Reinforcement Learning for Enhanced Path Tracking in Autonomous Vehicles: A Formula SAE Skid-Test Validation

Danilo MENEGATTI*, Francesco LUZI, Francesco PAPPALARDO, Antonio PIETRABISSA, and Alessandro GIUSEPPI

Abstract— **Accurate path tracking is one of the main challenges autonomous vehicles have to deal with. It is known that, when dealing with real hardware systems, the presence of parametric uncertainty and unmodelled aspects of the system dynamics affects all model-based control approaches, hindering their nominal performance guarantees. To compensate for this issue, data-driven schemes have drawn significant attention from the scientific community thanks to their inherent ability to learn from experience, thus automatically compensate for system uncertainties and time-varying behaviours. This work aims to develop a reinforcement learning-based longitudinal and lateral dynamics control introducing mismatch and motion penalization metrics, validating the resulting controller in a simulated Formula SAE skid-test scenario employing the Sapienza Fast Charge Formula SAE Electric Racing Team dynamical model.**

Index Terms— **Path Tracking, Deep Reinforcement Learning, Intelligent Systems.**

I. INTRODUCTION

Autonomous driving has emerged as a promising technology with the potential to completely reshape the future of transportation. By leveraging advancements in intelligent control systems and sensor technology, it aims to enhance road safety, reduce traffic congestion, and provide further mobility options. From an architectural perspective, being designed to navigate complex environments, autonomous vehicles comprise of three main components: environmental perception, path planning, and path tracking. While the former are related to surroundings awareness and optimal route determination respectively, the latter involves maintaining the vehicle within desired boundaries given a predefined trajectory to follow.

Numerous studies have focused on the path tracking problem aiming to reduce the error in following the planned path in terms of orientation, lateral and longitudinal motion. To this end, simple geometric algorithms have been proposed, such as the Pure Pursuit (PP) [1] and the Stanley [2] controller; the former leverages a look-ahead waypoint, that is a point of the planned path ahead of the vehicle, to determine the steering angle needed to adjust the orientation of the vehicle, while the latter computes it by minimizing both the cross-track error and the heading error to ensure the vehicle follows the path accurately.

Due to its simplicity of implementation, multiple attempts were made to improve the tracking accuracy of the PP algorithm which is strictly related to the choice of the look-ahead distance; the authors of [3] optimize it by means of swarm and Brownian motion algorithms, while in [4] a dynamic optimization of such a distance is proposed on the basis of goal points pre-selected along the trajectory. A combination of PP and Stanley controllers has been proposed in [5] for high speed and curvature scenarios, while the authors of [6] attempted to fine tune of Stanley's gain via genetic programming. Over the years, various Proportional Integral Derivative (PID) controllers have been developed to deal with the path tracking problem [7]. In particular, the authors of [8] developed an adaptive PID showing superior performance in S-shaped trajectories, while the authors in [9] proposed a linear parameter varying controller, and [10] introduced a fuzzy logic-based control system. Nevertheless, several more advanced control paradigms have been employed to tackle the tracking issue, such as sliding mode control [11], nonlinear control [12] [13] and artificial potential fields [14].

A crucial aspect of path tracking is the explicit inclusion of constraints in its formulation. This challenge has been addressed in various ways, and in the context of Linear Quadratic Regulators (LQR) the authors of [15] proposed an iterative algorithm to deal with non-convex constraints, while in [16] a LQR with feedforward and feedback action is developed, with the authors of [17] considering the same control architecture of [16] for high-curvature and low adhesion scenarios. Due to its inherent ability to deal with constraints explicitly, Model Predictive Control (MPC) has also been broadly employed to achieve lateral and longitudinal control of the vehicle in [18], [19], with the authors of [20] addressing the issue of road curvature at different speeds.

In this context, Reinforcement Learning (RL) has been recently investigated both as a standalone control synthesis method and in conjunction with other control schemes to tune some specific parameters, as seen in [21] where the authors used it to tune PID parameters, and in [22] where it was used to enhance the robustness of MPC. Regarding full RL controllers, we mention [23], where a RL-based real-time moving object tracking system is developed, and [24], where the authors warm start the RL agent training procedure by means of the imitation learning paradigm leveraging human driver's driving data.

This work deals with the path tracking problem by developing a RL-based control architecture which achieves the control of longitudinal and lateral dynamics of the vehicle by introducing a set of mismatch and motion penalization metrics which guide the training of the agent towards an optimal control law. In addition, the proposed approach is

The authors are with the Department of Computer, Control and Management Engineering "Antonio Ruberti" – Sapienza University of Rome.

*Corresponding author. Email: menegatti@diag.uniroma1.it

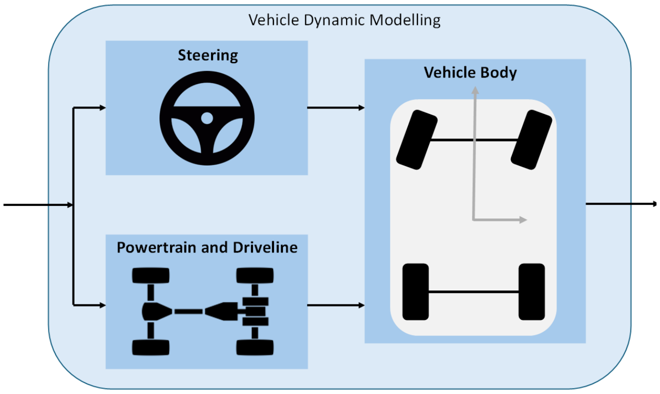

Fig. 1. High level dynamic architecture of the controlled vehicle.

validated over a simulated Formula SAE skid-test scenario, leveraging the Sapienza Fast Charge Formula SAE Electric Racing Team dynamical model.

To summarize, the main contributions of this work are:

- Formulation of a control strategy based on mismatch and penalty metrics within the RL paradigm tailored for autonomous driving scenarios.
- Implementation and enforcement of stringent requirements for both longitudinal and lateral vehicle dynamics.
- Evaluation of the proposed approach over a Formula SAE skid-test scenario.

The remainder of the manuscript is organized as follows: Section II describes the dynamic model of the system and defines the penalty and mismatch metrics; Section III offers the theoretical foundations of RL and details the proposed approach; Section IV presents the simulations results, highlighting the advantages of the proposed approach; Section V draws the conclusions and outlines future work.

II. System Modelling

The driverless racing system to be controlled consists of a four-wheel-drive vehicle with steering front wheels and it is comprehensively modelled through three modules, *Steering*, responsible for the steering process, *Powertrain and Driveline*, and *Vehicle Body*, as shown in Figure 1.

In order to model the driving process as accurately as possible, the Steering module is introduced to convert the steering angle received as input into the actual wheel angle, hence correlating steering wheel movements with those of the wheels. Being the front wheel steering parallel, the wheel angle is calculated as follows:

$$\delta_w = \frac{\theta_w}{\Gamma}, \tag{1}$$

where Γ is the steering ratio, θ_w is the steering angle, and δ_w the wheel angle.

The Powertrain and Driveline module is responsible for translating driving commands, such as acceleration and braking, into reference signals for each electric motor, such as power and shaft torque, in order to distribute the driving or braking force applied to the wheels, taking into account their condition, while the Vehicle Body module describes the vehicle lateral and longitudinal dynamics via the double track model [25], thus focusing on the horizontal dynamics while neglecting vertical roll, pitch, and yaw.

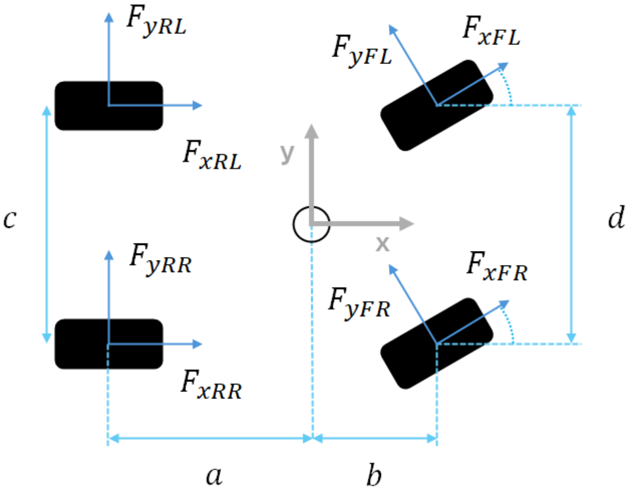

Fig. 2. Geometry of the double track model.

The resulting geometry is depicted in Figure 2, where a and b are distances from the rear and front axles to the vehicle center of gravity, which corresponds to the origin of the xy reference frame, c and d the length of the rear and front axels. In addition, the forces acting on each wheel are represented, where the first subscript refers to the axis along which they act, and the second one uniquely identifies the wheel.

As the car is self-driving, the generation of driving commands is delegated to a feedback-type controller from the continuous monitoring of the vehicle's behaviour. Specifically, it is responsible for the control of longitudinal and lateral dynamics.

Controlling the longitudinal dynamics means to manipulate the velocity of the vehicle with respect to a predefined reference value, that is to solve a set-point tracking problem. Let $v(t)$ be the magnitude of the velocity at time t and $\bar{v}$ the corresponding reference value, the mismatch $e_v(t)$ between the two is evaluated as follows:

$$e_v(t) = -\alpha_v \frac{\left(v(t) - \bar{v}\right)^2}{\left(v(t) - \bar{v}\right)^2 + \beta_v}, \tag{2}$$

where $\alpha_v \in \mathbb{R}$ and $\beta_v \in \mathbb{R}$ coefficients which ultimately influence the behaviour of the controller; while the former acts like a scaling factor for the error, the latter is related to the steepness of the curve, that is to the sensitivity of the controller with respect to changes in the velocity.

The lateral dynamics is related to the orientation of the vehicle, hence controlling it involves the manipulation of the steering angle so the system can follow a predefined desired trajectory, effectively solving a tracking problem with a time-varying reference. Let $\left(\tilde{x}(t), \tilde{y}(t)\right)$ be the position of the vehicle at time t with respect to a reference frame and let $\left(\bar{x}_i, \bar{y}_i\right)$ (for $i = 1, \ldots, N$, with $N \in \mathbb{N}$ being the number of points comprising the trajectory) be the point on the trajectory closest to the vehicle, measured in terms of the Euclidean norm. The control action is then derived on the basis of two error quantities, related to the steering angle error $e_w(t)$, given between the actual and desired orientation, and the lateral motion error $e_m(t)$, related to the lateral distance from the trajectory.

These two errors are, at least in part, caused by the curvature of the trajectory. Let $\left(\bar{x}_{i-1}, \bar{y}_{i-1}\right)$ and $\left(\bar{x}_{i+1}, \bar{y}_{i+1}\right)$ be the points on the trajectory immediately before and after $\left(\bar{x}_i, \bar{y}_i\right)$, which is the point closer to the position of the vehicle. These three points uniquely identify the osculating circle approximating the trajectory, with its curvature evaluated as follows:

$$\kappa = \frac{|\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)|}{\left(\dot{x}^2(t) + \dot{y}^2(t)\right)^{3/2}}, \tag{3}$$

with $\left(\dot{x}(t), \dot{y}(t)\right)$ and $\left(\ddot{x}(t), \ddot{y}(t)\right)$ being the first and second order derivatives of the trajectory $\left(x(t), y(t)\right)$, calculated as finite differences with respect to $\left(\bar{x}_i, \bar{y}_i\right)$, that is:

$$\begin{array}{ll} \dot{x}(t) = \dfrac{\Delta x}{\Delta t} & \dot{y}(t) = \dfrac{\Delta y}{\Delta t} \\ \ddot{x}(t) = \dfrac{\Delta \dot{x}}{\Delta t} & \ddot{y}(t) = \dfrac{\Delta \dot{y}}{\Delta t}, \end{array} \tag{4}$$

with $\Delta x = \bar{x}_{i+1} - \bar{x}_{i-1}$ and $\Delta y = \bar{y}_{i+1} - \bar{y}_{i-1}$.

The steering angle error $e_w(t)$ is defined as the difference between the actual orientation $\psi(t)$ of the vehicle and a desired one that is evaluated on the basis of the above mentioned differences, $\bar{\psi}(t) = \text{atan2}(\Delta y, \Delta x)$. Finally, the error is evaluated as follows:

$$e_w(t) = \psi(t) - \bar{\psi}(t). \tag{5}$$

In order to take into account potential discontinuities coming from trajectories that cross the coordinate axes, the error is defined as follows:

$$e_w(t) = \begin{cases} e_w(t) - 2\pi & \text{if } e_w(t) > \pi \\ e_w(t) + 2\pi & \text{if } e_w(t) < \pi \text{ .} \\ e_w(t) & \text{otherwise} \end{cases} \tag{6}$$

The lateral motion error $e_m(t)$ is related to the deviation of the vehicle from its desired trajectory, and it is evaluated in the integral form as follows:

$$e_m(t) = e_m(t_0) + \int_{t_0}^{t} \dot{e}_m(\tau) d\tau \tag{7}$$

with $e_m(t_0)$ being the deviation from the desired trajectory at time t_0. Without loss of generality, let $t_0 = 0$, $\dot{e}_m(t)$ is then evaluated as follows:

$$\dot{e}_m(t) = v_x(t)\, \dot{e}_w(t) + v_y(t), \tag{8}$$

where $\left(v_x(t), v_y(t)\right)$ are the components of the velocity $v(t)$, and $\dot{e}_w(t)$ corresponding by definition to:

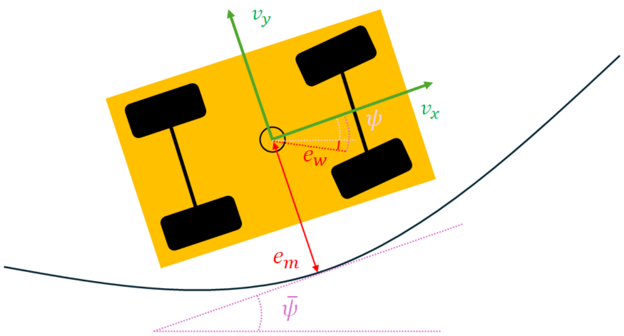

Fig. 3. Graphical representation of $e_m(t)$ and $e_w(t)$ at a given instant of time $\bar{t}$.

$$\dot{e}_w(t) = \dot{\psi}(t) - \dot{\bar{\psi}}(t), \tag{9}$$

with $\dot{\bar{\psi}}(t) = \kappa\, v_x(t)$.

III. RL-based Driverless Racing

A decision–making problem in the context of RL can be formally modeled as a Markov Decision Process (MDP). The latter is defined via a tuple

$$(\mathcal{S}, \mathcal{A}, P, R, \gamma), \tag{10}$$

where $\mathcal{S}$ is the finite set of states representing all possible configurations of the environment, $\mathcal{A}$ is the finite set of actions that the agent can take in each state, $P(s, a, s')$ is the transition probability function, representing the probability of transitioning from state s to state s' when taking the action a, $R(s, a, s')$ is the reward function, which assigns a scalar reward to the agent when it passes from state s to state s' through action a, and $\gamma \in [0,\ 1)$ is the discount factor, representing the agent's preference for immediate rewards ($\gamma \approx 0$) over the future ones ($\gamma \approx 1$).

The objective in a RL–based decision–making problem is to find a policy $\pi : S \rightarrow A$ that specifies which action to take in each state, in order to maximize the following function

$$V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s\right], \tag{11}$$

known as expected return or state–value function.

The agent learns the policy during the training phase which involves a trade–off between exploration (trying new random actions) and exploitation (choosing actions that are currently believed to be the best). At the end of the training procedure, the agent deploys the learned policy π^* on the environment, during the so–called execution phase.

This work introduces a novel RL-based Driverless Racing approach leveraging the Deep Deterministic Policy Gradient (DDPG) algorithm [26], which can tackle problems with continuous action and state spaces. DDPG is based on an actor-critic architecture, in which the actor network approximates the policy, and the critic network approximates the state-value function (11), which in turn estimates the expected

Algorithm 1 Trajectory Generator

```
Set parameter $N \in \mathbb{N}$
Randomly generate $T_0, T_f$

Trajectory_Generator
$x \leftarrow$ LINSPACE($T_0$, $T_f$, $N$)
$y \leftarrow$ SQUARE($x$)
return $\big(x(t), y(t)\big)$
```

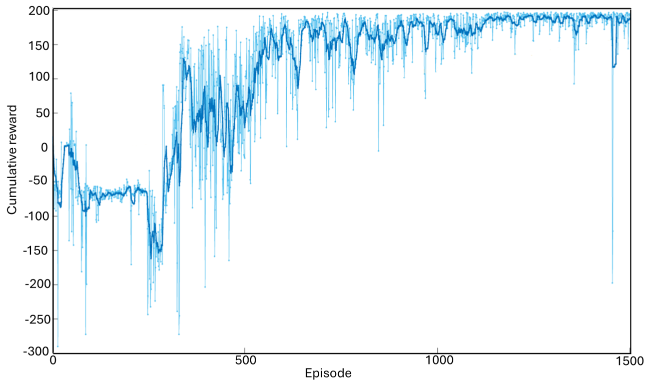

Fig. 4. Evolution of the episodic reward during training (thin line) and moving average (thick line).

cumulative reward obtained by starting from a certain state s and following a given control policy π. To enhance learning stability, DDPG introduces an experience replay buffer to store past experiences and typically adds a dynamic noise to the actions selected by the actor in order to efficiently explore the environment.

The state of the agent is chosen to be as representative and significant as possible, therefore considering lateral motion error $e_m(t)$ and its derivative $\dot{e}_m(t)$, steering angle error $e_w(t)$ and $\dot{e}_w(t)$, longitudinal velocity $v_x(t)$, speed error $e_s(t) = \bar{v}(t) - v(t)$, where $\bar{v}(t)$ is the speed reference, previous steering angle $\theta_w(t-1)$ and longitudinal acceleration $\dot{v}_x(t-1)$. Therefore, the state space is as follows:

$$\begin{aligned} \mathcal{S} = \Big\{ \big(& e_m(t), \dot{e}_m(t), e_w(t), \dot{e}_w(t), \\ & v_x(t), e_s(t), \theta_w(t-1), \dot{v}_x(t-1) \big) \Big\} \end{aligned} \tag{12}$$

The action space comprises of the commanded steering angle θ_w and longitudinal acceleration $\dot{v}_x(t)$, that is:

$$\mathcal{A} = \{ (\theta_w(t), \dot{v}_x(t)) \} \, . \tag{13}$$

The objective of the agent is to ensure that the vehicle's driving style is appropriate for the provided trajectory and velocity references, by controlling its longitudinal and lateral dynamics. Such a goal is achieved via the following multi-objective reward function:

$$\begin{aligned} R(t) = & - \big(w_1 \, e_w^2(t) + w_2 \, e_m^2(t) \big) \\ & + w_3 \, e_v(t) - w_4 \, |\Delta\theta_w(t)|^2 \\ & - w_5 \, |\Delta\dot{v}_x(t)|^2 - w_6 \, \delta_1(t) \\ & + w_7 \, \delta_2(t), \end{aligned} \tag{14}$$

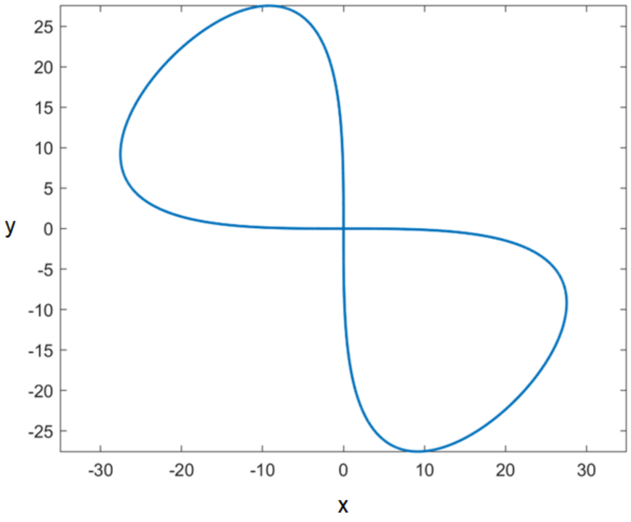

Fig. 5. Formula SAE's skid pad-like 8-shaped trajectory used for testing purposes.

with w_i $i = 1, \dots, 7$ being weights responsible for deciding a trade-off between the various objectives, $\Delta\theta_w(t) = \theta_w(t) - \theta_w(t-1)$, $\Delta\dot{v}_x(t) = \dot{v}_x(t) - \dot{v}_x(t-1)$, and $\delta_1(t)$ being a function that penalizes excessive lateral deviation as follows:

$$\delta_1(t) = \begin{cases} 1 & \text{if } e_m(t) \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad , \tag{15}$$

while $\delta_2(t)$ encourages the agent to keep the vehicle within a predefined lateral range as follows:

$$\delta_2(t) = \begin{cases} \gamma_1 & \text{if } e_m^2(t) < \eta_1 \\ \gamma_2 & \text{if } \eta_1 < e_m^2(t) < \eta_2 \\ 0 & \text{otherwise} \end{cases} \quad . \tag{16}$$

with η_i $i = 1, 2$ being two thresholds to be chosen.

IV. Simulations and Results

In order to validate the effectiveness of the proposed approach, we consider a scenario of an 8-shaped trajectory and a velocity reference value $\bar{v} = 10$ [m/s]. This scenario is typical of the skid-pad tests in Formula SAE, whose purpose is to assess the vehicle's performance in curves. The dynamical system simulated is based on the high-fidelity model employed to validate the Sapienza Fast Charge Formula SAE electric vehicle.

The RL agent is trained for $E = 1500$ episodes, each one lasting $T = 10$ [s], with Runge-Kutta 4th order method adopted to integrate the system dynamics with a time-step $dt = 0.04$ [s].

Training data are generated by tracking parabolic trajectories that are randomly generated according to Algorithm 1. Each trajectory is a two-dimensional curve composed of $N = 1000$ points, with starting point T_0 and ending point T_f randomly generated. The function LINSPACE generates a vector x of N linearly spaced points between T_0 and T_f,

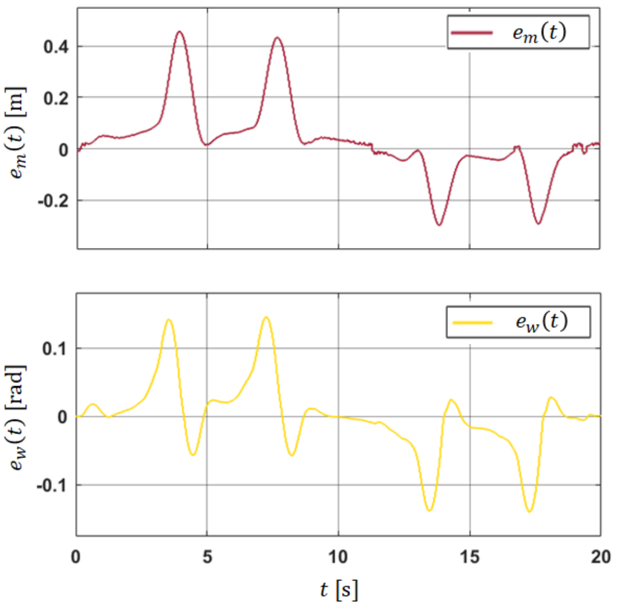

Fig. 6. Evolution of $e_m(t)$ (top) and $e_w(t)$ (bottom) during testing.

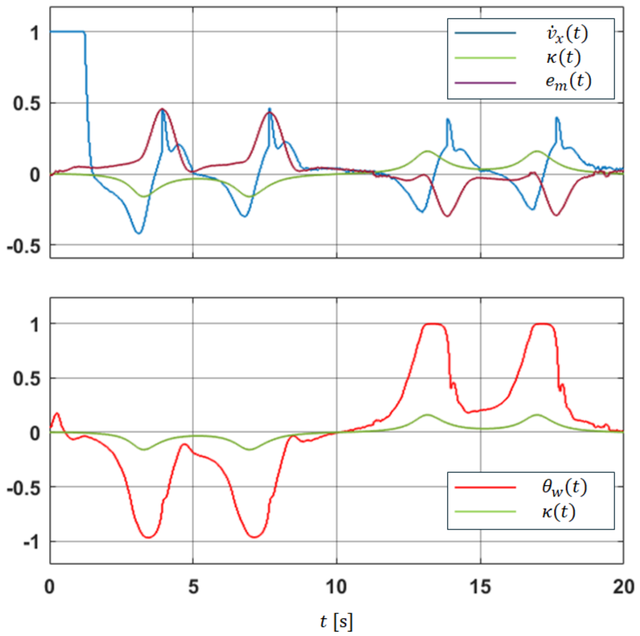

Fig. 7. Evolution of the action of the agent $(\theta_w(t), \dot{v}_x(t))$ with respect to the curvature of the desired trajectory $\kappa(t)$ and lateral deviation $e_m(t)$.

while SQUARE associates to each component of x its square, resulting in a trajectory $\big(x(t), y(t)\big)$.

The DDPG hyperparameters have been carefully tuned as follows: the actor learning rate $\beta_a = 5 \times 10^{-5}$, the critic learning rate $\beta_c = 1 \times 10^{-3}$, gradient threshold $\alpha = 1$, discount factor $\gamma = 0.99$, batch size $B = 32$, and memory capacity set as 5×10^4 transitions. The actor and critic share almost the same neural architecture composed of three layers, the first two of 256 and 256 neurons with ReLU [27] activation function, and the last one constituted by a single neuron with linear activation for the critic and two tanh neurons for the actor.

Ornstein-Uhlenbeck [28] noise is employed in the exploration phase, with a standard deviation of 10% for $\theta_w(t)$ and 30% for $\dot{v}_x(t)$. The mean value of the noise is 0, with a standard deviation decay rate of 1×10^{-5} and a minimum standard deviation of 0. The initial value of the noise is set to 0, and the mean attraction constant is 0.15.

The weights and hyperparameters of the reward function are chosen as follows, $w_1 = w_2 = 1$, $w_3 = 1.5$, $w_4 = w_5 = 2$, $w_6 = 100$, $w_7 = 10$, $\gamma_1 = 10$, $\gamma_2 = 5$, $\eta_1 = 0.01$, and $\eta_2 = 0.04$, and the function which penalizes excessive lateral deviation $\delta_1(t)$ serves as stopping criteria for each training episode. Moreover, with respect to (2), $\alpha_v = 3$ and $\beta_v = 2$.

The training process is carried out in about 1 hour using Matlab [29] on an Intel Core i7 platform with 8 GB of RAM. The evaluation is performed over the same machine.

Figure 4 shows the evolution of the reward attained by the agent at the end of each training episode (thin line) and its corresponding moving average (thick line). Following an exploration phase lasting approximately 500 episodes, the agents appear to become increasingly confident in learning the environment, reaching convergence after 1100 episodes.

Evaluation of the proposed approach is performed over the 8-shaped trajectory shown in Figure 5, starting from the $(0, 0)$ position running clock-wise. Although such a trajectory represents an unknown scenario to the agent, which has been trained over parabolic trajectories only, it exhibits excellent performance, as illustrated in Figure 6, which depicts the evolution over time of lateral motion $e_m(t)$ and steering angle $e_w(t)$ errors.

Focusing on $e_m(t)$ (top), it appears that the maximum lateral deviation from the desired trajectory $|e_m^{max}| \approx 0.41$ [m], that is coherent with the penalization threshold in (15). In addition, four oscillations with greater elongation compared to the others are identified. These occur at the four curves with the greatest curvature, which the agent is able to manage quickly and effectively, bringing the vehicle back close to the desired trajectory. Such an oscillating behaviour can be identified also in $e_w(t)$ (bottom), however the maximum deviation from the desired orientation $|e_w^{max}| \approx 0.13$ [rad], hence the agent precisely controls the orientation $\psi(t)$ of the vehicle.

Figure 7 compares the evolution of $\dot{v}_v(t)$ (top) and $\theta_w(t)$ (bottom) against the curvature of the trajectory $\kappa(t)$. From the top plot it is possible to notice that the agent accelerates violently during the initial instants (blue line) in an attempt to reach the desired speed in the shortest possible time, and then it modulates its acceleration by adapting it to the changes in curvature (green line) of the trajectory. In particular, each significant variation of the latter corresponds to an equally significant deceleration; the resulting decrease in speed allows the vehicle to make the turn as to keep the lateral deviation (purple line) as bounded as possible. Similar considerations can be made on the steering angle side (bottom). In both cases, the agent exhibits human-like behaviour, reaching satisfactory performances.

V. Conclusions

This paper has presented a RL-based controller tailored for the real-life operation of Formula SAE driverless vehicle. The proposed control law governs the longitudinal acceleration and steering angle on the basis of error metrics involving velocity and orientation mismatches, to satisfy predefined requirements in terms of lateral deviation and reach human-like performances.

The mixture of penalization and error terms within the reward function results in excellent performance over Formula SAE's skid pad-like test track. Future works shall be related to performance improvements via reward shaping and sensor integration.

Acknowledgment

The authors would like to thank Sapienza Fast Charge Formula SAE Electric Racing Team for the fruitful discussions which provided significant inputs for this work.

References

[1] C. Coulter, "Implementation of the pure pursuit path tracking algorithm," 1992.

[2] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing," in *2007 American Control Conference*, p. 2296–2301, IEEE, July 2007.

[3] R. Wang, Y. Li, J. Fan, T. Wang, and X. Chen, "A novel pure pursuit algorithm for autonomous vehicles based on salp swarm algorithm and velocity controller," *IEEE Access*, vol. 8, p. 166525–166540, 2020.

[4] E. Horvath, C. Hajdu, and P. Koros, "Enhancement of pure-pursuit path-tracking algorithm with multi-goal selection," in *2019 1st IEEE International Conference on Gridding and Polytope Based Modeling and Control (GPMC)*, p. 13–18, IEEE, Nov. 2019.

[5] M. Cibooglu, U. Karapinar, and M. T. Soylemez, "Hybrid controller approach for an autonomous ground vehicle path tracking problem," in *2017 25th Mediterranean Conference on Control and Automation (MED)*, IEEE, July 2017.

[6] L. Wang, Z. Zhai, Z. Zhu, and E. Mao, "Path tracking control of an autonomous tractor using improved stanley controller optimized with multiple-population genetic algorithm," *Actuators*, vol. 11, p. 22, Jan. 2022.

[7] F. A. A. Cheein, C. De La Cruz, T. F. Bastos, and R. Carelli, "Slam-based cross-a-door solution approach for a robotic wheelchair," *International Journal of Advanced Robotic Systems*, vol. 6, Jan. 2009.

[8] P. Zhao, J. Chen, Y. Song, X. Tao, T. Xu, and T. Mei, "Design of a control system for an autonomous vehicle based on adaptive-pid," *International Journal of Advanced Robotic Systems*, vol. 9, Jan. 2012.

[9] M. Corno, G. Panzani, F. Roselli, M. Giorelli, D. Azzolini, and S. M. Savaresi, "An lpv approach to autonomous vehicle path tracking in the presence of steering actuation nonlinearities," *IEEE Transactions on Control Systems Technology*, vol. 29, p. 1766–1774, July 2021.

[10] C. Hu, Y. Chen, and J. Wang, "Fuzzy observer-based transitional path-tracking control for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, p. 3078–3088, May 2021.

[11] J. Guldner, V. Utkin, and J. Ackermann, "A sliding mode control approach to automatic car steering," in *Proceedings of 1994 American Control Conference - ACC '94*, vol. 2 of *ACC-94*, p. 1969–1973, IEEE.

[12] M. Schorn, U. Stahlin, A. Khanafer, and R. Isermann, "Nonlinear trajectory following control for automatic steering of a collision avoiding vehicle," in *2006 American Control Conference*, p. 6 pp., IEEE, 2006.

[13] L. Sabattini, N. Chopra, and C. Secchi, "Distributed control of multi-robot systems with global connectivity maintenance," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 2321–2326, IEEE, Sept. 2011.

[14] J. C. Gerdes and E. J. Rossetter, "A unified approach to driver assistance systems based on artificial potential fields," *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, p. 431–438, Nov. 1999.

[15] J. Chen, W. Zhan, and M. Tomizuka, "Autonomous driving motion planning with constrained iterative lqr," *IEEE Transactions on Intelligent Vehicles*, vol. 4, p. 244–254, June 2019.

[16] Z. Qin, L. Chen, M. Hu, and X. Chen, "A lateral and longitudinal dynamics control framework of autonomous vehicles based on multi-parameter joint estimation," *IEEE Transactions on Vehicular Technology*, vol. 71, p. 5837–5852, June 2022.

[17] K. Kritayakirana and J. C. Gerdes, "Using the centre of percussion to design a steering controller for an autonomous race car," *Vehicle System Dynamics*, vol. 50, p. 33–51, Jan. 2012.

[18] H. Wang, B. Liu, X. Ping, and Q. An, "Path tracking control for autonomous vehicles based on an improved mpc," *IEEE Access*, vol. 7, p. 161064–161073, 2019.

[19] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on Control Systems Technology*, vol. 15, p. 566–580, May 2007.

[20] L. Tang, F. Yan, B. Zou, K. Wang, and C. Lv, "An improved kinematic model predictive control for high-speed path tracking of autonomous vehicles," *IEEE Access*, vol. 8, p. 51400–51413, 2020.

[21] X. Yu, S. Xu, Y. Fan, and L. Ou, "Self-adaptive lsac-pid approach based on lyapunov reward shaping for mobile robots," *Journal of Shanghai Jiaotong University (Science)*, Aug. 2023.

[22] P. Lin, Y. S. Quan, J. H. Yang, C. C. Chung, and M. Tsukada, "Safety tunnel-based model predictive path-planning controller with potential functions for emergency navigation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, p. 3974–3985, Apr. 2023.

[23] Z. Li, J. Zhou, X. Li, X. Du, L. Wang, and Y. Wang, "Continuous control for moving object tracking of unmanned skid-steered vehicle based on reinforcement learning," in *2020 3rd International Conference on Unmanned Systems (ICUS)*, IEEE, Nov. 2020.

[24] M. Liu, F. Zhao, J. Yin, J. Niu, and Y. Liu, "Reinforcement-tracking: An effective trajectory tracking and navigation method for autonomous urban driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, p. 6991–7007, July 2022.

[25] K. Kahraman, M. Senturk, M. T. Emirler, I. M. C. Uygan, B. Aksun-Guvenc, L. Guvenc, and B. Efendioglu, "Yaw stability control system development and implementation for a fully electric vehicle," 2020.

[26] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on Machine Learning* (E. P. Xing and T. Jebara, eds.), vol. 32 of *Proceedings of Machine Learning Research*, (Bejing, China), pp. 387–395, PMLR, 22–24 Jun 2014.

[27] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[28] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Physical Review*, vol. 36, p. 823–841, Sept. 1930.

[29] I. The MathWorks, "Matlab (version 9.13.0 r2022b)," 2022.