ASCETISM –

Autonomous and Connected Vehicle Testing using Infrastructure Sensor Measurements

Public report

Project within FFI Electronics, Software and Communication
Author Katarina Boustedt, editor; project team, contributors
Date 31 Mar 2023

Jan 2019

FFI in short

FFI is a partnership between the Swedish government and automotive industry for joint funding of research, innovation and development concentrating on Climate & Environment and Safety. FFI has R&D activities worth approx. €100 million per year, of which about €40 is governmental funding.
For more information: www.vinnova.se/ffi

Contents

1 Summary

Today, there are cameras monitoring traffic at strategic locations in the Swedish road system. This project used data from such cameras to automatically create realistic test scenarios. In combination with driver behaviour modelling and prediction, these scenarios will improve and simplify the testing of functions for autonomous vehicles. This leads to increased safety and fewer persons killed or seriously injured in traffic.

Sensors placed in the infrastructure, cameras in this project, was gathering traffic information which then serves as input to generating scenarios for testing autonomous driving. In essence, real traffic situations were then influencing the assessment of future safety functions in vehicles. The outcome was more realistic testing and thus safer vehicles, both to drivers and passengers and persons outside the vehicle in question. The resulting information could also be valuable to for example authorities and cities when planning new roads and evaluating sections of infrastructure which are accident-prone and should be reconstructed. The hope is to assist the authorities with AD function safety assessment, providing relevant feedback to stakeholders to make a judgement call before releasing AD function to market.

The consortium consists of partners representing the relevant toolchain for a successful implementation. Viscando is supplying cameras and the related know-how. Chalmers contributes with driver behaviour modelling. Zenseact provides the project with a vehicle platform to test with a function included. Volvo supplies closed-loop simulation integration and evaluation of scenario formats and driver behaviour models. The coordinator is AstaZero, implementing the scenario generation in its test track offering.

The project has run from 1 April 2021 to 31 March 2023, with a total budget of SEK 13 136 844 of which SEK 6 567 391 in public funds.

2 Sammanfattning på svenska

Idag finns kameror som övervakar trafiken på strategiska platser i det svenska vägsystemet. Detta projekt har använt data från sådana kameror för att automatiskt skapa realistiska testscenarier. I kombination med modellering och förutsägelse av förarbeteende kommer dessa scenarier att förbättra och förenkla testningen av funktioner för autonoma fordon. Det leder till ökad säkerhet och färre döda eller allvarligt skadade i trafiken.

Sensorer placerade i infrastrukturen, kameror i detta projekt, har samlat in trafikinformation som sedan fått fungera som input för att generera scenarier för att testa autonom körning. I huvudsak påverkar verkliga trafiksituationer då bedömningen av framtida säkerhetsfunktioner i fordon. Resultatet är mer realistiska tester och därmed säkrare fordon, både för förare och passagerare och personer utanför fordonet i fråga. Den resulterande informationen kan också vara värdefull för till exempel myndigheter och städer när de planerar nya vägar och utvärderar delar av infrastrukturen som är olycksbenägna och bör byggas om. Förhoppningen är att hjälpa myndigheterna med AD-funktionssäkerhetsbedömning, ge relevant feedback till intressenter för att göra ett bedömningssamtal innan AD-funktionen släpps ut på marknaden.

Konsortiet består av partners som representerar den relevanta verktygskedjan för en framgångsrik implementering. Viscando levererar kameror och tillhörande know-how. Chalmers bidrar med modellering av förarbeteende. Zenseact förser projektet med en fordonsplattform att testa med en funktion inkluderad. Volvo levererar simulering, integration och utvärdering av scenarioformat och förarbeteendemodeller. Koordinator är AstaZero, som implementerar scenariogenereringen i sitt testbaneerbjudande.

Projektet har pågått från 1 april 2021 till 31 mars 2023 med en total budget på 13 136 844 kronor, varav 6 567 391 kronor i offentliga medel.

3 Background

3.1 Infrastructure-based traffic sensors

Using Infrastructure-based traffic sensors to collect data has been done before, such as in the public dataset collected by US Department of Transportation, NGSIM [1] which is used in traffic behaviour research [2]. However, this dataset is just 90 minutes in total, too short for any statistically certain conclusions on scenario parameter occurrences. Furthermore, it does not offer possibility to acquire up-to-date traffic data from new locations, a requirement for AV projects.

Traffic surveillance cameras have been used to extract scenarios and behaviours [3]. This gives large amount of data from multiple cameras but has serious drawbacks: low quality of video and limited number of camera locations, and the longitudinal position and speed accuracies of detected objects using single cameras are low.

This inaccuracy has been mitigated in a recent study [4] where an experimental setup based on data fusion between camera and traffic surveillance radar was developed, achieving improved accuracies of object tracks in a highway merge scenario, at prototype level.

Several companies offer off-the-shelf systems for automated traffic data collection and analysis, for example Bosch [5]. However, these systems are primarily dimensioned for traffic analysis rather than recording single trajectories, which limits their potential usage for the AV data collection.

The project has used Viscando's stereovision and AI based sensor OTUS3D for 3D measurement of traffic and extraction of accurate trajectories of different object types, such as light and heavy vehicles, bikes, and pedestrians. The design of this system makes it hard to locate by the road users, eliminating any influence on their behaviour, which resolves a reported drawback of infrastructure-based sensors [6].

3.2 Driver behaviour models

The use of driver models in vehicular system testing was revolutionised in the 2010s and has since then been a very important tool for quantitative simulation of driver–vehicle–environment (DVE) scenarios. Chalmers, together with Volvo Cars and other actors, has since then participated actively in the research, especially through successful FFI projects such as QUADRA, dnr 2009-02766, and QUADRAE, dnr 2015-04863, for the purpose of increasing the understanding of human behaviour related to active safety and complex driving scenarios. Important contributions from Chalmers have been both to map the state-of-the-art [7] specifically related to critical near-crash scenarios, and to find fundamental properties of driver behaviour in novel basic research [8, 9]. Important findings were both related to the actual control of the vehicles, solving the 70-year mystery related to the *remnant* in human control [10], as shown to originate from intermittent *reaching* patterns when controlling the vehicle, and the importance of visual cues related to retinal flow and its special case for longitudinal control referred to as looming.

These findings are still considered state-of-the-art, and research is currently active to further explore the biological roots of driver behaviour, for the purpose of building quantitative driver models. One such active track of research is the influence of active driver gazing [11], and how that relates to retinal flow and driver behaviour. Another example of a similar research direction is how to better understand reaction times [12], where they would no longer be an effect of a symbolic and atomic event in the environment, but rather an effect of an accumulation of deviations from the expected. From these recent findings, it is clear that low-level visual ques are crucial for the understanding of human behaviour, rather than the traditional high-level object and trajectory-based approach. In the traditional approach, the behaviour of a single actor was very much seen from the perspective of a full cognitive awareness of the traffic environment and its future developments, where each other actor was clearly represented in the cognitive perceptual domain. However, by using the recent findings from retinal flow and active gaze modelling it can clearly be seen that driver action do exist without object-based perception, and that the combination of the retinal-flow field may influence driver action in ways that would not be predictable from a traditional objectification of the driving scene.

In this project, the modern approach of seeing human behaviour as an effect of low-level sensation, without mandatory conceptualisation of the traffic environment, will be employed for the purpose of (1) generating state-of-the-art realism in the model behaviour, and (2) to further bring the research and state-of-the-art forward.

3.3 Closed-loop simulations for development and verification of AD functions

Computer simulations play a crucial role in the development and verification of AD algorithms as it is impractical or even impossible to prove their safety by real driving [13]. Closed-loop simulations allow for virtually driving a great number of traffic scenarios by running in parallel on computer clusters, enabling efficient testing of AD functions. The use of such simulation environments is now common practice among automotive OEMs and start-ups aiming at the commercialisation of AD products [14].

Because of the many different players and methodologies involved in AD testing, the adoption of open standards is of key importance as it enables a shared understanding of the safety argumentation among industries and regulatory bodies. OpenDRIVE [15] and OpenSCENARIO [16] standards are among the most established standards to represent the static and dynamic content of simulation scenarios. The suitability of these two standards for industrial and research use cases was investigated by the FFI project Simulation Scenarios [17]. One of the deliverables of the project was the scenario engine esmini [18], an open-source, cross-platform application to run OpenSCENARIO and OpenDRIVE files. The application is currently being maintained and improved in functionality by the open-source community, with major contributions from ASCETISM project partners.

The traffic scenarios investigated by the Simulation Scenarios project include interaction between the host vehicle and other traffic actors in terms of scripted, pre-determined interactions (e.g., triggered manoeuvres, platooning), but do not investigate how to include accurate driver models in the OpenSCENARIO formulation. The proposed project will address this gap by investigating how to include accurate driver models into the OpenSCENARIO formulation and how to execute AD-relevant scenarios in the esmini environment.

3.4 Scenarios and standards

Scenario based verification and validation of autonomous driving (AD) functions is a popular area undergoing a lot of change. Large efforts are made towards harmonisation of test and evaluation processes for AD function validation. An example of this is the Pegasus project [19] which was completed in the autumn of 2019. A result of the project was a process for function verification called "The Pegasus Method" [20]. Based on data collected from real world test driving and accident databases, scenarios can be reconstructed and represented in a common format. The scenarios can then be used as part of validation test schemes aimed at safety assurance argumentation, where the AD functions are evaluated on a set of parameterised scenarios. The project aligned with other scenario format initiatives, such as OpenDRIVE [15] and OpenSCENARIO [16] to make use of already existing open-source formats. Today, both these formats are developed by ASAM [21] where the first official version has been released and ongoing work for the second version is on its way.

In 2019, a new ISO standard working group was formed to streamline the work towards safety assurance argumentation. The working group ISO/TC 22/SC33/WG09 "Test scenarios of automated driving systems" [22] has since then continued to work towards a scenario-based approach of verifying and validating AD functions, using the results from the Pegasus project. The group is looking to release four standards by 2023.

Work done in the Swedish national FFI research project ESPLANADE [23] has shown that certification of AD functions through already existing standards, such as ISO 26262 [24], is not feasible. The project, which delivered its final report in 2019, argued that a plausible way forward in evaluating safety is to apply a method called quantitative risk norm (QRN). QRN defines an acceptable frequency of incidents with different levels of severity and a mapping of incidents to different classes of definitions. This type of argumentation could lead to faster exposure of AVs on national roads, where an acceptable number of incidents are allowed. A scenario-based approach could potentially support this method by examining the incidents occurring in a defined set of testable scenarios in an ODD, evaluated in simulation and test track.

Tools for testing autonomous functions have been prototyped in the FFI project Chronos part 2, Dnr 2017-05501 [25], where the unpredictability of AVs must not make testing unfeasible. Based in scenarios, the tools developed showed capability of synchronously adapting actors to the action of the VUT on the test track, thus enabling the running of complex scenarios that could only be set up in simulations before, in a safe and repeatable way. Using these tools for test track execution will be an important part of the scenario format and driver behaviour model evaluation within this project.

4 Purpose, research questions and method

Scenarios and driver behaviour models were derived from recorded data and scrutinised for integration into simulation and test track testing. This project answered the following research questions:

- How can naturalistic data from infrastructure sensors be used in scenario-based verification and validation of autonomous vehicles?
- How do data properties, such as location, quantity, accuracy, number of recorded interactions, influence quality of simulations?

The partner contributions to the project are illustrated in Figure 1.

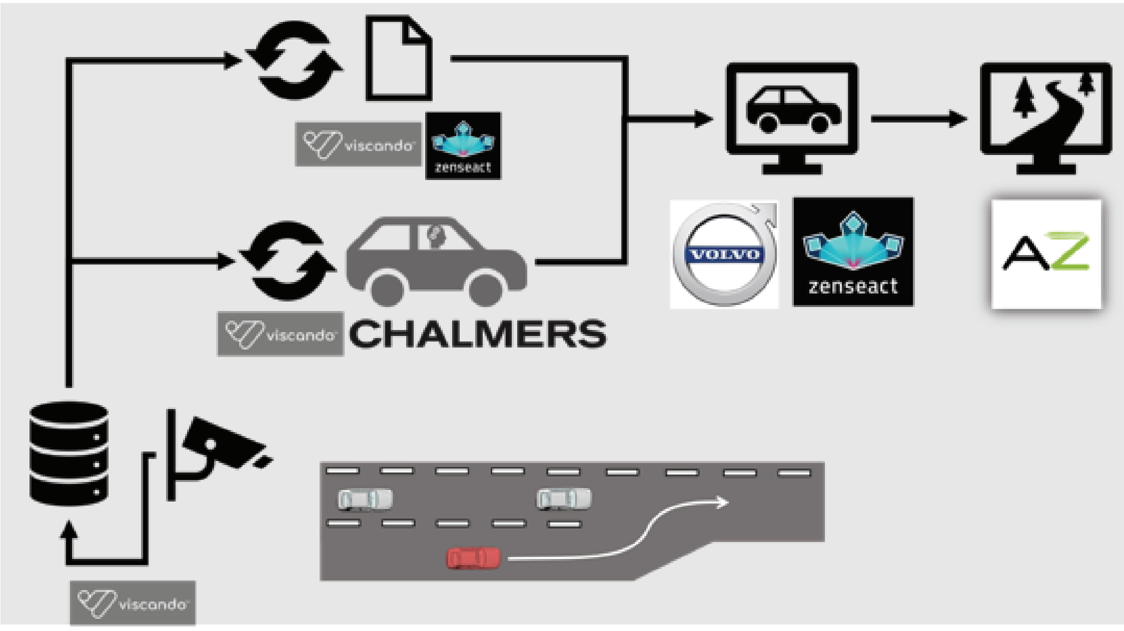


Figure 1. The ASCETISM project structure.

5 Objective

The objectives of the project, as described in the application, were to

1. Identify and deploy roadside sensor-based naturalistic data collection solution to collect a dataset for extraction of critical merging scenarios for AV development.
2. Model the behaviour of human actors in merging situations.

3. Demonstrate the concept of verification and validation toolchain of AVs using extracted merging scenarios and behaviour models.
4. Identify the gaps in the currently available data, such as quality and quantity and required improvements in data collection systems to enable scenario-based verification of AVs.

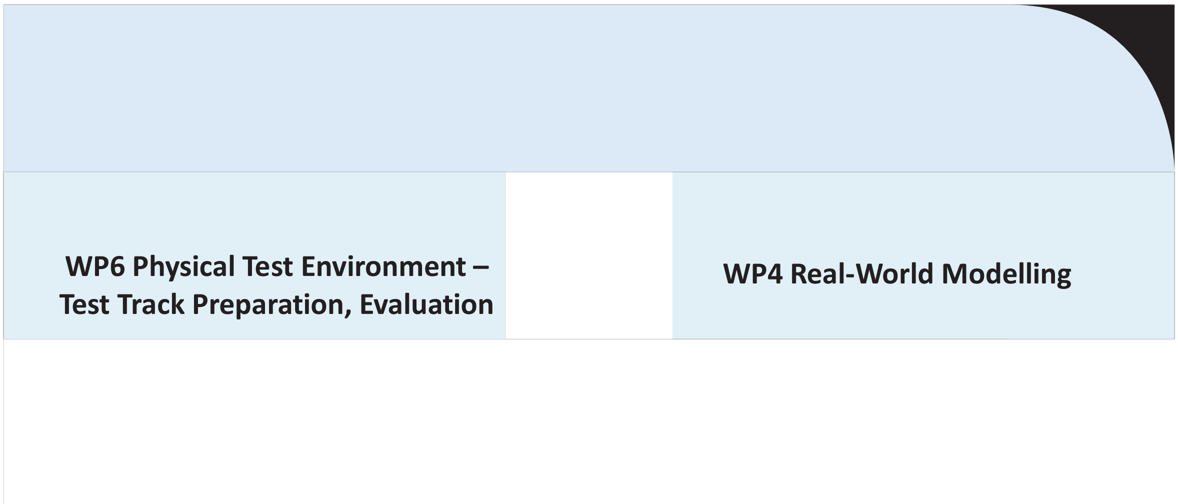


Figure 2. Work package structure of the ASCETISM project.

6 Results and Deliverables

6.1 The Chosen Scenario – Motorway On-Ramps

Requirements for data collection location have been derived as an outcome from workshops, project meetings and email communication between project members. The following aspects were lifted as the most important:

- Relevance for state-of-the-art AD technologies and intended operational design domains for motorway AD.
- Technical and economic feasibility of sensor installations and data collection – which would maximise the volume of collected data in the budget frame of the project.
- Possibility to generalise the collected data on other on-ramp locations.

6.1.1 Number of Locations

The project parts have decided to focus on a single location and maximise the amount of collected data within project budget, rather than run several smaller data collections in different locations.

Large amount of collected data allows for high statistical certainty in extracted scenarios and behaviour models. It will also allow studying of required amounts of data, potentially leading to reduced data collection needs in the following projects.

Transferability of results to other similar locations, which will require data collection in more places and comparison of scenarios and behaviour models, will be the focus of the follow-up projects.

6.1.2 Geometry and Traffic Rules

To maximise the relevance of collected data for the complete motorway ODD rather than a single specific ramp, the most common type of the motorway ramp was optimal for data collection. This transferred into the following requirements:

- Straight or almost straight motorway section without slope
- The geometry of the ramp in the merging section should be as depicted in Figure 3. The ramp becomes a full-width separate lane (“acceleration lane”) before merging with motorway, which gives incoming vehicles a runway for acceleration and smooth merging.
  The length of the merging section of the ramp should be between 50-100 meters. Shorter ramps without sufficient acceleration section are rare on motorways in Sweden - and on such ramps, it is more common that incoming vehicles stop and yield instead of accelerating to motorway speed and merging.
- There should not be any complex road infrastructure that can reduce the speed or limit the possibilities for merging and lane change (e.g., solid lines between motorway lanes, new ramps, lane split or merge, tunnels, or speed limit change). Figure 3 shows an example of a location that is unsuitable for these reasons (northern entrance to Tingstadstunneln in Gothenburg).
- The speed limit on the motorway should be at least 70 kilometres per hour.

Figure 3. Left: a schematic view of a ramp layout chosen for data collection. Right: An example of a ramp that is deemed unsuitable for data collection. (1) there is an entrance to a tunnel, (2) the acceleration section is not long enough, and its width starts d decreasing immediately at the start of merging section, and (3) the solid road marking prevents vehicles in the left lane to change it. Image source: Eniro, © Lantmäteriet/Infotrader i Ronneby AB.

6.1.3 Traffic Properties

- For capturing as much variation as possible, the traffic amounts should vary. Both light traffic, heavy traffic (including slow-moving queues) should take place at different times.
- Variety of vehicle types on the ramp. Occurrence of heavy vehicles and buses on the ramp is desired.

6.1.4 Geography

- Search for suitable location was in the first place limited to the area around Gothenburg.
  This is to reduce the cost and time of attending the systems for retrieving the collected data, and for collection of reference GNSS data using Zenseact test vehicle.

6.1.5 Roadside Infrastructure:

- Because of a long data collection, batteries could not be used. The location had to allow access for permanent electricity source.
- The location had to have poles suitable for installation of roadside sensors, e.g., lampposts. The alternative, setting up temporary poles, is not always possible due to regulations on motorway traffic safety and roadside layout; moreover, it implies expensive installation and high renting costs.

In the end, the lampposts are the best alternative as they allow both installation at optimal height (between 7-9 meters over ground) and access to electricity that feeds the lamps.

6.1.6 Authority Approval

The last but the most important requirement is that the installation and data collection is approved by the responsible road authority.

6.2 Data Collection

6.2.1 Chosen Location

After a thorough examination of the available motorway on-ramps in Gothenburg area, one was chosen as most suitable according to requirements listed above.
Street and satellite view images of this motorway crossing, named Fiskhamnsmotet, are shown in Figure 4. Western direction was chosen.

Being in central Gothenburg, it is characterised by large traffic volumes with peaks in rush hours (which is known by personal experience by several persons involved in ASCETISM). Moreover, it is the connection of the ferry terminal to the motorway, so large numbers of trucks are entering the motorway when a ferry arrives.
The speed limit on this section of the motorway is 70 kph.

The illumination poles placed in the middle of the motorway are suitable for installation of sensors. After presentation of the project scope and the installation layout, the Swedish Road Administration (Trafikverket) approved both installation of sensors and connection to the illumination power supply. Viscando expresses gratitude to Trafikverket for the approval and support in the installation process.

Figure 4. Fiskhamnsmotet, the on-ramp to Oscarsleden, Gothenburg, chosen as measurement location. Left: Street view of the western direction of the motorway, with the relevant on-ramp on the right. Right: Satellite view of the same location, with suitable poles marked in red, and the road stretch where measurement was desired shown in pale orange. Image source: Eniro, © Lantmäteriet/Infotrader i Ronneby AB.

6.2.2 Measurement System: 3D vision and AI-based Infrastructure Sensors OTUS3D

The solution for location-based data collection is a grid of Viscando's own developed infrastructure sensors, OTUS3D. These sensors are installed on stationary infrastructure and use 3D vision and artificial intelligence to detect, classify and track all types of road users (currently, supported classes are pedestrians, two-wheelers, light and heavy vehicles).

Figure 5. Left: a photo of Viscando OTUS3D sensor installed on a pole. Right: example of tracking of different road users in an urban environment.

Vision data are processed in the embedded computational unit and removed within 20 ms from being captured. Thus, fully anonymous data comprising object positions, velocities, 3D rectangular bounding boxes, and road-user types are stored, ensuring full GDPR compliance because any personal information was removed. However, in research and development projects, there is also a possibility to store low-resolution and anonymized video, where identification of persons and vehicles is impossible.

6.2.2.1 Viscando Data Content

Viscando delivery to scenario analysis and behaviour modelling work packages consisted in datasets of road user object tracks. Each detected vehicle has an individual trajectory (with a unique object ID), and the following measured 20 times per second during the whole period the object was observed:

- Local position (in ground-based coordinate system)
- GNSS-based georeferenced position
- Velocity over ground
- Lane assignment (left lane/right lane/ramp)
- D# bounding box: length, width, height, direction

Figure 6 illustrated the trajectories plotted over a bird-view of the measurement area, and bounding boxes plotted both in top-view and in camera-view.

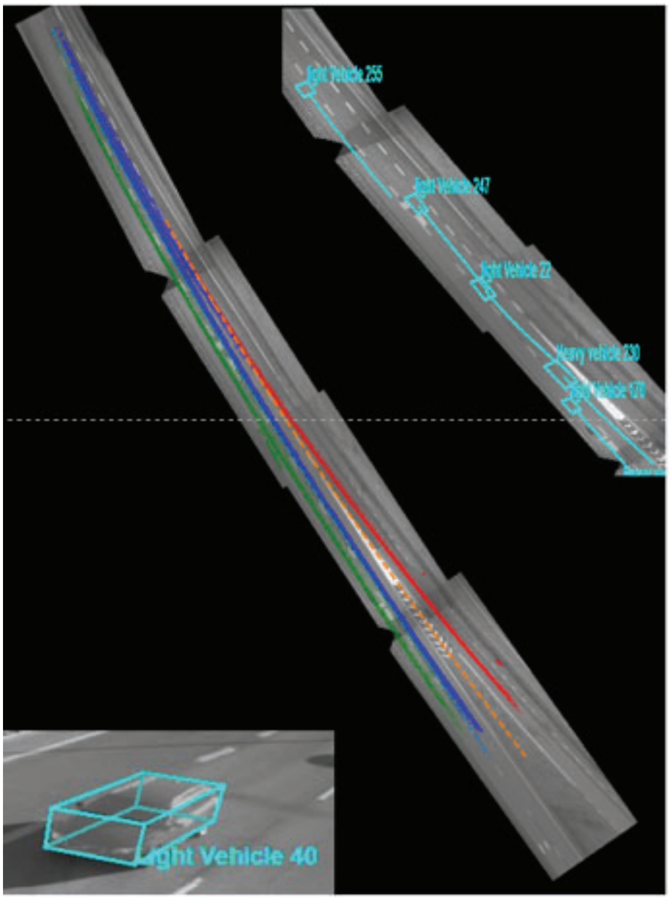


Figure 6. Illustration of the object track datasets delivered by Viscando. Middle: tracks plotted over a bird-view of the measurement road stretch. Blue, green, and red colours are used for tracks in the right lane, left lane and the ramp. Top right: bounding boxes of vehicles in bird-view. Bottom left: Example of identified 3D bounding box.

6.2.2.2 Data Processing

Figure 7 shows the data processing steps in Viscando 3D&AI sensors.

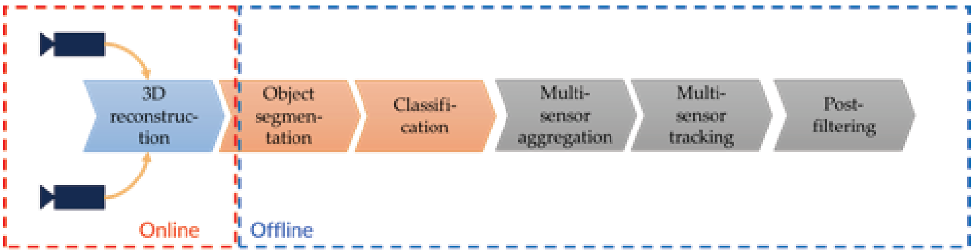


Figure 7. Data processing toolchain for Viscando 3D&AI sensors

3D stereovision processing of data was done online in the sensors, while the further steps leading to object tracks required for scenario analysis and behaviour modelling, were done at Viscando servers.

The processing implied the following steps:

- Segmentation of individual objects from the 3D point cloud
    Bounding box estimation: computation of an accurate 3D bounding box representation of vehicle geometry, was done in this step.
- Classification of the object types
    Viscando applies the following object types: pedestrian, two-wheeler, light vehicle, heavy vehicle, unknown.
- Aggregation of detection data from individual sensors
- Filtering of noise detections and detections coming from outside the area of interest
- Multisensor object tracking, which implied identification of detections belonging to the same object moving along the road, and Bayesian computation of their positions and velocities, considering inaccuracies in both detection positions and kinematic motion parameters for the objects.
- Post-filtering of the tracks: removing the tracks that obviously did not represent consistent vehicle tracks.

While it is possible to run most of the steps online in the sensors in real time, offline processing was chosen in this project. This allowed to re-process already collected 3D data with improved algorithms as those were developed and introduced during the project.

6.2.2.3 Improvements in Sensor Data Processing in the Scope of the ASCETISM Project

In the course of the project, Viscando has been working on improvement of all steps in the data processing toolchain, in order to increase the accuracy of the trajectory data for AD applications. The following are the most important improvements introduced within the scope of the project:

- Improved separation of relevant objects from background, and between individual objects in 3D point clouds yielded from stereovision algorithms.
- Improved tracking algorithms for multisensor grids, through more efficient processing of detections in the zones where sensor fields of view overlap
- New primary classification method introduced, based on Histogram of Gradients (HOG) and Support Vector Machine (SVM). Moreover, deep neural network (DNN) based classification has been tested using several openly available and pre-trained networks such as Detectron v.2 and GluonCV. To simplify future transition to DNN classification, a framework for integration of DNN into Viscando onboard computational unit developed.

It is worth to mention that the effect of the improvements had a strong effect on the need for post-filtering of the trajectories. Initially, many incomplete tracks with obviously unrealistic trajectories could be observed, as seen in Figure 8 (left). The primary reason for these was presence of large numbers of false detections stemming from noise, vehicles whose detections were split in multiple objects, etc. The post-filtering had to remove noise tracks based on the following criteria:

1. Unrealistic accelerations
2. Unrealistic direction of travel (with a large angle to the motorway direction)
3. Start- and finish points of a track (it should cover sufficiently large part of the motorway or/and ramp)

After introduction of improvements mentioned above, considerable improvement was observed (Figure 8, right). This alleviated the need for post-filtering, so only criterion 3 (sufficient track length) was applied in the final data processing.

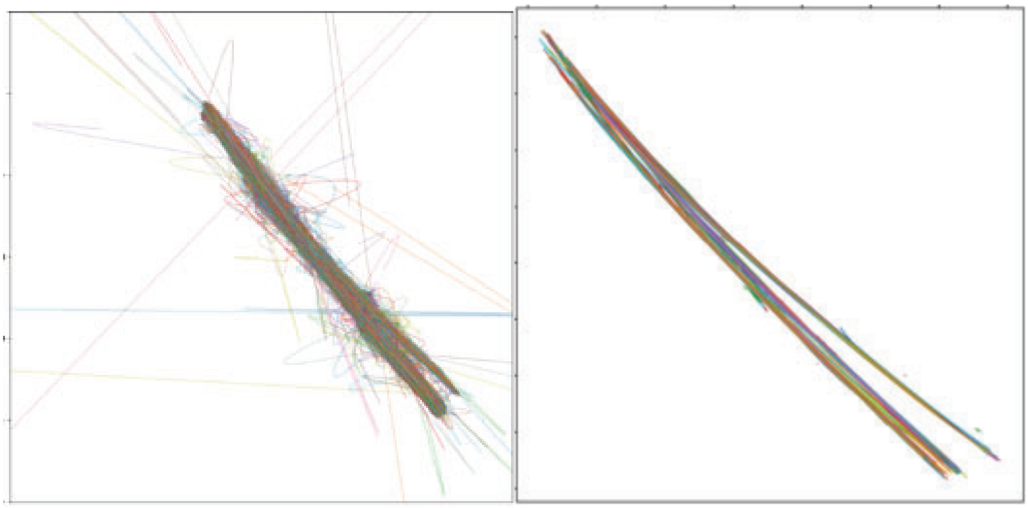


Figure 8. Examples of top-view plots of vehicle trajectories over a time period, before post-filtering. Left: before introduction of improvements in data processing; right: after introduction of improvements.

6.2.3 Motorway Measurement

To ensure desired coverage of the motorway and the on-ramp, a layout with four sensors installed on two poles was chosen. Figure 9 shows the exact location and orientation of sensors, as well as the combined field of view.

The total length of the motorway/ramp covered by Viscando sensors was 240 meters, from approximately 100 meters before the first point where merging is allowed to the point where the ramp/acceleration lane completely disappears. Therefore, both behaviour of vehicles adapting their speed in advance before the ramp, and late merges could be captured by the measurement system.

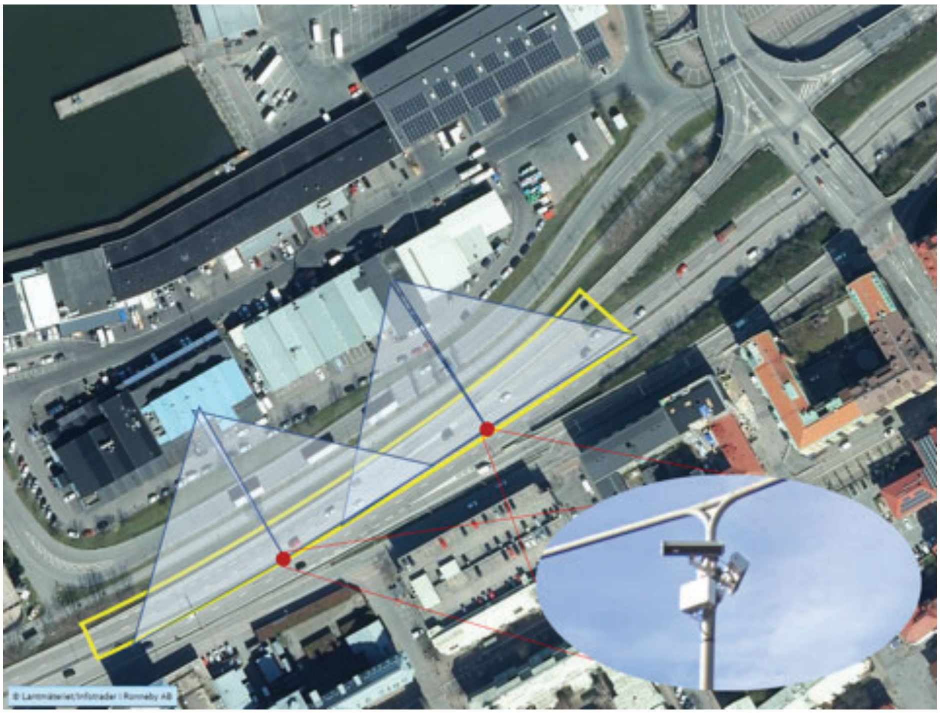


Figure 9. Measurement stretch of Oscarsleden motorway, Fiskhamnsmotet, western direction. Yellow contour shows the measurement area. Blue triangles show the directions and approximate coverage of the four installed sensors. Red circles are the illumination poles. Inlay shows a photo of installed sensors and WiFi box on the pole. Background image source: Eniro, © Lantmäteriet/Infotrader i Ronneby AB.

The collected data (3D data after stereo-computation) was transferred from each individual sensor via local WiFi to a nearby logging unit, which is then brought to Viscando office where the data is downloaded to the server and post-processed. Post-processing involves offline object tracking across all sensors and additional filtering steps.

6.2.4 Data Delivery and Validation

6.2.4.1 Georeferencing

For georeferencing and validation purposes, Viscando tracking data was compared to trajectories of Zenseact vehicle measured using precise on-board GNSS unit from Oxford Technical Solutions with down to 2 cm position accuracy.

Georeferencing implied that the local ground-based coordinate system of Viscando measurement setup was matched to GNSS coordinates. This allowed delivery of trajectories in GNSS coordinates by Viscando measurement system.

6.2.4.2 Data Extent and Distribution over the Year

During the period the sensors were mounted at the measurement stretch, September 2021 to November 2022, the data collection was run under 5 periods, as shown in Table 1. Note that the measurement times over day were different in different periods. These were decided so that the times with major traffic are included, while the night-time measurements are limited (as night-time traffic was not deemed relevant in the project).

Table 1. Summary of measurement periods

Date		Time				
Start	Stop	Start	Stop	Diff	Nof days	Nof hours
2021-09-22	2021-09-28	04:00:00	20:00:00	16:00:00	7	112:00
2021-11-09	2021-12-17	06:30:00	16:00:00	09:30:00	39	370:30
2022-03-02	2022-03-31	05:30:00	18:00:00	12:30:00	30	375:00
2022-04-23	2022-05-10	04:00:00	18:00:00	14:00:00	18	252:00
2022-06-16	2022-07-04	02:00:00	19:00:00	17:00:00	19	323:00

Totalt	1432:30

In total, 1432 hours of data were collected for 114 days. Despite uneven distribution of the hours over different months (Figure 10, left), the data is balanced over the summer vs. winter period (Figure 10, right).

It is worth to note that the estimated number of vehicles passing the measurement location per day is between five to six thousand. One can roughly estimate the total recorded driving distance over the course of data collection, to 100000 km.

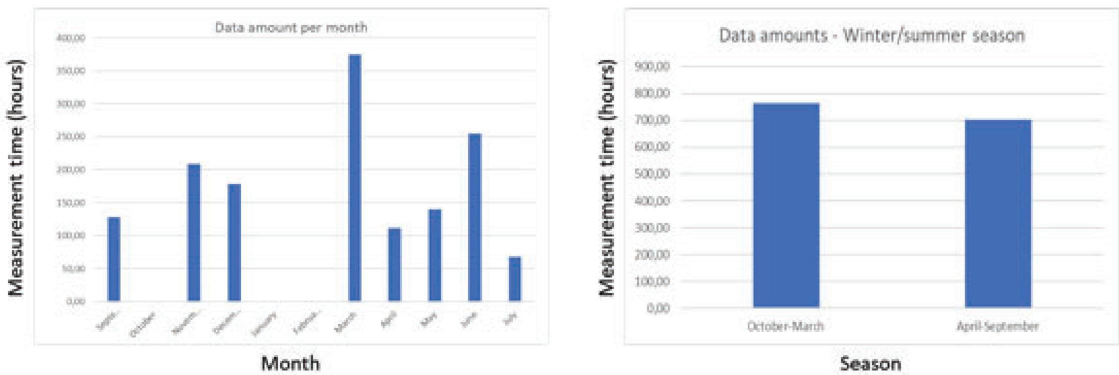


Figure 10. Distribution of measurement time over months (left) and seasons (right)

6.2.4.3 Validation of Viscando data accuracy

Zenseact vehicle position measured by Viscando sensors was compared to accurate GNSS position data to evaluate the accuracy of the former. It was found that the average position error was around 0.5 meter, as shown in Figure 11.

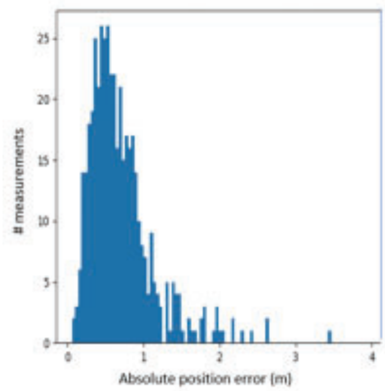


Figure 11. Absolute position error for Zenseact vehicle, calculated as difference between positions measured by Viscando sensors and the precise GNSS unit onboard the vehicle.

Moreover, completeness of the trajectory data was assessed using comparison of the measurements to a manual annotation of all objects in an 8-minutes anonymised video.

Of 200 vehicles identified in annotation, the data after post-filtering contained 169 tracks, that is, 15.5% less than the correct amount. The total number of recorded tracks before post-filtering contained 260 tracks, some of which represented stationary objects not relevant to the motorway traffic, trailers, or vehicles which the system failed to track as a single object throughout the whole motorway stretch. The latter part was identified as important to improve in the following projects, as some vehicles (and hence some interactions) can be missed, which may have an influence on scenario analysis and behaviour modelling.

6.3 Scenario Extraction, Scenario Catalogue; Safety Surrogate Measures and Completeness Analysis

The concept of Operational Design Domain (ODD) plays a substantial role in a scenario-based verification framework to evaluate the performance of the automated functionality in motor ways. The notion of the ODD is used to confine the limits for development and verification of Automated driving systems (ADS) [26]. It defines a trade-off between the availability and the implementation cost of the ADS. This helps to avoid unnecessary safety margins and thus enable the development of a cost-efficient and yet performant ADS. In [26], it is proposed that the ODD should be defined using statistical models, in contrast to physical limits of scenarios, based on collected data. In order to get the full benefits of such a statistical approach, there exists few crucial factors which should be kept in mind when designing the scenario ontology:

1. Scenarios are collectively exhaustive (i.e., complete with respect to data), and
2. Scenarios can quantitatively represent the available data from real traffic,

3. Scenarios can be extrapolated to generate rare scenarios that cannot be observed in the data.

In the past researchers have proposed different approaches to achieve such completeness. K. Czarnecki [27, 28] defines an operational world model (OWM) which can be used to define the ODD. The OWM defines a holistic model for the world around the ego vehicle. However, the OWM for road users does not allow a full quantification of traffic scenarios. [29] presents a model-based scenario specification approach to make the temporal and spatial description of the driving scenarios. However, this approach is not detailed enough to be able to populate scenarios from data. Bagschik et al. [30, 31] proposes a knowledge-based ontology to produce scenes for testing of automated vehicles. The ontology is formulated based on both data as well as expert knowledge. In a similar fashion, Wheeler et al. [32, 33] develop a factor graph in order to generate scenes with distributions similar to the true distributions. Here, a scene refers to “the arrangement of vehicles on a given road network”. However, it is not clear how the time evolution is specified for the generated scenes to form a scenario. The focus is on creating realistic simulations and not to create a complete ODD model for analysis.

Gyllenhammar et al. [26] present a detailed model for the actions of vehicles in the surrounding of ego vehicle when driving on motorways. They define a set of fundamental vehicle actions for motorway driving which quantitatively captures the driving interactions of vehicles. This approach provides a modular framework for analysis of field data. The fundamental actions proposed are derived by acknowledging the fact that a vehicle can move in the direction of travel, i.e., longitudinally, and perpendicular to the direction of travel, i.e., laterally. The actions proposed are possible to combine to create multi vehicle scenarios. The main benefits of the proposed fundamental actions are that they (1) are based on dynamics describing the elementary movements of the vehicle, and (2) provide a modular framework. These benefits not only facilitate completely modelling vehicle interactions on highway motorways, but also allow for analysis of correlation and independence of scenarios at an arbitrary granularity level. Also, the proposed fundamental actions support the definition of the ODD and thus can be used to construct simulation scenarios for validation and also to estimate and monitor the residual risk after the introduction of ADS to market.

The PEGASUS project is a European initiation to build up a scenario database as part of a sign-off process for the launch of ADS. In [34], the vehicle scenarios are defined by considering a hypothetical collision between the ego vehicle and another object, so called the challenger. The initial position of the challenger in the scenario and the final collision position with the ego vehicle is used to define nine base scenarios. This set of base scenarios present a “complete modelling” of potential collision scenarios. To extend the base scenarios to include the contribution of additional objects, the authors propose that the additional objects might make a scenario more difficult by (1) constraining the actions of the ego vehicle, (2) occluding the challenger, (3) simultaneously acting as a challenger and (4) conducting actions that eventually lead the challenger to create the challenging situation. Weber et al. [35] tackle the combinatorial complexity by focussing on modelling the actions of the vehicle challenging the ego vehicle. The remaining objects in the surrounding of the ego vehicle are modelled with lower fidelity to capture the essential impact that they might have to the scenario.

TNO is another initiative to build up a huge database of scenarios though join collaboration of big industrial actors. To this end, they propose the Streetwise approach which gives some suggestions on how tree structures can be used to classify field data into scenarios and events and, hence, describing the format of entries to scenario database. However, this initiative misses to address how the scenarios can be compared, analysed, and extrapolated.

6.3.1 Scenario Mining from Data

The scenario of interest in this project is the on-ramp merging scenario with a vehicle merging onto the mainline from an on-ramp lane. Because the merging vehicles can influence the traffic flow, and because those situations can readily become critical due to reasons like occluded field-of-view, a short on-ramp lane, high occupancy of the mainline, and high relative velocities.

On-ramp merging manoeuvre can be treated as a lane change scenario but in a different context. In fact, a vehicle performing on-ramp manoeuvre is assumed to drive on the acceleration lane of a highway's on-ramp section and performs a lane-change to the mainline. In (Klitzke, Gimm, Koch, & Köster, 2022), the on-ramp scenario is considered to be comprised of several sequential driving states, the driving primitives, i.e., idle, approach, cross, and change. This allows to break down the complexity of such manoeuvres to basic driving actions.

A vehicle can move arbitrarily between the driving primitives. For example, a driver on the acceleration lane can have an incorrect estimation of the speed of the vehicle in the mainline and, thus, need to abort the merging manoeuvre and swerve back to the on-ramp lane. Therefore, not only the completed merging manoeuvres are of interest but also the aborted ones.

To identify the on-ramp scenarios from data, [36] proposes to use Hidden Markov Models (HMMs) with hidden states are the driving primitives and the transition probability between the hidden states are identified based on the data. However, the method fails to identify around 5% of the merging scenarios available in the dataset.

Although, the use of unsupervised techniques is motivated to identify a specific type of scenarios from accelerated dataset, but generally these methods suffer from a lack of accuracy and, hence, one might miss to identify rare scenarios from the dataset. Therefore, in this project we propose to use a rule-based technique to identify on-ramp scenarios from data. The triggering condition to look for an on-ramp scenario, is when the vehicle driving on the acceleration lane crosses the lane marking. The start of the scenario is considered to be the location of the vehicle three seconds before crossing the lane marking and the end of the scenario defined as the moment when the vehicle is fully merged to mainline on highway, i.e., the lateral velocity relative to road approaches zero.

6.3.2 Safety Surrogate Measures to Investigate Conflicts in Motorway On-Ramps

The existing surrogate safety measures cannot fully evaluate the level of conflict on motorway on-ramps. To overcome these shortcomings, [37] proposes a new metric called Conflicting Merging Headway (CMH), which is defined as the time interval between a ramp merging vehicle arriving at the merging point (T_1) and the time that mainline vehicle that directly follows it and arriving at the same position (T_2). This measure can be viewed as a variant of Post Encroachment Time (PET) as it adopts the idea time difference between moments that the two vehicles occupying the same conflicting area. However, in CMH, the merging point is a spatial zone that the conflict can begin to arise, Figure 12.

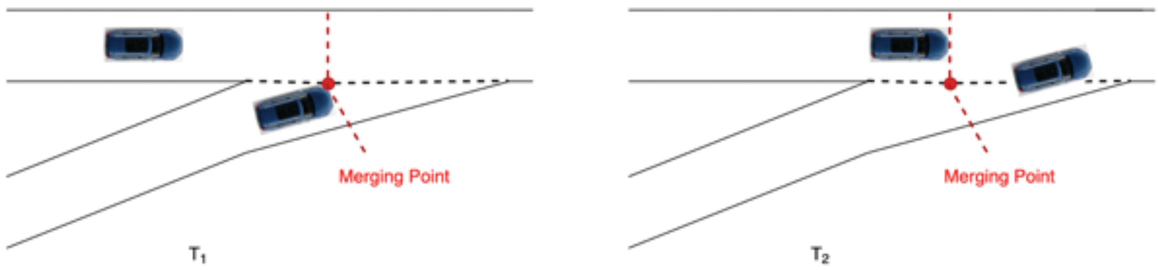


Figure 12. Illustrating T_1, the time at which the vehicle in highway merge arrives at margining point, and T_2, the time at which the vehicle at highway mainline arrives at the merging point.

6.3.3 Data Completeness

Before an ADS can be released to public roads, one needs to make sure that the traffic scenarios to which the ADS will be exposed to in the intended ODD are known sufficiently well. This necessitates the need for collecting field data. Generally, the data collection during the lifecycle of the ADS consists of four phases. The first phase is the data collection prior to release of ADS. The purpose of this phase is to quantify the traffic scenarios with sufficient accuracy and estimate the exposure to such scenarios with sufficiently low uncertainty. This phase could in turn be split into two stages, one stage is the data collection with human driver behind the steering wheel, and the second stage is to collect data when the ADS controls the behaviour of the ego vehicle. The purpose of the second stage is to verify the exposure rates obtained from the first phase. The second phase is the data collection after release of AD feature to ensure that the exposure data collected with the AD feature in production vehicles is not statistically different than the data recorded prior to release and to increase the confidence on the exposure estimates. The third phase is to monitor the traffic to detect changes in the traffic environment, potentially due to feature updates or societal changes. The last phase could be to characterise the traffic scenarios outside the boundaries of the release ODD for later ODD expansions or increasing the use case for the ADS.

The data collected in each phase should be gathered according to a Real-World User Profile (RWUP) in the intended ODD. This means that the collected data is representative for the intended ODD. Therefore, once can assume that not only all traffic scenarios are reflected in the collected data but also every important factor affecting the dynamic of traffic scenarios is also captured in the dataset. Given this let us define the data completeness problem as follows:

"Data can be considered complete when the newly added data contains very little new information and only reinforces what is already known/estimated to be known."

In **a statistical sense**, the data collection can be interpreted as sampling from the true but unknown underlying distribution describing the traffic scenarios and, hence, the data completeness is about ensuring that the estimation of the true distribution shall not:

- underestimate or significantly overestimate the probability of unfavourable parameter values
- be shifting (be non-stationary) when the new data batch is included

Thus, the solution to the data completeness analysis boils down to check for the stationarity of the underlying empirical distribution estimated from the collected data. To this end, we propose to first identify the traffic scenarios from the collected data, then quantify the dynamic of each scenario by fitting a six-order polynomial to the trajectory describing the motion of the traffic participants, and finally apply statistical methods to check for the stationarity of the empirical distribution of the parametrized scenarios. The following sections elaborates our approach in more details.

6.3.4 Indicators for stationarity analysis

From the collected data, values of parameters/variables are extracted, which are assumed to represent the characteristics of exposure within an ODD sufficiently well. Examples include:

- Six polynomial coefficients for lateral and longitudinal trajectory of merging vehicles,
- The relative starting position of merging vehicle in relation to ego vehicle,
- The duration of scenario,
- The speed of the go vehicle at the start of the scenario,

- Safety surrogate measures such as PET and CMH,
- Any other combined variable which is a function of all scenario parameters, e.g., projecting scenario parameters to principal components

The analysis discussed here is performed on such parameters/quantities. **The choice of those parameters is of utmost importance** and should be reconsidered from time to time, whether the selected quantities are representative enough for the purpose.

6.3.5 Stationarity analysis

The following statistical methods are used for determining data completeness:

- Jensen-Shannon divergence (JSD)

JSD [38] is a measure, based on Kullbach-Leibler divergence metric, for comparing two probability distributions. JSD takes values between 0 and 1, where the former (JSD= 0) corresponds to distributions that are equal and the latter (JSD=1) distributions that are not similar at all. In practice it may be possible to assume that distributions are sufficiently similar if a threshold (e.g., 0.1) is reached.

- Bayesian parameter estimation (BEST)

The BEST [39] approach, which is based on a Bayesian estimate of distribution parameters, is used for comparing two distributions. In this approach, a t-distribution is fitted to two datasets. From this, one can derive the probability that the t-distribution parameters for the two distributions being equal to each other. If the probability converges to 1, this test concludes that the two distributions are equal.

6.3.6 Analysis Approaches In Different Data Collection Phases

One way to analyse the stationarity of the empirical distribution captured by the collected data is to split the data randomly into two parts and fitting a distribution to both splits. The two distributions can then be compared using statistical methods mentioned above. If the distributions obtained for both splits are sufficiently similar, it can be assumed that the collected data is sufficient to characterize the dynamic of highway scenarios.

When comparing the distributions, it should be noted that the computed metrics using either JSD or Bayesian approach are random. Therefore, it is suggested to use an iterative approach to able to reliably compare the distribution over the course of the data collection. This approach begins with randomly creating two data batches of equal size from the complete dataset, each of which containing very few data points. Intuitively, the probability distribution underlying these two data batches are not similar, and computing the JSD and BEST metric should also confirm this assumption. In the next iterations, more data points are successively and randomly added to each of the data batches until the size of each bath reaches the half of the complete dataset, see Figure 13. At each iteration, the JSD and BEST metrics are calculated for the indicators of interest. The idea here is that if the distributions are similar the value of the BEST and JSD metrics would gradually converge to a value close to 1 and 0, respectively.

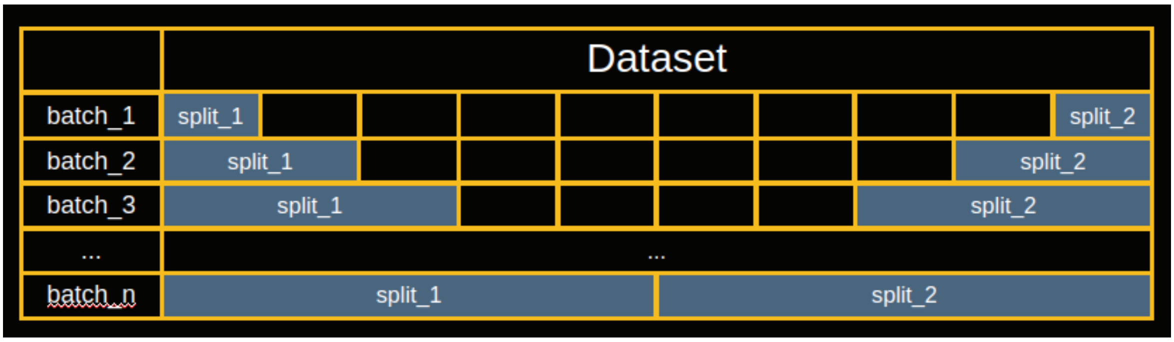


Figure 13. The iterative approaches to calculate the distance metric between empirical distributions for random subsets of data.

6.3.7 Results

This section presents the results for applying the methods mentioned in the previous sections on 192 hours of data collected suing infrastructure camera at Fiskhamnsmotet. The first step to identify the highway merging scenarios from the data is to extract the road geometries from the HAD map, Figure 14, and feed the road boundaries to the scenario identification toolchain using OpenDRIVE format, Figure 15.
The next step is to project data collected by the infrastructure camera to the same coordinate system as the road geometries and feed them to the scenario mining tool to extract the trajectories of the interacting vehicles at highway merges, i.e., the vehicle driving trying to merge to highway and the vehicle on the right most lane on highway to which the behaviour of the merging vehicle is adjusted to. Figure 16 depicts the number of merging scenarios extracted from the data during the first 19 hours.

Figure 14. Extracted road boundaries from the HAD map overlaid on the helicopter view of the data collection site taken from Google maps.

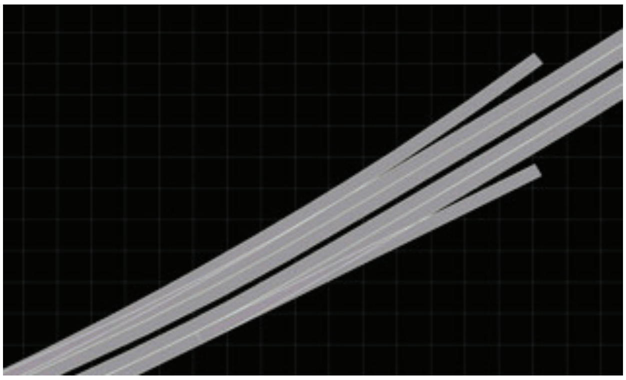


Figure 15. Road boundaries constructed using the OpenDRIVE format visualised in OpenDRIVE Viewer.

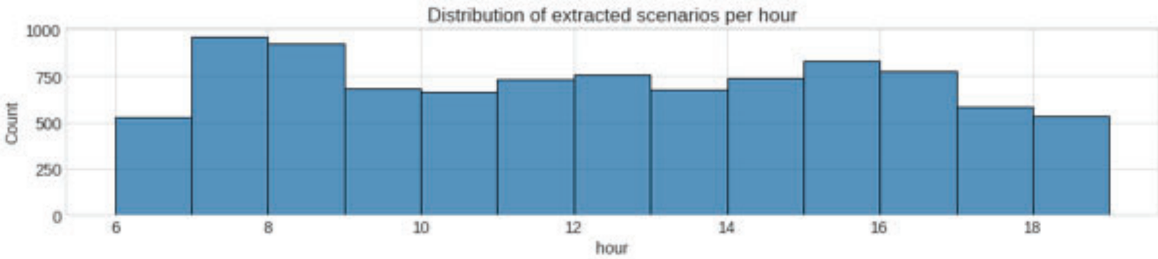


Figure 16. Number of merging scenarios identified in the 19 hours of data collection.

As mentioned in the previous section, one needs to first quantify the mined scenarios using polynomial fitting. Based on expert knowledge, a polynomial of order 5 suffices to fully describe the lateral and longitudinal displacement and velocity profile of the vehicles during the merging scenarios. Figure 17 illustrates the scatter plot of joint distribution of four scenario parameters, namely, scenario duration, initial longitudinal velocity of vehicle on highway, lateral position of merging vehicle relative to road at start of scenario and longitudinal position of merging vehicle relative to road at start of scenario.

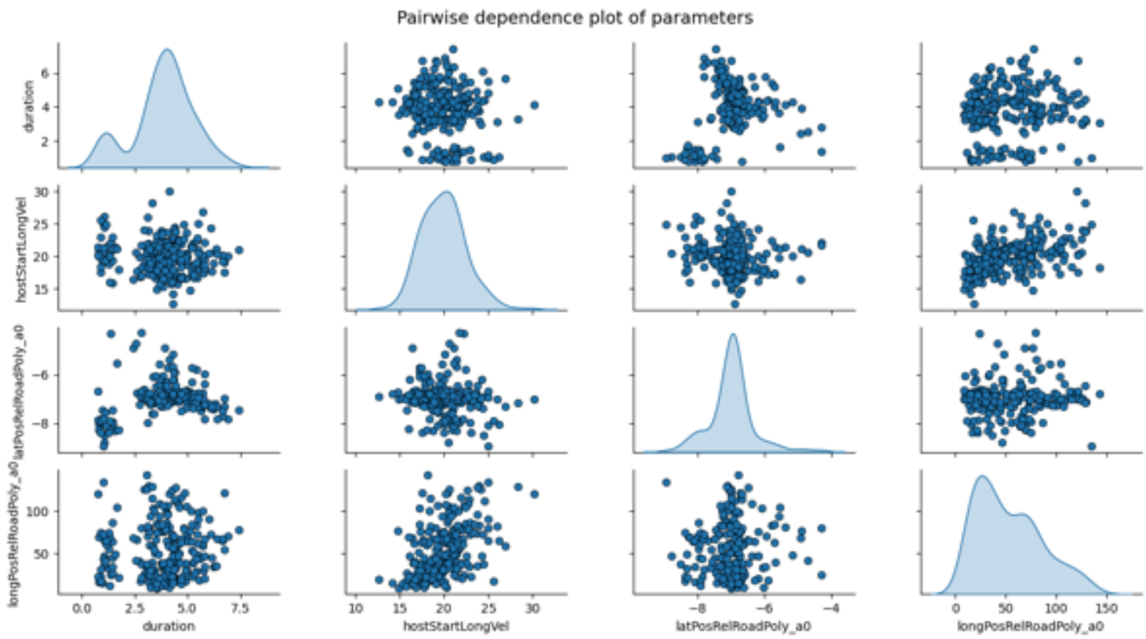


Figure 17. Scatter plot of the joint distribution of few selected scenario parameters.

In theory, the metrics which are selected to analyse the similarity of distributions in this research can be used to compare distributions of any dimensions. However, in practice distributions are normally compared in one dimension. Hence, we apply the JSD and BEST on the marginal distribution of all scenario parameters, see Figure 18. However, a similar marginal distribution does not necessarily mean the joint distributions are also similar. This motivates to select few additional parameters which are a function of all scenario parameters and perform the similarity analysis on the marginal distribution of these parameters. To this end, in this study, in addition to the safety surrogate metrics, i.e., PET and CMH, the scenario space is transferred to the space spanned by the principal components of the original scenario space in which each dimension is a function of all the scenario parameters. Figure 19 and Figure 20 depict the results of the similarity analysis for the principal components and CMH, respectively.

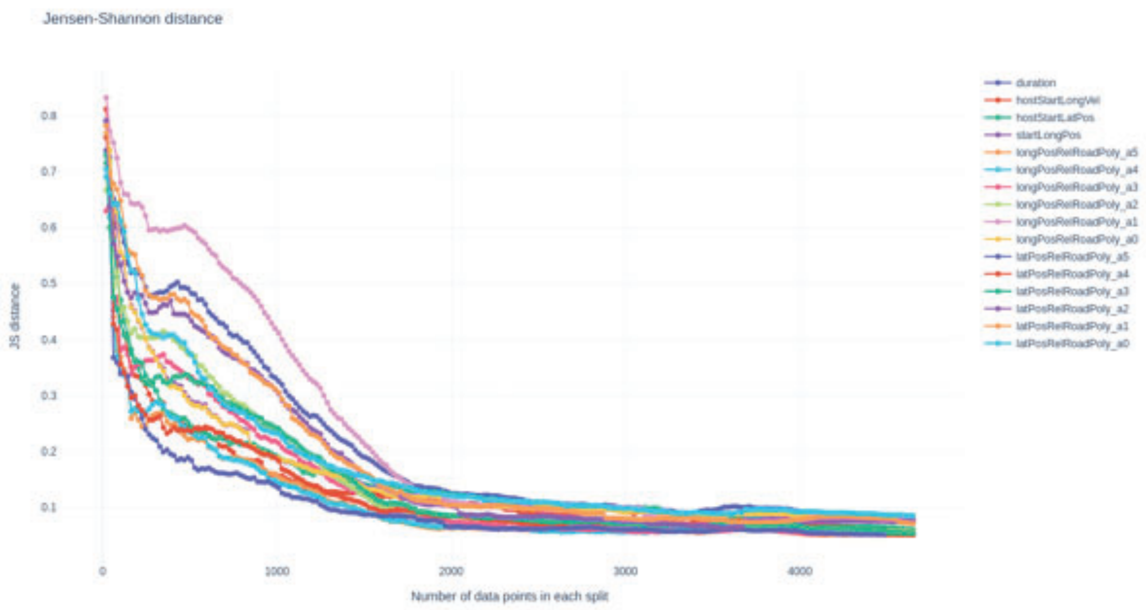


Figure 18. The JSD values as a function of the number of data points in the data batches for all of the scenario parameters.

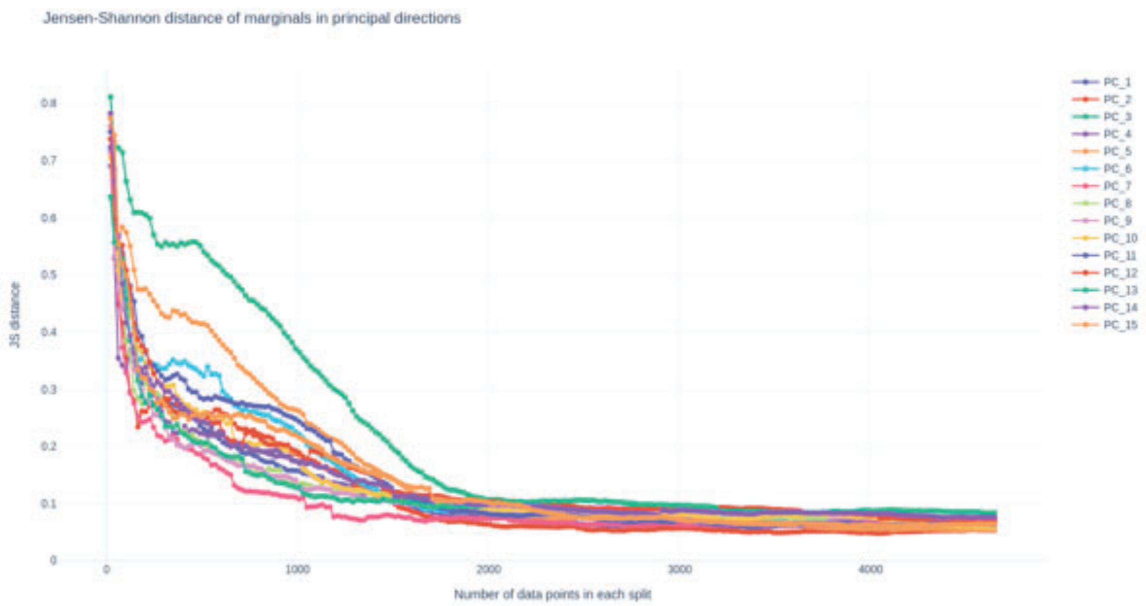


Figure 19. The JSD values as a function of the number of data points in the data batches for principal components of the scenario space.

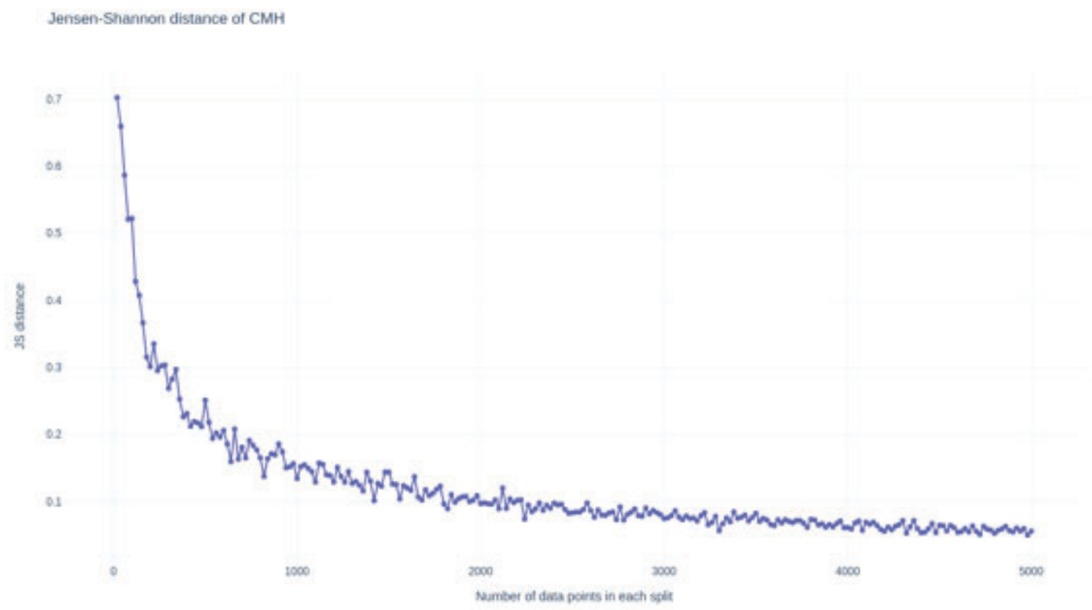


Figure 20. The JSD value for the empirical distribution of the CMH as a function of number of data points in the data batches.

JSD is a non-parametric approach to compare distributions and as a result it very susceptible to the randomness in the data set. To address this challenge, we have used another method to compare the distributions, i.e., BEST which is a parametric method. Figure 21 to Figure 24 demonstrate the results when BEST is applied on the distribution of scenario duration and CMH.

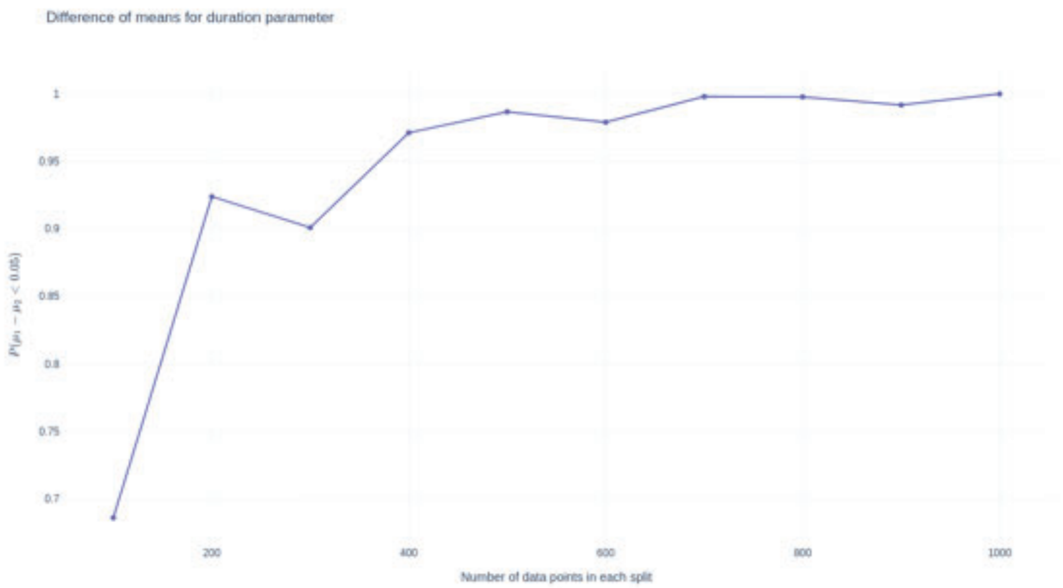


Figure 21. The BEST value for the mean of the empirical distributions of the scenario durations as a function of number of data points in the data batches.

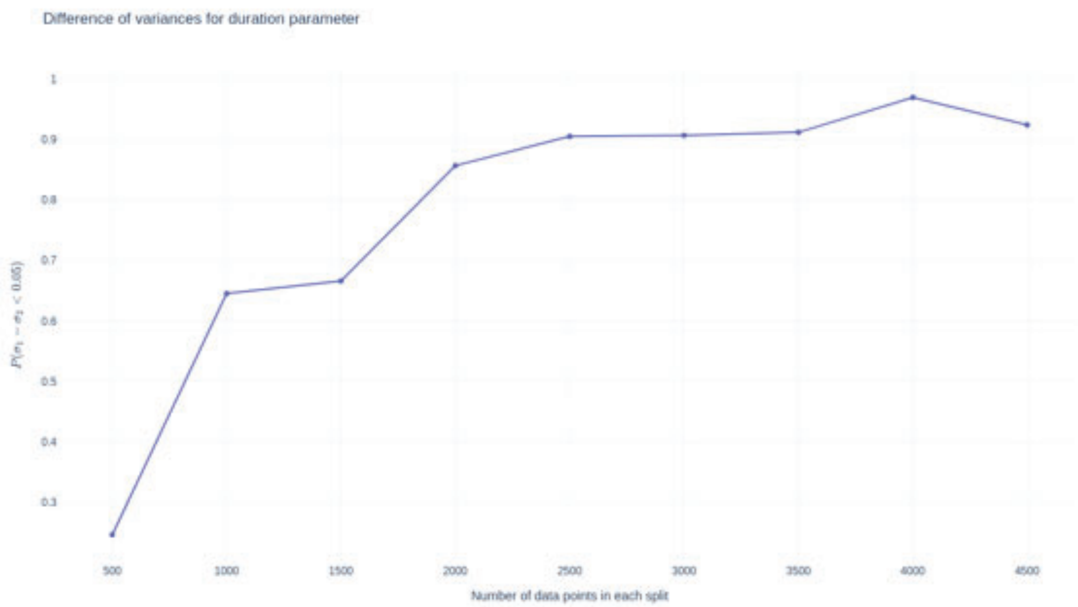


Figure 22. The BEST value for the variance of the empirical distributions of the scenario durations as a function of number of data points in the data batches.

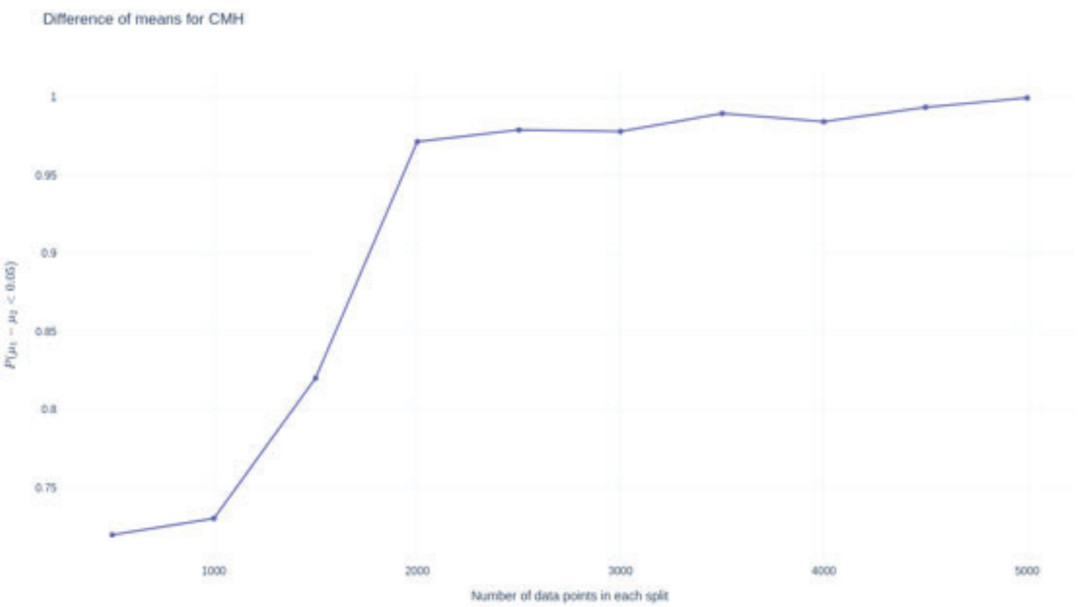


Figure 23. The BEST value for the mean of the empirical distributions of CMH as a function of number of data points in the data batches.

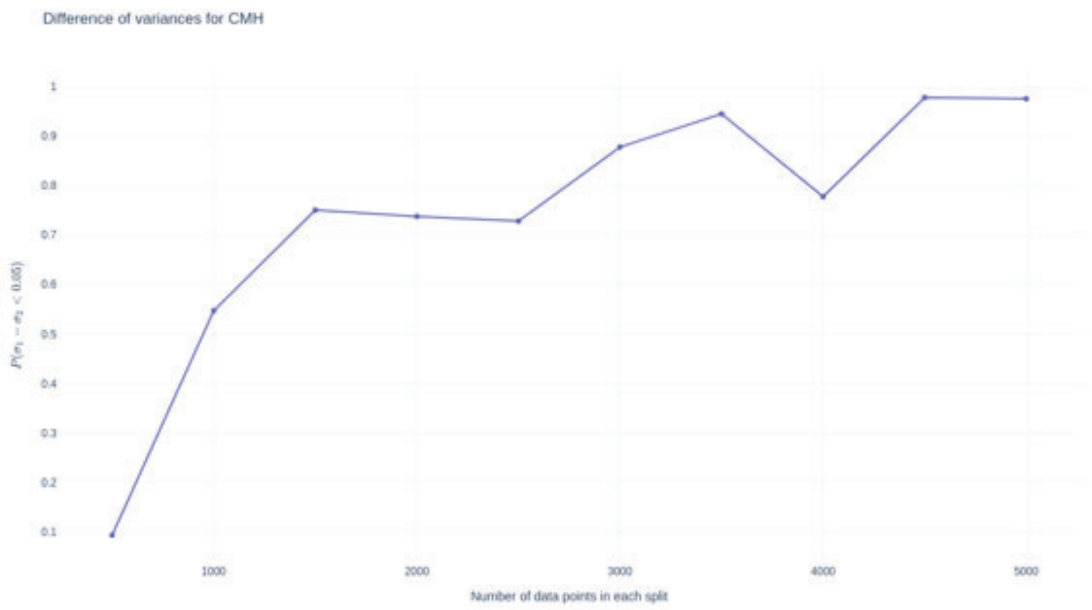


Figure 24. The BEST value for the variance of the empirical distributions of CMH as a function of number of data points in the data batches.

All in all, the analyses performed in this research project reveals that the 192 hours of data suffices to statistically describe the patterns, or the traffic dynamics, at highway on-ramps. However, the methods presented here are based on the entire range of the marginal distribution of the indicators mentioned previously. This means that these results are substantially affected by stationarity of the centre of the distributions and thus do not show the stability of the distribution tails. This remains as a future work for this project.

The mined on-ramp scenarios were further synthesised in order to simulate the performance of an ego vehicle equipped with the off the shelf ADAS features in esmini. Figure 25 shows the histograms of two common threat metrics for the simulated scenarios and extracted scenarios from the data. One can see that the basic ADAS feature in esmini resembles the human behaviours captured during the data recording sessions.

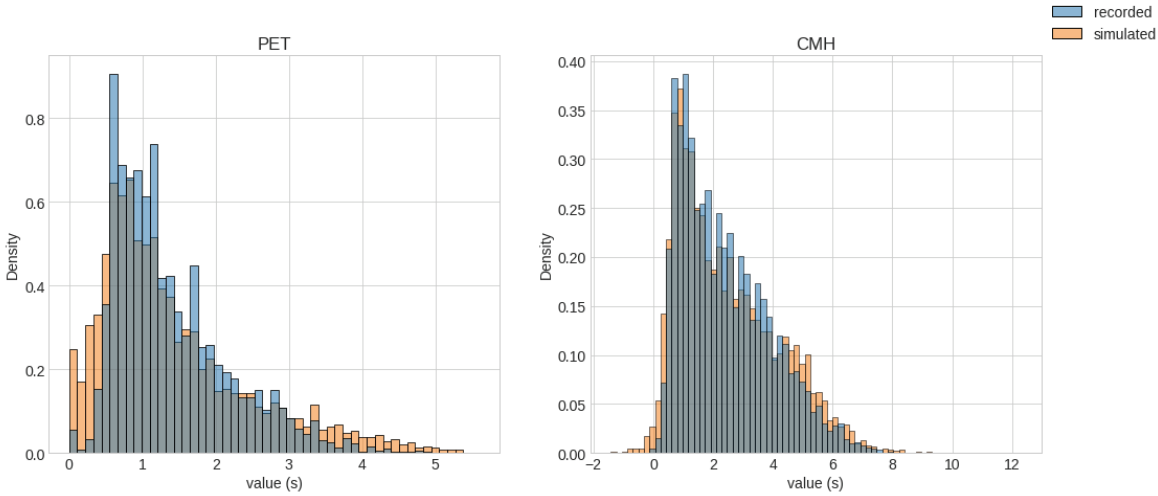


Figure 25. The histogram of PET and CMH for the identified scenarios from data vs simulated scenarios.

6.4 Scenario extraction and safety surrogate modelling

To extract relevant scenarios from the data provided by Viscando, safety surrogate measures had to be collected and measured as well as the behaviour of the individual vehicles. To do this the road network over the domain had to be modelled.

To begin the process of creating a road network model, a satellite photo of the location was used. This photo was then converted into a georeferenced digital map format in the *ASAM OpenDrive* format, see Figure 26. ASAM OpenDrive is a file format designed for the representation of road networks, including roads, lanes, intersections, and other features of road networks. It is a widely used standard for road network modelling in the automotive industry.

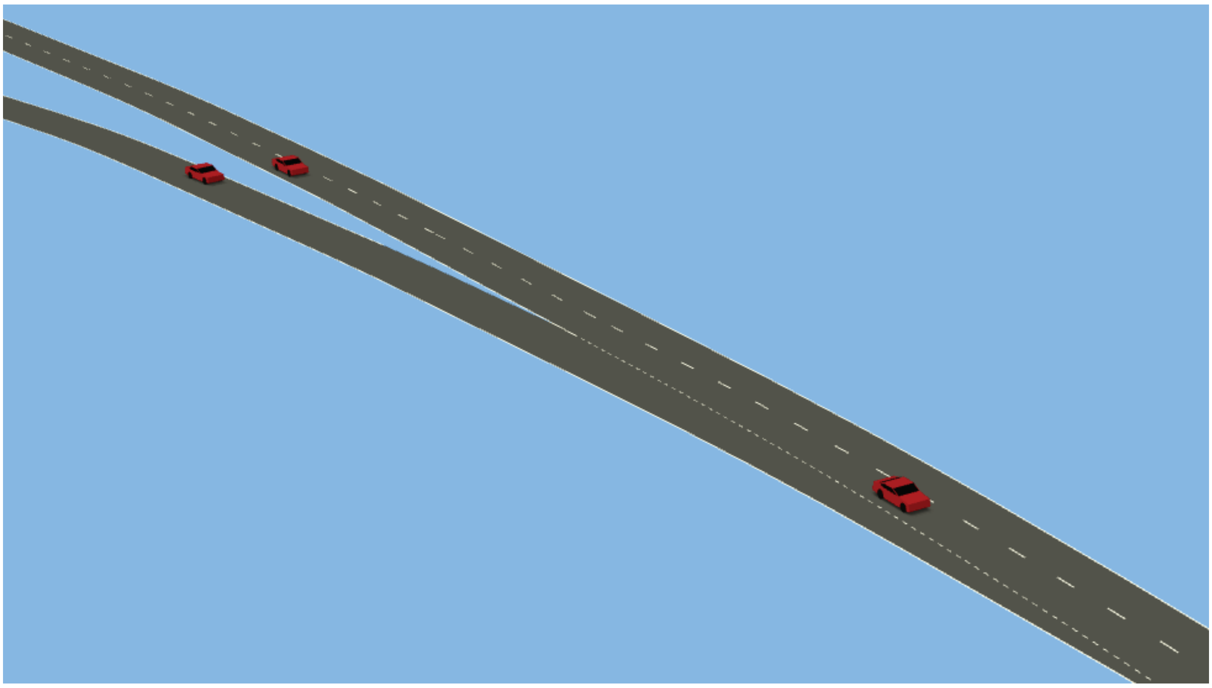


Figure 26. OpenDrive over Fiskhamnsmotet.

Once the digital map was created, it accurately reflected the location where the data was collected. This allowed each individual data point to be mapped to a location on the road network. By doing so, it was possible to determine the location of each vehicle on the road in relation to each other.

To extract relevant scenarios from the data, some measure of what is "relevant" must be inferred. To do this, the vehicle dynamics as well as safety surrogate measures have been extracted.

The following has been collected and measured:

Vehicle Dynamics:

- Longitudinal and lateral velocities of vehicles in the data, Figure 27, Figure 28
- Longitudinal and lateral accelerations of vehicles in the data, Figure 29, Figure 30

The vehicle dynamics that have been collected and measured provide information about how the vehicles are moving and changing direction on the road.

Safety Surrogate Measures:

For all vehicles:

- Distance to Collision
- Time to Collision
- Inter-Vehicular Time

For vehicles on the road merging into the highway, the following has been collected as well:

- Longitudinal Free Space

Distance to Collision is the distance between the front of a vehicle and the closest point of another vehicle or obstacle, Figure 31. Time to Collision is the time it will take for two vehicles to collide if they maintain their current trajectory and speed, Figure 32. Inter-Vehicular Time is the time it takes for a vehicle to travel the distance to a vehicle in front of it, Figure 32. In addition to these measures, specific safety surrogate measures have been collected for vehicles merging onto the highway. The Longitudinal Free Space measures the distance available to a merging vehicle, which is helps determining if the merging vehicle has sufficient space to safely merge onto the highway without causing a collision or impeding the flow of traffic, Figure 33.

Vehicle Dynamics Distributions:

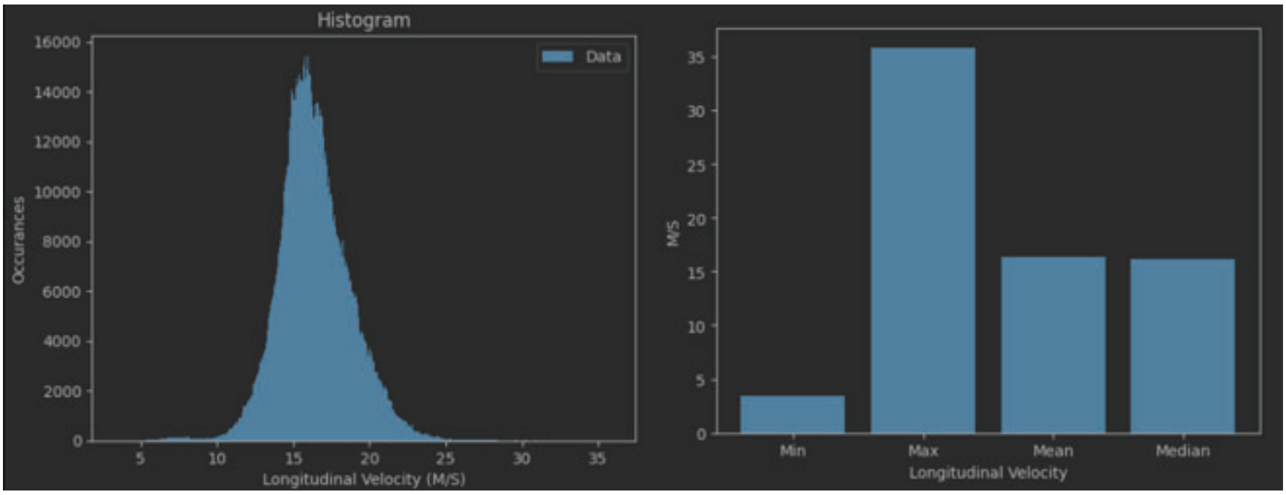


Figure 27. Longitudinal velocity observed in the data.

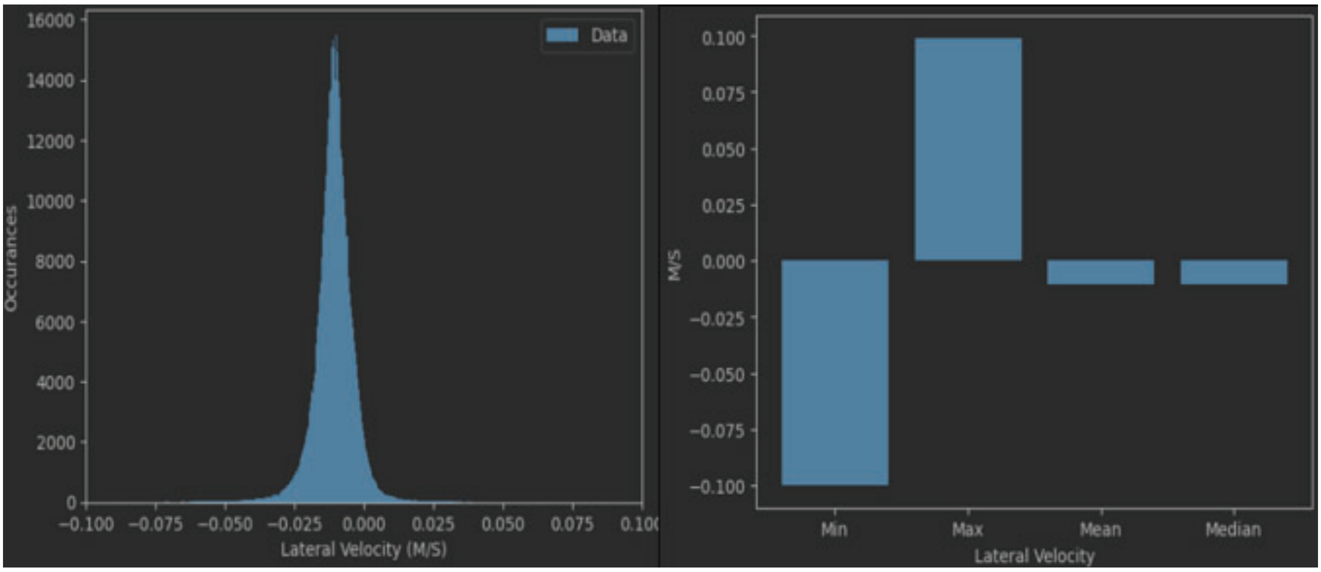


Figure 28. Lateral velocity observed in the data.

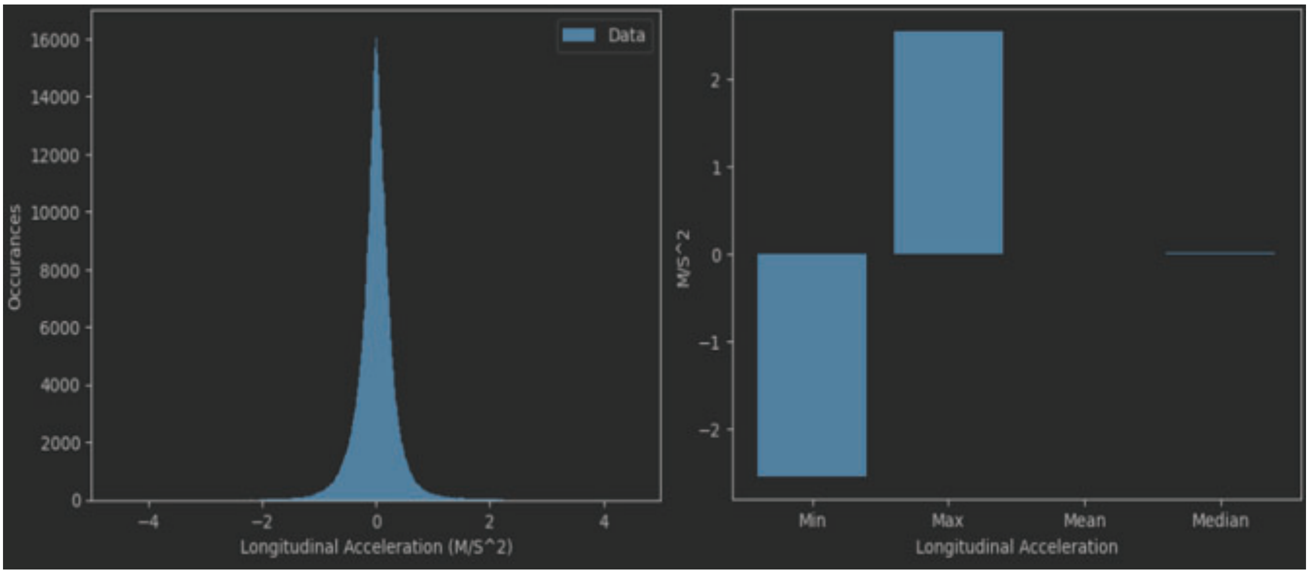


Figure 29. Longitudinal acceleration observed in the data.

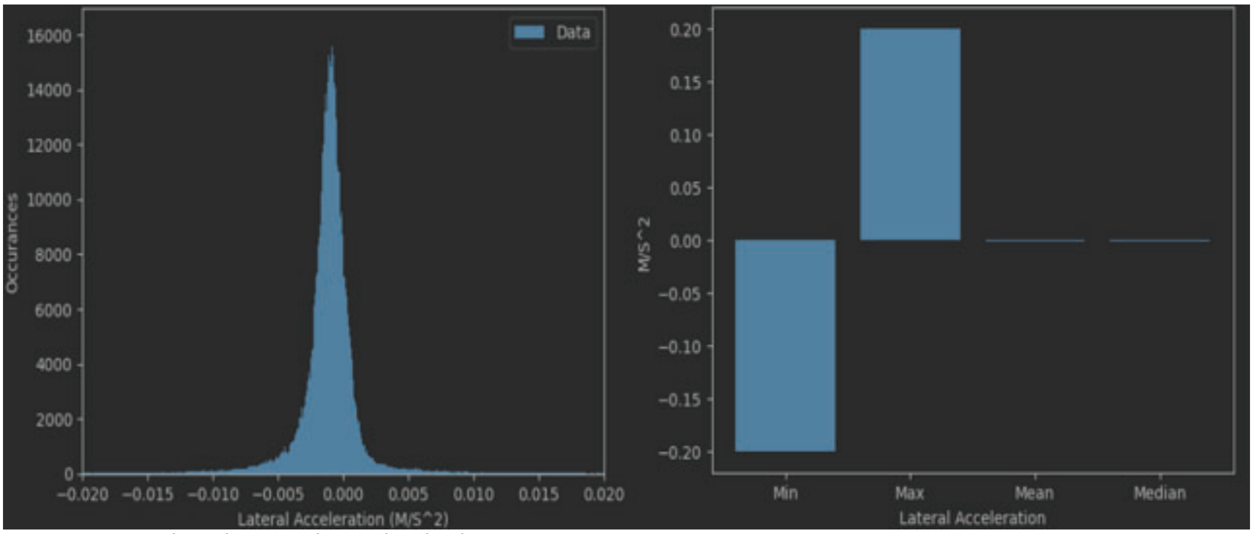


Figure 30. Lateral acceleration observed in the data.

Safety Surrogate Measures:

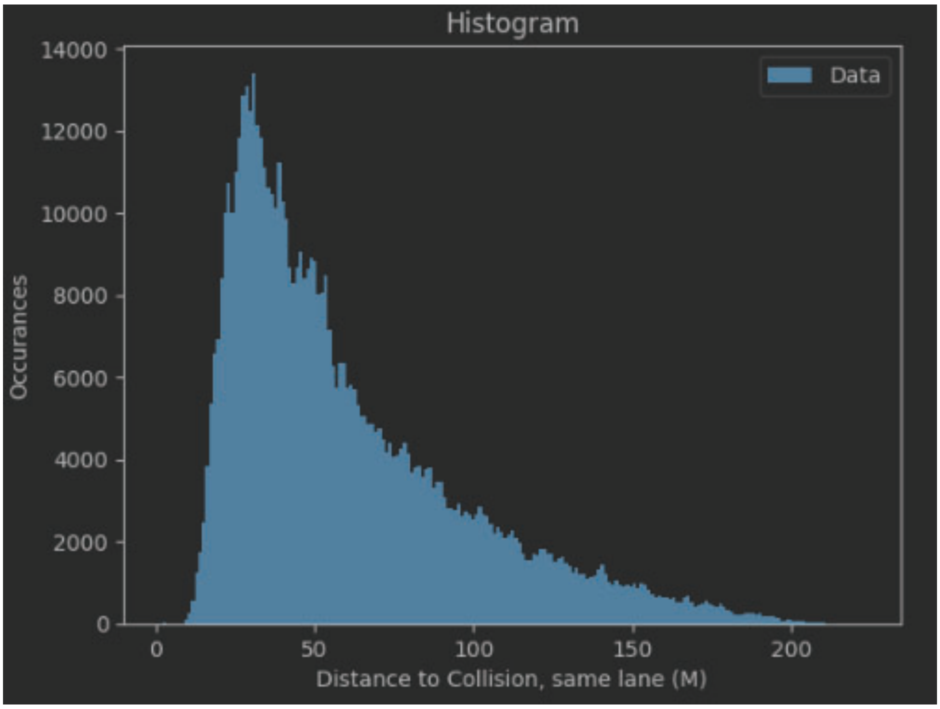


Figure 31. Distance to collision between vehicles in the data.

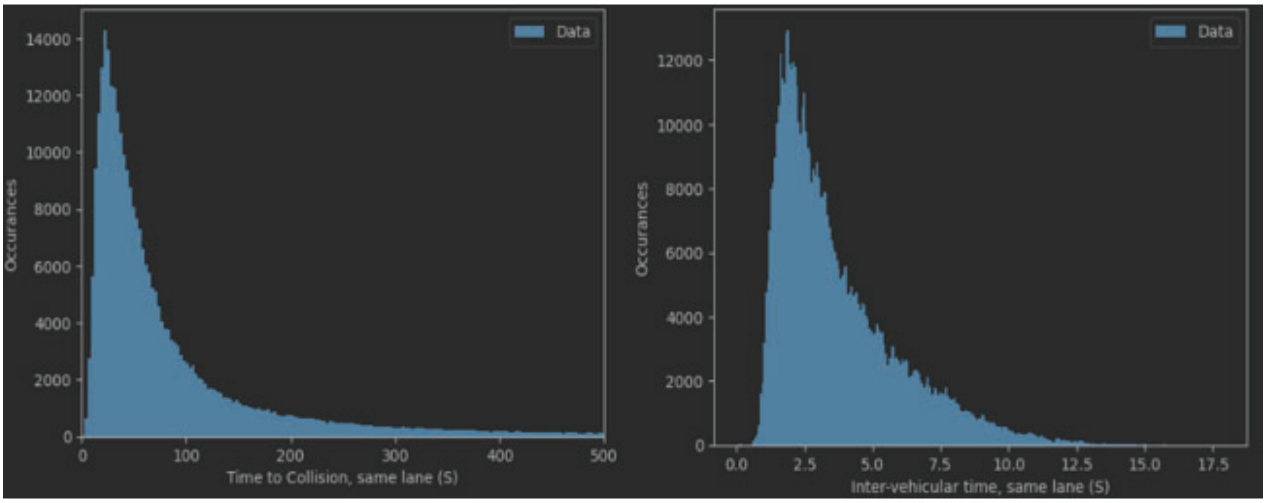


Figure 32. Time to Collision and Inter-vehicular Time between vehicles in the data.

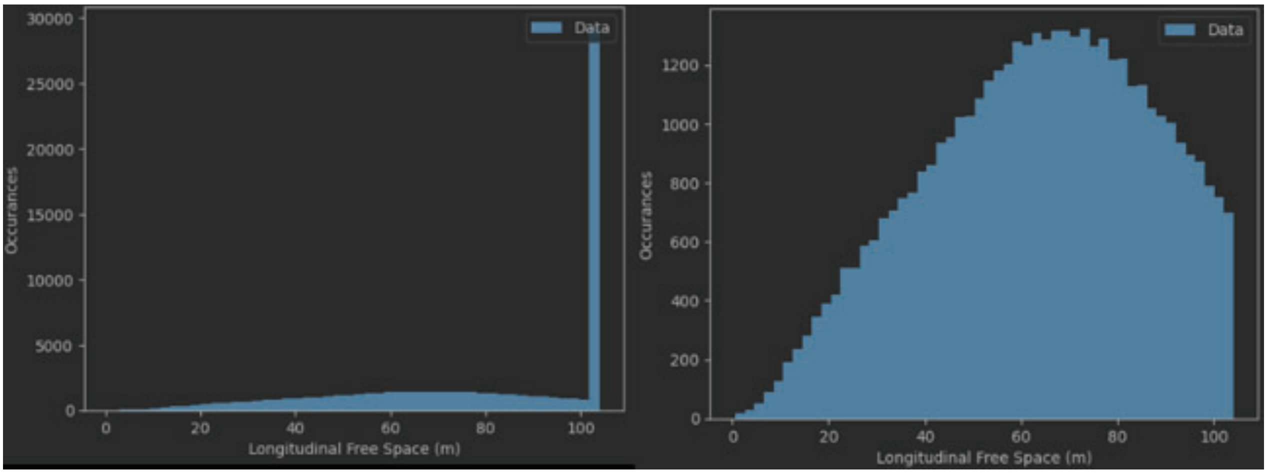


Figure 33. Longitudinal Free Space observed for vehicles on the ramp on merging. The first sub-figure is for all vehicles merging scenarios in the data, the second sub-figure is for all merging scenarios in the data where there is at least one vehicle ahead or behind in the lane being merged into.

One can identify potential scenarios for vehicle dynamics and safety surrogate measures by analysing the collected data. New scenarios can be generated from the data using machine learning algorithms, simulations, and predictive analysis. These scenarios can aid in identifying areas where improvements can be made to t the autonomous driving system. By evaluating observed and generated scenarios in a simulation.

Once these scenarios have been identified, these scenarios can be saved. This is achieved through a wrapper around the open-source repository scenariogeneration [40], which provides an interface for generating an ASAM OpenSCENARIO file. This file is a standardised format for specifying the behaviour of vehicles in a driving scenario, including their positions, speeds, and accelerations etc. To generate the OpenSCENARIO file, one needs to feed in timestamps from the original data set. These timestamps are used to identify the relevant vehicle behaviour data for a particular scenario. Once the OpenSCENARIO file has been generated, it can be used to simulate the scenario visually in a simulator such as esmini.

This is important, as virtual simulations are rapidly gaining popularity as a means of testing and assessing autonomous driving systems. The primary benefit of this approach is that it enables developers to evaluate their systems in a secure and controlled environment. By generating scenarios based on actual driving data and simulating them in a virtual setting, developers can gain a deeper understanding of how their systems will perform under a range of real-world conditions.

6.5 Behaviour modelling development by Chalmers and Viscando

Behaviours and reactions of road users surrounding an autonomous driving vehicle (AV) on a motorway, are a source of a large variety of situations, including critical incidents, that AV needs to understand and mitigate without compromising traffic safety and flow. Indeed, AV needs to both understand present actions and intentions of human road users, and act in a way that indicates its intentions in a clear and understandable way. Unnatural, unexpected behaviour or AV will result in both dangerous situations (direct accident risk) and sudden, critical reactions of surrounding road users leading to even more critical and unpredictable situations.
Within scenario-based AV verification concept, AV design safety is achieved by directed testing over an extensive catalogue of scenarios that represent the scope of possible situations in the AV's operational design domain (ODD). This implies that the spectrum of possible reactions of human road users to (possibly flawed) AV actions must be possible to recreate. Hence, for the directed testing results to be representative for real road driving, it is paramount that realistic behaviour models for road users are included in the test environments (traffic simulation and test track).

6.5.1 Main approaches in behaviour modelling

Behaviour models are used to calculate the reactions of a driver to the traffic situation, the latter includes the dynamics of both own motion and the observed states of other road users. The tactical goal of the driver (e.g., to change lane) and their personal characteristics (e.g., risk-taking) and state (e.g., sleepy) play an important role in reality and therefore would be desirable to include into the model.

Three main approaches to behaviour modelling in traffic can be separated:

- Rule-based
- Neuromechanics-based
- AI-based

Rule-based approach parametrises dynamic responses of the driver based on the dynamics of the traffic, using explicit equations for the responses. Behaviours in complex situations require complicated models with large amounts of parameters and hence become intractable. This approach is described in more detail in [34]
Neuromechanics-based methods focuses on the neural responses and muscular mechanics of the driver. This allows for accurate account of physical limitations of driver reactions (e.g., reaction time) but does not address the decision making in complex situations.
AI-based methods train AI models to generalise behaviours of modelled road users based on either explicit rules that reward or penalise certain actions and outcomes (Reinforcement learning (RL)) or the real traffic behaviour data ("expert demonstrations)" (imitation learning (IL) or inverse reinforcement learning (IRL)). Literature reviews of both approaches are provided in [34] and [41]

In the ASCETISM project, two approaches have been investigated independently, namely:

- Looming controller that falls under neuromechanics-based models, as it is based on modelling of human visual apparatus.
- IRL-based approach, which uses observed interactions to train an AI agent to drive and interact with other road users in different situations.

6.5.2 Looming controller

The angular expansion of objects on the retina is crucial to how humans and animals monitor and react to relative motion, such as during locomotion. To further study the importance of looming and retinal flow in merging scenarios, data from a recent experiment designed specifically to expose the subjects to a one-dimensional continuous tracking task, as is the essence of simple constant-speed car driving, was analysed. In order to study the details of the perceptual cues experienced by the drivers, the retinal optic flow was recreated from numerical optic flow estimation and gaze compensation. The result showed the instantaneous retinal optic flow path as experienced by the driver, see Figure 34. Optic flow and retinal optic flow are shown using a colour code, as illustrated in the upper left corner. Every pixel represents a moving visual point, i.e., a planar motion vector field, where direction is shown by hue and magnitude by intensity. The first image shows pure optic flow emergent from relative motion, while the second and third image show retinal optic flow using two different gaze points. For the second image the measured (actual) gaze point was used, and for the third a shifted point was used for demonstration purposes.

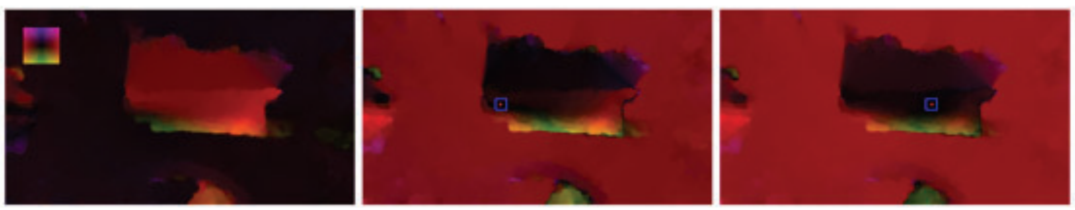


Figure 34. The instantaneous retinal optic flow path as experienced by the driver.

Using the numerical flow calculation, the path could then be related to individual gaze fixations and steering corrections, as seen below. A typical sequence of snapshots after a steering correction is shown in Figure 35. The three vertical lines in the plots denote the three instances in time from which the images are captured, starting from the oldest at the top. The second image is zoomed in around the gaze point (red) and the third image is the retinal optic flow. The blue line is the future GNSS track projected into the images. In the leftmost plot, the gaze is shown as grey for the Y-axis and black for the X-axis, and the step plot show when the subject blinks. The plot in the middle depicts the head movement in yaw rotation. The rightmost plot shows the steering wheel rate, and the circles mark the beginning of a steering correction.

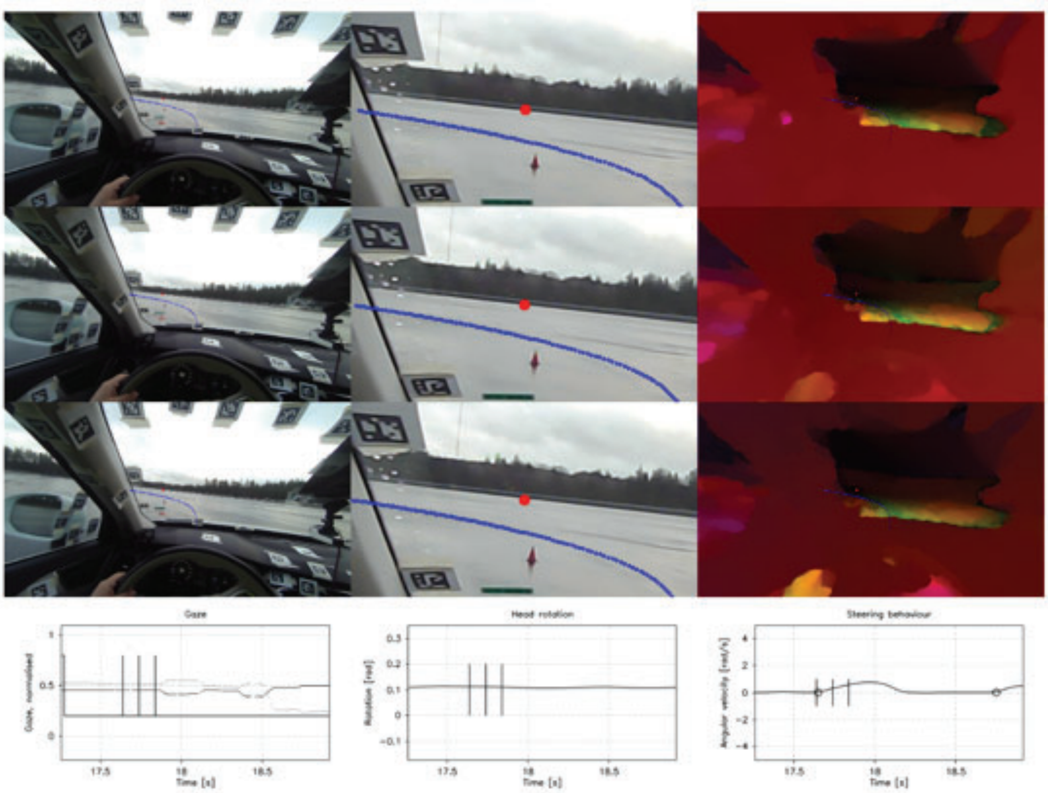


Figure 35. Optic flow and retinal optic flow are shown using a colour code, as illustrated in the upper left corner. Every pixel represents a moving visual point, i.e., a planar motion vector field, where direction is shown by hue and magnitude by intensity. The first image shows pure optic flow emergent from relative motion, while the second and third image show retinal optic flow using two different gaze points. For the second image the measured (actual) gaze point was used, and for the third a shifted point was used for demonstration purposes..

The analysis was then done by focusing on the steering corrections as motivated by the intermittent control theory. The theory identifies both individual and super-positioned corrections (i.e., corrections formed by several overlapping reaching movements), but it was decided to first focus on correction onset in order to isolate any perception-action response. By then merging the basic principles of the theory with that of retinal optic flow in locomotion, it was assumed that individual steering corrections, in general, are applied only after the gaze fixation process has identified the instantaneous retinal flow path from the flow fields. The hypothesis from nulling-strategy, is then that if the instantaneous retinal flow path differs from the intended path a steering correction is initiated within a 50 to 100 millisecond window. Also note that, in the experimental setup, where drivers drove on an S curve, a single specific steering wheel angle could be found for each of the two curves that would bring the road vehicle perfectly around the track. However, since the exact motion dynamics of the vehicle and limitations in the perception-action loop, it is expected that the subject is required to correct the steering wheel a few times along the path. In the figure above, a representative example of a correction is shown. One can see in the top image, that the retinal optic flow path is not perfectly aligned with the intended path, but after the steering correction was initiated the flow field is aligned with the future path. The pattern is typical to all correction onsets, for the first time showing that retinal optic flow and intermittent control theory can be combined to fully account for the perception-action loop of locomotion.

Continuing, by looking deeper into the data it was found that the super-positioned steering corrections in fact could be explained from retinal flow as well, as shown below. In the figure, the correction seems to be composed of at least five individual reaching movements, and in literature no detailed explanation to the reason for such super-position has been given so far. However, from this study it seems clear that each steering correction was initiated from a specific gaze fixation, that was initiated during the previous steering action. This is especially interesting as it points at two things: (i) it is clear that the steering action is an open-loop response, as predicted by the intermittent control theory, and (ii) the concept of so-called waypoints recently proposed in high-impact literature is too simplified, as the locomotor system should rather be seen as a collection of parallel processes, attracted and initiated directly by the flow field.

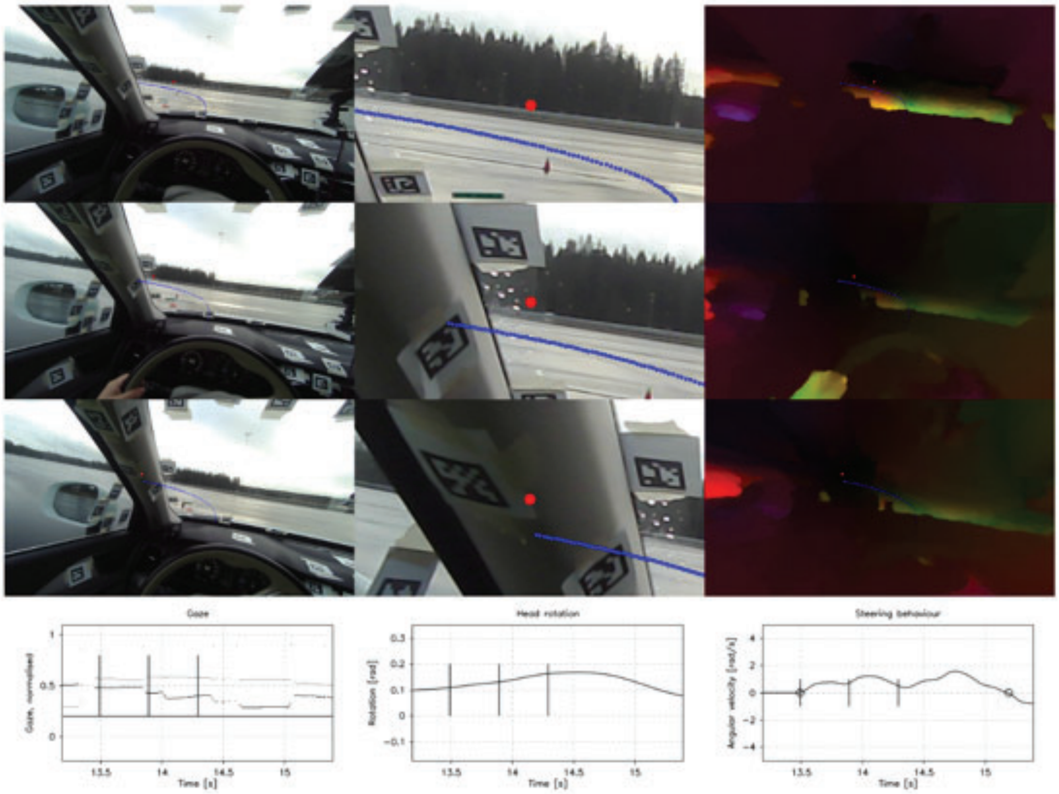
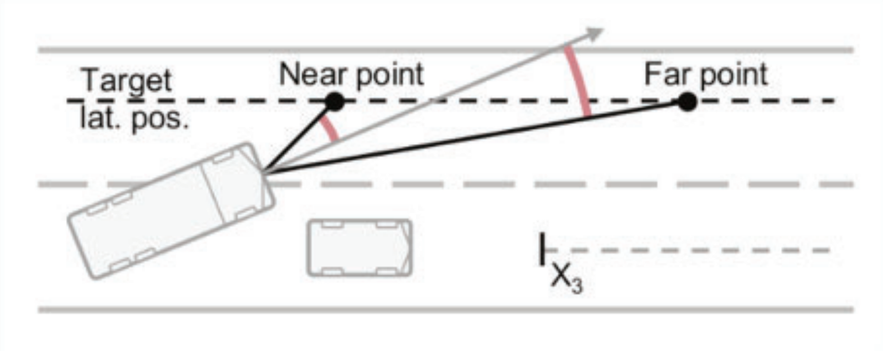


Figure 36. The sequence shows the interaction between gaze and intermittent control for a super-positioned steering response. See Fig. 3 for further details. Note that the head-mounted camera is located approximately above the right eye of the subject, possibly giving a slightly misleading view on the third row, i.e. it might very well be the case that the left eye can see past the A-pillar.

From these findings, a much more detailed understanding of the human perception-action loop during locomotion in scenarios relevant to ASCETISM was provided. The important finding both further highlights the relevance and importance of intermittent control from antagonistic neuromuscular systems, and how such control connects to external stimuli. The full retinal flow model is not suitable for the ASCETISM dataset, gathered at Fiskhamnsmotet, as the lack of vehicle on-board sensors make it impossible recreate the full flow. However, the strength of the data is that it contains a very large number of interactions between vehicles, especially when drivers are visually negotiating their space. As a simplification, retinal flow was reduced to only contain looming, the angular expansion of the individual vehicles. The looming is used both to understand speed selection of individual drivers, but also to negotiate between drivers (merging and overtaking behaviour). In addition, an aim point driver model was incorporated, in particular the Salvucci and Gray *two-point model of steering,* to achieve lateral control. The two-point model is special as it models the time derivative of rate of turn (yaw rate) instead of simply rate of turn. The benefit is that steady states of the driver more easily can be modelled which is crucial for sharp turns (to avid unstable behaviour). The model uses two aim points, one referred to as the near point located close to the vehicles on the desired path centre line, and one far point located at the gaze of the driver. The model then includes three control laws: the near point should be steered so that it stays right in front of the driver, and any movement of the near and far points should be counter-acted with steering. Specifically, the model is defined as in Figure 37 below (delta dot is the steering rate, i.e., relating to yaw acceleration).



$$\dot{\delta} = k_{\mathrm{nP}}\dot{\theta}_{\mathrm{n}} + k_{\mathrm{f}}\dot{\theta}_{\mathrm{f}} + k_{\mathrm{nI}}\theta_{\mathrm{n}}$$

Figure 37. The two-point model of steering. The near and the far point guides the driver and is a simplification of the retinal flow. The control law models the steering rate delta dot (i.e., related to yaw acceleration). The k values are tuneable parameters, and the thetas are the angles to the near and far points, where the dots denote rate of change. From the model, the only way to achieve no steering is if the vehicle is correctly aligned towards the near point, and that the far and near points are stationary in the field of view.

Using the data from the Ascetism project, and the modelling frameworks described above, a simulation was created. The simulation mixes recorded vehicle traces, with simulated closed-loop drivers. The closed-loop drivers react to surrounding traffic from the current looming landscape, including merging and overtaking behaviour (given a desired speed, and the desired paths). Specific driver parameter analysis was not done, but from visual inspection the behaviour looks realistic indicating that flow-based driver models work well in traffic scenarios in which agents do spatial negotiation. The specific rules for each driver's lateral behaviour are (for merging):

- Follow the current path, at the desired speed.
- Set the desired lane:
  - If current lane is on-ramp: Desired lane is right lane (moving left)
  - If current lane is right lane, and looming or max visual angle above threshold: Desired lane is left lane
  - If current lane is left lane, and no right lane looming: Desired lane is right lane

The detailed lateral control is given by the two-point model, where the far and near points are fixed according to the above rules. There is also a maximum steering wheel angle set in the simulation. For the longitudinal behaviour, the rules are as follows:

- There is a desired speed (target)
- High looming causes braking
- There is a max visual angle of the car in front.

The specific lanes were extracted from the Ascetism data, by looking at the mean path in each lane (we did not use the centre line of the lane, rather the actually driven path). The simulation is shown in the picture below.

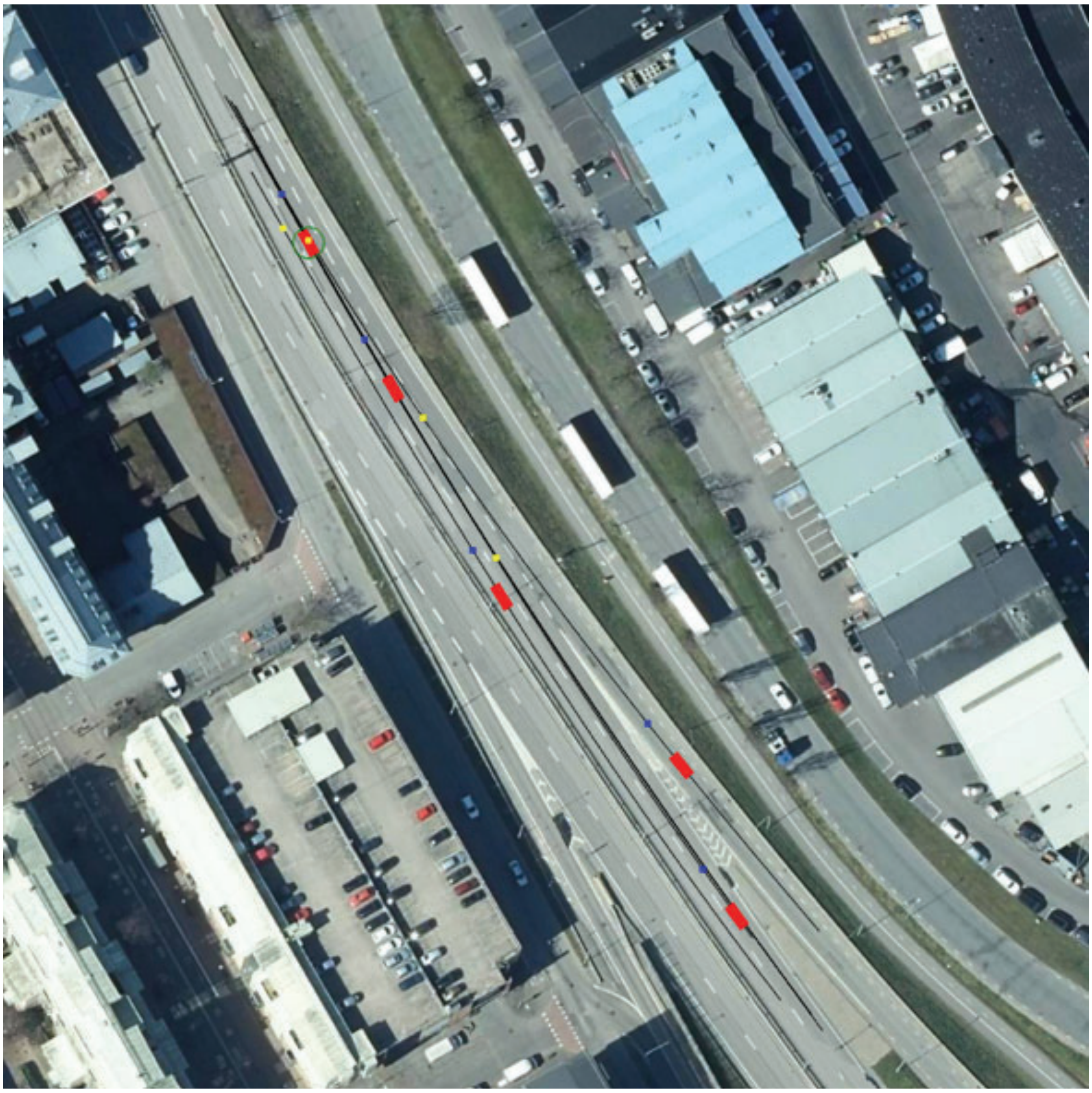


Figure 38. The simulation environment. The red rectangles represent cars, where non-circled cars are replayed directly from data, and the circled car is simulated in closed loop, with a initial state extracted from data. The blue points are the near points, and the yellow the far points. The black lines are the mean path for the three desired paths (left lane, right lane, on-ramp).

6.5.3 Inverse Reinforcement Learning

6.5.3.1 Choice of AI Method

To be qualified for use in design validation of AVs, road user behaviour models must be valid over a wide range of situations and render variability of both surrounding traffic behaviours and driver responses (different persons will react differently in the same situation) with statistical validity.

This makes AI-based models trained on real driving data a good choice. The methods that are most commonly used in behaviour modelling and prediction are:

- Imitation learning (IL)
- Reinforcement learning (RL)
- Inverse reinforcement learning

Basic principles behind these three classes of models are illustrated in Figure 39.

RL has a great potential to optimise the motion in traffic by training the strategies (policies) to maximise the rewards (which could be avoiding conflicts and disturbances in flow). While generally, the definition of a reward function is a non-trivial task, it has been shown to be especially challenging to define reward functions for human drivers, because of

- Often unexplainable, spontaneous behaviour of humans in traffic
- Less than perfect reactions and attentiveness to traffic situation
- Variations in the reactions
- Difference between different countries and roads

This is because of, on one hand, huge number of variables to take into account, and, on the other hand, far from obvious motivations behind far-from-perfect behaviour of human drivers.

Data-driven models, IL and IRL, are better solutions to the above challenges as these train from how real drivers behave and generalise these behaviours in previously unseen traffic situations.

While IL tries to find the behaviour that is closest to the expert demonstrations, IRL tries to identify the underlying reasoning for these behaviours, in terms of reward functions – under a simplifying assumption that human decisions are taken in a Markovian manner in the driving scenarios. Due to this, IRL is capable of learning more complex and varying behaviour patterns compared to IL. As it was demonstrated in [42], IRL can even learn binary decision making (cyclist passing before a vehicle or waiting for it to pass).

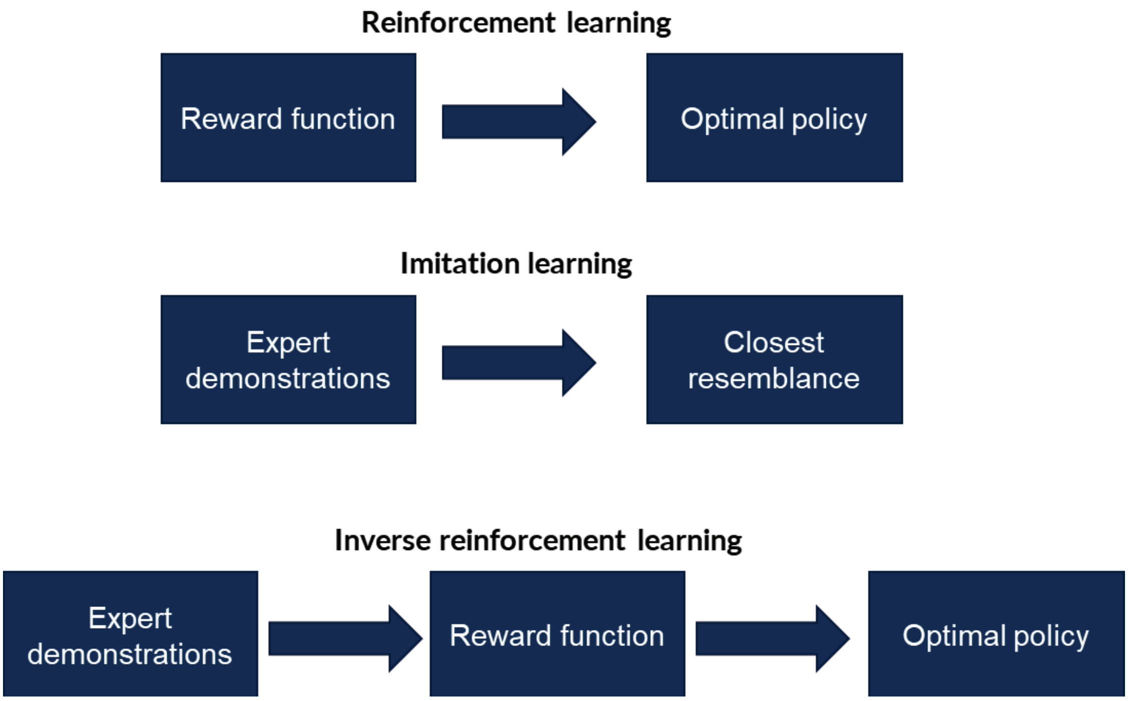


Figure 39. Comparison of basic operation principles for Reinforcement Learning, Imitation Learning, and Inverse Reinforcement Learning.

As compared to IL, IRL method is believed to be transferable to new locations, as it learns the underlying rewards instead of trying to mimic trajectories specific for a location (see [42] and references therein).
This is a very important property for the type of data collection used in ASCETISM: as stationary sensors can only be installed in a limited number of ramps, the models trained in these locations need to be re-usable for the remaining ramps. Therefore, it is important to investigate the potential and limitations of transferability of IRL models for the particular use case of motorway AD; this is planned for the follow-up projects where data will be collected in more locations.
For a more detailed comparison, and review of the applications of AI behaviour models, see [41] and [42].

6.5.3.2 Driver Model Integration in AV Simulation.

Figure 40 illustrates how the driver behaviour model is envisioned to be included in the traffic simulation. The IRL-based *ramp agent* represents the vehicle on the ramp. At each time instance, it decides its action (displacement) based on its own state (position and velocity) on the ramp, and on the states of the *obstacles* – vehicles on the motorway in its vicinity. As in real life, we expected that properly trained ramp agent would both manage to take the correct trajectory to get onto motorway, and correctly adapt speed and position to avoid conflict with vehicle(s) on the motorway.
In AV simulation, the main obstacle to the ramp vehicle is the AV. Software-in-loop component of AV stack integrated in the simulation environment would control the

movement of AV in presence of surrounding vehicles (including ramp agent). Thus, one can view IRL model and simulation environment as running “against” each other.

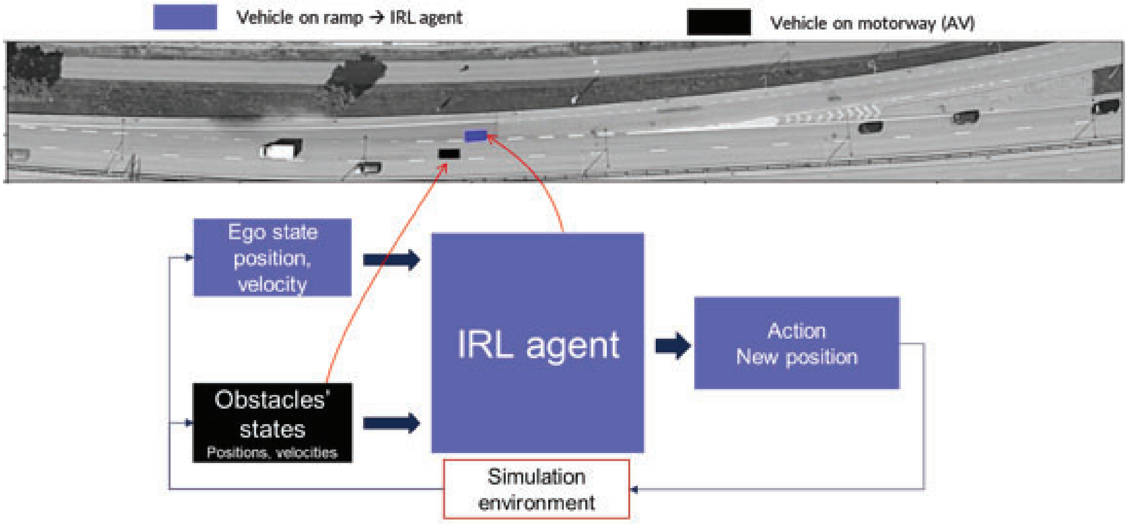


Figure 40. Illustration of the behaviour model integration in AV simulation.

However, integration and running of production-intended AV software in simulation environments was outside of the ASCETISM scope. Therefore, the IRL behaviour model integration was only tested on recorded data for human-driven vehicles from Viscando measurements.

6.5.3.3 IRL Behaviour Model Design.

The design and implementation of the IRL-based behaviour model is described in detail in [42]. In short, variational, generative adversarial network (GAN) - based formulation of IRL was chosen based on literature studies, schematically presented in Figure 41. The IRL model consists of generator and discriminator modules.

- Discriminator separates recorded trajectories from those generated from current policy.
  The modified discriminator explicitly approximates the reward function and uses it to distinguish between real and generated trajectories. This is not done in the standard discriminator which does not output an explicit value for the reward function.
- Generator optimises policy to make generated trajectories indistinguishable from recorded ones.

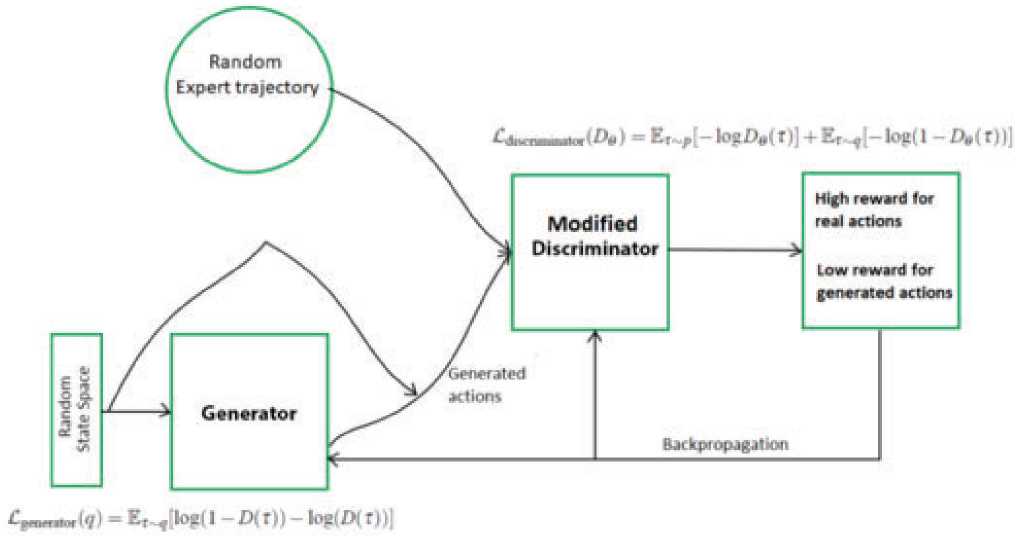


Figure 41. Schematic view of the GAN-based IRL model design used for modelling ramp agents.

Merging Scenario Choice.

The experiments described in literature, were in most cases run in the same static scenario, with a fixed number of agents, timesteps and interactions. The goal was also very specific and obvious. Moreover, expert demonstrations were in most cases generated using a “baseline” RL algorithm whose reward function IRL tried to recreate from these demonstrations in the second step.

In contrast, the real-road traffic scenario is characterised by large variety of the numbers of involved objects, variable number of time steps, non-stationary problem with intentions that change at every time step (aborted lane change manoeuvres happen, and these depend on traffic situation development in the middle of the manoeuvre), and inaccuracies in the expert demonstrations. These present a challenge for IRL and constitute a novelty in behaviour modelling research in ASCETISM.

To simplify the implementation, it was decided to limit the scope of scenarios to a situation where there is only one vehicle on the motorway that interacts with the ramp agent, as illustrated in Figure 42. Situations where vehicle on motorway changed to left lane to leave the right lane empty for the merging vehicle, were not considered either. These situations were filtered out from the catalogue of recorded trajectories and used for training and testing IRL model.

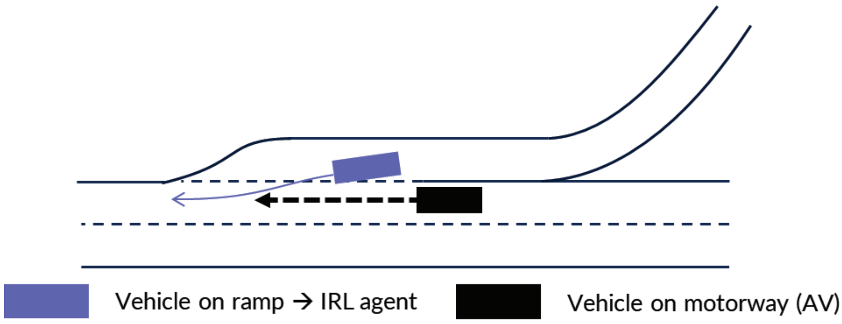


Figure 42. Merging situations with a single motorway vehicle, considered for behaviour modelling in ASCETISM project.

6.5.3.4 IRL Model Performance Analysis.

IRL model has been trained on a catalogue of merging situations described above. So far, the training dataset has been limited to one single day. Extending the training and validation dataset to more days and investigating the effect of larger training sets on model performance is planned to be done in future projects.
As the integration with simulation environment with a production-intent AD SIL component was not available in the project, the model was tested in a simple environment described in detail in [42]. The environment played back recorded interactions; the vehicles on motorway were moving forward according to their recorded positions and velocities. The vehicle on ramp controlled by the trained IRL model that uses the motorway vehicle position at each time step as input.
Figure 43 below shows snapshots of the recorded trajectories (left) and simulated with IRL-based ramp agent (right), for one merging event from the recorded dataset.
The example illustrates a general finding: trained ramp agents are capable of following the ramp and merging onto motorway. The gap between the vehicles is slightly smaller in the simulated case as can be seen from the figure.

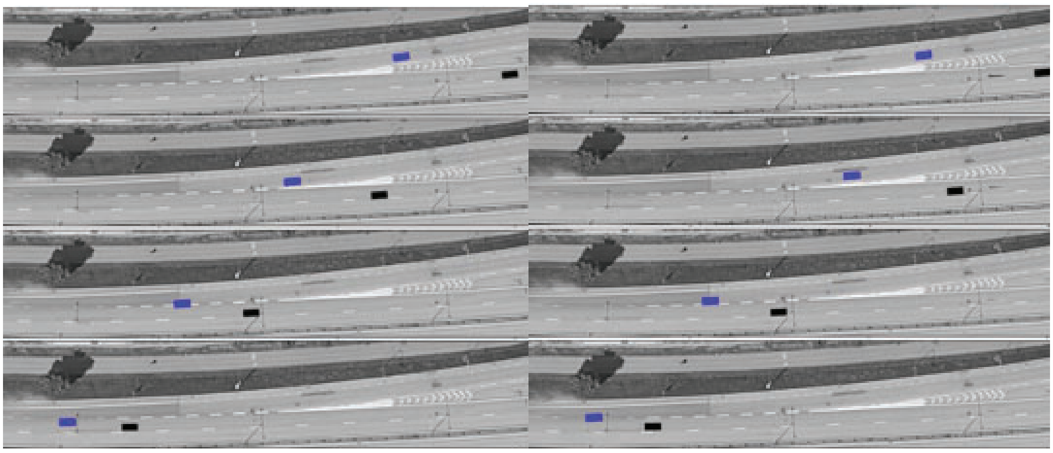


Figure 43. Left: a merging situation recorded in the data; Right: Result from running IRL model on the same scenario. In both cases, the motorway vehicle (black) follows a recorded trajectory. The ramp agent (blue) follows IRL model in the right image.

6.5.3.5 Safety Performance and Surrogate Safety Measures.

To get a more complete assessment of the performance of behaviour model, it is not feasible to compare single trajectories. Instead, overall safety performance of the model over a statistically representative set of realistic interactions should be compared. Driver model should ideally be as safe as human drivers – that is, show similar performance in normal and risky situations, including faulty behaviours that lead to incidents. If driver models are too cautious and safe, and hence are better than humans in avoiding conflicts, then they cannot be used to generate realistic interactions for Avs in simulations. On the other hand, if driver models are less safe than human drivers, then they may pose unrealistically high requirements on AV capabilities, which cannot be satisfied (and do not need to be satisfied) in practice. Therefore, the optimal behaviour models should be similar, or possibly slightly worse than humans in safety performance.

Surrogate safety measures (SSM) are widely used to assess the safety performance of vehicles. These measures quantify interactions according to how close these interactions are to an actual crash. Interactions with low SSM value are those which nearly ended up in crash – often referred to as incidents or close encounters. It is commonly assumed that the crash probability is correlated to the ratio of interactions with low SSM – therefore, crash risk can be compared for different locations, or times of day, by comparing distributions of SSM.

6.5.3.6 Post-Encroachment Time.

Post-encroachment time (PET) is often used as a SSM for interactions that involve crossing paths. It is the time gap between first actor leaving the area where the trajectories cross, and the second one arrives into it. PET equal to zero obviously implies a collision.
Figure 44 illustrates how PET is calculated in a classical crossing interaction (left) and how for motorway merge interactions, as we have defined in ASCETISM (right).

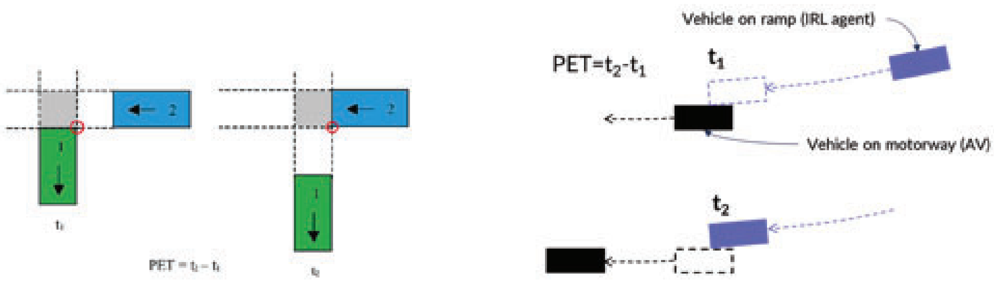


Figure 44. Post-encroachment time definition for "classical" crossing situation (left) and for merge interaction at motorway ramp (right).

PET distributions have been calculated and compared between recorded interactions used to train the IRL model, and interactions simulated using the model. The PET distributions are shown in Figure 45.

Qualitatively, the distributions do not show any critical dissimilarity; however, the share of more dangerous interactions (with PET close to zero) is increased for generated data, which indicates that the IRL model is not as good in avoiding close encounters as real drivers.

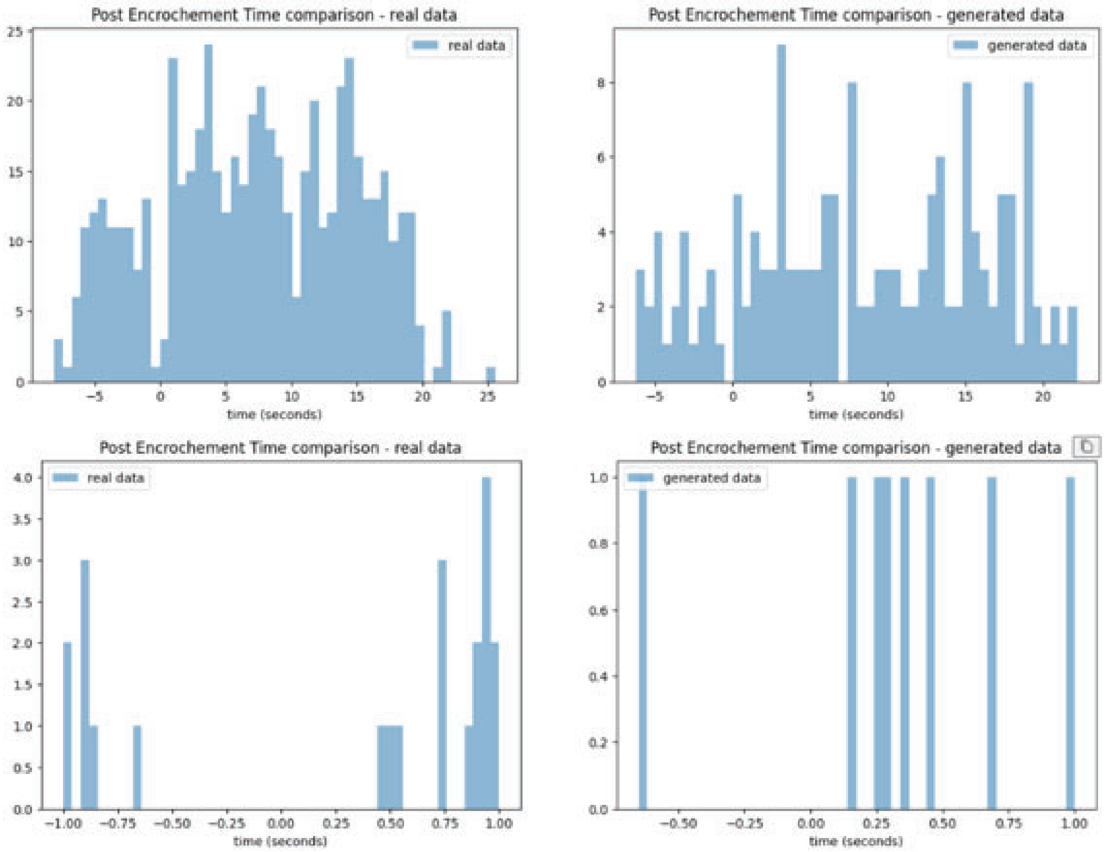


Figure 45. PET distribution for recorded merging interactions (left) and ones simulated with trained IRL behaviour model (left). Positive PET corresponds to vehicle on motorway passing first, and negative one to vehicle on ramp passing first. The lower images show zoom-in to PET values around zero, corresponding to near encounters.

Furthermore, how the behaviour model acts in interactions of different severity level was investigated. In Figure 46, the PET for a simulated interaction is plotted as a function of PET of recorded interaction. It is obvious from the plot that IRL model acts very similarly to the recorded event in all events (as the points lie around the diagonal line). Still, it can be noticed that the points mostly lie below the line y=x. For the upper right quarter of the plot, corresponding to the vehicle on motorway passing first, this means that the IRL-controlled vehicle keeps lower distance (thus lowering simulated PET). For the lower left quarter, which corresponds to the vehicle on ramp passing first, the absolute value of simulated PET increases, which means that the vehicle on ramp goes faster, thus creating a higher gap with vehicle on motorway. The exact reason for this is not clear, and more investigation is in the scope of future work.

It is also worth to mention that there are no points in the 2nd and 4th quadrant of the plot. This means that the sign of PET does not change in simulation. In other words, the vehicle that passes first in the recorded event, does the same in the simulated event. This might be an indication of IRL model making the right decision in the merging situations – however, more analysis with a larger dataset with more close encounters is required to confirm this.

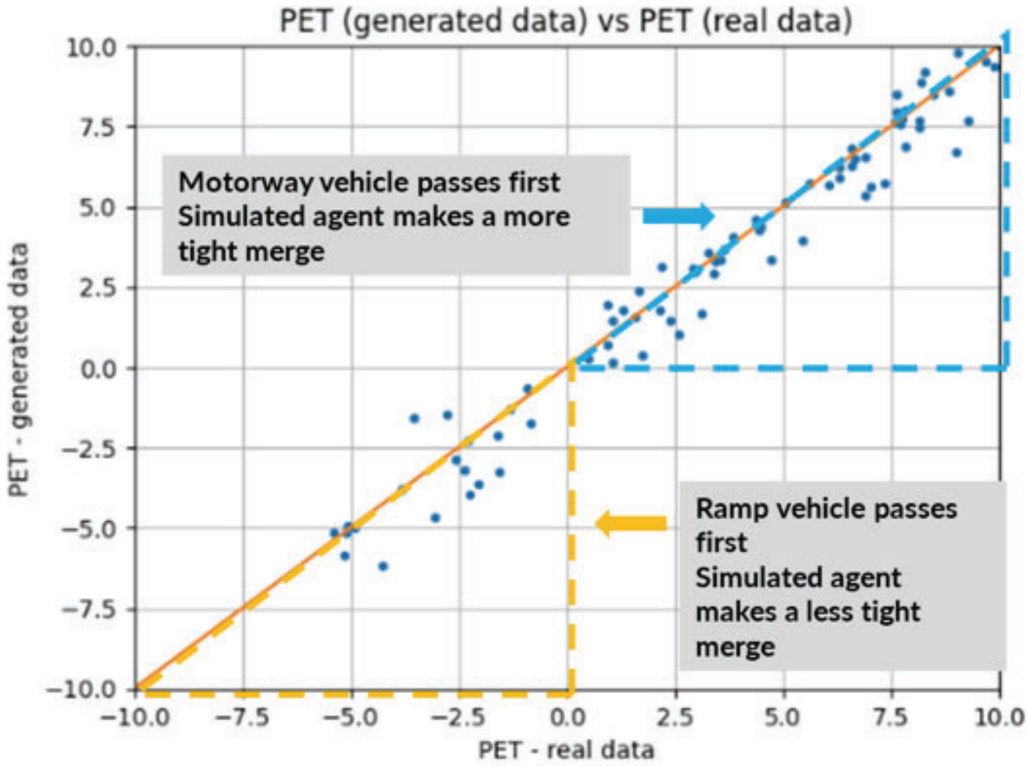


Figure 46. PET values from interactions simulated with IRL driver model, plotted versus PET values for recorded data.

6.5.3.7 Interacting IRL models.

One potential source of discrepancies between safety performance between simulated and recorded data can originate from the fact that we used recorded data for vehicle on motorway. It could not react to the action of IRL-based driver in a natural way, which, as we hypothesize, may lead to its unrealistic behaviour.

Therefore, an attempt has been made to use a separate IRL model for the motorway vehicle, and to let these two IRL models run against each other. This study was performed as a course project at Mathematical Modelling course at Chalmers University of Technology [42]. The project report, appended to this report, gives a thorough summary of the method, implementation and experiments performed to identify whether two IRL models interact in an expected way. Although the project duration was too short for definite conclusions, the authors showed that the IRL models reacted to each other and to artificially created obstacles on the road.

6.5.3.8 Time To Merge Metric.

While PET metrics is easy to use and intuitive, it has certain disadvantages. One of these is that low PET values may not always correspond to dangerous situations. A pedestrian may wait for the car to pass, and then cross the road immediately with a very little time margin. A merging vehicle at an on-ramp can adapt its speed and headway to the vehicle on the motorway and make a tight yet fully conscious and relatively safe merge. In both cases, this will lead to PET close to zero – despite low collision risk. Moreover, PET does not consider the dynamics before the crossing point. A vehicle may have been driving with a constant speed or had a last-second hard braking to avoid crash, with same value of PET for both cases.

There is an interest for other metrics that would better reflect the intentions and dynamics of interaction before the meeting point. In case of the on-ramp use cases, the metrics should ideally reflect:

- Whether vehicles are on the collision course initially
- If so, who is taking action to avoid collision (e.g., by changing the speed)
- How late and sharp the avoidance action is.

With these needs in mind, Viscando has proposed new metrics called Time To Merge (TTM) and Difference in Time To Merge (DTTM). This metric and following analysis of the ramp interactions are described in detail in a course project report [43]. Below is a shortened summary of the main definitions and findings.

TTM is defined as the time required for the vehicle to reach the line perpendicular to the driving direction, at the point of merging, as illustrated in Figure 47.

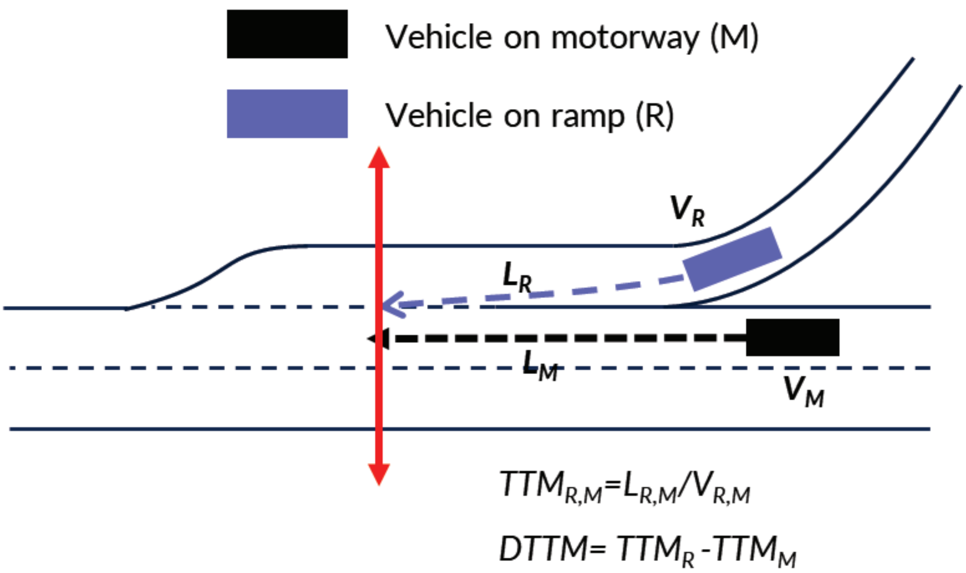
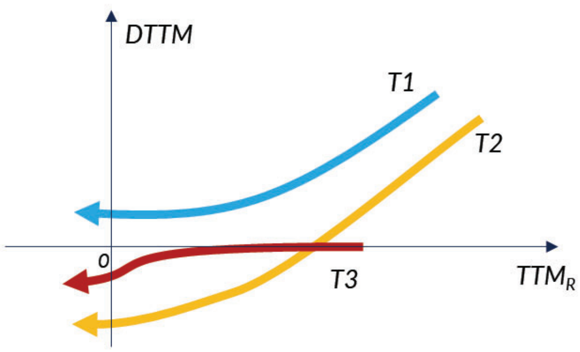


Figure 47. Definition of Time To Merge.

TTM is computed continuously for vehicle on motorway (TTM_M) and on ramp (TTM_R). DTTM is then defined as a difference between TTM for ramp and motorway vehicles. The meaning of DTTM at a particular time step is the time gap between vehicles when they come to the merging point if their speed would not change. Low DTTM means that vehicles are currently on a collision course. When there is a sufficient time before they reach that point (TTM is large), this may not be risky; however, at low TTM this means less and less time for the vehicles to avoid the collision.

Plotting DTTM as a function of TTM of one of actors (for example, TTM_R) helps illustrating different interaction dynamics and associated risks. Figure 48 shows a schematic drawing of three different cases.

- Trajectory T1 (blue) corresponds to ramp vehicle that stays behind the motorway vehicle all the time, merging after it.
- Trajectory T2 (orange) shows the situation where the vehicle on ramp is first behind (while DTTM is positive) but then accelerates and merges in front of the motorway vehicle (the part where DTTM is negative).
- Trajectory T3 (red) corresponds to probably the riskiest case. Vehicles are moving so that they are coming to the meeting point at the same time (DTTM is zero) until very late. The negative "bump" in the curve corresponds to an abrupt change of speed – either ramp vehicle accelerating or motorway vehicle braking hard. As a result, the merging is made with a narrow time margin (small DTTM value at TTM=0).



Blue: Ramp agent passes after MW agent

Orange: Ramp agent accelerates and passes before MW agent

Red: Collision course, but ramp agent accelerates to pass in front

Figure 48. Three different cases of interaction dynamics and associated risks.

Thus, DTTM-TTM plots can illustrate merging behaviours in a particular ramp. A heatmap of the DTTM-TTM curves corresponding to interactions recorded from Viscando data is shown in Figure 49 (top), while Figure 49 (bottom) shows the corresponding heatmaps for the simulation results with IRL driver model, for the same interactions.

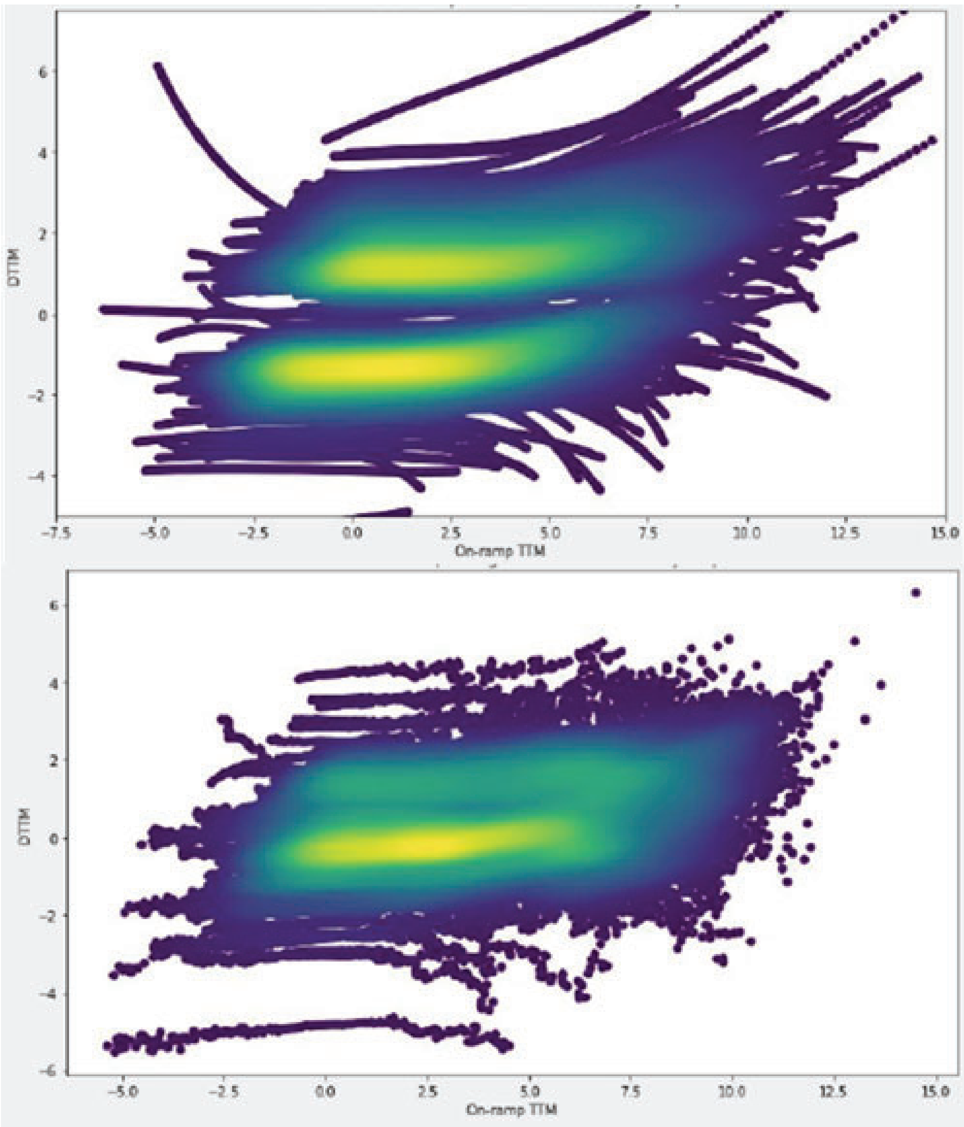


Figure 49. Density plots of DTTM vs TTMR curves for recorded merging interactions (top) and simulated ones using IRL driver model (bottom).

Quantitative comparison of the plots shows a clearer difference between recorded and simulated data than only looking at TTC distributions. For example, the gap in the density around (0,0) expected in the absence of collisions, is very weakly pronounced, and shifted in the simulated data.
The conclusions drawn are that:

- DTTC-TTC curves are a good visual aid in both understanding the interactions at a motorway on-ramp.

- IRL-based behaviour model needs further improvements to show the similar behaviours at a ramp, as illustrated by the same DTTC-TTC plots.

6.5.3.9 Integration of IRL model into esmini

A proof-of-concept integration of IRL-based driver behaviour model in the simulation environment esmini. This environment and the interface for external agent controllers are introduced in the following chapter, "esmini overview".

The IRL model received information about the agent's own state and surrounding agents and provided the updates in agent position and speed via UDP communication interface for agent controller. Necessary data pre- and post-processing logics was introduced for conversion between esmini signals and the IRL model outputs, as illustrated in Figure 50. The code for the implementation, with necessary documentation, will be public [44].

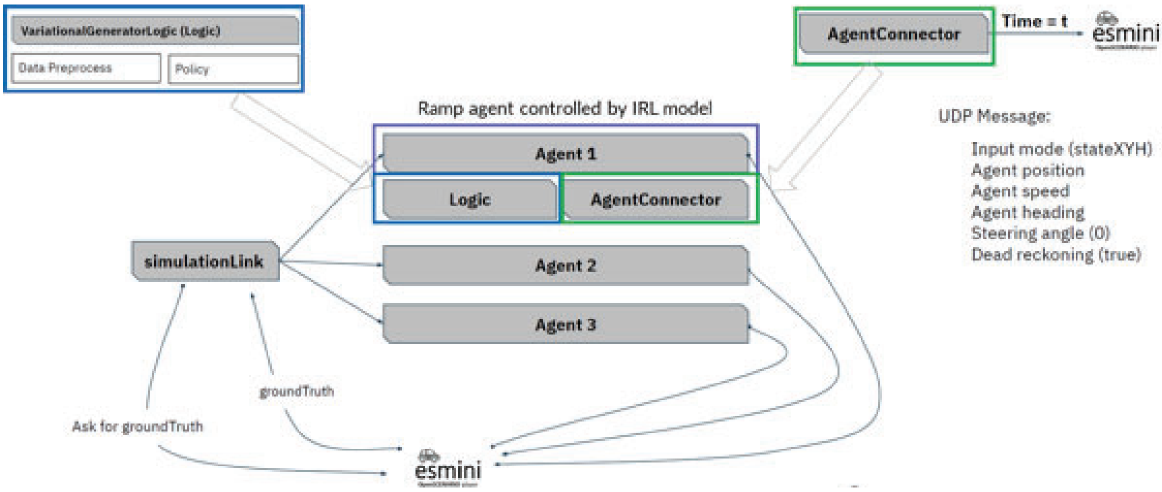


Figure 50. Schematic drawing of IRL driver model integration as external agent controller in ESMINI. Agent 1 is the ramp agent controlled by the IRL model.

While testing the implementation, certain issues with the performance of the scenario playback with IRL model as external controller, have been identified. These issues are well visible in Figure 51 where displacement, velocity, and acceleration of the ramp agent are plotted as function of time, together with recorded trajectory from the same scenario.

Firstly, the IRL model gave considerably faster advance of the ramp agent, while velocity was on average the same as the recorded one. The most credible source of this discrepancy is erroneous outputs from the IRL model when it receives the input from esmini, presumably due to the format of the data. More investigation is required. Secondly, the velocity and acceleration exhibit large spikes in the IRL model. This has been thoroughly investigated, leading to a conclusion that the velocity and acceleration computation in esmini was not fully compatible with an external controller running at lower update frequency. Between the model updates, esmini assumed the object to

stay in the same place. After the model update, the agent position was updated in a step-like manner, which caused a spike in object speed and acceleration.
The solution for this issue has been identified and proposed to the esmini developer community: to introduce a “dead reckoning” agent movement option, which propagates an agent using the last known velocity and acceleration as long as there are no updates from the controller. This improvement has been later introduced in esmini and are expected to solve the identified issues. However, this could not be tested in the on-ramp scenarios due to lack of time.

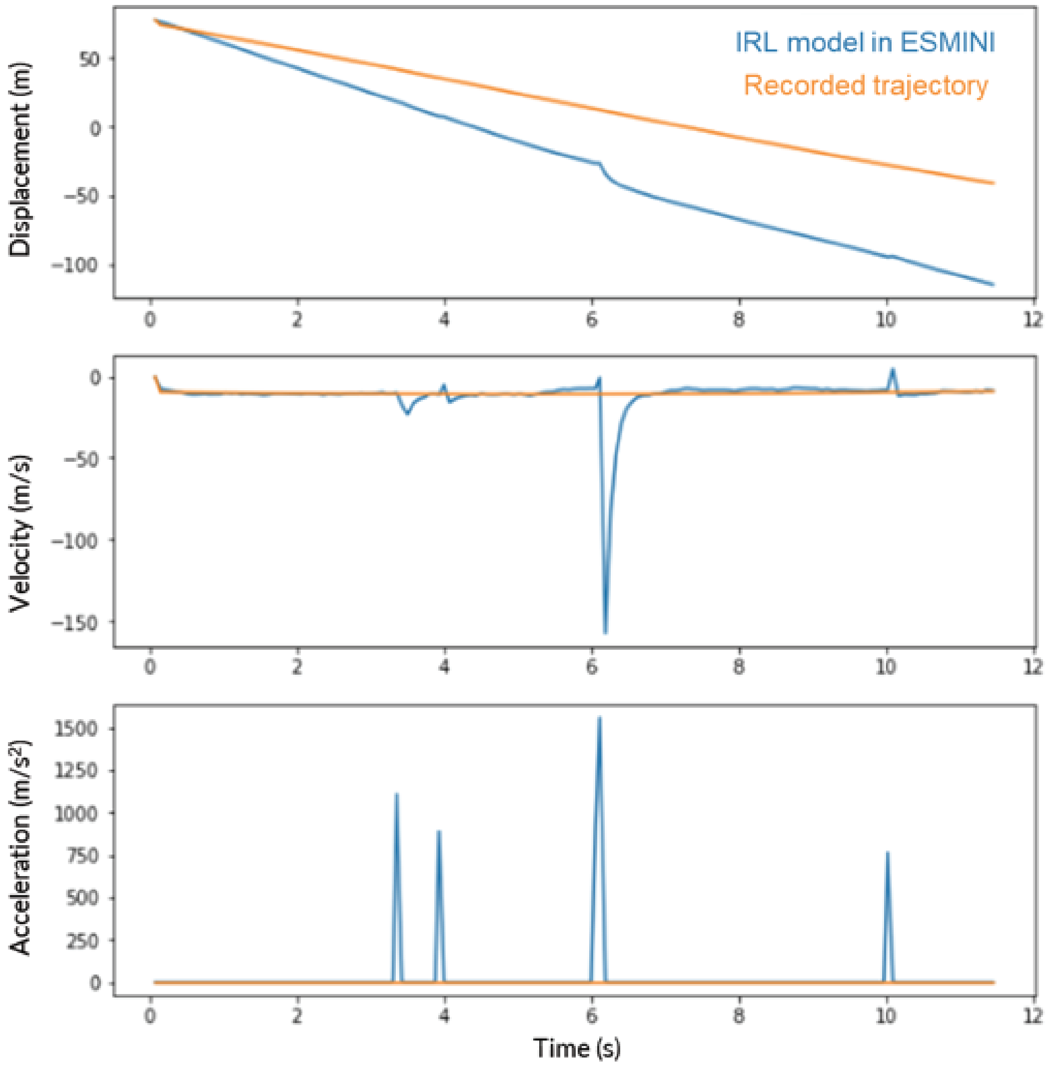


Figure 51. Displacement (top), velocity (middle) and acceleration (bottom) of ramp object steered by IRL model in esmini, with recorded data for the same scenario as a reference.

6.6 esmini overview

esmini [18] is an open-source traffic simulation tool based on C^{++} that executes and visualises traffic scenarios described in the ASAM OpenSCENARIO [16] and OpenDRIVE [15] formats. It is being used as standalone application or as a library linked to custom test frameworks. esmini is available for Windows, Linux, and Mac.

The simulation includes road network and traffic users. It can be either random traffic or deterministic scenarios. The main intended area of use is testing of autonomous drive functions.

A typical use case is to connect esmini to a vehicle simulator. The vehicle simulator handles the ego (Volvo) car including the software under test. esmini simulates the environment, including road network and other road users, communicating the complete "ground truth" back to the vehicle simulator for further processing in sensor models before stimulating the AD/ADAS software, Figure 52. The scenarios can be designed to expose ego vehicle to variations of critical situations.

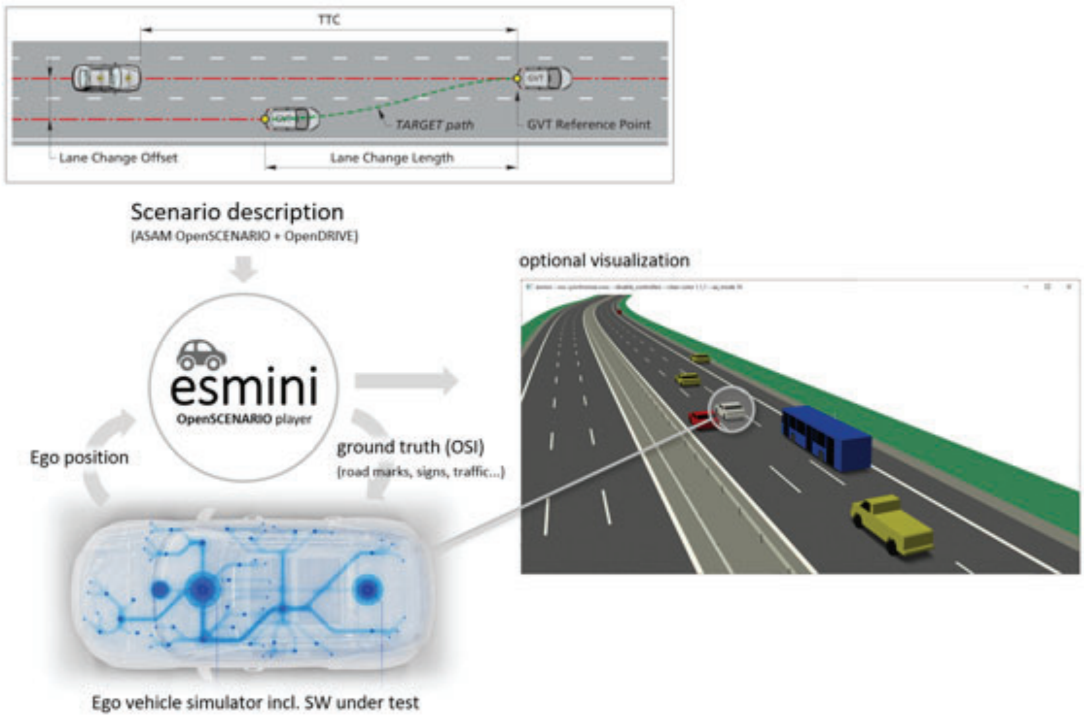


Figure 52. The typical use case is to integrate esmini with a vehicle simulator including the software under test.

The initial version of esmini was a result of the Swedish research project Simulation Scenarios [45], co-financed by VINNOVA. It has since then been continuously developed, mainly by Volvo Cars, and widely used within industry and academia.

6.6.1 Typical integration

The common way of integrating esmini is to link esmini shared library with the custom test platform, Figure 53. The communication of ground vehicle and ground-truth data is then handled in terms of API function calls:

- from esmini: Ground truth (road, other road users)
- to esmini: Ego pose (at least x, y and heading)

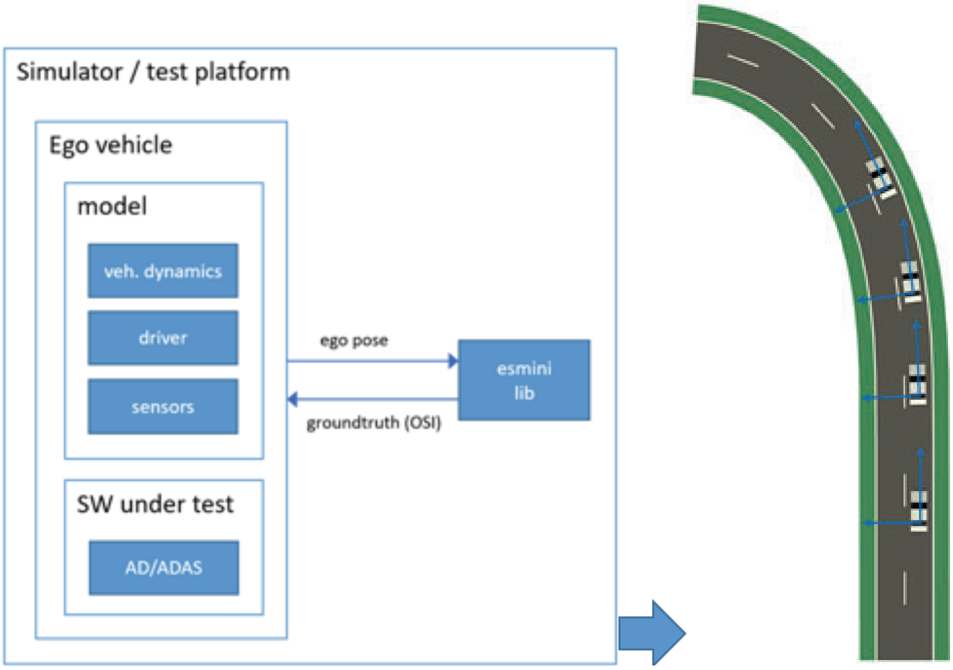


Figure 53. esmini integrated as linked library.

This way of integrating esmini is limited to supported platforms. Another limitation is that if multiple driver model instances are to co-simulate, they all need to be part (i.e., threads) of a main process which owns the esmini instance, although the communication, via function calls, can either be handled per thread or centralized to the main process.

6.7 esmini UDP driver model interface

Use cases addressed by this project involves separate driver model processes, representing one vehicle each, that should interact with the same traffic simulation. A clear requirement for esmini were raised: Provide a flexible interface supporting multiple driver models to manoeuvre separate vehicles in the same simulation session. The driver models may run on various platforms with various update frequency.

The solution implemented in the context of this project establishes such an interface in terms of an OpenSCENARIO controller, which is a standardised way of handing over control of a simulation entity from the standard behaviour defined by OpenSCENARIO to a custom process. To make it fully flexible, the UDP network protocol is utilised for the actual communication of information between esmini and the driver models.

The driver model can be implemented in any programming language and platform supporting UDP communication, which is basically any. For example, it can be an application based on C++, Python or C# running on a PC (Windows, Linux, Mac…), Raspberry Pi or other mobile devices.

The driver model can receive ground-truth information including the road network (road geometry, road marks, signs…) and all other road users, Figure 54. This information is used as input for the driver model, which results in some driver output affecting the motion of the vehicle. This output is communicated to esmini which will handle the actual update of the vehicle state, including position and orientation.

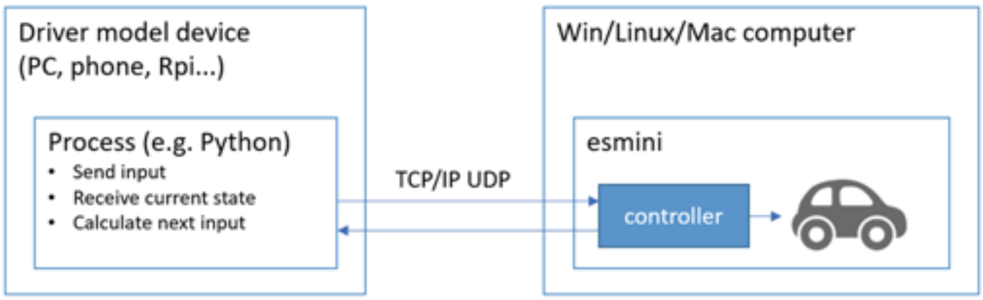


Figure 54. esmini UDP driver model controller overview.

From the driver model perspective, the following steps are performed:

- Assign the controller to any scenario vehicle (in the OpenSCENARIO file)
- Loop:
  - Send input to esmini controller,
  - The vehicle will be updated by the controller accordingly (bypassing default controller)
  - Receive ground-truth from esmini (OSI over UDP)

6.7.1 Input modes

To support various use cases, with different modes of operation, multiple input modes had to be supported.

Input mode 1: Pedals and steering, Figure 55

High level driver input in terms of throttle, brake, and steering angle. esmini will apply these inputs to a simple 2D vehicle model. Latest values will be used for dead reckoning until next message is received.

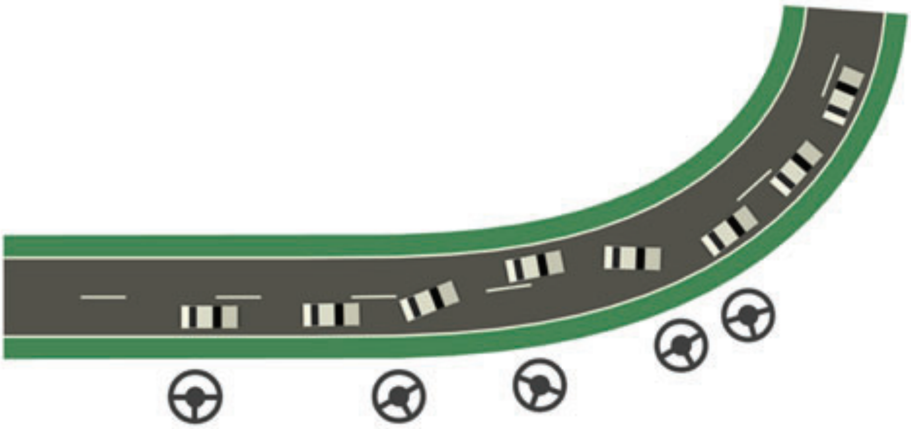


Figure 55. Input mode 1 - pedals and steering

Input mode 2: Exact pose
Exact position and orientation, including position (X, Y, Z) and orientation (Head, Pitch, Roll). This mode is useful when the external driver model either is based on exact trajectory following or do include simulation of complete vehicle powertrain and chassis dynamics, Figure 56. In those cases, the actual final pose is propagated to esmini. Optionally dead reckoning can be activated in which case esmini will update based on latest reported position, orientation, and speed, Figure 57.

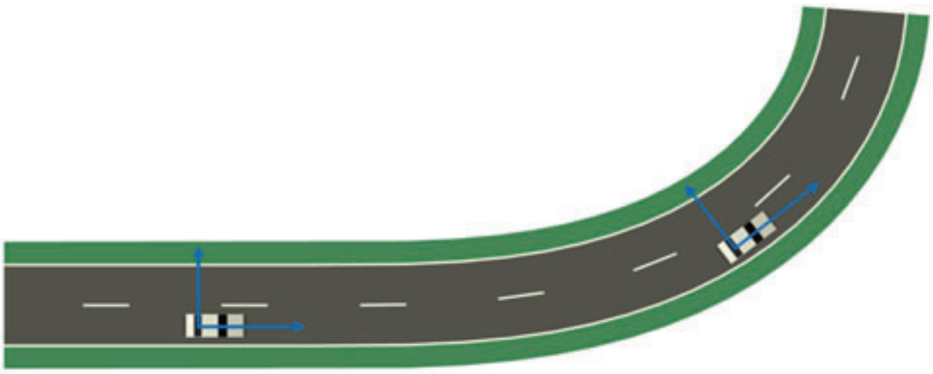


Figure 56. Input mode 2, without dead reckoning.

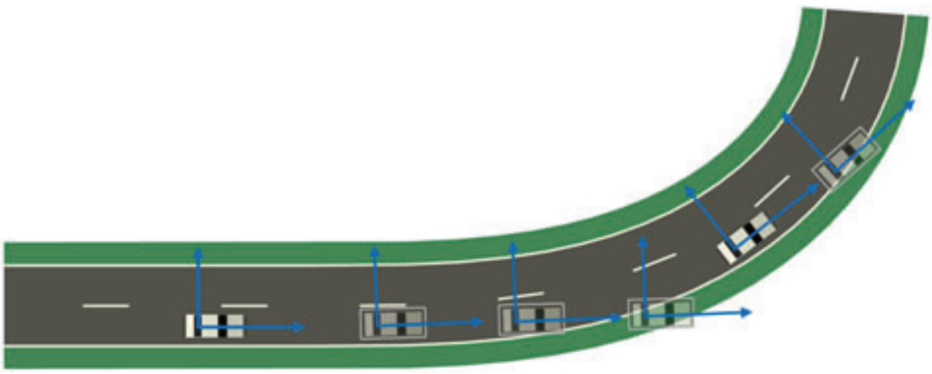


Figure 57. Input mode 2, with dead reckoning.

Input mode 3: Aligned pose

Like input mode 2, but omitting elevation (Z), pitch and roll which instead are calculated by esmini to align with the road geometry, Figure 58.

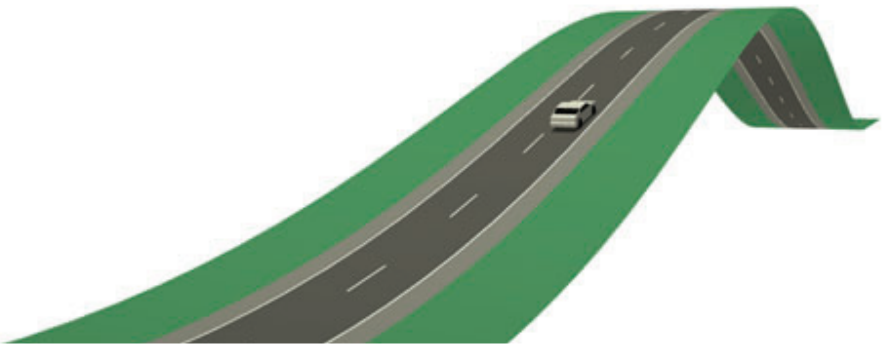


Figure 58. Input mode 3 - esmini takes care of road alignment.

Input mode 4: Heading and speed

This mode is similar to input mode 1. But instead controlling speed and direction with pedals and steering these values are reported directly.

6.7.2 Message implementation

The various message types are defined in terms of simple structs with primitive (universal) data types for maximum portability, Figure 59.

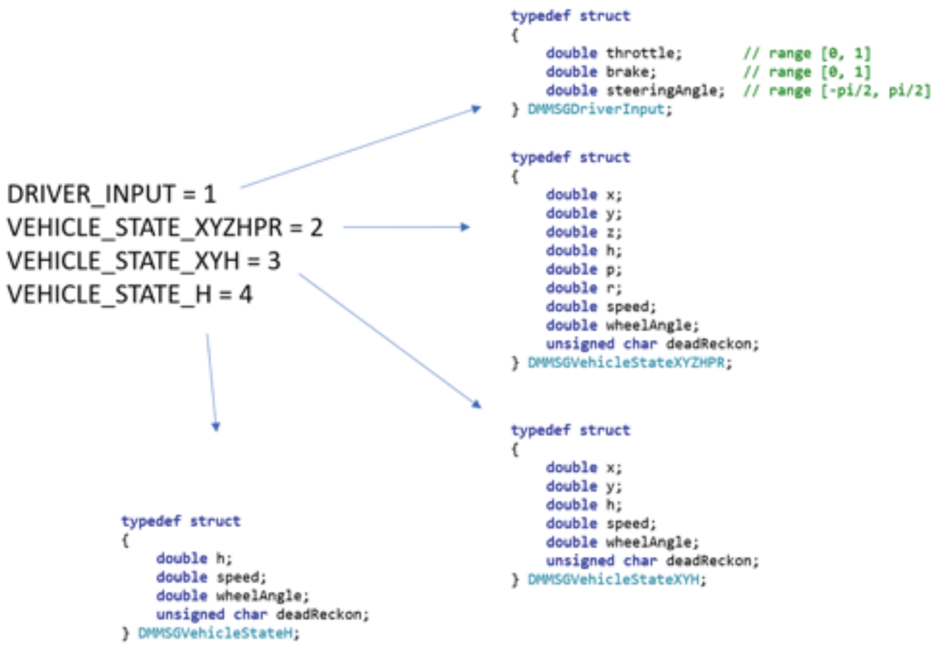


Figure 59. Definition of the input modes.

For complete headers, including full message composition, see [46].

6.7.3 Examples

esmini provides several examples in Python to demonstrate how to use the UDP driver model controller [47] and a video clip is available [48].

The testUDPDriver.py script [49] includes a GUI to control two cars by the various input modes, Figure 60. Instructions on how to run are found in the top comment section of the script file.

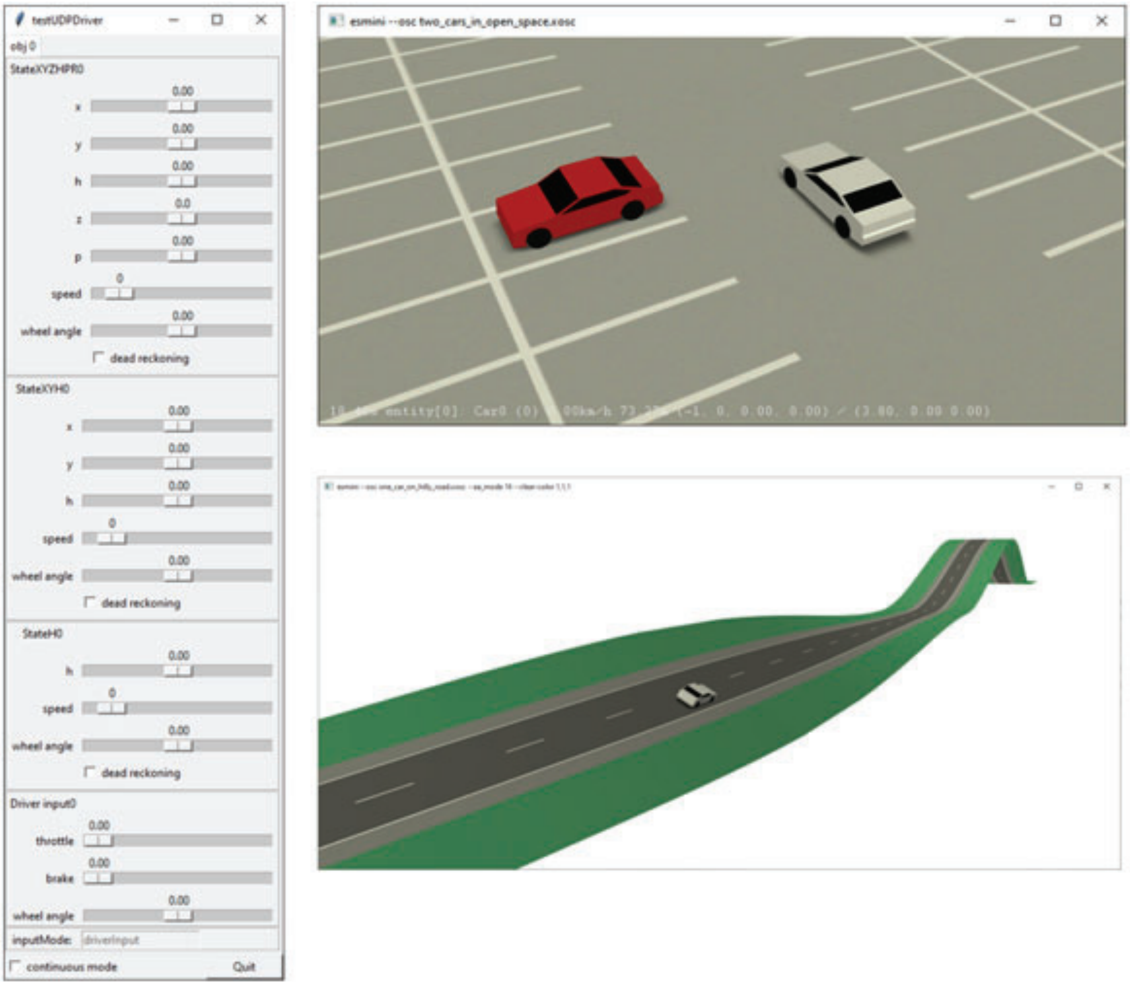


Figure 60. esmini example script "testUDPDriver.py" demonstrating the various input modes.

A video showing the looming controller on highway can be found in [50], and one of the esmini looming controller on narrow road in [51].

6.8 Driver model integration in ATOS

A corresponding driver model interface was added to the ATOS [52] open-source test control system, developed and used by AstaZero. The aim was to enable test equipment to behave according to driver models collected from public road data or parameterised according to literature. To facilitate simple integration with both esmini and ATOS, the same API structure was chosen i.e., the esmini UDP driver model message (DMM) as input for control signals and OSI GroundTruth as output for state data. The test equipment is controlled via a standardised protocol, ISO 22133, meaning that all equipment supporting this protocol can controlled via driver model and generate dynamic environment input for later driver model actions.

The below diagram, Figure 61, shows the essential data flow when controlling test equipment with a driver model. The test equipment (right) is controlled in real time based on the input from the driver model (left). In addition to what is shown, safety data is continuously transmitted and control input is only passed on to the equipment if the test is in the “running” state.

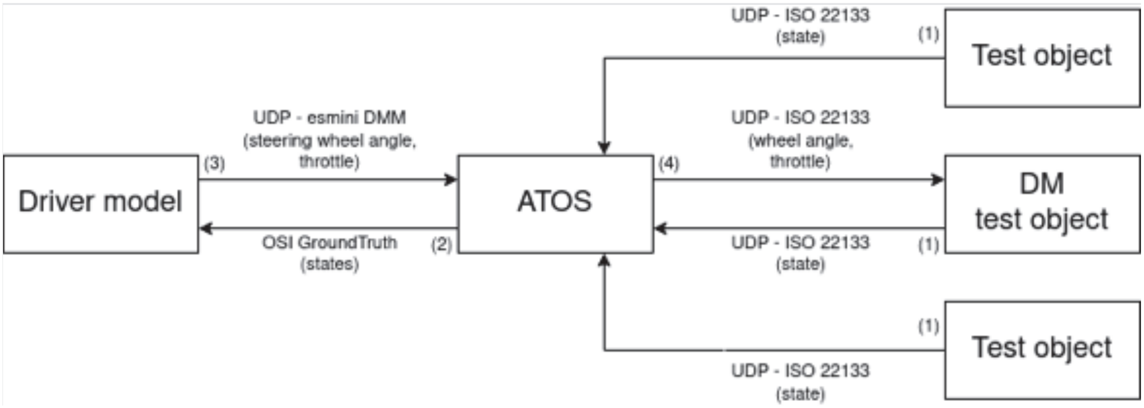


Figure 61. A diagram of the data flow when controlling test equipment with a driver model.

Repeatedly and in order, the sequence of actions is as follows:

1. The test equipment asynchronously reports position, orientation, velocity, and acceleration at 100 Hz. The data is time stamped with onboard GPS clocks.
2. ATOS collects state data into one OSI GroundTruth update. Since the object data arrives asynchronously, ATOS must extrapolate positions of test equipment to the current GPS time based on last known state. The OSI output rate is configurable.
3. The driver model generates a control input based on its environment (populated based on the OSI message) and sends back a DMM.
4. ATOS converts the steering wheel angle and throttle control input into the ISO 22133 format and sends it to the specific test equipment which shall act according to the driver model.

The system was not evaluated in closed loop – instead one of the esmini test GUIs for manually generating UDP driver messages was used, together with an OSI visualiser. This was tested with an RC-car – an interesting future work would be to evaluate the same in closed loop with commercial test equipment.

6.9 SIL toolchain overview

The simulation platform consists of basically four different modules: the input creation, the world engine, the Vehicle Simulator, and the Software under test. An overview can be seen in Figure 62.

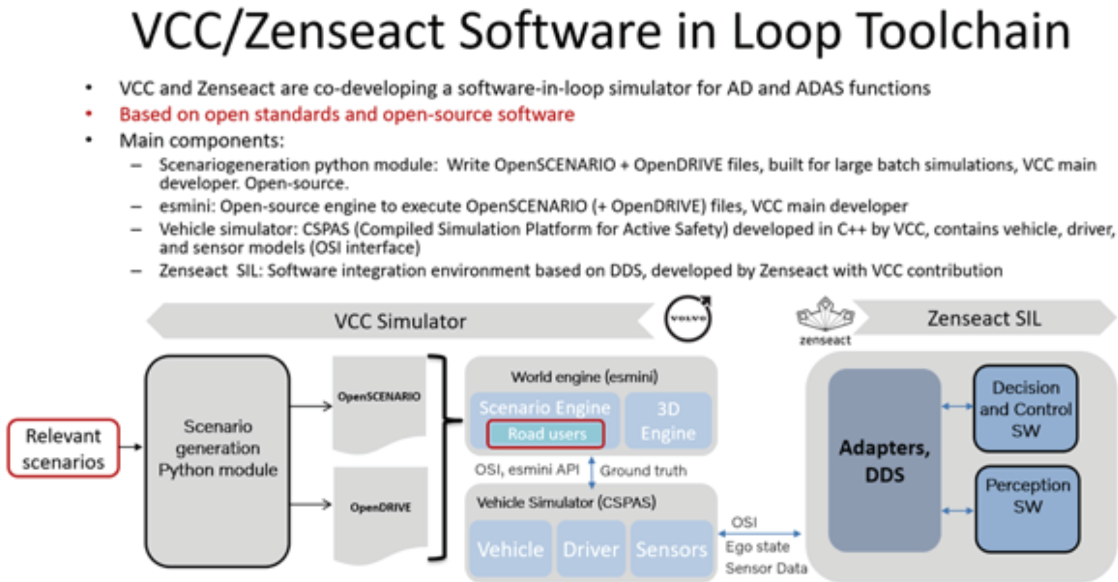


Figure 62. VCC/Zenseact SIL toolchain

In this toolchain, both the input creation and world engine are open-source projects mainly developed by Volvo Cars: the scenario generation python package, and esmini. The Vehicle Simulator is developed in-house at Volvo Cars (CSPAS), and the software under test at Zenseact.

The important part of this simulation platform is that the communication between all these modules is mostly utilising the ASAM standards OpenDRIVE, OpenSCENARIO, and Open Simulation Interface, making the interfaces as standardised as possible.

6.9.1 OpenDRIVE vs OSI

In esmini, the OpenDRIVE is one of the inputs describing the road-network. However, for usage in an AD simulator, another standard must be used as well: The Open Simulation Interface (OSI). It represents the “ground truth” of the simulation, including road-network and all objects moving in the scenario.

OpenDRIVE and OSI do not describe the road-network in a similar manner. OpenDRIVE has represents the road-network with: "geometries", "road", "lane-sections", "lane" and "junction", while OSI only has the concept of lanes.

For the type of highway entry chosen in the ASCETISM project, the options to create such an entry in OpenDRIVE is not unique, since both "Junction" or "Direct Junctions" can be used. In Figure 63, the esmini representation of the OpenDRIVE can be seen on the top, where the road starts are represented by the circles. The middle figure shows the OSI representation of the lanes, on the same road.

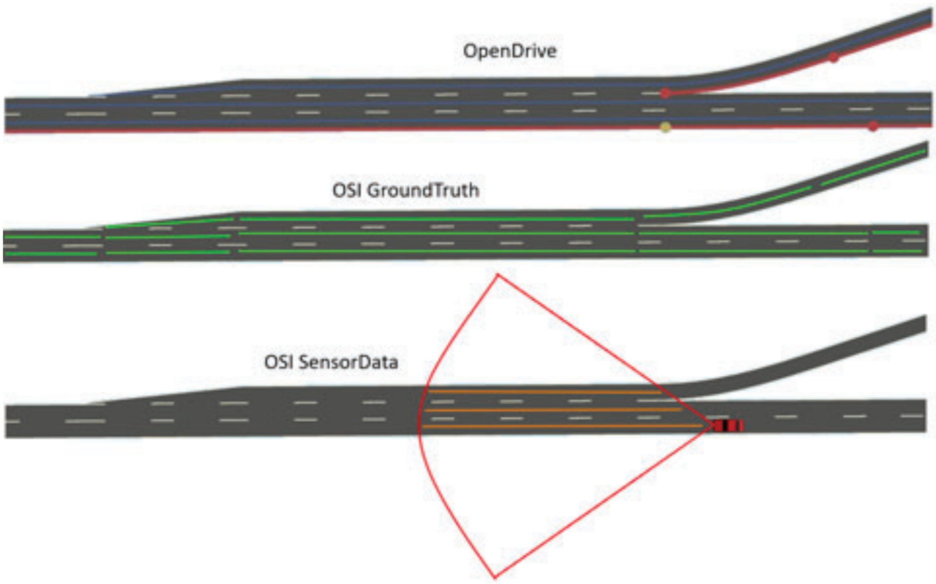


Figure 63. Various representations of a highway entry junction.

Finally, this OSI GroundTruth information is processed in a sensor model, to make out what the sensor actually sees, to form the OSI SensorData that later can be used to feed the AD function. This representation can be seen in the bottom figure of Figure 63. Note that the same consideration must be taken for the lane-markings, but the same principles apply.

6.10 esmini integration – open scenario playback on test track, AstaZero

Several candidate libraries were evaluated for reading OpenDRIVE and OpenSCENARIO into the ATOS control system.

- ad-xolib. A MIT-licensed library for parsing OpenDRIVE 1.7 and OpenSCENARIO 1.1.1 in C++. This was the one finally chosen on the merit that it was actively maintained and designed to be integrated as a separate parsing library.
- pyoscx / scenariogeneration. A python-based scenario generation tool, supporting OpenSCENARIO 1.2.0 and OpenDRIVE 1.7.1. Mainly intended for generating scenarios and not to be used as a parser. As the library is based on python, it would mean for more integration issues with the C++ based ATOS.
- Direct integration of esmini's OpenSCENARIO and OpenDRIVE parsing capabilities into ATOS. While attractive as an option this was ultimately not chosen due to the difficulties in integrating esmini in this way.

An interesting future work, which is ongoing in the project Digital Vehicle Testing at AstaZero (Vinnova dnr. 2022-01648) and was presented at the ASCETISM final demo, is to integrate the whole of esmini as a scenario library.

7 Dissemination and publications

How are the project results planned to be used and disseminated?	Mark with X	Comment
Increase knowledge in the field	X	
Be passed on to other advanced technological development projects	X	
Be passed on to product development projects	X	
Introduced on the market	X	Viscando planning to commercialise both enhanced data processing algorithms and AI-based behaviour models for autonomous driving applications
Used in investigations / regulatory / licensing / political decisions		

Some of the data and code developed in this project will be made available publicly under a common license in the ESTECO git repository [44]

- A trained model
- Integration tools developed in ASCETISM
- A dataset of scenarios used to train the model, based on eight hours of data.

Two successful master's degree projects, and seven course projects at Chalmers and KTH were carried out, which both contributed to ASCETISM deliverables and gave students valuable expertise of working with real-life data and AI models.

8 Conclusions and future research

8.1 Conclusions

The data provided was comprehensive enough to implement effective methods for identifying, analysing, and simulating relevant scenarios observed in the data. By utilizing the data and applying the ASAM OpenDRIVE and OpenSCENARIO standards, one managed to model the road network accurately but also to replay the scenarios in real-time within a simulator. This simulation offered a detailed representation of vehicle dynamics, safety surrogate measures, and interactions among vehicles, thereby providing valuable insights into how human drivers interact with each other.

The real-time simulation enabled us to observe the longitudinal and lateral velocities and accelerations of vehicles, as well as safety surrogate measures such as distance to collision, time to collision, inter-vehicular time, and longitudinal free space. By extracting scenarios based on actual driving data and simulating them, we can effectively assess the robustness and reliability of autonomous driving systems under a wide range of real-world conditions.

Furthermore, the collected data is intended for generating new, synthetic data and, consequently, new scenarios. One of the primary reasons behind the importance of synthetic data lies in the reduction of both time and cost associated with collecting and processing large amounts of real-world data. Gathering data from diverse driving scenarios, road conditions, and traffic situations can be expensive and time-consuming. In contrast, synthetic data can be generated more efficiently and with fewer resources. Moreover, synthetic data allows for the creation of a virtually unlimited number of unique scenarios and enables us to identify dangerous scenarios or other edge cases for testing purposes.

The main conclusions from development of AI-based behaviour model are summarised below.

- IRL was chosen as a behaviour modelling technique in this project as, according to literature, it is the most transferable of the available AI-based methods, capable of generalising human behaviours and identifying hidden motivations behind these.
- Viscando has implemented a novel IRL-based driver behaviour model to simulate the behaviour of vehicles on the on-ramp interacting with vehicles on the motorway.
- This model was trained and tested on recorded on-ramp scenarios. Scenarios were limited to cases where a single on-ramp vehicle is managing the merge with a single vehicle on the motorway.
- Qualitatively, the IRL based behaviour model shows promising results, as it manages to drive the vehicle on ramp and merge to motorway correctly.
- More comprehensive analysis of the model was done by comparing the safety performance of it against the recorded data. The surrogate safety measures, PET and TTM/DTTM ones (the latter proposed and introduced in the ASCETISM project) were used to visualize and quantify the safety performance.
- The IRL-based behaviour model clearly shows capability to interact with the other road users; however, discrepancies can still be observed. Further work is required,

which would both imply training the model on larger datasets and introduction of larger and more advanced networks realising the IRL model.

8.2 Future work

Viscando has an ambition to continue the development of the IRL-based behaviour model, eventually bringing it to industrialisation. More extended analysis of the model performance is required. This analysis will include quantification of decision making by the model, reaction to variations and quantitative comparison of the surrogate safety metrics distribution.

As more data for different on-ramps will be available, transferability of the model between different road layouts will be studied. Furthermore, knowledge of traffic rules and understanding of traffic infrastructure will be introduced, which we believe will make the model more easily transferable to new locations.

AstaZero intends to conduct a comparative analysis between the data collected in this study and data gathered from other ramps and road networks. This analysis will help identify patterns, similarities, and differences in driver behaviour across various locations and traffic situations. Furthermore, AstaZero will focus on generating synthetic data based on the collected data, aiming to create a multitude of dangerous and edge case scenarios specifically tailored for autonomous driving systems.

9 Participating parties and contact persons

AstaZero AB	Katarina Boustedt, Lukas Wikander, Adam Eriksson, Seif Bourogaa
Zenseact AB	Majid Khorsand Vakilzadeh and Team Hades
Viscando AB	Yury Tarakanov
Volvo Personvagnar AB	Emil Knabe
Chalmers tekniska högskola AB	Ola Benderius

10 References

1. J. Colyar and J. Halkias, NGSIM - US Highway 101 Dataset, https://www.fhwa.dot.gov/publications/research/operations/07030/07030.pdf
2. B. Coifman and L. Li, "A critical evaluation of the Next Generation Simulation (NGSIM) vehicle trajectory dataset," Transportation Research Part B: Methodological, vol. 105, pp. 362 – 377, 2017
3. X. Ren, D. Wang, M. Laskey, K. Goldberg, "Learning Traffic Behaviors by Extracting Vehicle Trajectories from Online Video Streams", *14th IEEE Int. Conf. on Automation Science and Engineering (CASE)*, Munich, 2018, pp. 1276-1283
4. D. Notz, F. Becker, T. Kühbeck, D. Watzenig, (2020). Extraction and Assessment of Naturalistic Human Driving Trajectories from Infrastructure Camera and Radar Sensors. arXiv preprint arXiv:2004.01288.
5. Bosch Smart City, https://cds.bosch.us/smart-city
6. R. Krajewski, J. Bock, L. Kloeker, L. Eckstein, "The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems," 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, 2018, pp. 2118 - 2125
7. G. Markkula, O. Benderius, K. Wolff, M. Wahde, A Review of Near-Collision Driver Behavior Models, Human Factors: The Journal of the Human Factors and Ergonomics Society, Vol. 54 iss. 6, 1117-1143, 2012
8. O. Benderius, G. Markkula, Evidence for a fundamental property of steering, Human Factors: The Journal of the Human Factors and Ergonomics Society, Volume: 58 issue: 1, 884 - 888, 2014
9. G. Markkula, O. Benderius, M. Wahde, Comparing and validating models of driver steering behaviour in collision avoidance and vehicle stabilisation, Int. J. of Vehicle Mechanics and Mobility, 52:12, 2014
10. O. Benderius, Modelling driver steering and neuromuscular behaviour, Ph.D. Thesis, Chalmers U. Tech., 2014
11. S. Tuhkanen, J. Pekkanen, P. Rinkkala, C. Mole, R. M. Wilkie, O. Lappi, Humans Use Predictive Gaze Strategies to Target Waypoints for Steering, Scientific Reports volume 9, Article number: 8344 (2019
12. G. Markkula, J. Engström, J. Lodina, J. Bärgman, T. Victor, A farewell to brake reaction times? Kinematics-dependent brake response in naturalistic rear-end emergencies, Accident Analysis & Prevention, Volume 95, Part A, October 2016, pp. 209 - 226
13. N. Kalra, S. M. Paddock. "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?", Transportation Research Part A: Policy and Practice 94 (2016): 182 - 193.

14. Waymo blog post, https://blog.waymo.com/2020/10/revealing-our-approach-to-safety.html
15. OpenDRIVE, https://www.asam.net/standards/detail/opendrive/
16. OpenSCENARIO, https://www.asam.net/standards/detail/openscenario/
17. FFI project Simulation Scenarios, ref. 2016-05495, https://www.vinnova.se/en/p/simulation-scenarios/
18. Esmini, https://github.com/esmini/esmini
19. The Pegasus project, https://www.pegasusprojekt.de/en/about-PEGASUS
20. The Pegasus method, https://www.pegasusprojekt.de/en/pegasus-method
21. ASAM, https://www.asam.net
22. ISO/TC 22/SC33/WG09 "Test scenarios of automated driving systems", https://www.iso.org/committee/5383785.html
23. FFI Project ESPLANADE, https://esplanade-project.se
24. ISO 26262, https://www.iso.org/standard/43464.html
25. Chronos part 2, https://www.astazero.com/chronos-part-2/
26 . M. Gyllenhammar, C. Zandén and M. Törngren, "Defining fundamental vehicle actions for the development of automated driving systems.," in SAE technical paper series, 2020.
27. K. Czarnecki, "Operational World Model Ontology for Automated Driving Systems--Part 1: Road Structure," Waterloo Intelligent Systems Engineering Lab (WISE) Report, Waterloo, 2018
28. K. Czarnecki, "Operational World Model Ontology for Automated Driving Systems--Part 2: Road Users," Waterloo Intelligent Systems Engineering Lab (WISE) Report, Waterloo, 2018.
29. J. Bach, S. Otten and E. Sax, "Model based scenario specification for development and test of automated driving functions," in *IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, 2016.
30 . G. Bagschik, T. Menzel and M. Maurer, "Ontology based Scene Creation for the Development of Automated Vehicles," in *Proceedings of 2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, China, 2018.
31. A. Pütz, A. Zlocki, J. Bock and L. Eckstein, "System validation of highly automated vehicles with a database of relevant traffic scenarios," *Situations,* vol. 1, p. E5, 2017.
32. M. Kochenderfer and T. Wheeler, "Factor Graph Scene Distributions for Automotive Safety Analysis," in *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, 2016.
33. J. K. Kruschke, "Bayesian estimation supersedes the t test," *Journal of Experimental Psychology: General,* vol. 142, no. 2, p. 573, 2013.
34. Camerota Verdù, F.J., Inverse Reinforcement Learning for Autonomous Vehicle Validation and Traffic Safety., Master's Degree thesis in Data Science and Scientific Computing, Dept. of Mathematics and Geosciences, University of Trieste, 2020.

35. H. Weber, J. Bock, J. Klimke, C. Roesener, J. Hiller, R. Krajewski, A. Zlocki and L. Eckstein, “A framework for definition of logical scenarios for safety assurance of automated driving,” *Traffic Injury Prevention,* vol. 20, pp. S65-S70, 2018.
36. L. Klitzke, K. Gimm, C. Koch and F. Köster, “Extraction and Analysis of Highway On-Ramp Merging Scenarios from Naturalistic Trajectory Data,” in *In 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022.
37. Y. Zhang, Y. Zou, Y. Wang, L. Wu and W. Han, “Understanding the merging behavior patterns and evolutionary mechanism at freeway on-Ramps,” *Journal of Intelligent Transportation Systems,* pp. 1-14, 2022.
38. J. Zhu and I. Tasic, “Safety analysis of freeway on-ramp merging with the presence of autonomous vehicles,” *Accident Analysis and Prevention,* vol. 152, p. P. 105966, 2021.
39. F. Nielsen, “A family of statistical symmetric divergences based on Jensen's inequality,” *arXiv preprint arXiv:1009.4004 ,* 2010.
40. open-source repository “scenariogeneration” https://github.com/pyoscx/scenariogeneration
41. Corredor, S.E.A., Modelling of Human Behaviour in Traffic Interactions Using Inverse Reinforcement Learning. Master’s Degree thesis in Engineering Mathematics and Computational Science. Chalmers University of Technology, 2021
42. Dahlén, F., Gkritsis, A., Nasir, S., “AI against AI: simulation of merging interactions between road user agents trained using novel DNN methods”, Project report, Mathematical and statistical modelling course, Chalmers University of Technology, 2023. This report is not available publicly but is appended to the ASCETISM report.
43. Benson, L., Statistical assessment of Inverse Reinforcement Learning model performance in traffic merging interactions. Course report, Mathematical and Statistical modelling course, Applied Mathematics, Chalmers University of Technology, 2023. The report is not available publicly, so it is appended to the ASCETISM report.
44. Code and documentation for the implementation of the IRL model in esmini, https://github.com/ESTECO-SPA/gemini.git
45. Vinnova project Simulation Scenarios, https://sites.google.com/view/simulationscenarios
46. Esmini on github, https://github.com/esmini/esmini/blob/master/EnvironmentSimulator/Modules/Controllers/ControllerUDPDriver.hpp
47. UDP driver model controller, https://github.com/esmini/esmini/tree/master/scripts/udp_driver
48. Video clip of the testUDPDriver.py script, https://youtu.be/IuwuGE3BzKw
49. The testUDPDriver.py script, https://github.com/esmini/esmini/blob/master/scripts/udp_driver/testUDPDriver.py
50. A video showing the looming controller on highway, https://youtu.be/X1D4b2xZiJc
51. Video of esmini looming controller on narrow road, https://youtu.be/t2SjoaIpGpo
52. ATOS – Automated vehicle Test Operating System, an open-source project, https://github.com/RI-SE/ATOS