Preprints, 10th IFAC International Symposium on
Advances in Automotive Control
Columbus, Ohio, USA, August 28-30, 2022

Vehicle Control with Cloud-aided Learning Feature: an Implementation on Indoor Platform

Balázs Németh *, **Zoltán Antal** *, **Attila Csaba Marosi** *, **Róbert Lovas** *, **Máté Fazekas** *, **Péter Gáspár** *

* *Institute for Computer Science and Control (SZTAKI) Eötvös Loránd Research Network (ELKH), Kende u. 13-17, H-1111 Budapest, Hungary. E-mail: [balazs.nemeth;zoltan.antal;atisu;robert.lovas;mate.fazekas;peter.gaspar]@sztaki.hu*

Abstract: Safe motion together with improved economy and traveling performance levels are important requirements against automated vehicles. Thus, the design of enhanced control systems is requested, which contain conventional model-based controllers and the use of unconventional approaches, e.g., learning features and cloud-based methods. This paper proposes a hierarchical vehicle control design method with learning functions, which incorporates control in two levels, such as in cloud level and in vehicle level. The control on the cloud level is designed by using reinforcement learning, with which the maximum speed for the vehicle is achieved. The vehicle level contains a robust controller and a supervisor, with which the collision avoidance of the vehicle is guaranteed. The hierarchical control guarantees performance requirement of safe motion, i.e., collision avoidance in all scenarios, even if the connection with the cloud is lost. The proposed control on indoor Hardware-in-the-Loop platform is implemented. The effectiveness of the control and the safe motion of the vehicle under various scenarios with and without cloud connection are demonstrated.

Keywords: automated vehicles, vehicle-cloud connection, learning features, roundabouts

1. INTRODUCTION AND MOTIVATION

Increasing number of vehicles poses the challenge of their safe motion under varying environmental conditions. Moreover, the improvement of effectiveness requires the fast motion of the vehicles, which can be facilitated through their coordination. These challenges motivate the use of cloud technologies, with which the high amount of information can be processed, stored and used for control purposes.

A recent hot topic of vehicle and mobile robot control is cloud-aided learning. There are high number of cloud solutions, which can vary on tools and technologies used in the build-up of such systems (Dawarka and Bekaroo [2022]). The aim of clouds for vehicles and mobile robots is to use a centralized server for performing high complexity computation process, whose realization requires the optimization of data transfer on the network (Chinchali et al. [2021]). One of the purpose of learning feature on the cloud is to provide optimal resource allocation, e.g., Liu et al. [2018] proposes a resource allocation scheme based on reinforcement learning (RL). A further solution is to utilize the computational capability of the mobile agent, and simultaneously, to consider the latency and CPU availability, which through a RL-based deep Q-network can be achieved (Penmetcha and Min [2021]). The problem of best quality service against stochastic communication delays and task deadline in automotive context can also be found (Li et al. [2018]).

The topic of this paper is related to the control application type of cloud-aided learning, i.e., to improve performances of vehicles through learning-based features. Especially, it focuses on the motion control of vehicles on an indoor platform, which must result in collision-free and time-saving motion profile. Although learning-based approaches have high efficiency to solve these control tasks (He et al. [2021], Aradi [2020]), a crucial problem is the verification of the resulted control agent from the aspect of safety performance requirements (Hewing et al. [2020], Tran et al. [2020]). Therefore, in the last years several papers in the topic of safe learning have been proposed, e.g. using model predictive framework (Muntwiler et al. [2020], Rosolia and Borrelli [2018]), using Hamilton-Jacobi reachability methods (Fisac et al. [2019]) or using Satisfiability Modulo Theory (Huang et al. [2018]).

This paper proposes the implementation of a novel cloud-aided safe learning method for vehicles, focusing on the

[1] The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems National Laboratory Program. The presented work of R. Lovas was also supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. On behalf of Project ARNL we thank for the usage of ELKH Cloud (https://science-cloud.hu/) that significantly helped us achieving the results published in this paper.

implementation of the method for indoor vehicles. The predefined safety specifications through the architecture of the robust control system are guaranteed. The control of the vehicle in a hierarchical architecture, i.e., vehicle level and cloud level, is designed. The aim of the cloud-level is to achieve enhanced control performances using the high computation capacity of the cloud. Thus, reinforcement learning on the cloud level for achieving maximum speed of the vehicle is implemented. Moreover, on the vehicle level the safety requirement, i.e., collision avoidance, are guaranteed. The advantage of the solution is that the safe performance specifications even at the degradation of the communication in the network can be guaranteed.

The paper is organized as follows. The hierarchical control structure with the design on the vehicle level and on the cloud level is presented in Section 2. The focus of Section 3 is the implementation of the presented control design on indoor vehicles. Thus, Section 3 proposes the Hardware-in-the-Loop (HiL) environment with augmented reality feature, the architecture and the orchestration of the cloud and some results of roundabout scenarios of vehicles. Finally, the paper in Section 4 is concluded.

2. HIERARCHICAL DESIGN FOR MOTION CONTROL OF INDOOR VEHICLES

In this section the hierarchical control design for indoor vehicles is proposed. First, the architecture of the control is presented. Second, the design of the road level control and third, the learning feature on the cloud level are proposed.

The architecture of the hierarchical control with each levels in Figure 1 is illustrated. The goal of the control is to provide single motion input $u(k)$ for a given individual vehicle, i.e., longitudinal acceleration command $a_1(k)$, with which the vehicle moves along its route. $u(k)$ is computed by the supervisor, such as $u(k) = u_K(k)+\Delta(k)$, where $u_K(k)$ is the output of the robust controller on the vehicle level. $\Delta(k) \in \mathbf{\Delta}$ is an additional term of the control input and $\mathbf{\Delta}$ is the finite domain of $\Delta(k)$. In the control architecture, $u_L(k)$ is a candidate control input, which is suggested by the RL-based controller. The value of $\Delta(k)$ is computed to minimize the difference $(u(k) - u_L(k))^2$, and simultaneously, to avoid the collision of the vehicle with another vehicle.

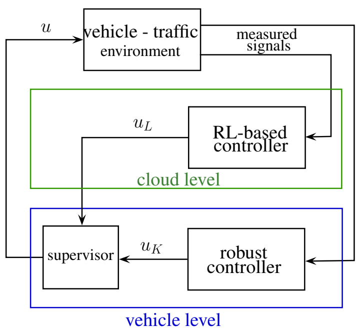


Fig. 1. Illustration of the control architecture

It is necessary to distinguish the roles of the RL-based and the robust controllers. The robust controller is designed to provide $u_K(k)$, with which the collision avoidance for all scenarios is guaranteed, i.e., $u_K(k), \Delta(k)$ pairs for all scenarios can be found. It requires the incorporation of the domain $\mathbf{\Delta}$ in the design of the robust controller, which is interpreted as an input uncertainty domain in the context of the robust control, see Németh and Gáspár [2021]. The role of the RL-based controller is to provide $u_L(k)$ candidate control input, with which further non-safety performance requirements can be considered, e.g. maximization of the vehicle speed or the limitation of the lateral acceleration to protect delivered goods. Since the RL-based controller through a training process with high number of varying episodes is resulted, $u_L(k)$ can be unable to guarantee collision avoidance for all scenarios in itself.

2.1 Control design on the vehicle level

The goal of the control on the vehicle level is to provide $u(k)$, with which collision avoidance can be guaranteed, and the suggestion of the cloud, i.e., $u_L(k)$ is considered as much as possible. Since the design of the robust control in a preliminary work (Németh and Gáspár [2021]) is available, this subsection focuses on the formulation of the strategy in the supervisor.

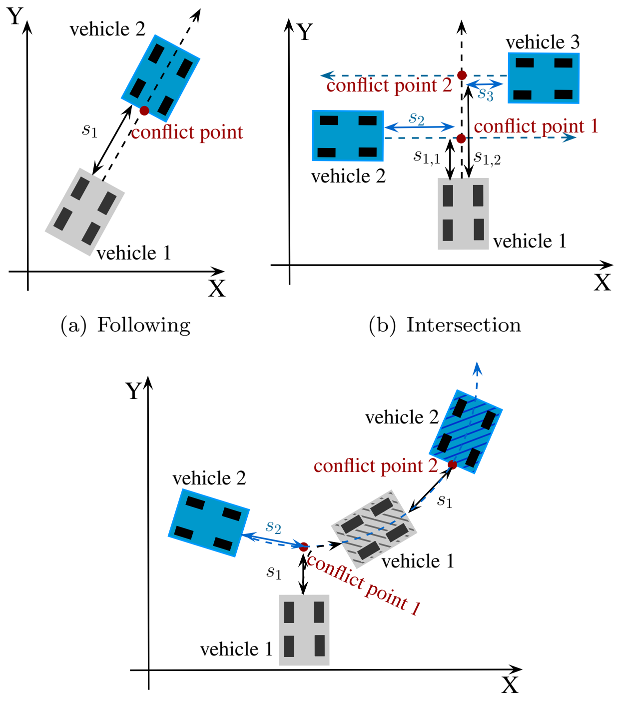
The goal of the supervisor is to provide $\Delta(k)$, with which the collision avoidance of the vehicles can be guaranteed. Figure 2(b) illustrates the planar motion of vehicles on the ground, whose routes are intersected. The intersection of the routes is defined as the conflict point. Considering point-mass model of vehicles, the control problem is independent from the direction of vehicle motion. Therefore, avoiding vehicle collision problem can be transformed to the problem of finding trajectory outside of a circle (Németh and Gáspár [2021]). In the control system, the supervisor must provide $\Delta(k)$ additional term for $u(k) = u_K(k)+\Delta(k)$, with which the following constraint is guaranteed:

$$s_1(k+1, u(k))^2 + s_2(k+1)^2 \geq s_{safe}^2, \quad (1)$$

where s_{safe} is a predefined safety distance, which must be kept during the motion of the vehicles. In the selection of s_{safe} the physical sizes of the vehicles must be considered, which is able to compensate the insufficiency of the point-mass model. Distance $s_2(k)$ and the speed of the uncontrolled vehicle are considered to be measured (e.g., through an indoor motion capture system) and a prediction through $s_2(k+1) = s_2(k) + v_2(k)T$ is given, where T is the time step between k and $k+1$. Thus, $s_1(k+1, u(k)), s_2(k+1)$ mean distances from the conflict point at the next time step. The motion of the controlled vehicle is formed as

$$s_1(k+1) = s_1(k) + v_1(k)T + a_1(k)\frac{T^2}{2}, \quad (2)$$

where $a_1(k) = u(k)$, $v_1(k)$ is longitudinal velocity. The formulation of the constraint can be provided for various types of vehicle interactions, see Figure 2. Following a preceding vehicle can be handled that actual position of the preceding vehicle is the critical point, i.e., $s_2 \equiv 0$ and $s_1(k)$ is the distance between them (Figure 2(a)). Thus, (1) is turned to $s_1(k+1, u(k)) \geq s_{safe}$. Intersection scenarios (Figure 2(b)) with n number of vehicles through multiple constraints for $s_{1,i}, s_i(k)$, $i \in [1; n]$. In this case the conflict points can be defined as the middles of the intersections. Furthermore, roundabout is a complex scenario with intersections and vehicle following tasks, see Figure 2(c)). Safe motion in roundabout requires the modification of the conflict point during the motion of the vehicle. Thus, guaranteeing safe motion of a vehicle requires a function, which selects the current conflict point in the region of interest concerning the surrounding vehicles. From these information $s_i(k)$ values are computed for (1).



(a) Following

(b) Intersection

(c) Roundabout

Fig. 2. Typical vehicle interactions

The output of the supervisor is $u(k)$, which is recommended to be close to the candidate control input $u_L(k)$ and similarly, the safety constraints (1) for n number of vehicles must be satisfied. It leads to a quadratic optimization task:

$$\min_{\Delta(k)} \Big(u(k) - u_L(k)\Big)^2 \tag{3a}$$

subject to

$$s_{1,i}(k+1, u(k))^2 + s_i(k+1)^2 \geq s_{safe}^2, \forall i \in [1, n], \tag{3b}$$

$$\Delta \in \boldsymbol{\Delta}. \tag{3c}$$

The formed constrains (3b) for avoiding collision has quadratic form, which can pose the requirement of increased computation effort. Therefore, it can be beneficial to reformulate (3b) to disjunctive linear inequalities (Belotti et al. [2011]), see its method in Németh and Gáspár [2021].

The solution of the optimization problem (3) with the linearization of the constraint (3b) can lead to appropriate longitudinal acceleration of the automated vehicle. Nevertheless, in case of some special traffic scenarios, e.g., a surrounding vehicle unexpectedly moves inside of s_{safe} distance, the optimization problem (3) do not have solution. Its reason is that $[s_1(k), s_2(k)]$ is inside of the circle and thus, tangent lines cannot be found. In these scenarios for safety reasons, the minimum of $\Delta(k)$ must be selected, which leads to the stopping of the automated vehicle.

2.2 Learning feature on the cloud level

In this section achieving of learning function on the cloud-level is presented. The goal of the learning is to improve non-safety performance level of the controlled system, thus the speed of the vehicle along its route is maximized. The training of the RL-based controller on the environment, which contains the vehicle model and the vehicle level control, through deep deterministic policy gradient method (DDPG) is performed.

In case of the learning process, the environment for reinforcement learning involves the vehicle-traffic environment, the robust controller and the supervisor, see Figure 1. The input, i.e., the observation of the RL-based controller is represented by $S(k)$, the output of the controller is $u_L(k)$. The observation contains measured signals on the vehicle ($v_1(k)$, $s_1(k)$) and on the surrounding n_s number of vehicles ($v_j(k)$, $s_j(k)$, $j \in [1; n_s]$), where in this paper $n_s = 3$ is selected. The output of the controller is selected to be limited, i.e., $u_L(k) \in [u_{L,min}, u_{L,max}]$, where the limits express the physically feasible minimum and maximum acceleration commands.

A selected DDPG method for learning results in an actor-critic RL-based controller that computes an optimal policy for maximizing long-term value of reward function $r(k)$ (Lillicrap et al. [2016]). In the method actor and critic approximators are used. Both approximators use the observations, which are represented by S. The purpose of the actor approximator $\mu(S)$ is to find action A with $u_L(k)$, which maximizes the long-term future reward. The role of critic $\mathcal{Q}(S, A)$ is to find the expected value of the long-term future reward for the task.

The reward function in the learning process of the paper contains two terms, such as longitudinal speed of the vehicle and difference between $u(k)$, $u_L(k)$:

$$r(k) = v_1(k) - Q(u(k) - u_L(k))^2, \tag{4}$$

where Q is a positive scalar weighting factor, whose role is to provide a balance between the terms in $r(k)$. The reward can be interpreted as follows. $v_1(k)$ in $r(k)$ represents the performance requirement of increasing vehicle speed along its route. If $r(k) = v_1(k)$ is selected as a reward function, the solution for $u_L(k)$ is trivial, i.e., $u_L(k) = u_{L,max}$ for all k. It results in that in this case $u_L(k)$ along the route of the vehicle is often overwritten. Its reduction requires the consideration of the term $(u(k) - u_L(k))^2$ in $r(k)$ and Q sets the balance between the two terms. If Q is selected for a high value, $(u(k) - u_L(k))^2$ has increased importance during the training process. In this case $u_L(k)$ is able to approximate $u(k)$, which leads to the reduction of the speed of the vehicle. Therefore, it is requested to find a balance between the two terms, which can be achieved through the appropriate selection of Q.

The training can require performing high episode number. Since the training process requests the environment, containing with the vehicle, the learning can be expensive or unfeasible in practice. Instead of the real physical system it is recommended to use digital twin of the vehicle, with which the costs of the training process can significantly reduce. Since in the given example of this paper, the vehicle moves with limited speed in an indoor environment, digital twin of vehicle motion through kinematic representation is created, see (2).

3. IMPLEMENTATION OF THE MOTION CONTROL ALGORITHM

The goal of this section is to propose the implementation of the motion control algorithm. First the HiL environment with the visualization of the augmented reality is presented. Since cloud has high importance in the learning issue, its architecture and orchestration are detailed. Finally, effectiveness of the controlled system on some examples are demonstrated.

3.1 HiL environment with augmented reality

The scheme of the HiL architecture is illustrated in Figure 3. The communication between the elements through their connection to Robot Operating System (ROS) is realized. The vehicles, the motion capture system, the cloud and the computer, the tablet as nodes into the ROS network are connected.

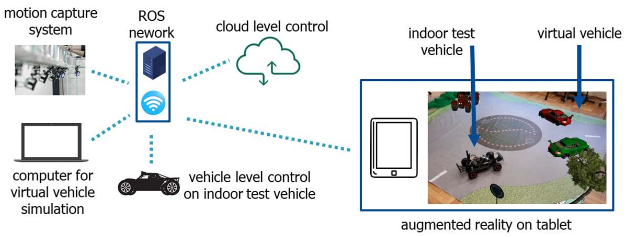


Fig. 3. Scheme of the HiL architecture

The role of the OptiTrack motion capture system is to measure the position and the speed of the vehicles, which to the ROS server are transmitted. The system contains 6 cameras, the motion of the vehicles through passive markers are calculated and the error of the positioning is below $\pm 0.2mm$.

In the given setup F1TENTH type of 1/10 sized wheeled RC vehicles are used. The vehicle contains Jetson Xavier NX computer and various sensors, i.e., camera, LiDAR, IMU etc. In this paper the signals of these sensors are not used for control purposes, but in the cloud are stored. The candidate control input $u_L(k)$ and the position of surrounding virtual and physical vehicles for the controlled vehicle through Wi-Fi on the ROS network are transmitted. Note that lateral steering control of the vehicle also on this level is implemented. The applied PID-based steering controller uses the difference of vehicle position and of route centerline as inputs, and thus, computes front wheel steering angle.

The motions of the virtual vehicles on a computer using Matlab based on the vehicle model (2) are simulated. The advantage of virtual vehicles is that high number of vehicles without their expensive physical realization in learning and evaluation process can be incorporated. The augmented reality, with which the virtual vehicles can be visualized, is implemented on a tablet with Android. For the visualization an application based on Vuforia engine in Unity is developed. The operation requires a marker (e.g. on the floor) for the positioning of the tablet, and position, orientation information on the virtual vehicles for the tablet are transmitted.

Finally, the cloud, containing the cloud level of the control architecture, to the ROS network is also connected. For achieving RL-based controller, in this paper the Reinforcement Learning Toolbox of Matlab is used (Mat [2020]).

3.2 Architecture and orchestration of the cloud

In the framework of this research the Cloud and Big Data-based Research Platform are being developed on the ELKH Cloud, which is an OpenStack based Infrastructure-as-a-Service science cloud, representing a pool of scalable computing, storage and network resources. The connection between the local environment and the cloud-based one is provided via secure VPN connection ("4. VPN Host"). The Reinforcement Learning Toolbox of Matlab is running on a dedicated virtual machine (VM) due to performance considerations. Additionally, two other components are present. First, an ingestion component ("7. ROS Ingestion") is running that acts as a bridge for ingesting metrics from the ROS network into the Research Platform for further analysis. Second, a Python-based RL agent is being evaluated as an alternative of the Matlab. The architecture consists of several other components (e.g., hosts depicted as "Platform-A" to "Platform-E", network components such as routers and external network connections shown in Figure 4).

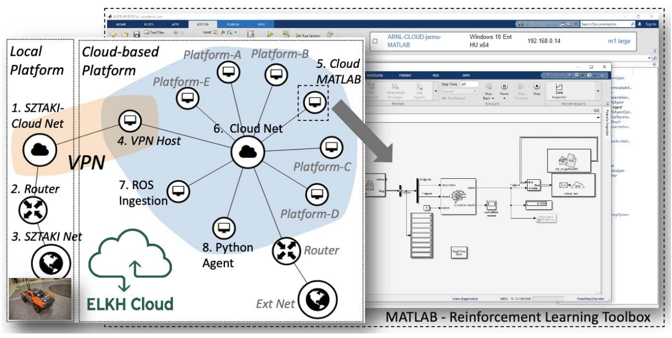


Fig. 4. High-level architecture presenting the local to cloud connection and components

Figure 5 provides a data-flow and platform oriented perspective, where *HiL Cloud Level Ctrl.* group contains the HiL cloud components. The remaining groups are part of the core platform, which is designed to support multiple use cases. It is loosely based on the Lambda-architecture pattern (Marz [2011]), open source, software container-based, and builds on previous experience building IoT data platforms (Lovas et al. [2018]). Its *Staging & Data Lake* stage is responsible for data ingestion via standardized interfaces. Currently, MQTT and native Kafka for streaming data, Amazon S3 for non-structured or blob data, and ODBC for relational data are supported. In our experience MQTT is common ground for IoT devices and similarly, S3 is for blob data. For the HiL environment a ROS to MQTT bridge acts as the only data ingestion component. However, we are planning to connect other data sources such as *rosbags*. Additionally, the first stage acts as a staging area for all incoming data. Here data is retained until explicitly removed, or a default expiry is reached. The *(Pre-)Processing* stage is responsible for data transformation. Here Apache NiFi serves as the "glue" between stages, it is responsible for the data-flow, and also for lightweight transformations. For more complex processing tasks Kafka Streams tasks are also available for streaming data, and custom tasks for batch computation. All tasks are executed as software-containers via Docker and Kubernetes. The *Data Warehouse* stage is responsible for storing and serving data. We rely on TimescaleDB for storing time-series data. It is an extension on top of PostgreSQL, which is in turn used for storing relational data and data warehousing functionality overall. The *Business Intelligence (BI)* stage is responsible for providing dashboard and visualization capabilities. Currently, Apache Superset has been selected, although other BI tools (such as Power BI) can be connected via an ODBC interface. Additionally, ad-hoc queries and self-service data exploration are also supported either via Superset or via Jupyter notebooks. All platform components are monitored using the open source Prometheus monitoring system and Grafana for monitoring dashboards.

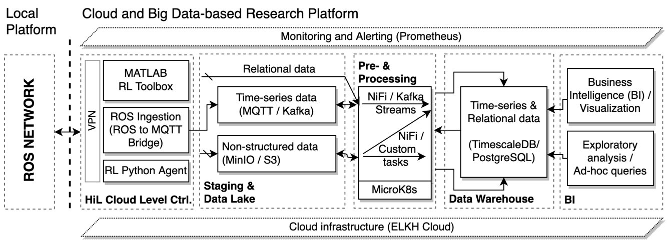


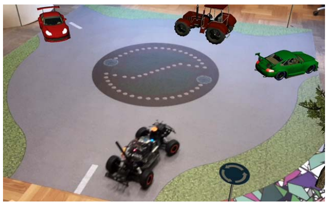
Fig. 5. Architecture of the Cloud and Big Data-based Research Platform: Core and HiL components

Finally, manually deploying and maintaining complex container-based micro-services and computation tasks are not feasible at scale, thus, it is a further challenge to use the application-level cloud orchestrator MiCADO (Kiss et al. [2019]), and release the final version after testing and benchmarking as a reference architecture (Nagy et al. [2021]).

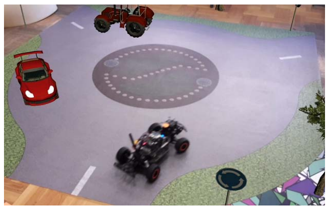
3.3 Demonstration of the control operation

Finally, the effectiveness of the controlled system on the examples of roundabout scenarios with virtual and real four-wheeled vehicles, see Figure 6. The goals of the examples are to show (i) the impact of Q on the result of the learning and (ii) the safe motion of the vehicle, even if the cloud connection has been lost.

In the first scenario the vehicle enters into roundabout at the entrance on the bottom (Figure 6(a),(b)), takes a round and exits also on the bottom of the roundabout. During its route three virtual vehicles also move in the roundabout. In this scenario the RL-based control is designed to have high priority on speed maximization, i.e., $Q = 7.5$ is selected. Moreover, two cases in this scenario are compared, such as with and without cloud connection.



(a) With cloud connection (Scen. 1)

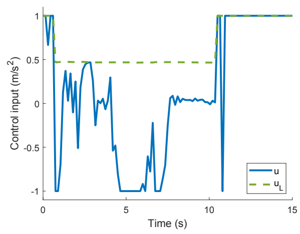


(b) W/o cloud connection (Scen. 1)

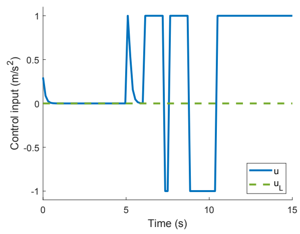
(c) With cloud connection (Scen. 2)

(d) W/o cloud connection (Scen. 2)

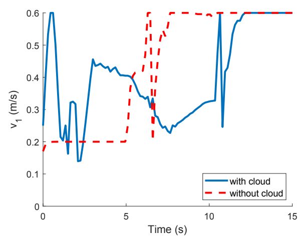
Fig. 6. Visualization of the scenarios



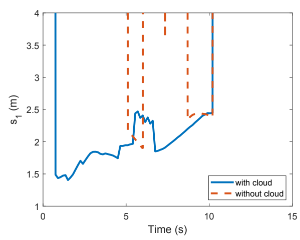
(a) With cloud connection



(b) Without cloud connection



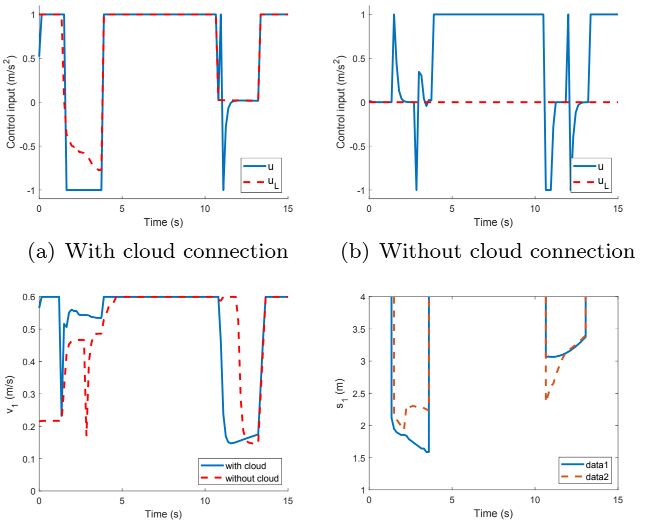
(c) Speed of the vehicle



(d) Distance from the conflict point

Fig. 7. Results on Scenario 1

Figure 6(a),(b) show the moment when the vehicle decides to enter into the roundabout. In the case of cloud connection (Figure 6(a)) the vehicle earlier enters into the roundabout, as in the case without cloud connection (Figure 6(b)). Its reason is that the RL-based control facilitates the increase of speed (Figure 7(a)), but if the connection has been lost, $u_L = 0$ during the scenario (Figure 7(b)). It results in difference in the speed profile, see Figure 7(c). Distances between the vehicle and the actual conflict point in each scenario are found in Figure 7(d). It illustrates that the cloud level control facilitates the vehicle to move closer to the actual closer vehicle, but the $s_{safe} = 1m$ in both cases is kept. It is achieved through the modification of u_L (see e.g. Figure 7(a)) to guarantee constraints on collision avoidance. Thus, the safe motion of the vehicle in both cases is guaranteed.



(a) With cloud connection (b) Without cloud connection

(c) Speed of the vehicle (d) Distance from the conflict point

Fig. 8. Results on Scenario 2

In the second scenario the vehicle inters into the same entrance, but exits on the second after it (Figure 6(c),(d)). The RL-based controller with $Q = 12$ is trained, which means that it is requested to reduce the difference between u and u_L. Figure 8(a) illustrates that through the learning process it is successfully achieved. It results in that if the connection to the cloud has been lost, the differences of the cases in u is small. Thus, the speed profile (Figure 8(c)) of both cases are close to each other. It results in almost the similar motion, see Figure 8(d). Thus, through appropriate selection of Q it can be achieved that the motion of the vehicle in the connected and unconnected cases are close to each other. It can have benefits under real operation circumstances. For example, if the connection for a short period during the operation of the vehicle has been lost, it can have reduced impact on the logistic or transport process.

4. CONCLUSIONS

The paper has proposed a hierarchical motion control for vehicles with cloud-aided learning feature. The method using four-wheeled indoor vehicle and using communication with cloud is implemented. The operation of the control under various scenarios is demonstrated. As a contribution, the proposed control strategy is able to guarantee safe motion of the vehicle, i.e., collision avoidance, even if the connection with the cloud is lost. Moreover, the RL-based control on the cloud level is able to increase the speed of the vehicles.

A future challenge of the research is to provide motion control methods for the coordination of high number of vehicles. In the framework of the hierarchical control the coordination on the cloud level can be effectively achieved. Its advantage is that the high computation capability of the cloud is used for the coordination, and thus, increasing computation requirements against the vehicle are not posed.

REFERENCES

Szilárd Aradi. Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Tr. on Int. Transp. Sys.*, pages 1–20, 2020.

Pietro Belotti, Leo Liberti, Andrea Lodi, Giacomo Nannicini, and Andrea Tramontani. *Disjunctive Inequalities: Applications And Extensions*. American Cancer Society, 2011.

S. Chinchali, A. Sharma, J. Harrison, Amine Elhafsi, Daniel Kang, Evgenya Pergament, Eyal Cidon, Sachin Katti, and Marco Pavone. Network offloading policies for cloud robotics: a learning-based approach. *Auton Robot*, 2021.

Viraj Dawarka and Girish Bekaroo. Building and evaluating cloud robotic systems: A systematic review. *Robotics and Computer-Integrated Manufacturing*, 73:102240, 2022.

J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2019.

Zichen He, Jiawei Wang, and Chunwei Song. A review of mobile robot motion planning methods: from classical motion planning workflows to reinforcement learning-based architectures, 2021.

Lukas Hewing, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):269–296, 2020.

X. Huang, M. Kwiatkowska, S. Wang, and M. Wu. Safety verification of deep neural networks. *Proc. of the International Conference on Computer Aided Verification. CAV 2017.*, 10426:3–29, 2018.

Tamas Kiss, Peter Kacsuk, Jozsef Kovacs, Botond Rakoczi, Akos Hajnal, Attila Farkas, Gregoire Gesmier, and Gabor Terstyanszky. MiCADO - microservice-based cloud application-level dynamic orchestrator. *Future Generation Computer Systems*, 94:937–946, 2019.

Zhaojian Li, Tianshu Chu, Ilya V. Kolmanovsky, Xiang Yin, and Xunyuan Yin. Cloud resource allocation for cloud-based automotive applications. *Mechatronics*, 50:356–365, 2018.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *Int. Conf. on Learning Representations*, 2016.

Hang Liu, Shiwen Liu, and Kan Zheng. A reinforcement learning-based resource allocation scheme for cloud robotics. *IEEE Access*, 6:17215–17222, 2018.

R. Lovas, A. Marosi, Márk Emödi, Á. Kisari, E. Simonyi, and P. Gáspár. Paas-oriented iot platform with connected cars use cases. *2018 Int. Conf. on Sensor Networks and Signal Processing*, pages 409–420, 2018.

Nathan Marz. How to beat the cap theorem, 2011. URL `http://nathanmarz.com/blog/how-to-beat-the-cap-theorem.html`. Accessed: 2021-09-12.

Reinforcement Learning Toolbox User's Guide. The MathWorks, Inc., March 2020.

Simon Muntwiler, Kim P. Wabersich, Andrea Carron, and Melanie N. Zeilinger. Distributed model predictive safety certification for learning-based control. *IFAC-PapersOnLine*, 53(2):5258–5265, 2020. 21st IFAC World Congress.

Enikő Nagy, Róbert Lovas, István Pintye, Ákos Hajnal, and Péter Kacsuk. Cloud-agnostic architectures for machine learning based on apache spark. *Advances in Eng. Software*, 159:103029, 2021.

Balázs Németh and Péter Gáspár. The design of performance guaranteed autonomous vehicle control for optimal motion in unsignalized intersections. *Applied Sciences*, 11(8), 2021.

Manoj Penmetcha and Byung-Cheol Min. A deep reinforcement learning-based dynamic computational offloading method for cloud robotics. *IEEE Access*, 9:60265–60279, 2021.

U. Rosolia and F. Borrelli. Learning model predictive control for iterative tasks. A data-driven control framework. *IEEE Transactions on Automatic Control*, 63(7):1883–1896, July 2018.

Hoang-Dung Tran, Weiming Xiang, and Taylor T. Johnson. Verification approaches for learning-enabled autonomous cyber-physical systems. *IEEE Design Test*, pages 1–1, 2020.