEE Transactions on intelligent transportation systems*, vol. 12, no. 4, pp. 1248–1260, 2011.

[12] G. Ling, K. Lindsten, O. Ljungqvist, J. Löfberg, C. Norén, and C. A. Larsson, "Fuel-efficient model predictive control for heavy duty vehicle platooning using neural networks," in *2018 Annual American Control Conference (ACC)*, pp. 3994–4001, IEEE, 2018.

[13] A. Coppola, D. G. Lui, A. Petrillo, and S. Santini, "Eco-driving control architecture for platoons of uncertain heterogeneous nonlinear connected autonomous electric vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24220–24234, 2022.

[14] W. Zhao, D. Ngoduy, S. Shepherd, R. Liu, and M. Papageorgiou, "A platoon based cooperative eco-driving model for mixed automated and human-driven vehicles at a signalised intersection," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 802–821, 2018.

[15] T. Chu and U. Kalabić, "Model-based deep reinforcement learning for cacc in mixed-autonomy vehicle platoon," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 4079–4084, IEEE, 2019.

[16] C. Chen, J. Jiang, N. Lv, and S. Li, "An intelligent path planning scheme of autonomous vehicles platoon using deep reinforcement learning on network edge," *IEEE Access*, vol. 8, pp. 99059–99069, 2020.

[17] W.-H. Hucho, "Aerodynamics of road vehicles, 1998," *Warrendale, PA: Society of Automotive Engineers*, 1998.

[18] V. S. Dolk, J. Ploeg, and W. M. H. Heemels, "Event-triggered control for string-stable vehicle platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3486–3500, 2017.

[19] J. Ploeg, N. Van De Wouw, and H. Nijmeijer, "Lp string stability of cascaded systems: Application to vehicle platooning," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 786–793, 2013.

[20] J. K. Hedrick, M. Tomizuka, and P. Varaiya, "Control issues in automated highway systems," *IEEE Control Systems Magazine*, vol. 14, no. 6, pp. 21–32, 1994.

[21] P. Seiler and R. Sengupta, "An $\mathcal{H}_\infty$ approach to networked control," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 356–364, 2005.

[22] M. Wang, S. P. Hoogendoorn, W. Daamen, B. van Arem, B. Shyrokau, and R. Happee, "Delay-compensating strategy to enhance string stability of adaptive cruise controlled vehicles," *Transportmetrica B: Transport Dynamics*, vol. 6, no. 3, pp. 211–229, 2018.

[23] T. Oguchi, M. Katakura, and M. Taniguchi, "Available concepts of energy reduction measures against road vehicular traffic," in *Intelligent Transportation: Realizing the Future. Abstracts of the Third World Congress on Intelligent Transport SystemsITS America*, 1996.

[24] M. R. Khan, *Advances in clean hydrocarbon fuel processing: Science and technology*. Elsevier, 2011.

[25] P. A. Ioannou and C.-C. Chien, "Autonomous intelligent cruise control," *IEEE Transactions on Vehicular technology*, vol. 42, no. 4, pp. 657–672, 1993.

[26] T. R. Gonçalves, V. S. Varma, and S. E. Elayoubi, "Vehicle platooning schemes considering V2V communications: A joint communication/control approach," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2020.

[27] V. Vukadinovic, K. Bakowski, P. Marsch, I. D. Garcia, H. Xu, M. Sybis, P. Sroka, K. Wesolowski, D. Lister, and I. Thibault, "3GPP C-V2X and IEEE 802.11p for vehicle-to-vehicle communications in highway platooning scenarios," *Ad Hoc Networks*, vol. 74, pp. 17–29, 2018.

[28] S. M. Ross, J. J. Kelly, R. J. Sullivan, W. J. Perry, D. Mercer, R. M. Davis, T. D. Washburn, E. V. Sager, J. B. Boyce, and V. L. Bristow,

Stochastic processes, vol. 2. Wiley New York, 1996.
[29] M. L. Puterman, "Markov decision processes: Discrete stochastic dynamic programming," 1994.
[30] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.
[31] S. T. Kaluva, A. Pathak, and A. Ongel, "Aerodynamic drag analysis of autonomous electric vehicle platoons," *Energies*, vol. 13, no. 15, p. 4028, 2020.

Tiago Rocha Gonçalves received his B.S. degree (*summa cum laude*) in electrical engineering from the Federal University of Rio Grande do Norte, Brazil, in 2016. In 2018, he received his M.S. degree in control from the University of Campinas, Brazil. He obtained his Ph.D. degree from the University of Paris-Saclay in 2021. He joined the Driving Assistance Research Team at Valeo in France in 2023. His research interest includes platooning systems, reinforcement learning, and control and communication interaction.

Rafael Fernandes Cunha received his B.S. degree in electronic engineering from the Technological Institute of Aeronautics (ITA), Brazil, in 2008. From 2009 to 2017, he worked as a Petroleum Engineer at Petrobras, Brazil. In 2018, he received his M.S. degree in control from the University of Campinas, Brazil. He is currently pursuing his Ph.D. at the University of Groningen, the Netherlands, where he has also been a Lecturer in Artificial Intelligence since 2023. His research interest includes reinforcement learning, multi-agent systems, and control theory.

Vineeth Satheeskumar Varma is a CNRS researcher at the Centre de Recherche en Automatique de Nancy (CRAN) in Nancy (France). He received his dual Master's degree in Science and Technology from Friedrich-Schiller-University of Jena in 2009 and Warsaw University of Technology in 2010. He obtained his Ph.D. degree from LSS-University of Paris Saclay in 2014. His areas of interest are energy efficiency in telecommunication, multi-agent systems and the interface of automatic control and communication.

Salah Eddine Elayoubi received his M.S. degree in telecommunications from the National Polytechnic Institute of Toulouse, France, in 2001, and his Ph.D. and Habilitation degrees in computer science from the University of Paris VI, France, in 2004 and 2009, respectively. From 2004 to 2013 he was with Orange Labs in France. Since January 2018, he is a full professor at CentraleSupélec, France. His research interests include radio resource management, modeling, and performance evaluation of mobile networks.