

# JOAN: a framework for human-automated vehicle interaction experiments in a virtual reality driving simulator

Niek Beckers<sup>ID</sup><sup>\*</sup><sup>1</sup>, Olger Siebinga<sup>ID</sup><sup>†</sup><sup>1</sup>, Joris Giltay<sup>1</sup>, and André van der Kraan<sup>1</sup>

<sup>1</sup> Human-Robot Interaction group, Department of Cognitive Robotics, Faculty 3mE, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands

DOI: [10.21105/joss.04250](https://doi.org/10.21105/joss.04250)

## Software

- [Review](#) <sup>↗</sup>
- [Repository](#) <sup>↗</sup>
- [Archive](#) <sup>↗</sup>

Editor: [Patrick Diehl](#) <sup>↗</sup> <sup>ID</sup>

## Reviewers:

- [@humanfactors](#)
- [@imcatta](#)

Submitted: 14 February 2022

Published: 30 January 2023

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

With the rapid development of automated driving systems, human drivers will soon have to share the road, and interact with, autonomous vehicles (AVs). To design AVs that can be safely introduced in our mixed traffic, research into human-AV interaction is needed. Driving simulators are invaluable tools, however, existing driving simulators are expensive, implementing new functionalities (e.g., AV control algorithms) can be difficult, and setting up new experiments can be tedious. To address these issues, we present JOAN, a framework for human-AV interaction experiments. JOAN connects to the open-source AV-simulation program Carla and enables quick and easy implementation of simple driving experiments. It supports experiments in VR, a variety of human input devices, and haptic feedback. JOAN is easy to use, flexible, and enables researchers to design, store, and perform human-factors experiments without writing a single line of code.

## Statement of need

Real-world traffic is rapidly changing: automated vehicles, even autonomous vehicles, are rapidly being introduced on the roads interacting with other road users. It is likely that traffic will be mixed for the coming decades, with manually driven vehicles, vehicles of various levels of automation, and vulnerable road users (e.g., pedestrians and cyclists) sharing the same road ([Di & Shi, 2021](#)). Therefore, investigating the interaction between AVs and other road users is critical for the development of safe and acceptable AV behavior. Driving simulators are invaluable tools to study such human-AV interactions.

State-of-the-art driving simulators use motion platforms, immersive graphics ([WIVW, 2020](#)), and/or high-fidelity haptic feedback ([Mulder et al., 2012](#)) to create simulations with high levels of realism ([Slob, 2008](#)). However, such simulators require dedicated and often expensive hardware, limiting their availability for algorithm development and evaluation. Furthermore, they require a steep programming learning curve and lack the flexibility to quickly implement customized experiments.

Alternatively, virtual-reality-based driving simulators offer reasonable levels of realism, require relatively low-cost hardware, and are generally more flexible in terms of rapid algorithm development or experiment evaluation. Traffic simulators based on the Unity game engine have been developed for studying human-human interaction in mixed traffic ([Bazilinsky et al.,](#)

---

<sup>\*</sup>co-first author

<sup>†</sup>co-first author

2020; Kearney & Noyce, 2018). However, both lack the framework to implement AV driving models, which is necessary to simulate mixed-traffic scenarios.

Several game engine-based simulators exist that only specifically focus on AV algorithm development (e.g., Shah et al. (2018), LG Electronics America R&D Center (2020), Dosovitskiy (2017)). While such simulators provide possibilities for implementing, training, and evaluating fully automated driving algorithm, they do not support human-in-the-loop experiments. A popular open-source simulators is CARLA (based on the Unreal-Engine); it provides a flexible and well-documented API, pre-trained automated driving models, common roadmap standards (OpenDrive), and is actively maintained.

Silvera et al. extended Carla to support human-in-the-loop experiments (Silvera et al., 2022). They created an affordable driving simulator primarily implemented in C++ and python that can be used for human-AV interaction experiments which leverages the benefits of CARLA for AV development. However, because their software is not set up with a more general experiment framework that allows customizability, their software is restricted to specific hardware (both for human input and VR headsets) and requires extensive programming to implement customized experiments.

To address these issues, we created JOAN, an open-source framework for conducting human-in-the-loop driving experiments. JOAN interfaces with Carla, is written in Python, is fully customizable through code and graphical user interfaces (GUI). JOAN can be used with a variety of human input devices, including game console controllers (e.g., Xbox or PlayStation), generic USB controllers (e.g., Logitech G920), or a SensoDrive high-fidelity steering wheel (SensoDrive, 2022) (for including haptic feedback). JOAN includes a framework for experiment design to create and execute experiments, and provides reliable data acquisition. These features can be accessed through a user-friendly interface, which means no extensive (knowledge of) programming nor a lot of time is required to set up new experiments.

For detailed instructions on how to install and use JOAN, see our online documentation<sup>1</sup>. JOAN can be found in a GitLab repository<sup>2</sup>, a video with more information can be found on YouTube<sup>3</sup>.

## Software functionality

We set the following requirements for JOAN based on the needs of a typical human-in-the-loop driving experiment. JOAN should:

- provide reliable data acquisition,
- enable researchers to connect multiple types of user input devices,
- facilitate the design and execution of experiment protocols; create repeatable traffic scenarios involving multiple agents,
- provide an easy way to design experiments by providing access to most functionality from a GUI,
- be flexible and extensible by design, and
- include support for haptic feedback.

JOAN consists of modules, each with a functionality, to address the flexibility and extensibility requirements. These modules can be enabled or disabled based on the needs of a researcher, if needed they can easily be adapted, and researchers can create new modules to implement new functionality. The standard modules included with JOAN:

- log and plot experiment data,
- handle hardware inputs,

---

<sup>1</sup><https://joan.readthedocs.io>

<sup>2</sup><https://gitlab.tudelft.nl/tud-cor-hri/joan-framework/joan>

<sup>3</sup><https://www.youtube.com/watch?v=TLLw48isYJU>

- design and execute experiments with multiple conditions,
- implement traffic scenarios with dynamic triggers for events,
- provide a human with haptic feedback.

Each module has its own GUI which appears if the module is enabled. These interfaces provide access to all standard functionality of the module. A researcher can design and perform a simple human-in-the-loop experiment without writing a single line of code. The documentation provides more information for researchers who want to adapt existing modules (e.g. to implement new hardware), or who want to write a new module (e.g. to implement specific controllers).

## Usage in Science and Education

JOAN was used to quickly develop and perform several human-in-the-loop experiments (including various MSc students, for a published example, see (Melman et al., 2021)), illustrating that JOAN provides a framework that is flexible and quick to implement (most projects are completed in 6 months). Furthermore, a core research focus of our department is the haptic interaction between human drivers and their vehicles (e.g., Abbink et al. (2018), Mulder et al. (2012)), for which we implemented a module to control a haptic force-feedback steering wheel. The `SteeringWheelController` module enables implementation of control algorithms for force feedback and comes with two example implementations: a standard PD controller (Mulder et al., 2012) and a Four Design Choice Architecture controller (FDCA) (van Paassen et al., 2017).

## Acknowledgements

We thank Timo, Cedric, Sam, Peter, Mark, and Tarbiya for their support in the initial stages of software development.

## References

- Abbink, D. A., Carlson, T., Mulder, M., de Winter, J. C. F., Aminravan, F., Gibo, T. L., & Boer, E. R. (2018). A Topology of Shared Control Systems—Finding Common Ground in Diversity. *IEEE Transactions on Human-Machine Systems*, 48(5), 509–525. <https://doi.org/10.1109/THMS.2018.2791570>
- Bazilinsky, P., Kooijman, L., Dodou, D., & de Winter, J. C. F. (2020). Coupled simulator for research on the interaction between pedestrians and (automated) vehicles. *19th Driving Simulation Conference (DSC)*, 7.
- Di, X., & Shi, R. (2021). A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning. *Transportation Research Part C: Emerging Technologies*, 125(July 2020), 103008. <https://doi.org/10.1016/j.trc.2021.103008>
- Dosovitskiy, A. (2017). CARLA: An Open Urban Driving Simulator. *Proceedings of the 1st Annual Conference on Robot Learning*, 1–16. <http://proceedings.mlr.press/v78/dosovitskiy17a/dosovitskiy17a.pdf>
- Kearney, J. K., & Noyce, D. A. (2018). *Multi-Modal Distributed Simulation Combining Cars, Bicyclists, and Pedestrians* (p. 31).
- LG Electronics America R&D Center, A. P. L. at the. (2020). *LGSVL simulator*. <https://www.lgsvlsimulator.com/>

- Melman, T., Visser, P., Mouton, X., & Winter, J. de. (2021). Creating the Illusion of Sportiness: Evaluating Modified Throttle Mapping and Artificial Engine Sound for Electric Vehicles. *Journal of Advanced Transportation*, 2021, 1–15. <https://doi.org/10.1155/2021/4396401>
- Mulder, M., Abbink, D. A., & Boer, E. R. (2012). Sharing Control With Haptics: Seamless Driver Support From Manual to Automatic Control. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 54(5), 786–798. <https://doi.org/10.1177/0018720812443984>
- SensoDrive. (2022). *SensoDrive force feedback*. <https://www.sensodrive.de/products/force-feedback-products.php>
- Shah, S., Dey, D., Lovett, C., & Kapoor, A. (2018). *AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles* (pp. 621–635). [https://doi.org/10.1007/978-3-319-67361-5\\_40](https://doi.org/10.1007/978-3-319-67361-5_40)
- Silvera, G., Biswas, A., & Admoni, H. (2022). *DReyeVR: Democratizing Virtual Reality Driving Simulation for Behavioural & Interaction Research*. <http://arxiv.org/abs/2201.01931>
- Slob, J. (2008). *State-of-the-art driving simulators, a literature survey* (DCT Report No. 107).
- van Paassen, M. M., Boink, R., Abbink, D. A., Mulder, M., & Mulder, M. (2017). Four Design Choices for Haptic Shared Control. In M. A. Vidulich, P. S. Tsang, & J. Flach (Eds.), *Advances in Aviation Psychology, Volume 2: Using Scientific Methods to Address Practical Human Factors Needs* (First, pp. 237–254). Routledge. <https://doi.org/10.4324/9781315565712>
- WIVW. (2020). *Driving simulation and SILAB*. <https://wivw.de/en/silab>