Case Study: Safety Controller for Autonomous Driving on Highways *

Michael Leuschel, Fabian Vu, and Kristin Rutenkolk

Heinrich-Heine-Universität Düsseldorf
Faculty of Mathematics and Natural Sciences
Institute of Computer Science
{leuschel, fabian.vu, kristin.rutenkolk}@uni-duesseldorf.de

Abstract. This requirements document presents the case study for the ABZ conference 2025. The case study is about a safety controller for autonomous driving on a highway. The description contains two variations of the case study. First, in the simpler setting, we just consider a single-lane highway where each vehicle can accelerate and brake. The goal is to keep a safe distance to the preceding car. Second, we consider a multi-lane highway where each vehicle can also change lanes.
The challenge is to model the system and its environment, and to derive assumptions and model a controller for which the safety can be guaranteed. The challenge is also to present the safety case in such a way that it is convincing to readers not entirely familiar with the formal method employed.
The case study is designed such that the formal model can be used as a safety shield within a highway simulation environment. We provide pre-trained (unsafe) AI agents, which can be used to experiment with. This part of the case study is optional.

Keywords: Formal Methods, Autonomous Driving, Case Study, Highway, Artificial Intelligence

Revision History

Version	Date	Modifications
01	16.10.2024	First version of case study
02	13.11.2024	Added Summary in Section 4, Added introduction of Section 2.1: Vehicles, Clarify braking behavior in Section 2.1: Vehicles, Clarify FASTER in Section 2.1: Vehicles, Clarify in Section 2.1: Vehicles what a cycle is Clarify RSS formula in Section 2.4: Safety Requirement

* The work of Fabian Vu is part of the KI-LOK project funded by the "Bundesministerium für Wirtschaft und Energie"; grant # 19/21007E, and the IVOIRE project funded by "Deutsche Forschungsgemeinschaft" (DFG) and the Austrian Science Fund (FWF) grant # I 4744-N.

1 Introduction and Motivation

This requirement document presents the case study for the ABZ conference 2025, which is about a safety controller for driving vehicles on a highway (motorway in UK English). In practical use, the safety controller could be employed as an assistant for a human driver, or can also be added to an artificial intelligence (AI) component to obtain a safe autonomous driving system.

In practice, cameras and sensors are used to observe the environment of the vehicle. In the case study, the perception system is abstracted away, i.e., the controller has access to the vehicle's position as well as the positions of all cars in the vicinity. There are no other obstacles on the highway. The challenge of the case study is to model the driving system with appropriate safety rules that guarantee safety, i.e., the absence of collisions.

For demonstration but also empirical evaluation purposes, the case study is combined with a simulated highway environment along with trained reinforcement learning AI agents based on [6]. We consider two environments specifically: a single-lane highway where each vehicle can accelerate and brake, and a multi-lane highway where each vehicle can also change lanes.

With this case study, we ask the following questions:

- Which strategy and assumptions do we need for safe driving?
- Under which conditions is it possible to guarantee complete safety?
- How can we implement those conditions on an autonomous driving system, and verify and validate the safety?

2 Requirements

This section presents the details of the vehicles, the (single-lane and multi-lane) environments, and the safety requirements that are considered for this case study. We build on the highway environment presented by Leurent [6] and configure it accordingly. The technical details and configuration of the parameters correspond to [6] as well.

2.1 Vehicles

In the following, we provide environmental requirements for the vehicles.

- **VEH1**: Every vehicle has a length of l meters.
- **VEH2**: Every vehicle has a width of w meters.
- **VEH3**: A vehicle has a maximum speed of v_{max} m/s
- **VEH4**: A vehicle has a minimum speed of 0 m/s, i.e., it cannot move backwards.
- **VEH5**: A vehicle has a maximum acceleration of a_{max} m/s^2.
- **VEH6**: A vehicle has a maximum braking deceleration of b_{max} m/s^2.

– **VEH7**: A vehicle has a minimum guaranteed braking deceleration of b_{min} m/s^2, i.e., if it is braking than the braking deceleration will be between b_{min} and b_{max}, until the point it stops.

Concluding from **VEH3**, **VEH4**, and **VEH5**, the range for the speed is thus $[0, v_{max}]$ m/s. Concluding from **VEH6** and **VEH7**, the range of the acceleration is thus $[-b_{max}, a_{max}]$ m/s^2, with $b_{max} > 0$ and $a_{max} > 0$.

Note that there are edge cases where the acceleration can be in $[-b_{min}, 0]$ as well, e.g., when the difference between a full stop (0 m/s) and the current speed is less than b_{min}. For all other cases, the acceleration is in $[-b_{max}, b_{min}]$ when braking and $[0, a_{max}]$ when accelerating.

When using the highway environment from [6], the concrete values for the parameters above are:

– **VEH1-ENV**: Each vehicle has a length l of 5 meters.
– **VEH2-ENV**: Each vehicle has a width w of 2 meters.
– **VEH3-ENV**: Each vehicle has a maximum speed v_{max} of 40 m/s (= 144 km/h).
– **VEH4-ENV**: Each vehicle has a minimum speed of 0 m/s.
– **VEH5-ENV**: Each vehicle has a maximum acceleration of 5 m/s^2.
– **VEH6-ENV**: Each vehicle has a maximum braking deceleration of 5 m/s^2.
– **VEH7-ENV**: Each vehicle has a minimum guaranteed braking deceleration of 3 m/s^2.

In the following, we describe actions that a controller can perform to control one or multiple vehicles in the environment. We will use the term *cycle* as a time interval in which a vehicle observes its environment, and decides to perform an action until reaching the next cycle, i.e., until the next observation and decision.

– **ACT1**: Accelerate (`FASTER`): This action increases the speed (up to v_{max}) with an acceleration up to a_{max} m/s^2. Once the car reaches the v_{max} the acceleration is $0m/s^2$.
– **ACT2**: Brake (`SLOWER`): This action brakes with a braking deceleration of b_{min} up to b_{max} m/s^2. Once the car stops the braking deceleration is $0m/s^2$.
– **ACT3**: Idle (`IDLE`): This action reduces the (braking) acceleration close to 0 m/s^2.
– **ACT4**: Change lane to left (`LANE_LEFT`): This action changes the current lane of the vehicle to the lane directly left of it within the current cycle. The acceleration behaves like `IDLE`.
– **ACT5**: Change lane to right (`LANE_RIGHT`): This action changes the current lane of the vehicle to the lane directly right of it within the current cycle. The acceleration behaves like `IDLE`.

Note that the other vehicles also perform these actions, but at different times. For example, another vehicle could brake for the first half of the cycle and accelerate in the second half. In Section 3, we provide trained agents configured as single agents. Instead/additionally, one can also train and configure multi-agents.

Also note, that there is no guarantee that `FASTER` will use the maximal accleration a_{max}.

Regarding the controller, the following requirement applies:

– **CON1**: All controlled vehicles observe the environment in a specific time interval of t, i.e., the response time is t seconds.

Concerning the environment, the requirement is:

– **CON1-ENV**: All controlled vehicles observe the environment every second, i.e., the response time is 1 second.

2.2 Single-Lane Environment

Figure 1 shows a visualization of a single-lane environment. Regarding the environment, the following assumption can be made:

– **ENV1**: At any time, there are n_{ve} vehicles on the highway with $n_{ve} \geq 1$.
– **ENV2**: All vehicles drive in the same direction.

In the single-lane environment, the relevant actions are `FASTER`, `SLOWER`, and `IDLE`.

Fig. 1: Visualization of Single-Lane Environment; figure is created while simulating in [6].

2.3 Multi-Lane Environment

Figure 2 shows a visualization of a multi-lane environment (with 4 lanes). **ENV1** and **ENV2** also apply to the multi-lane environment. Additionally, the following assumption can be made about the environment:

– **ENV3**: The multi-lane environment consists of a fixed number of lanes n_{la} with $n_{la} \geq 2$.

This means that the number of lanes does not change over time. In addition to `FASTER`, `SLOWER`, and `IDLE`, actions to change lanes to the left (`LANE_LEFT`) or the right (`LANE_RIGHT`) are also relevant.

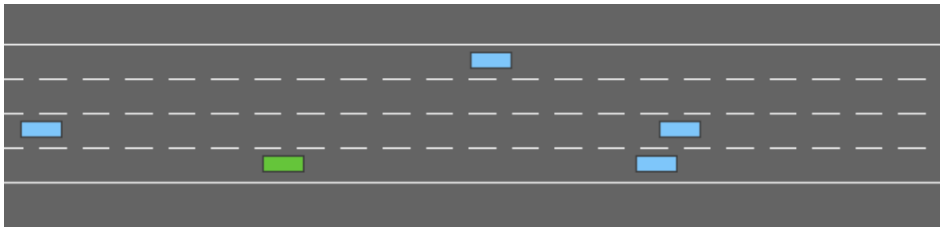

Fig. 2: Visualization of Multi-Lane Environment (with 4 lanes); figure is created while simulating in [6].

2.4 Safety Requirement

This section presents the main safety requirement and a formula to maintain the safety distance.

The most important safety requirement for the case study is:

– **SAF**: All controlled vehicles must avoid collisions.

SAF can be achieved by maintaining a safety distance. For the single-lane environment, one has to consider the distance to the vehicle behind and to the vehicle in front. For the multi-lane environment, one has to consider lane changes, and possibly even more rules. The Responsibility-Sensitive Safety (RSS) model [10] presented by Shalev-Shwartz et al. [1] can be used to maintain safety distances. In particular, the first rule [10] of RSS defines the computation of the safety distance as:

$$d_{min} = [v_r * \rho + \frac{1}{2} * a_{max} * \rho^2 + \frac{(v_r + \rho * a_{max})^2}{2 * \beta_{min}} - \frac{v_f^2}{2 * \beta_{max}}]_+$$

using the notation $[x]_+ := max\{x, 0\}$ and with

– ρ - response time
– v_r - speed of rear vehicle
– v_f - speed of front vehicle
– a_{max} - maximum acceleration of rear vehicle before braking
– β_{max} - maximum braking acceleration of front vehicle
– β_{min} - braking acceleration of rear vehicle (reaction to braking of front vehicle)

This formula was for example used in [2] with Isabelle to prove safety or combined with goals in [3, 4]. For the case study, one can also consider other formulas or assumptions (additionally or instead of RSS) for computing the safety distance.

[1] More details available at: https://www.mobileye.com/technology/responsibility-sensitive-safety/

3 Simulation in AI Environment

This section provides additional material, in case you wish to use and evaluate your safety controller as a safety shield [5] for an AI system. As such, we provide several reinforcement learning (RL) agents that were trained in the highway environment [6]. The requirements above were designed in such a way that the formal model integrates with the abstraction provided by this highway environment.

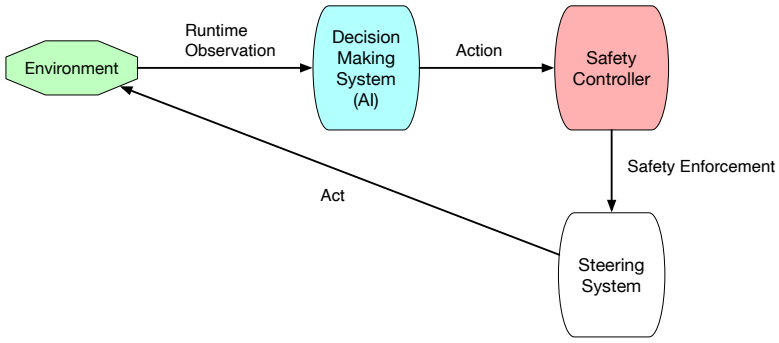

Fig. 3: Components of Autonomous Driving System

3.1 Overview

Figure 3 depicts using a safety controller for an autonomous driving system. In practice, the perception is done by cameras and sensors; this is abstracted away in our case study. We suppose we obtain position and speed information about vehicles in the vicinity (see Section 3.3 below).

Based on the observations, the decision-making system decides which actions to execute next. The safety controller checks whether the actions made by the decision-making system are safe, and intervenes/corrects the decisions accordingly. The corrected action is then provided to the steering system for execution.

3.2 Trained Agents

We trained agents for both the single-lane and the multi-lane environment with Deep Q-learning (DQN) [7]. For both environments, we present two agents: an agent which was trained with penalties for collisions, and another agent which behaves adversarially, i.e., it is rewarded for collisions.[2]

The trained agents are available at: https://github.com/hhu-stups/abz2025_casestudy_autonomous_driving.

[2] Additionally, you can also train more agents if required.

Single-Lane Environment. For the single-lane environment, we use the standard configuration for training, and modify them for both agents as follows:

- The first agent, called BASE, is trained with a penalty for collisions, and a reward v_{cur}/v_{max} for the current speed (the faster the better).
- The second agent, called ADVERSARIAL, is trained with a reward for collisions, and again a reward depending on the current speed.

Multi-Lane Environment. For the multi-lane environment, we also use the standard configuration for training, and modify them for both agents as follows:

- The first agent, called BASE, is trained with a penalty of for collisions, a right-lane reward (i.e., the car is rewarded if it drives on the right to let other cars pass), and again a reward for the current speed.
- The second agent, called ADVERSARIAL, is trained with a reward for collisions, again a reward for the current speed, and no right-lane reward but a lane change reward.

3.3 Observing and Controlling Vehicles in Highway Environment

In the following, we provide information that are only relevant to implement Figure 3, i.e., to implement an adapter between the agents' observations and the safety controller. The details relate to how a controlled vehicle observes its environment.

Within the highway environment, an agent observes the environment in each cycle (of 1 second by default) and performs an action (**ACT1**–**ACT5** from Section 2.1). Each observation contains the presence, the positions, and the speeds of all vehicles. The position and speed of the controlled vehicle are absolute, while the positions and speeds of the other vehicles are relative to the controlled vehicle. A controlled vehicle can only observe other vehicles within a distance of 100m. More details about the environment are available at: https://highway-env.farama.org/observations/.[3] Such an observation for a single agent is shown in Figure 4.

$$
\begin{array}{l}
 \\
ControlledVehicle \\
Vehicle_2 \\
Vehicle_3 \\
Vehicle_4 \\
Vehicle_5
\end{array}
\begin{array}{c}
\begin{array}{ccccc} Presence & x & y & v_x & v_y \end{array} \\
\left(\begin{array}{ccccc}
1.0 & 0.89 & 0.50 & 0.31 & 0.0 \\
1.0 & 0.09 & -0.50 & -0.04 & 0.0 \\
1.0 & 0.21 & 0.00 & -0.02 & 0.0 \\
1.0 & 0.33 & 0.00 & -0.04 & 0.0 \\
1.0 & 0.43 & -0.25 & -0.04 & 0.0
\end{array}\right)
\end{array}
$$

Fig. 4: Example: Observation in Highway Environment

[3] There is also a multi-agent setting to control multiple vehicles where the state is represented by an array of observations: https://highway-env.farama.org/multi_agent/.

3.4 Metrics

In the validation process, one can consider more metrics to evaluate the quality of an autonomous driving system: the accident rate, the expected time until collision, the distance traveled, the speed, the cumulative reward (of the reinforcement learning agent), the time spent on right-most lane according to the *keep right requirement* in many countries. Some metrics are described in [12]; they are only relevant when the safety controller is adapted to the RL agents.

3.5 Some Related Works

The idea of using a simple system to control a complex system was originally introduced by Sha [9], and later expanded to reinforcement learning applications [11] in the neural simplex architecture [8]. Figure 3 works similarly to post-shielding [1] where the AI's decisions are corrected. Another approach is pre-shielding [1] which provides safe actions the AI can choose from.

4 Summary

We expect contributions which

- formalise the behaviour of the vehicles and the effect of the different control actions (`FASTER`, `SLOWER`, ...),
- derive a set of assumptions and rules for which the system is safe,
- formally show the safety of the system under these rules and assumptions.

For this case study we would like to put particular emphasis on a clear exposition of the models and of the safety argument. Ideally, your argument should convince somebody not familiar with the particular formal method used of the safety of the system.

Your solution can target one or both of these settings:

- a single line setting without lane changes,
- a multi-lane setting with possible lane changes.

Another motivation of our case study is applying formal methods to AI to improve the safety. The main goal here is to develop a formal model that can superivse an existing AI system. To this end, we provide trained AI agents for our case study, which can be run in a highway simulation environment and which can be combined with your formal model (or code generated from your formal model). We thus encourage to develop a solution

- that can be used as a safety-shield of an AI agent in the highway environment,
- thereby improving safety or even guaranteeing safety,
- while achieving good practical performance (e.g., in terms of total distance travelled).

Acknowledgements

We thank Amel Mammar and Atif Mashkoor for useful feedback.

References

1. Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., Topcu, U.: Safe reinforcement learning via shielding. In: Proceedings AAAI. pp. 2669–2678. AAAI Press (2018). https://doi.org/10.1609/aaai.v32i1.11797
2. Crisafulli, P., Taha, S., Wolff, B.: Modeling and analysing cyber-physical systems in HOL-CSP. Robotics Auton. Syst. **170**, 104549 (2023). https://doi.org/10.1016/J.ROBOT.2023.104549
3. Hasuo, I., Eberhart, C., Haydon, J., Dubut, J., Bohrer, R., Kobayashi, T., Pruekprasert, S., Zhang, X.Y., Pallas, E.A., Yamada, A., Suenaga, K., Ishikawa, F., Kamijo, K., Shinya, Y., Suetomi, T.: Goal-aware rss for complex scenarios via program logic. IEEE Transactions on Intelligent Vehicles **8**(4), 3040–3072 (2023). https://doi.org/10.1109/TIV.2022.3169762
4. Kobayashi, T., Bondu, M., Ishikawa, F.: Formal modelling of safety architecture for responsibility-aware autonomous vehicle via event-b refinement. In: Proceedings FM'2023. pp. 533–549 (2023), https://doi.org/10.1007/978-3-031-27481-7_30
5. Könighofer, B., Lorber, F., Jansen, N., Bloem, R.: Shield Synthesis for Reinforcement Learning. In: Proceedings ISoLA. pp. 290–306. LNCS 12476 (2020), https://doi.org/10.1007/978-3-030-61362-4_16
6. Leurent, E.: An Environment for Autonomous Driving Decision-Making. https://github.com/eleurent/highway-env (2018)
7. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. nature **518**(7540), 529–533 (2015), https://doi.org/10.1038/nature14236
8. Phan, D.T., Grosu, R., Jansen, N., Paoletti, N., Smolka, S.A., Stoller, S.D.: Neural simplex architecture. In: Proceedings NFM. pp. 97–114. LNCS 12229 (2020), https://doi.org/10.1007/978-3-030-55754-6_6
9. Sha, L.: Using simplicity to control complexity. IEEE Software **18**(4), 20–28 (2001). https://doi.org/10.1109/MS.2001.936213
10. Shalev-Shwartz, S., Shammah, S., Shashua, A.: On a formal model of safe and scalable self-driving cars. CoRR **abs/1708.06374** (2017), http://arxiv.org/abs/1708.06374
11. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
12. Vu, F., Dunkelau, J., Leuschel, M.: Validation of reinforcement learning agents and safety shields with ProB. In: Proceedings NFM'2024. pp. 279–297. LNCS 14627, Springer (2024)