The TOR Agent: Optimizing Driver Take-Over with Reinforcement Learning

Conference Paper · September 2021

DOI: 10.1145/3473682.3480262

3 authors:

Jakob Kuen
TU Wien

Clemens Schartmüller
Technische Hochschule Ingolstadt

Philipp Wintersberger
TU Wien

The TOR Agent: Optimizing Driver Take-Over with Reinforcement Learning

Jakob Kuen
TU Wien
Vienna, Austria
jakkuen@gmail.com

Clemens Schartmüller
CARISSMA, Technische Hochschule Ingolstadt
Ingolstadt, Germany
clemens.schartmueller@thi.de

Philipp Wintersberger
TU Wien
Vienna, Austria
philipp.wintersberger@tuwien.ac.at

Figure 1: Learning process of the reinforcement learning agent. The agent observes a set of 10 coordinates (x/y, white boxes in the images) lying on the lane center of the upcoming road segment in exponentially increasing distance (i.e., the immediate future is represented in higher resolution). Given these observations, the agent can either issue TOR or postpone it to a later moment. By receiving the drivers' lateral performance after a TOR as reward signal, the agent learns which road trajectories are more suitable for TORs than others. Given the implemented artificial driver model, it would issue TOR in the straight segment (left), but postpone it in curvy sections (right).

ABSTRACT

Various factors influence drivers' response to Take-Over Requests in automated driving, and a wide range of designs have been proposed to improve transitions. Still, little research has investigated how systems could deliver Take-Over Requests at the best moment in time. In this paper, we sketch the idea of a reinforcement learning agent that learns to deliver Take-Over Requests at the right time so that drivers' performance gets optimized, which could help to increase driving safety. We implemented such a system in Unity to evaluate this approach using a simple driver model. Our agent receives coordinates of the upcoming road segment and learns to deliver a Take-Over Request at an appropriate moment within a short time window. The reward function is composed to minimize the lateral deviation in the subsequent phase of manual driving. The initial results obtained are promising, and we will evaluate the concept with real human users soon.

AutomotiveUI '21 Adjunct, September 9–14, 2021, Leeds, United Kingdom

ACM ISBN 978-1-4503-8641-8/21/09.
https://doi.org/10.1145/3473682.3480262

CCS CONCEPTS

• **Human-centered computing** → **User interface management systems**; *Interaction techniques*; • **Computing methodologies** → *Machine learning approaches.*

KEYWORDS

take-over requests; automated driving; conditional automation; intelligent user interfaces; adaptive automation

ACM Reference Format:
Jakob Kuen, Clemens Schartmüller, and Philipp Wintersberger. 2021. The TOR Agent: Optimizing Driver Take-Over with Reinforcement Learning. In *13th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '21 Adjunct), September 9–14, 2021, Leeds, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3473682.3480262

1 INTRODUCTION

Take-Over Requests (TORs, i.e., transitions from automated to manual control) are a prominent research topic for automotive user interfaces [6, 15, 29]. Much research has focused on imminent TORs (where drivers must respond timely to an emergency), but it was argued that a majority of transitions would be "scheduled" [7, 26]. Scheduled transitions can occur in both SAE levels 3 and 4 [21] when the automated driving system leaves its operational design domain (for example, a highway driving system at a highway exit),

but they could also happen in the future to maintain drivers' situation awareness and driving skills [17]. It has even been argued that connected vehicles could allow scheduling in some emergencies [27]. Compared to imminent situations, scheduled transitions allow for planning within a certain time window. For example, recent works have looked into pre-alerts ("priming" [24]) and multi-stage warnings [5] to improve drivers' TOR responses. However, not many works have addressed the question of when exactly a user interface should communicate a TOR to the driver.

"Determining when to communicate" [10] is an increasingly important question in user interface design. We argue that a sophisticated interface would need to determine the best moment for a TOR since timing influences response performance and drivers' well-being [27]. Many factors have been shown to affect transitions, such as traffic volume [8], road curvature [20], TOR modality and presentation patterns [19], driver state [25], or secondary tasks [15]. Given the fast-changing dynamic driving environment, the question of when (within a given time window) a TOR should be issued can be seen as a probabilistic problem that allows for optimization. For example, when the need for a transition emerges in a curve, it could be worth waiting for a straight section – provided a sufficient time window and that a TOR would be less degrading (with respect to driving performance) on a straight road.

We propose to model the problem of deciding when a TOR should be issued within a given time frame as a reinforcement learning (RL) problem. Using machine learning to improve the TOR process has been suggested multiple times, mostly to predict human reactions. Proposed systems have included parameters such as traffic situation and secondary tasks [1], heart rate and eye movements [4], video streams of drivers [3], and vehicle data [18]. For example, such predictions could be used for driver monitoring systems that alert drivers to maintain situation awareness.

Our approach is novel in a way that it aims to determine the potentially best timing to communicate TORs to drivers. Such a system could become a valuable extension for in-vehicle interfaces, e.g., for multi-stage TORs [5]. Although the challenge of determining an appropriate timing to interrupt drivers has been proposed by empirical works both in the context of side activities [22] and transitions [28], we are not aware of any project investigating how such interfaces could be implemented. We have set up a driving simulation in Unity and modeled the scenario as an RL problem for our investigations. The initial results presented in this work have been obtained with simulated drivers (based on various assumptions), but we aim to include real human users later in the project.

2 A REINFORCEMENT LEARNING AGENT TO OPTIMIZE TAKE-OVERS

RL is a sub-branch of machine learning that does not require prelabeled training data. In an RL problem, an agent senses subsequent configurations of an environment (states of a Markov-decision process) and performs actions that may (or may not) receive a reward. Over time, by randomly "trying out" actions in different environment states (exploration), the agent eventually builds up a policy that maximizes future rewards as it learns which actions are most promising [23]. We argue that the probabilistic process of issuing a TOR at an appropriate moment within a given time frame can be formulated as an RL problem.

To test this idea, we have set up a driving simulation in Unity 3D using the "Windridge city" asset and integrated a simple automated vehicle model, which automatically drives and navigates through the city (i.e., algorithmic steering to follow lines/splines in the lane center with a speed of 20-30km/h and randomly selecting the next road segment at any junction). During this "endless" (no dedicated start or endpoint) drive, the car signals the need for a TOR at random moments. This activates an RL agent to determine the proper TOR timing within a 10-second window. We utilized the "ML-Agents" machine learning library for Unity [13] and implemented a simple model for our initial investigations:

- **Environment:** We modeled the state space as a list of (discretized) x/y coordinates describing 10 points lying on the upcoming trajectory (with exponentially increasing distance from the car, $dist(n) = 1.6^n meters$), with the vehicle acting as the origin of the coordinate system. Since the upcoming driving trajectory passes through these 10 points, the agent learns which coordinate sets are more suitable for TORs. Near points describe the trajectory that the human driver would have to react (i.e., "pass-through") to when the TOR is issued immediately, while points further away describe road segments that will become relevant when the TOR is delayed (see Figure 1).
- **Actions:** The agent can perform two actions: (1) issuing a TOR, or (2) wait/postpone to a later moment. In case a TOR is issued, the car hands over control to the human driver immediately. Postponing keeps the automated driving system active for another second before the agent is queried for the next decision.
- **Reward:** The agent receives the drivers' performance after completing the TOR. Given that this performance is dependent on the environment states, the agent will learn to issue the TOR notification at a suitable environment state where a high reward can be expected. We aimed at minimizing a drivers' lateral deviation in the phase after the TOR (i.e., the first phase of "manual driving"). Therefore, we calculated the standard deviation of lateral position (SDLP [14], a common indicator of maneuvering safety according to SAE J2944 [9]) of the 10s segment following the TOR. The reward function was defined as $1 - SDLP * 7$ (the scaling factor of 7 was determined based on experimentation) so that less lateral deviation from the lane center would translate to a higher reward. If the agent keeps postponing until the end of the 10s window and never issues a TOR, it received a highly negative reward (−4).

A trial for the RL-agent looks as follows (see Figure 2): While driving around the track (a) the agent receives the task of issuing a TOR within a 10-second window; (b) while progressing on the track, once per second the agent receives the current configuration of the environment (i.e., the set of coordinates) and either waits or issues the TOR; (c) after a TOR was issued and the "driver" has

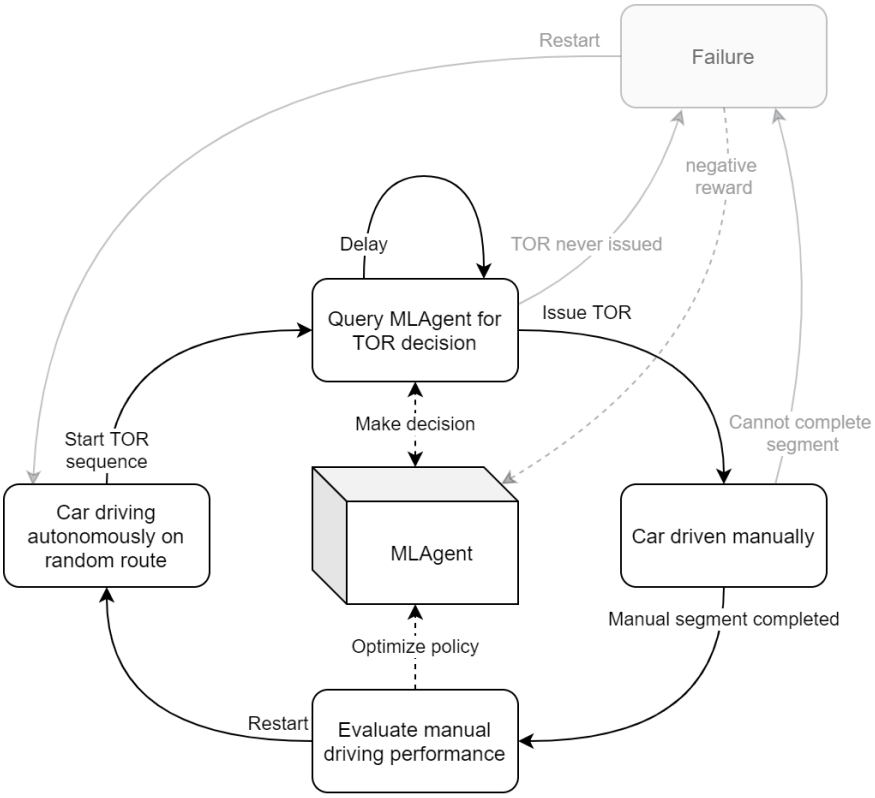

Figure 2: Architecture of the RL-agent that aims at determining suitable moments for TOR over time in an iterative training process. The agent receives the lateral deviation (SDLP) in the phase of "manual driving" following a TOR as a reward signal or gets punished in case of a failure (vehicle getting out of the lane or TOR not issued within the given time window).

maneuvered the vehicle for 10 seconds, the reward is calculated and forwarded to the agent. Summarized, our RL-agent aims at minimizing the lateral deviation (SDLP) in the response, given that this value is depending on the road layout. Before including real users (i.e., training and testing the system with driving simulator user studies), we utilized a simplified self-developed driver model simulating the following properties: The model shows little to no lateral deviation in straight segments but higher SDLP when driving through curves (implemented by varying the number of steering actions per second). This means, when a TOR is issued, after a simulated reaction time of 1 second, the driver model takes over control of the vehicle. Consequently, our agent should optimize in a way that TORs are issued when straight road segments are ahead, rather than curvy segments (which would lead to higher SDLP and thus less reward).

3 RESULTS

We trained our agent with Proximal Policy Optimization, and our best performing model ($\lambda = .9, \gamma = .998, \beta = .001, \eta = .002$, buffer size = 4096, batch size = 1024, two-layer neural network with 128 units per layer with a Sigmoid activation function) could achieve an average return of 0.72 (SD=0.378) after 10 million training steps (although the model quickly converges and becomes relatively stable after 500 thousand steps, see Figure 3). Further, we applied a trained model and issued 100 thousand TORs while the simulated vehicle was randomly driving in the Windridge city environment. Figure 4 shows that the trained policy would issue TORs mostly in straight sections and avoids issuing TORs before curves.

4 DISCUSSION, LIMITATIONS, AND FUTURE WORK

The RL-agent developed demonstrates the desired behavior (given the driver model, learning that TORs should be issued in straight road segments rather than curves, see Figure 4) by reducing the (simulated) lateral deviation, which highlights the potential of the approach. Still, the results also indicate that most of the time, only 2-3 seconds of the time budget was exploited (see Figure 3), potentially because of Windridge city's dense road layout. Consequently, we will update our driver model to become more realistic (i.e., gradually increasing driving performance after responding to TOR [16], variable reaction times, etc.) and build another simulation that generates random road layouts such that all potential environment configurations (i.e., an arbitrary set of road coordinates) are included in training and evaluation. We expect this will allow the policy to become more versatile. We will then continue training and evaluation of the agent before extending the environment model and including human users.

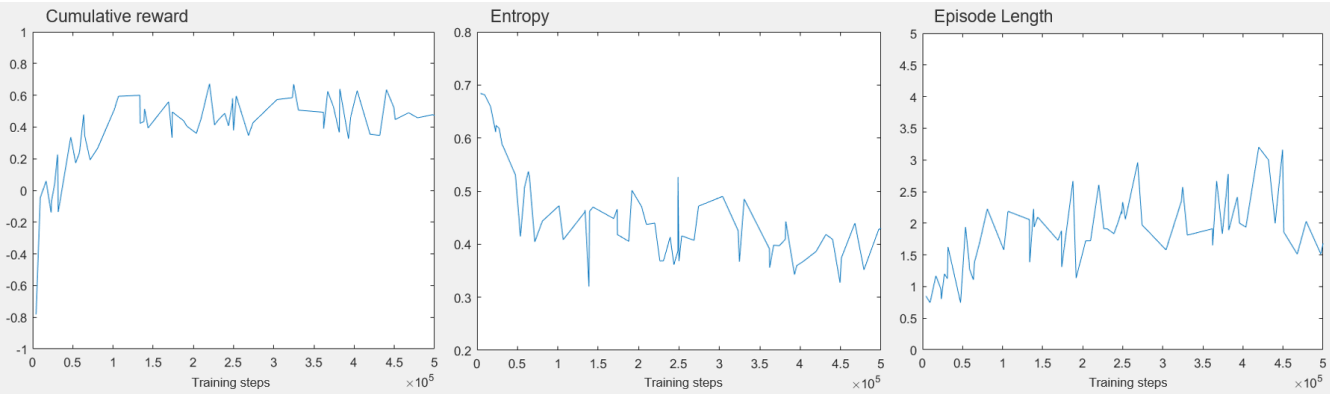

Figure 3: Results of the RL training progress, showing a fast increase in the reward (minimize SDLP) and an Entropy decrease over 500 thousand training steps (left, center). Still, the trained agent does not fully exploit the 10s time budget and delays TORs up to 3s (on average), potentially due to the dense road network of the training environment.

Figure 4: Application of a trained model (100 thousand TORs) in the Windridge city environment. The image on the left shows the driving directions of the individual driving lanes (green arrows). The image on the right shows the distribution of issued TORs over the map. Larger blue bubbles indicate a higher number of issued TORs compared to smaller bubbles. Investigation of the individual segments (under consideration of the corresponding driving direction) shows that the desired behavior has been trained successfully: TORs were mostly issued on straight segments and avoided shortly before curvy segments.

Currently, the model only includes the road layout, but future "real" systems could include other suitable sets of parameters. TORs were shown to be influenced by a wide range of factors, such as traffic volume, driver states, secondary tasks, etc. [15] that could be included in the model. Also, the reward function could be extended – for example, instead of just optimizing drivers' lateral driving performance after a TOR, the reward function could include drivers' reaction times and utilize physiological measurements to minimize driver stress.

After completing the evaluations with the simulated drivers (environment extensions, hyperparameter tuning, etc.), we will repeat the training process with data from real users in a driving simulator study. Then, we will evaluate it with a different user group to find out if the agent will optimize for real drivers' strengths and weaknesses. To deal with the high number of samples needed (500 thousand steps for the results in Figure 3), we could (a) increase the learning rate of the model, (b) implement a more sophisticated driver model based on data from human drivers (applying algorithmic variations or imitation learning [11]), or (c), not train from scratch but instead trying to adapt a model pre-trained with simulated drivers.

Finally, also the underlying concept has some limitations. For example, when an agent would, over time, issue TORs solely in situations that are easy to solve and "optimal" for drivers, the system itself could be another source of deskilling. Still, we argue that such issues could be resolved by appropriately configuring parameters and/or reward functions.

We still believe that the general concept behind this idea is very promising and could become a valuable extension for multi-stage warnings in scheduled transitions (see Figure 5). Appropriate timing

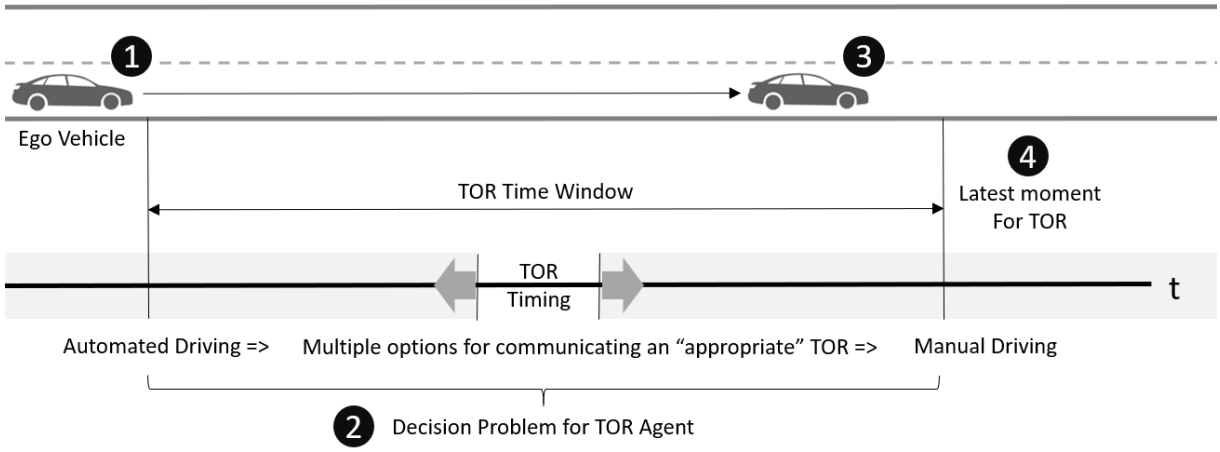

Figure 5: Application of the proposed idea in a multi-stage TOR UI. The driver is informed that a transition will occur soon (1), and the system will determine the ideal moment for the transition (2). The final TOR will be delivered when appropriate within the given time window (3) or at the end (with a proper lead time) when no suitable timing is found earlier (4).

of transitions will be important to maintain the safety and convenience for multi-tasking drivers [12, 22, 27], which is a precondition for other activities in the car – for example, realizing concepts such as the "mobile office" [2].

5 CONCLUSION

In this paper, we have sketched the idea of a reinforcement learning agent that aims at delivering Take-Over Requests at the right time, such that drivers' performance (i.e., reaction times, manual driving performance) is optimized. We have implemented the system in Unity 3D using the ML-agents library [13] and performed the first tests with a simulated driver. Our comparably simple model (based on ten coordinates describing upcoming road segments) aims at issuing TORs at the appropriate moment within a 10 second time window so that the lateral deviation in the subsequent phase of "manual driving" (in quotation marks as we used a simple driver model instead of real human users) is minimized. The early results are promising, and we will train/evaluate the concept in a driving simulator soon.

ACKNOWLEDGMENTS

Clemens Schartmüller was supported by the "Innovative Hochschule" program of the German Federal Ministry of Education and Research (BMBF) under Grant No. 03IHS109A (MenschINBewegung).

REFERENCES

[1] Christian Braunagel, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2017. Ready for take-over? A new driver assistance system for an automated classification of driver take-over readiness. *IEEE Intelligent Transportation Systems Magazine* 9, 4 (2017), 10–22.

[2] Lewis L. Chuang, Stella F. Donker, Andrew L. Kun, and Christian P. Janssen. 2018. Workshop on The Mobile Office. In *Adjunct Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Toronto, ON, Canada) *(AutomotiveUI '18)*. Association for Computing Machinery, New York, NY, USA, 10–16. https://doi.org/10.1145/3239092.3239094

[3] Nachiket Deo and Mohan M Trivedi. 2019. Looking at the driver/rider in autonomous vehicles to predict take-over readiness. *IEEE Transactions on Intelligent Vehicles* 5, 1 (2019), 41–52.

[4] Na Du, Feng Zhou, Elizabeth M Pulver, Dawn M Tilbury, Lionel P Robert, Anuj K Pradhan, and X Jessie Yang. 2020. Predicting driver takeover performance in conditionally automated driving. *Accident Analysis & Prevention* 148 (2020), 105748.

[5] Sandra Epple, Fabienne Roche, and Stefan Brandenburg. 2018. The sooner the better: Drivers' reactions to two-step take-over requests in highly automated driving. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. SAGE Publications Sage CA: Los Angeles, CA, Los Angeles, 1883–1887.

[6] Alexander Eriksson, VA Banks, and NA Stanton. 2017. Transition to manual: Comparing simulator with on-road control transitions. *Accident Analysis & Prevention* 102 (2017), 227–234.

[7] Alexander Eriksson and Neville A. Stanton. 2017. Takeover Time in Highly Automated Vehicles: Noncritical Transitions to and From Manual Control. *Human Factors* 59, 4 (2017), 689–705. https://doi.org/10.1177/0018720816685832

[8] Christian Gold, Moritz Körber, David Lechner, and Klaus Bengler. 2016. Taking over control from highly automated vehicles in complex traffic situations: the role of traffic density. *Human factors* 58, 4 (2016), 642–652.

[9] Paul Green. 2013. Standard Definitions for Driving Measures and Statistics: Overview and Status of Recommended Practice J2944. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Eindhoven, Netherlands) *(AutomotiveUI '13)*. Association for Computing Machinery, New York, NY, USA, 184–191. https://doi.org/10.1145/2516540.2516542

[10] Barbara Grosz. 2012. What question would turing pose today? *AI magazine* 33, 4 (2012), 73–73.

[11] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. 2017. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)* 50, 2 (2017), 1–35.

[12] Christian P Janssen, Shamsi T Iqbal, Andrew L Kun, and Stella F Donker. 2019. Interrupted by my car? Implications of interruption and interleaving research for automated vehicles. *International Journal of Human-Computer Studies* 130 (2019), 221–233.

[13] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, et al. 2018. Unity: A general platform for intelligent agents.

[14] Gwendolin Knappe, Andreas Keinath, Klaus Bengler, and Cristina Meinecke. 2007. Driving simulator as an evaluation tool–assessment of the influence of field of view and secondary tasks on lane keeping and steering performance.

[15] Anthony D McDonald, Hananeh Alambeigi, Johan Engström, Gustav Markkula, Tobias Vogelpohl, Jarrett Dunne, and Norbert Yuma. 2019. Toward computational simulations of behavior during automated driving takeovers: a review of the empirical and modeling literatures. *Human factors* 61, 4 (2019), 642–688.

[16] Natasha Merat, A Hamish Jamson, Frank CH Lai, Michael Daly, and Oliver MJ Carsten. 2014. Transition to manual: Driver behaviour when resuming control from a highly automated vehicle. *Transportation research part F: traffic psychology and behaviour* 27 (2014), 274–282.

[17] Siobhan E Merriman, Katherine L Plant, Kirsten MA Revell, and Neville A Stanton. 2021. Challenges for automated vehicle driver training: a thematic analysis from manual and automated driving. *Transportation research part F: traffic psychology and behaviour* 76 (2021), 238–268.

[18] Erfan Pakdamanian, Shili Sheng, Sonia Baee, Seongkook Heo, Sarit Kraus, and Lu Feng. 2021. DeepTake: Prediction of Driver Takeover Behavior Using Multimodal Data. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 103, 14 pages. https://doi.org/10.1145/3411764.3445563

[19] Ioannis Politis, Stephen Brewster, and Frank Pollick. 2015. Language-based Multimodal Displays for the Handover of Control in Autonomous Cars. In *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Nottingham, United Kingdom) *(AutomotiveUI '15)*. ACM, New York, NY, USA, 3–10. https://doi.org/10.1145/2799250.2799262

[20] Shadan Sadeghian Borojeni, Susanne C.J. Boll, Wilko Heuten, Heinrich H. Bülthoff, and Lewis Chuang. 2018. Feel the Movement: Real Motion Influences Responses to Take-over Requests in Highly Automated Vehicles. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 246, 13 pages. https://doi.org/10.1145/3173574.3173820

[21] SAE. 2018. SAE J3016 Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems.

[22] Rob Semmens, Nikolas Martelaro, Pushyami Kaveti, Simon Stent, and Wendy Ju. 2019. Is Now A Good Time? An Empirical Study of Vehicle-Driver Communication Timing. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300867

[23] Richard S Sutton and Andrew G Barto. 2018. Reinforcement learning: An introduction.

[24] Remo M.A. van der Heiden, Shamsi T. Iqbal, and Christian P. Janssen. 2017. Priming Drivers before Handover in Semi-Autonomous Cars. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 392–404. https://doi.org/10.1145/3025453.3025507

[25] Katharina Wiedemann, Frederik Naujoks, Johanna Wörle, Ramona Kenntner-Mabiala, Yvonne Kaussner, and Alexandra Neukum. 2018. Effect of different alcohol levels on take-over performance in conditionally automated driving. *Accident analysis & prevention* 115 (2018), 89–97.

[26] Philipp Wintersberger, Paul Green, and Andreas Riener. 2017. Am I Driving or Are You or Are We Both? A Taxonomy for Handover and Handback in Automated Driving.

[27] Philipp Wintersberger, Andreas Riener, Clemens Schartmüller, Anna-Katharina Frison, and Klemens Weigl. 2018. Let Me Finish before I Take Over: Towards Attention Aware Device Integration in Highly Automated Vehicles. In *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Toronto, ON, Canada) *(AutomotiveUI '18)*. Association for Computing Machinery, New York, NY, USA, 53–65. https://doi.org/10.1145/3239060.3239085

[28] Philipp Wintersberger, Clemens Schartmüller, and Andreas Riener. 2019. Attentive User Interfaces to Improve Multitasking and Take-Over Performance in Automated Driving: The Auto-Net of Things. *International Journal of Mobile Human Computer Interaction (IJMHCI)* 11, 3 (2019), 40–58.

[29] Bo Zhang, Joost de Winter, Silvia Varotto, Riender Happee, and Marieke Martens. 2019. Determinants of take-over time from automated driving: A meta-analysis of 129 studies. *Transportation research part F: traffic psychology and behaviour* 64 (2019), 285–307.