250 | https://doi.org/10.3311/PPtr.23455
Creative Commons Attribution

Periodica Polytechnica Transportation Engineering, 53(3), pp. 250–259, 2025

Safe Robust Framework for Reinforcement Learning-based Control of Indoor Vehicles

Attila Lelkó[1,2*], Balázs Németh[1,2]

[1] Institute for Computer Science and Control (SZTAKI), Hungarian Research Network (HUN-REN), Kende u. 13–17., H-1111 Budapest, Hungary
[2] Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary
[*] Corresponding author, e-mail: attila.lelko@sztaki.hun-ren.hu

Received: 15 September 2023, Accepted: 09 May 2025, Published online: 29 May 2025

Abstract

The paper presents the design of a safe data-aided steering control for indoor vehicles using the robust supervisory framework. The goal of the method is to achieve the combination of the effective motion with reinforcement learning (RL) based control and the guaranteed safe motion with robust control. The RL-based control through the Proximal Policy Optimization (PPO) method is designed in which actor and critic agents are used. The supervisory robust control is selected in the form with which robust stability against an additional input disturbance can be guaranteed. The effectiveness of the combination through simulations and experimental test scenarios is illustrated. For test purposes, an F1TENTH type small-scaled test vehicle is used, whose lap time is minimized through the proposed control system.

Keywords

adaptive and robust control of automotive systems, autonomous vehicles

1 Introduction and motivation

Nowadays, due to the appearance of fast hardware tools for solving learning problems, data-based methods are becoming more popular and efficient in the solution of complex control problems. One of the typical examples is autonomous vehicle control in which sensing, perception, decision and control problems must be solved in continuously varying traffic environment. Various performance specifications can be defined in relation to autonomous vehicle control systems. Usually, due to safety reasons, there are primary performance specifications, such as guaranteeing stable vehicle motion, reliability, or obeying different traffic regulations. These specifications must be kept in all cases. Moreover, several further non-safety performance requirements can be defined, which have lower priority, e.g., providing passenger comfort, achieving economic motion, minimization of travel times, etc. Lots of performance criteria together with the complex vehicle environment led to challenging problems for robust and optimal control design methods.

In recent years various solutions to autonomous vehicle control problems have been proposed, i.e., by utilizing a large amount of data in the control design process. It has led to the data-aided enhancement of the classical control methods, e.g., Model Predictive Control Kabzan et al. (2019); McKinnon and Schoellig (2019); Rosolia and Borrelli (2020), model-free control Fényes et al. (2022); Fliess and Join (2021), robust and Linear-Parameter Varying Németh and Gáspár (2021b); Sename (2021); Bao and Velni (2022) methods. Consequently, through a data-aided design enhanced performance levels on comfort and energy consumption can be achieved, and similarly, the safety performance level of the autonomous vehicles can also be guaranteed Nageshrao et al. (2023). Using data-aided robust control tools Varga et al. (2023) in autonomous vehicle context has high relevance, due to different noises, disturbances, and unmodeled dynamics during the motion of the vehicle. The impacts of these unwanted effects can be handled using robust design methods, e.g., in Khosravani et al. (2014) a robust controller has been designed in a driver-in-the-loop scenario. In the work of Chen et al. (2021) a robust driver assistance system for handover scenarios was

Cite this article as: Lelkó, A., Németh, B. (2025) "Safe Robust Framework for Reinforcement Learning-based Control of Indoor Vehicles", Periodica Polytechnica Transportation Engineering, 53(3), pp. 250–259. https://doi.org/10.3311/PPtr.23455

demonstrated, or Shao et al. (2021) proposed model predictive solutions for achieving rollover prevention.

In some of the proposed methods data has been used for training neural network-based agents, which results in a complex control structure. These methods can utilize the training process to achieve an optimal solution to the vehicle control problems, i.e., Brüggemann and Possieri (2021); Hegedűs et al. (2021). Moreover, using the resulting complex control structure various performance specifications in the operation of the system can be involved, e.g., Aradi (2022); Kuutti et al. (2021). Nevertheless, it is a current challenge to find control structures with which safe and high-performance operation for systems with neural networks can be achieved Németh and Gáspár (2021a), especially for race vehicles Betz et al. (2022). A solution to this problem has been found by designing a robust control framework for the learning-based agent using a supervisor Németh and Gáspár (2021a).

The work of this paper has followed the path of the latter method, i.e., designing a complex control for achieving high-performance steering capability for an F1TENTH Babu and Behl (2020) type small-scaled indoor test vehicle. The agent is designed through reinforcement learning (RL) with Proximal Policy Optimization (PPO). Moreover, a safe framework is established by the design of a robust controller and a supervisor, whose role is to provide stable motion for the vehicle. The contribution of the paper is an enhanced high-performance control method, which can provide steering angle for achieving minimum lap time regarding the indoor test vehicle. The designed control system was tested in various scenarios in simulations and in a highly complex and noisy real-life environment.

The presentation of the design of the RL-based agent is found in Section 2. Section 3 provides the method for the safe robust framework design. Validation of the designed control system through simulation and test scenarios is found in Section 4. Finally, the summary of the work is found in Section 5.

2 Training of the RL-based vehicle control system

The training process of the RL-based control for achieving a steering system is presented in this section. The goal of the controller is to find the steering angle of the vehicle to complete the racetrack with minimum lap time without leaving the path. In this section, the selection of the most important design parameters is presented.

2.1 Vehicle model and measurements

An efficient and fast learning process requires the use of a simple model on the vehicle that is accurate enough for characterizing control performance specifications. Regarding the steering control design task, the following considerations were made concerning the application and real-life tests.

1. The longitudinal velocity of the vehicle is limited due to the indoor application and, moreover, there is inevitable delay and limited sampling frequency in the real-life positioning, communication, and actuation.
2. Due to the surface roughness of the indoor vinyl floor and tire friction, the cornering stiffness is large enough regarding the minimal cornering radius and the maximal vehicle speed, the resulting slip is small even in the worst-case scenario.

The previous assumptions lead to the formulation of a kinematic bicycle model:

$$\dot{X}_v = v_{\text{Long}} \cos\psi, \tag{1}$$

$$\dot{Y}_v = v_{\text{Long}} \sin\psi, \tag{2}$$

$$\dot{\psi}_v = v_{\text{Long}} \frac{\tan\delta}{l}, \tag{3}$$

where X_v and Y_v are the corresponding vehicle coordinates at the rear wheels of the car, and ψ_v is the yaw angle of the vehicle. Longitudinal velocity v_{Long} is kept constant with a low-level controller, and the controller input is the δ steering angle. The dimensions of the vehicle determine that $l = 0.3$ m.

The measurements, i.e., inputs for the RL-based control agent are determined not only by the vehicle model, but also by the path. Thus, the points of the track centerline on a horizon with N equidistant points ahead of the vehicle must be measured. Although the selection of N for a long horizon can result in highly efficient control intervention due to the large amount of information on the track, but it can overcomplicate the neural network and, consequently, training time is significantly increased. In the case of an F1TENTH vehicle, the selection of $N = 5$ can be enough. The input information on the relative position of the vehicle related to the track is computed as $(X_{vi}, Y_{vi}) = (X_i - X_v, Y_i - Y_v)$, where (X_v, Y_v) are the actual vehicle coordinates, and (X_i, Y_i) are the coordinates of the i^{th} closest checkpoint ahead. The input of the actor neural network based on these is:

$$\begin{bmatrix} X_{v1} & Y_{v1} & \dots & X_{v5} & Y_{v5} & \psi_v & \delta_{t-1} \end{bmatrix}^T, \tag{4}$$

where ψ_v is the actual vehicle orientation and δ_{t-1} is the steering angle from the previous step. Measurement of δ_{t-1} is motivated by preliminary test evaluations, i.e., the rate of change on δ must be limited. If δ changes fast, lateral acceleration can reach extreme values and moreover, it can result in unwanted oscillations in straight track sections.

2.2 Structure selection for actor and critic networks

The applied RL-based process PPO uses two neural networks for training the RL-based controller, i.e., actor and critic network, (Schulman et al., 2017). The actor network takes the agent observation as an input, and it results in the mean value of the control action and its standard deviation as outputs. Thus, the agent action is a probability distribution, which ensures that the agent may act differently, which may lead to a previously undiscovered situation. It can enforce the training process through the exploration of the system operation. Using a normal distribution as the output during training is a conventional practice to promote exploration.

The structure of the actor network can be seen in Fig. 1 (a). The network uses ReLU activation, except on the outputs. The reason for that is hyperbolic tangent is a bounded function, its range is within the interval [−1, 1]. This makes it possible to limit steering intervention, e.g., in the case of the F1TENTH vehicle to $\delta \in [-0.35, 0.35]$ rad. The output activation at the standard deviation branch is a hyperbolic tangent as well. Similarly, it is used to limit the deviation of the control action. There is a scaling layer that ensures that the standard deviation stays in the interval $[0, \sigma_{\max}]$. The parameter $\sigma_{\max}$ is constant, which must be set before the training.

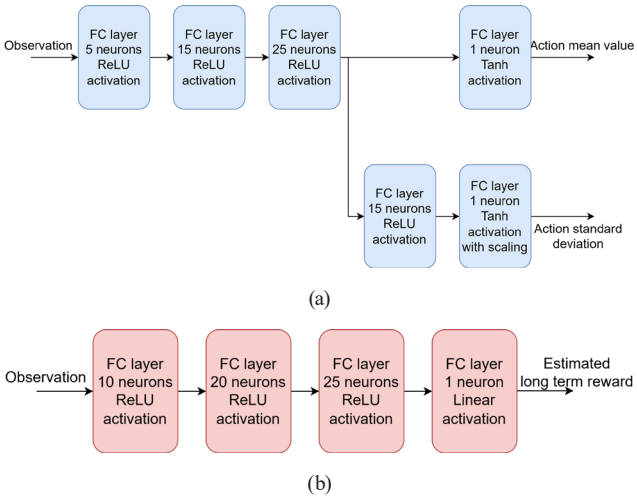


Fig. 1 Structure of the networks in the RL-based control process
(a) actor, (b) critic

Another neural network used during the training is the critic network, whose role is to estimate the goodness of the actor agent. It takes the observation as the input, and it results in the estimated long-term reward as an output. During training, the agent acquires more and more information about the goodness of the specific state, and how much reward can be collected based on a particular observation. The output of the critic network is used to evaluate the performance of the actor agent. If the collected reward after a training episode has a higher value than the estimation of the critic function, then this is considered a better action sequence. Otherwise, it is a worse sequence.

The structure of the used critic network can be seen in Fig. 1 (b). It has a simpler structure than the actor. The reason for it is that its main goal is only to estimate the long-term reward to help evaluate the performance of the actor. The output is a single value that is not necessarily bounded and thus, a linear activation function in the output layer is used.

2.3 Selection of reward function

In the design process of the RL-based agent Proximal Policy Optimization is used Schulman et al. (2017). The reason behind its selection is its fast-training capability, compared to Trust Region Schulman et al. (2015) or Policy Gradient Sutton et al. (1999) methods. The aim of this method is that for training purposes it uses a clipped surrogate objective function, which limits the variation of actions between two steps, i.e., a penalty for having too large policy update is applied. The training process is performed through simulation episodes in which the vehicle must move on given tracks. The simulations implemented in the training process are based on the simple kinematic model of the vehicle from Eqs. (1)–(3), due to the limitation of the velocity in later tests. The tracks are generated through linear and arc segment primitives with predetermined widths.

The role of the reward function is to involve performance specifications in the control design, i.e., it represents the expected controller behavior. In this case, a parametric reward function is selected as:

$$R(s,a) = -Ax_{\text{Lat,err}^2} - B\psi_{\text{err}^2} - C\Delta_\delta^4 + f(s,a), \tag{5}$$

where $x_{\text{Lat,err}}$ denotes lateral path tracking error, ψ_{err} is the orientation error, $\Delta\delta$ is the difference between the actual and previous step steering angles. A, B and C are the corresponding weights and:

$$f(s,a) = \begin{cases} 1, & \text{if the vehicle reached a next checkpoint} \\ -1, & \text{if the vehicle reached a previous checkpoint.} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

This last term rewards the agent if the vehicle moves further on the track and reaches new checkpoints and punishes the agent if it moves in the wrong direction and reaches previously visited checkpoints again.

The checkpoints are designated points on the centerline of the track, the motion along the track and the lateral and orientation errors are estimated based on these points. The driving behavior of the agents is greatly influenced by the weights of reward functions. The most typical example is if one chooses weights A and B large, then the result will be an agent that follows the center of the track accurately. But, if these weights are small compared to $f(s,a)$, then faster progress will be more important and the agent will tend to cut corners aggressively to reduce lap time. The result of a high C value is the reduction of steering angle oscillation. Moreover, if the vehicle leaves the track during the training process, the current scenario is interrupted and a large punishment through reward functions are applied.

Fig. 2 illustrates an example of the selection of tracking-related terms in the reward function. Based on the characteristics of the reward, small deviations from the centerline do not reduce the accumulated reward, but if the lateral error exceeds a threshold (e.g., 0.05 m in both directions in the example) the corresponding reward decreases through a quadratic function. Thus, small tracking errors are not punished and unnecessary steering interventions are avoided. Similar characteristics on the heading angle error are also defined.

2.4 Illustration of the results of the RL-based control design

Finally, the effectiveness of the learning process is illustrated through an illustrative example. Table 1 contains

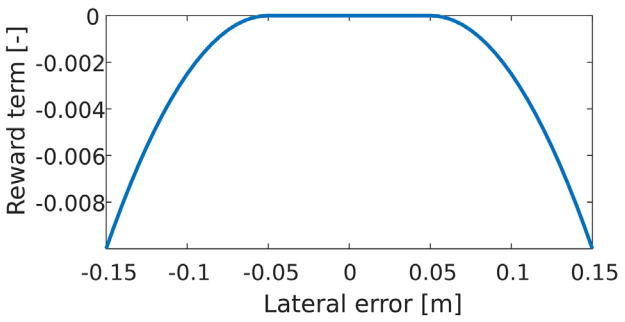


Fig. 2 An example of the lateral error reward term in the reward function

Table 1 Training options in case of a successful PPO training

Hyperparameter	Value
Max episode length	1,000 steps
Episode step time	0.05 s
Plant model integration time	0.005 s
Actor learning rate (η_a)	0.0005
Critic learning rate (η_c)	0.005
Max action standard deviation (σ_{max})	0.4
Experience horizon	2,000
Minibatch size	500
PPO clip factor	0.2
Entropy loss weight	0.1
Number of PPO epochs	4
Discount factor	0.97

the most important training parameters in the applied PPO algorithm.

The collected reward in each training episode (a set of a maximum of 500 consecutive steps in the environment) during the training process can be seen in Fig. 3. Darker blue color shows the moving average, while the lighter blue is the unfiltered reward. A cumulated reward over 30 means that the agent is performing very well, i.e., the vehicle can complete multiple laps if there is enough time given. In the example, the training scenario requires several thousands of episodes to converge, from around 4,000 episodes the agent can complete its task, and the change is small afterward. The episode number needed is typical, most of the time it is within the same order of magnitude.

The operation of the RL-based agent as a steering controller using F1TENTH-based test evaluations is shown in Fig. 4. In the illustration two examples with different reward parameters are found, i.e., prioritizing centerline

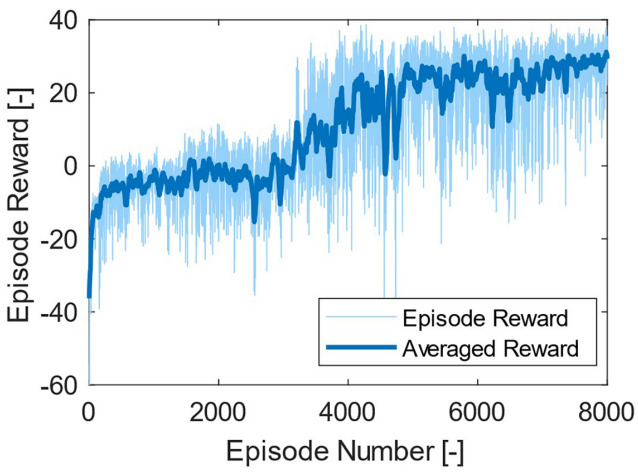
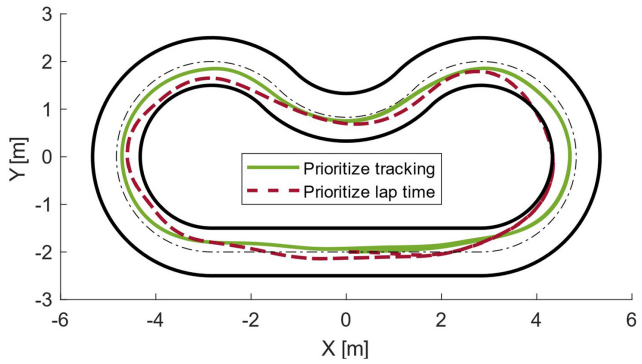
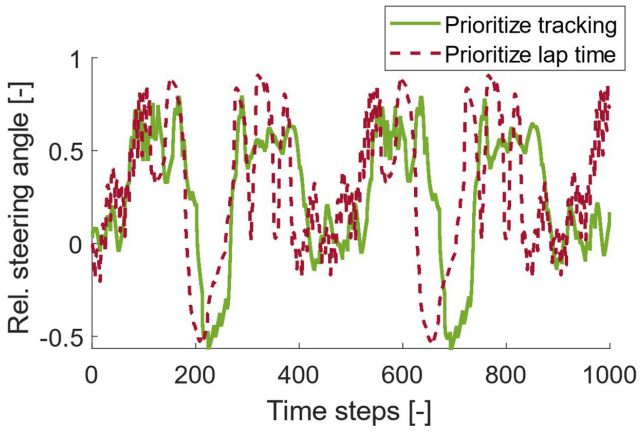


Fig. 3 The rewards during the example training process



(a)



(b)

Fig. 4 Illustration on the operation of the RL-based agent (a) vehicle trajectories, (b) steering interventions

tracking or lap time reduction performances. The controller, trained with a reward function containing large A and B weights, tends to follow the centerline accurately, see green vehicle trajectory in Fig. 4 (a). But if these weights are low, then the term $f(s,a)$ dominates, resulting in faster movement by cutting the corners, see red trajectory in Fig. 4 (a). Fig. 4 (b) shows δ for the two scenarios. Prioritizing lap time results in larger variation in the steering angle and more aggressive turning, see e.g., at time steps 200 and 650.

3 Design of safe control framework for the RL-based control agent

In this section, the RL-based control agent is augmented with a supervisory robust control framework to guarantee stable motion for the vehicle. The structure of the closed-loop system can be seen in Fig. 5. In the structure the role of the supervisor is to provide control input u.

The goal of the safe control framework is to provide u, which is close to u_L, but the stable motion is guaranteed. In this framework stable motion is achieved through robust stability, i.e., the difference between u and u_R is

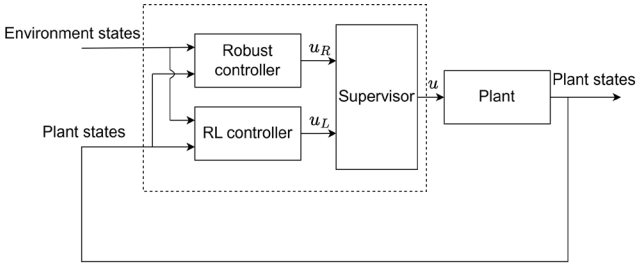


Fig. 5 Block diagram of the complete supervised control system

formed as a disturbance in the system. Therefore, the difference between u and u_R is bounded, and the bound is involved in the robust control design specifications.

The idea of the control operation above is formed as follows. The goal of the supervisor is to minimize the difference between the actual and the learning-based control signal:

$$|u - u_L| \to \min. \tag{7}$$

This criterion through a simplified rule is maintained in which $u = u_L$ in a bounded region defined as below. The control input u of the system is formed as:

$$u = u_R + \Delta_L, \tag{8}$$

where u_R is the output of the robust controller and Δ_L is a bounded additive control input disturbance:

$$\|\Delta_L\| \le \Delta_{\max}. \tag{9}$$

The minimization Eq. (7) can be achieved, when $u = u_L$, i.e., $\Delta_L = u_L - u_R \le \Delta_{\max}$. Otherwise, u ≠ uL, and thus, the difference is higher than 0. In the latter cases, the minimization is maintained through the selection of the closest bound of Δ_L to $\Delta_L = u_L - u_R$, such as:

$$\Delta_L = \begin{cases} u_R - u_L & \text{if} \quad -\Delta_{\max} \le u_L - u_R \le \Delta_{\max}, \\ -\Delta_{\max} & \text{if} \quad \Delta_{\max} > u_L - u_R, \\ \Delta_{\max} & \text{if} \quad \Delta_{\max} < u_L - u_R, \end{cases} \tag{10}$$

which selection rule is illustrated in Fig. 6. Remark that $\Delta_{\max}$ is a design parameter, which significantly influences the operation of the closed-loop. The selection of a larger $\Delta_{\max}$ results in a more conservative robust controller and in the domination of the RL-based controller, while in the case of smaller $\Delta_{\max}$ values the robust control signal will be used most of the time.

The structure of the robust controller for the F1TENTH type vehicle is chosen to be simple due to real-time implementation purposes. An appropriate solution can be achieved with a P controller with two inputs, which is formed as:

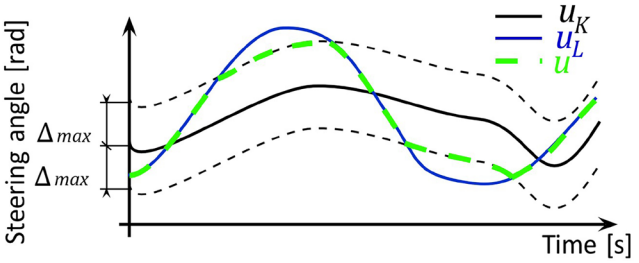


Fig. 6 Illustration of the selection strategy of u

$$u_R = -P_1 x_{\text{Lat,err}} - P_2 \psi_{\text{err}}, \tag{11}$$

and thus, the corresponding steering angle is calculated from path tracking error and heading error. The robust stability of the controller P is evaluated through an analytic way.

3.1 Analysis of the stability of the closed-loop system

In this subsection, the supervisory control system is analyzed from the viewpoint of robust stability. The block diagram of the system for analysis purposes can be seen in Fig. 7.

In Fig. 7 signal $R(s)\Delta(s)$ indicates the difference between u_R and u_L. For the sake of simplicity and consistency, $\|R(s)\|_\infty \leq 1$ is considered, but it does not influence the stability of the control system. In this model $\Delta(s)$ is considered as input-additive uncertainty, whose exact structure is not fixed. However, $\|\Delta(s)\|_\infty \leq \Delta_{\max}$ is considered because the supervisor limits that difference by $\Delta_{\max}$, which is a design parameter. $G(s)$ is the transfer function of the linearized plant from Eqs. (1)–(3), and $P(s)$ denotes the controller, which in this case is a MISO transfer function:

$$P(s) = \begin{bmatrix} P_1 & P_2 \end{bmatrix}, \tag{12}$$

and the negative sign indicates the negative feedback. $Y(s)$ is the output signal of the plant.

The closed-loop transfer function of the above system is formed as:

$$W_{Y,R}(s) = \left(I + G(s)P(s)\right)^{-1} G(s)\Delta(s), \tag{13}$$

and each term in the expression can be further expressed or simplified. $\Delta(s)$ has the bounded-input, bounded-output

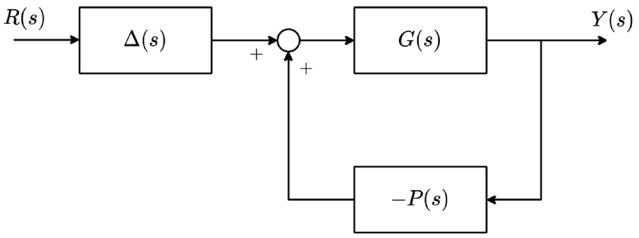


Fig. 7 Block diagram of the controlled system for stability analysis

stability property. Additionally, its output is bounded and, consequently, it does not influence stability. G(s) can be determined by linearizing the vehicle model around a stable trajectory. The lateral dynamics can be approximated with a linearized kinematic bicycle model form the path-tracking errors of which are:

$$\dot{x}_{\text{Lat,err}} = v\psi, \tag{14}$$

$$\dot{\psi}_{\text{err}} = \frac{v}{l}\delta. \tag{15}$$

The output of the plant is the lateral and orientation error and the input is the steering angle. The SIMO transfer function of the plant is:

$$G(s) = \begin{bmatrix} \frac{v^2}{ls^2} \\ \frac{v}{ls} \end{bmatrix}. \tag{16}$$

To analyze stability, the poles of the transfer function Eq. (13) must be calculated by solving:

$$\det\left(I + G(s)P(s)\right) = 0. \tag{17}$$

The poles can be expressed using the determinant and the system parameters as:

$$s_{1,2} = \frac{-P_2 v \pm |v|\sqrt{P_2^2 - 4lP_1}}{2l}, \tag{18}$$

which are stable if l, v, P_1, $P_2 > 0$. l is a physical distance, which is always positive, v is positive in the case of forward driving, and P_1, P_2 are design parameters that must be chosen to be positive to form negative feedback. Consequently, the stable motion of the vehicle within the validity region of the linearized plant can be guaranteed.

The worst-case disturbance amplification of the neural network controller can be characterized by:

$$|W(s)|_\infty = |\Delta(s)|_\infty \cdot |W_{G,P}(s)|_\infty \leq \Delta_{\max} \cdot |W_{G,P}(s)|_\infty, \tag{19}$$

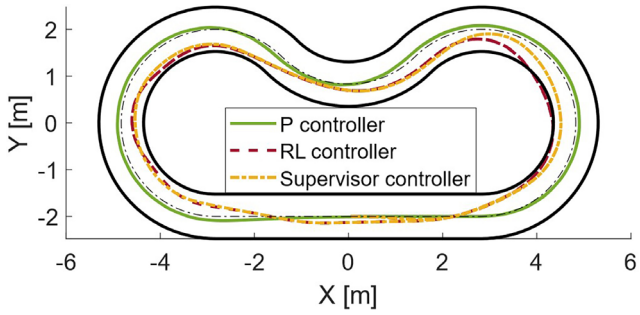
which is an inequality to estimate the disturbance attenuation. $\|W_{G,P}(s)\|_\infty$ can be determined by knowing the vehicle parameters and P_1, P_2. A smaller $\Delta_{\max}$ results in less disturbance, although, in that case, the robust controller is dominant more often. Increasing P decreases the disturbance, but increases the chance of the control output saturation, which is better to avoid.

4 Demonstration of the results

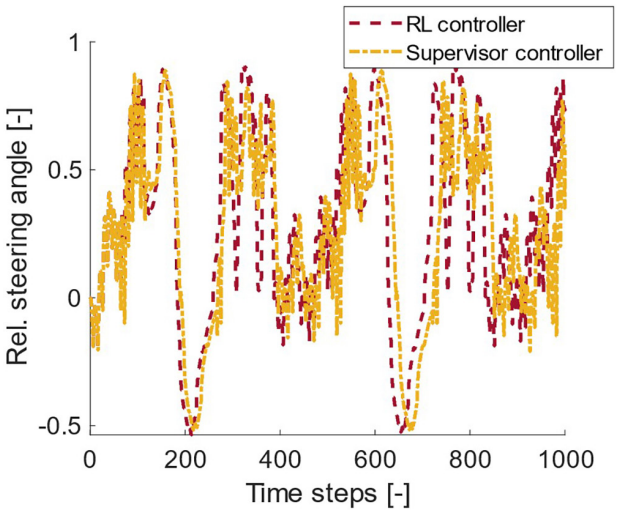
The effectiveness of the proposed control system is demonstrated through simulations and implementations on the F1TENTH small-scaled test vehicle.

4.1 Experiments in simulation

The efficiency of the control method is presented through a simulation example. Fig. 8 (a) shows the trajectories of the vehicle model using different controllers, i.e., individual robust *P* controller, RL-based controller, and the supervised controller with the combination of the individual controllers. The robust control results in accurate tracking of the centerline, while the RL-based controller leads to a motion close to the borders of the path (Fig. 8 (a)). Although it leads to the shortening of the lap time, it can lead to leaving of the path. Thus, through the combination of the control systems, the shortening of the lap time can be preserved, and similarly, the safe motion of the vehicle is also achieved, see the path in Fig. 8 (a). The resulting trajectories are similar, the supervisor tends to follow the output of the RL-based controller and only deviates in unsafe situations. These are typically the sharp turns on the right and the oscillations of the control action on the left.



(a)



(b)

Fig. 8 Resulting trajectories of the different controllers and the comparison of the control signal in the case with and without a supervisor (a) vehicle trajectories, (b) steering interventions

The control action during the testing of the supervisor controller can be seen in Fig. 8 (b). The difference between the two signals is minimal, the supervisor modifies the control action in critical situations only, and most of the time the RL-based controller is dominant. The unsafe RL-based controller is a little faster and thus, the signals start to shift after a time, although, their characteristics are similar during the simulation.

The performances with respect to lap time are summarized in Table 2. The individual *P* controller completes the track in the longest time because it provides reliable and stable path tracking with motion on the centerline. The difference between the RL-based control and the supervised control system is small, with a value of 0.5 s. Thus, the supervisory combined control can result in almost the same performance level as the RL controller, without leaving the track.

4.2 Experiments in small-scaled test vehicles

The effectiveness of the proposed supervised control system was also tested in a small-scaled test vehicle platform. During the evaluations, a camera-based positioning system was used to estimate vehicle position. The communication was carried out using WiFi-network. The communication system of the cameras, the vehicle, and the computers running the corresponding algorithms can be seen in Fig. 9. The blocks show separate devices, and the arrows indicate the direction of the communication, the communicated information, and the used devices. The cameras detect the image of the vehicle and send it to the image processing and positioning algorithm, which determines the position and orientation and sends it to the control algorithm through Robot Operating System (ROS). The control algorithm has feedback as the steering angle, it uses that to limit the rate of change of the steering angle. Finally, the control algorithm sends the control action, the steering angle to the vehicle through ROS as well.

The first experimental test was also carried out with three different controllers. In this example the track is only virtual, its points are predetermined in the same global

Table 2 Lap times in case of different controllers

Controller	Lap time
P controller	28.4 s
RL-based controller	22.2 s
Combined control system	22.7 s

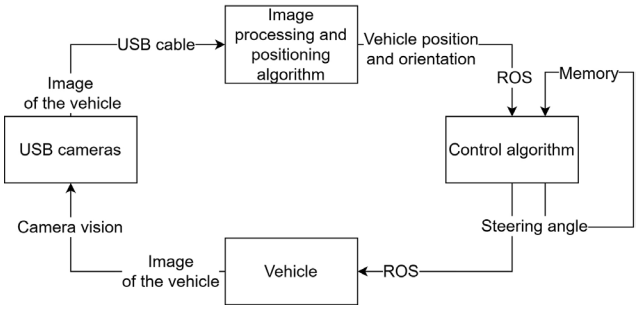


Fig. 9 The communication system used during testing

coordinate system where the vehicle is positioned by the camera-based positioning system. Since both the track and the vehicle is positioned in a global system, their relative position can be easily calculated. The longitudinal velocity of the vehicle was set to 1.5 m/s. The trajectory of the *P* controller is illustrated in Fig. 10 with a green line. This controller follows the centerline also under real circumstances. The RL-based controller is illustrated with a red line. A trajectory has similar characteristics, as it was shown in the simulation example. Especially on the right of the track, the vehicle cuts the corner and leaves the track. Using steering feedback in the measurement vector of the RL-based agent, oscillations in the path is eliminated, i.e., the trajectory is smooth.

The third solution is the supervisor-based controller with a yellow trajectory in Fig. 10. The supervisor modifies the control input of the RL-based controller to guarantee stable motion and to prevent the vehicle from leaving the track. The results are close to the trajectory of the RL-based controller. However, on the right, there is a small, but significant difference in the trajectories. The vehicle controlled by the supervisory controller does not leave the track. On the contrary, the vehicle with the RL-based controller leaves it almost every time on the right. This underlines the effectiveness of the supervisor controller in the real application as well.

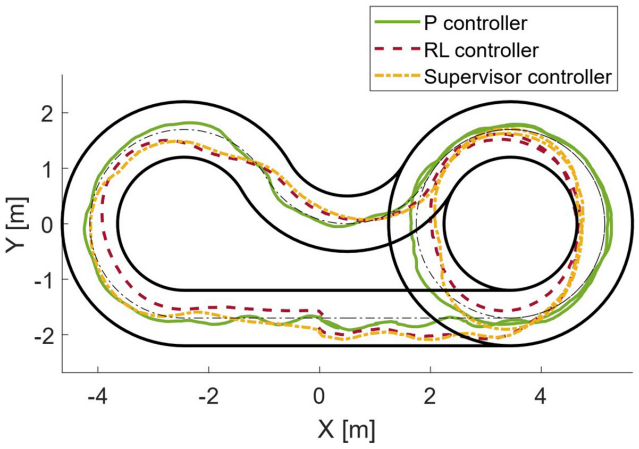
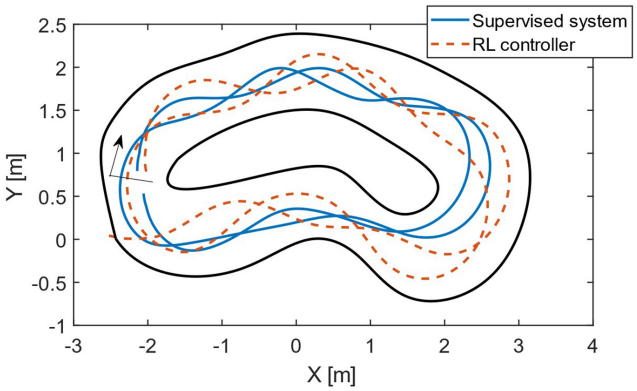


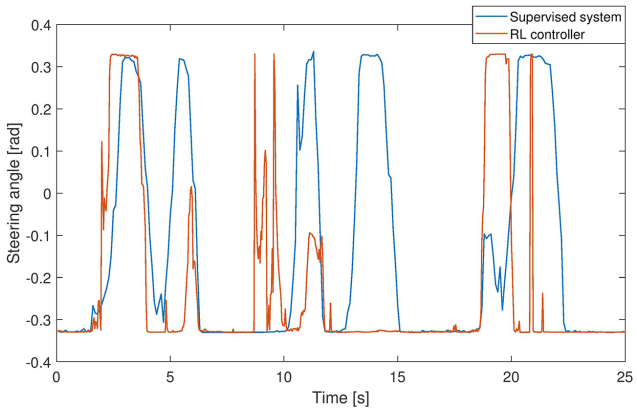
Fig. 10 Test scenario I - trajectories of the vehicles

Second test scenario, in which the control systems are compared, can be found in Fig. 11. In this example a significant difference compared to the previous test is that the track is not virtual, it is determined by traffic cones, and the onboard LiDAR sensor of the vehicle was used to calculate the relative coordinates of the points of the track.

Without the supervisor, the RL-based controller is only able to navigate one full lap and fails on the second by leaving the track. It can be avoided through lateral error prediction of the supervisor with which the vehicle is able to complete both laps safely (Fig. 11 (a)). It resulted from the differences in the steering interventions, i.e., at time 21 s the RL-based controller provides a sharp and intensive steering actuation, which leads to the leaving the track (Fig. 11 (b)). The operation of the supervised control results in a 9.5% increase of lap time, but in this case leaving the track is avoided.



(a)



(b)

Fig. 11 Test scenario II – (a) vehicle trajectories, (b) steering interventions

5 Conclusions

The paper proposes a safe robust framework for data-aided motion control of autonomous indoor vehicles. Using simulations and implementation on a small-scaled test vehicle, the effectiveness of the control system is illustrated. In the framework, the robust controller and the reinforcement learning-based control agent are designed independently, but a supervisory algorithm guarantees the safe and efficient operation of the closed-loop. The illustrations show through the presented examples that potentially unsafe control interventions of the RL agent can be avoided through the supervisory structure. Nevertheless, the high-performance operation of the RL-based control agent is only slightly reduced.

Acknowledgement

The research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems (RRF-2.3.1-21-2022-00002). The research was partially supported by the National Research, Development and Innovation Office (NKFIH) through the project 'Design of high performance safe autonomous vehicle systems via integrated robust control and learning-based methods' (2021-1.2.4-TÉT-2022-00065).

The work of Attila Lelkó was supported by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the EKÖP-24-3 funding scheme.

References

Aradi, S. (2022) "Survey of deep reinforcement learning for motion planning of autonomous vehicles", IEEE Transactions on Intelligent Transportation Systems, 23(2), pp. 740–759. https://doi.org/10.1109/TITS.2020.3024655

Babu, V. S., Behl, M. (2020) "f1tenth.dev – An open-source ROS based F1/10 autonomous racing simulator", In: 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), Hong Kong, China, pp. 1614–1620. ISBN 978-1-7281-6904-0 https://doi.org/10.1109/CASE48305.2020.9216949

Bao, Y., Velni, J. M. (2022) "Safe control of nonlinear systems in LPV framework using model-based reinforcement learning", International Journal of Control, 96(4), pp. 1079–1090. https://doi.org/10.1080/00207179.2022.2029945

Betz, J., Zheng, H., Liniger, A., Rosolia, U., Karle, P., Behl, M., Krovi, V., Mangharam, R. (2022) "Autonomous vehicles on the edge: A survey on autonomous vehicle racing", IEEE Open Journal of Intelligent Transportation Systems, 3, pp. 458–488. https://doi.org/10.1109/OJITS.2022.3181510

Brüggemann, S., Possieri, C. (2021) "On the use of difference of log-sum-exp neural networks to solve data-driven model predictive control tracking problems", IEEE Control Systems Letters, 5(4), pp. 1267–1272. https://doi.org/10.1109/LCSYS.2020.3032083

Chen, Y., Zhang, X., Wang, J. (2021) "Robust vehicle driver assistance control for handover scenarios considering driving performances", IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(7), pp. 4160–4170. https://doi.org/10.1109/TSMC.2019.2931484

Fényes, D., Hegedűs, T., Németh, B., Szabó, Z., Gáspár, P. (2022) "Robust control design using ultra-local model-based approach for vehicle-oriented control problems", In: 2022 European Control Conference (ECC), London, United Kingdom, pp. 1746–1751. ISBN 978-3-9071-4407-7 https://doi.org/10.23919/ECC55457.2022.9838107

Fliess, M., Join, C. (2021) "Machine learning and control engineering: The model-free case", Springer International Publishing, Cham, 1288, pp. 258–278. https://doi.org/10.1007/978-3-030-63128-4_20

Hegedűs, T., Fényes, D., Németh, B., Gáspár, P. (2021) "Improving sustainable safe transport via automated vehicle control with closed-loop matching", Sustainability, 13(20), 11264. https://doi.org/10.3390/su132011264

Kabzan, J., Hewing, L., Liniger, A., Zeilinger, M. N. (2019) "Learning-based model predictive control for autonomous racing", IEEE Robotics and Automation Letters, 4(4), pp. 3363–3370. https://doi.org/10.1109/LRA.2019.2926677

Khosravani, S., Khajepour, A., Fidan, B., Chen, S.-K., Litkouhi, B. (2014) "Development of a robust vehicle control with driver in the loop", In: 2014 American Control Conference, Portland, OR, USA, pp. 3482–3487. ISBN 978-1-4799-3274-0 https://doi.org/10.1109/ACC.2014.6858845

Kuutti, S., Bowden, R., Jin, Y., Barber, P., Fallah, S. (2021) "A survey of deep learning applications to autonomous vehicle control", IEEE Transactions on Intelligent Transportation Systems, 22(2), pp. 712–733. https://doi.org/10.1109/TITS.2019.2962338

McKinnon, C. D., Schoellig, A. P. (2019) "Learn fast, forget slow: Safe predictive learning control for systems with unknown and changing dynamics performing repetitive tasks", IEEE Robotics and Automation Letters, 4(2), pp. 2180–2187. https://doi.org/10.1109/LRA.2019.2901638

Nageshrao, S., Rahman, Y., Ivanovic, V., Jankovic, M., Tseng, E., Hafner, M., Filev, D. (2023) "Robust AI driving strategy for autonomous vehicles", In: Murphey, Y. L., Kolmanovsky, I., Watta, P. (eds) AI-enabled Technologies for Autonomous and Connected Vehicles, Springer International Publishing, Cham, pp. 161–212. ISBN 978-3-031-06780-8 https://doi.org/10.1007/978-3-031-06780-8_7

Németh, B., Gáspár, P. (2021a) "Guaranteed performances for learning-based control systems using robust control theory", Springer International Publishing, Cham, 984, pp. 109–142. https://doi.org/10.1007/978-3-030-77939-9_4

Németh, B., Gáspár, P. (2021b) "Ensuring performance requirements for semiactive suspension with nonconventional control systems via robust linear parameter varying framework", International Journal of Robust and Nonlinear Control, 31(17), pp. 8165–8182. https://doi.org/10.1002/rnc.5282

Rosolia, U., Borrelli, F. (2020) "Learning how to autonomously race a car: A predictive control approach", IEEE Transactions on Control Systems Technology, 28(6), pp. 2713–2719. https://doi.org/10.1109/TCST.2019.2948135

Schulman, J., Levine, S., Moritz, P., Jordan, M. I., Abbeel, P. (2015) "Trust region policy optimization", Proceedings of the 32nd International Conference on Machine Learning, 37, pp. 1889–1897 https://doi.org/10.48550/arXiv.1502.05477

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O. (2017) "Proximal policy optimization algorithms", arXiv: Learning, pp. 1–12. https://doi.org/10.48550/arXiv.1707.06347

Sename, O. (2021) "Review on LPV approaches for suspension systems", Electronics, 10(17), 2120. https://doi.org/10.3390/electronics10172120

Shao, K., Zheng, J., Huang, K., Qiu, M., Sun, Z. (2021) "Robust model referenced control for vehicle rollover prevention with time-varying speed", International Journal of Vehicle Design, 85(1), pp. 48–68. https://doi.org/10.1504/IJVD.2021.117154

Sutton, R. S., McAllester, D. A., Singh, S. P., Mansour, Y. (1999) "Policy gradient methods for reinforcement learning with function approximation", MIT Press, 12, pp. 1057–1063.

Varga, B., Kulcsár, B., Chehreghani, M. H. (2023) "Deep Q-learning: A robust control approach", International Journal of Robust and Nonlinear Control, 33(1), pp. 526–544. https://doi.org/10.1002/rnc.6457