Calibration Of Traffic Micro-Simulation For Modelling The Effects Of Autonomous Vehicle Integration On Traffic Safety

Diederik Huige[1], Ryo Sugimura[1], Teun van de Laar[1], Hicham Rahali[1], Xingyi Li[1], Vincent Kuipers [1]

***Abstract*—The testing of autonomous vehicles (AVs) is troublesome since testing in a real-world environment is unsafe. For this reason, agent-based traffic modelling software Aimsun Next is used to replicate a real-world traffic scenario from HighD trajectory data. Simultaneous Perturbation Stochastic Approximation (SPSA) and a loss function based on the Kullback-Leibler divergence test are used to calibrate a traffic micro-simulation. The application of SPSA yields car-follower parameters that lower the loss. However, the number of iterations taken to calibrate the system is found to be insufficient to conclude the convergence of the calibration. The penetration rate of AVs is tested in the calibrated simulation. For AV penetration rates until 50%, AVs do not contribute much towards improving traffic safety. However, increasing the penetration rate further than 50% increases safety.**

***Index Terms*—Agent-Based modelling, Traffic micro-simulation, Aimsun, HighD, Gipps driver model (Gipps), KL divergence test, Simultaneous Perturbation Stochastic Approximation (SPSA), Autonomous Vehicle (AV), Penetration rate**

I. Introduction

Traffic safety is a matter of critical importance in modern society. 1.3 million people die in road accidents globally each year, where traffic accidents are the leading cause of death for children and adults from the age of 5 to 29 [1]. According to previous as well as ongoing research, integration of autonomous vehicles (AVs) on the road could improve traffic safety substantially by removing the human factor from the equation [2]. The human factor relates to wilful violations of safety rules as well as inattention, fatigue, and intoxication [3]. Thereby reducing the element of human error in driving which is at the basis of many road traffic accidents.

However, the testing of AVs can usually not be conducted in a real-world environment, as the risk of causing accidents is too great. The outcome of a number of legislative cases related to accidents with AVs has demonstrated the ethical and legislative complexity of introducing such vehicles into the road network, even when strictly monitored [4] [5]. As a result, alternative methods are necessary to investigate the effects of AVs on the road. One of these methods would be the use of accurate traffic simulation environments.

In a previous study, Young et. al [6] give a comprehensive overview of the usefulness of current traffic safety simulation and previous research leading up to it. It discusses the advantages and disadvantages of the recent simulation models such as safety measures and driver behaviour modelling and the simulation's usefulness in road infrastructure areas. The usefulness of the simulation can be assured only if the simulation is a good representation of the driving environment and behaviour [6].

Modelling traffic scenarios

Modelling of traffic scenarios can be done on multiple levels, micro, meso and macro. A macroscopic model considers the relationship between traffic flow characteristics while a microscopic model considers the individual interactions between individual vehicles. Microscopic models are often simulated with an agent-based simulation, a representation of a system composed of interacting decision-making entities called agents. Agents update their state at each time step based on behavioural rules. In a traffic simulation, the agents are vehicles, cyclists, pedestrians, etc. Each of them has local rules, such as distance-keeping and car-following. When observing the simulation there are emergent properties which are of interest, examples are traffic intensity, density and speed averages. These traffic flow properties are a result of interacting agents during the simulation. Safety is a major component of traffic simulations, metrics to investigate safety are called Surrogate Safety Measures (SSMs). A commonly used metric is Time To Collision (TTC), the time before a collision will occur if all speeds stay the same.

In this project, a microscopic model is simulated in an environment based on real-world traffic data using Aimsun Next, a proprietary software in which multi-resolution models of real-world data can be simulated. These models can be of different sizes, from a single intersection to an entire region [7]. Aimsun Next uses a car-following model that computes the next step of the simulation to simulate the interaction between vehicles and their environment. The car-follower model used in this study is the Gipps driver model (Gipps model for short)[8][9]. The parameters inside this model dictate the vehicle's behaviour in the simulation. Emergent properties of traffic flow and SSM will be used to calibrate the model. When the model is calibrated it will mimic the real-world scenario for a specific dataset.

Problem statement

The purpose of this project is to investigate the effect of AVs on road safety. This is done by developing a system to calibrate

[1]*Dept. of Electrical Engineering*, *University of Technology Eindhoven*, Eindhoven, The Netherlands, [d.n.j.huige; r.sugimura; t.a.v.d.laar; h.rahali; x.li; v.b.f.kuipers]@student.tue.nl

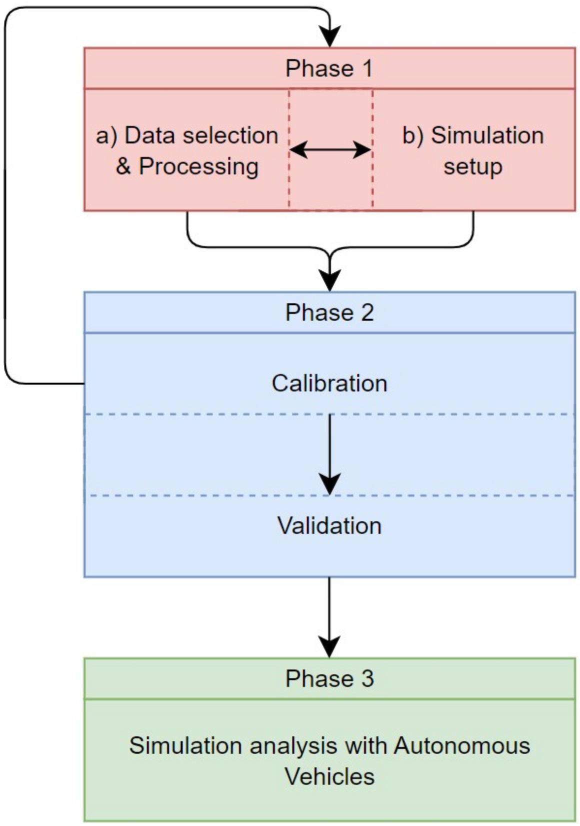


Fig. 1: Breakdown of the steps for the project.

a traffic micro-simulation from the perspectives of traffic and safety in relation to a real-world data set, then replacing human drivers with autonomous vehicles, and observing how indicators of safety and traffic flow are affected by the insertion of AVs.

II. Method

The initial steps consisted of investigating relevant state-of-the-art literature. With the knowledge of current literature, the project could be defined and presented in this chapter. The project is separated into 3 phases, the phases are shown in Fig. 1. The first phase consists of two steps which are performed in parallel. The first phase essentially consists of preparation work for the calibration and validation phase which follows. A calibrated and validated model is necessary in order to start with phase 3, the simulation analysis with AVs. In this section, the method for each of the phases is explained.

Phase 1 starts off with research into available datasets. The data is selected on certain criteria for it to be most suited for the task (Phase 1A). The selection criteria are:

- *Non-complex scenario*: the data should contain a minimal number of agent categories (cars, pedestrians, etc.) and a simple road layout.
- *Locality information*: it should be known where the data is recorded.
- *Trajectory data*: the trajectory data of each vehicle should be known.
- *Extractable SSMs*: some SSMs should already be included or be easily extractable.
- *Amount of (relevant) data*: the number of vehicles and recording time of the dataset should be sufficient to do a proper analysis.

Phase 1a: Data selection & Processing

Dataset: Out of the analysis, in which 22 datasets are considered, the HighD dataset [10] is chosen. The HighD dataset consists of drone-recorded data on the German highway at several locations. It is first of all chosen because the interactions between the agents are relatively simple: modelling a highway scenario is less complex than an urban scenario since there are more agents involved in an urban environment. The second argument for using this dataset is the excellent trajectory data. The typical positioning error is smaller than 10 cm [10]. A final benefit is the inclusion of several SSMs and the ease at which others could be extracted from the data. From the six available locations, one location is chosen (location 2) to be used for this paper. This location includes two lanes per direction, with a left and right driving direction. Only the left-heading side of the road is taken since this prevents any statistical interference from other road sections. The video recording of the chosen location is post-processed to extract the car and truck location, time, headway distance, etc. Fig. 2 shows the used layout.

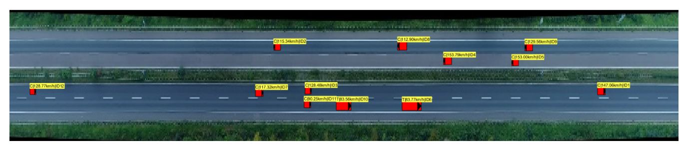


Fig. 2: Example image of the HighD dataset [11].

Parameters: Several parameters are selected for the calibration process: intensity, density, speed distribution (traffic flow parameters), TTC and Deceleration Rate to Avoid Crashes (DRAC) (SSMs). The selection of the traffic flow parameters is based on an extensive literature review [12] [13].

With regard to the safety metrics, TTC is widely used to define the severity of traffic conflicts and will be used to calibrate the simulation. The TTC is also directly given in the HighD dataset. For better calibration, DRAC is used besides TTC. On a highway at high speeds, the deceleration rate is important to identify severe traffic conflicts. DRAC can be calculated from the provided dataset. Both these safety metrics are commonly used in traffic micro-simulations [14].

Next, the formulas are given for all of these parameters. These are based on the formulas in the Aimsun documentation [8]. Intensity is calculated using the following equation:

$$F = \frac{N_{entering}}{I} \tag{1}$$

where I is the interval of statistics in hours, F is the intensity during period I in veh/h and $N_{entering}$ is the number of vehicles that enter the network during that period. Density is calculated using the following equation:

$$D = \frac{N}{L} \tag{2}$$

where D is the density of the network in veh/km, N is the total number of vehicles in the network and L is the total length of all lanes of all sections of the network in km. The speed of each vehicle at simulation step s is denoted as v_s, in km/h. The TTC is calculated using the following equation:

$$TTC = \frac{d}{v_{prec} - v} \tag{3}$$

where TTC is in seconds, d is the distance between the following and preceding vehicle in meters, (note that the car size is not taken into account), v and v_{prec} are the velocity of the following vehicle and the preceding vehicle in m/s, respectively. The obtained value belongs to the following vehicle. The equation for DRAC is as follows:

$$DRAC = \frac{(v_{prec} - v)^2}{2 \cdot d} \tag{4}$$

where $DRAC$ is the DRAC in m/s^2.

Correlation between traffic flow parameters: With the method described in the previous section, the traffic parameters of interest are extracted, in particular the Car speed, Truck speed, Intensity and Density. In this section, the correlation between these parameters is investigated. This is to understand the relationships between the parameters, this information can be used to better calibrate the traffic model. Intensity and Density are expected to be positively correlated to a certain degree. This is because they both measure the number of vehicles in a time interval, the difference is that Intensity uses length and Density area. Since a highway scenario with no exit or entrance ramps is evaluated all vehicles that enter the simulation will exit the simulation, based on this it is expected that both the Intensity and Density parameters will show similar behaviour and one of them is redundant.

To gain more insight into the correlation between the traffic parameters, Pearson's Product Moment Correlation Coefficient and Principle Component Analysis (PCA) will be used. This coefficient measures the degree of correlation there may be between two variables. A value of 0 means no correlation and -1 and 1 mean a perfect negative and perfect positive correlation, PCA is used to gain additional insight. The results of this analysis are given in Section III. The results indicate that using both Intensity and Density is not necessary, for this reason, only Density is considered for the rest of this paper.

Parameter sub-division: The aforementioned parameters of interest (density, speed, TTC, and DRAC) all need to be handled somewhat differently. Density is a macro-parameter, meaning that it is an effect of the interaction of all vehicles in a section. A single distribution of density values over the simulation time gives appropriate information. For speed, on the other hand, the different vehicle types, i.e. cars and trucks, are assessed separately because they have different characteristics. To avoid contributions from slower vehicles dominating as a result of being in the simulation for longer amounts of time, the mean speed is taken for each vehicle in the distribution, rather than every value per frame.
To calculate the TTC of a vehicle, the distance to the vehicle in front is needed, irrespective of the vehicle type. Therefore, the distributions are not separated in this case. Again, to avoid slower vehicles dominating, not all values per vehicle are taken into account over the whole trajectory. To emphasise the critical TTC values, the lowest positive value over the trajectory is taken for each vehicle. Similarly, for DRAC, critical situations occur when the values are high, so only the maximum values are taken into account.

Phase 1b: Simulation setup

The simulation setup is based on the available data from the first HighD dataset. The road is 420 meters long and has two lanes. The initial traffic demand in the simulation is based on the total amount of cars and trucks that passed during the recording. For each vehicle type (car and truck), the mean, standard deviation, minimum and maximum length and width are retrieved and used as input for the dimensions of the simulated vehicles. The initial desired speed is based on the mean speed of each vehicle type. This initial setup is the baseline on which calibration will be performed. Fig. 3 shows the start of the modelled road including the centroid used to generate vehicles. On the other side, another centroid is used to attract vehicles.

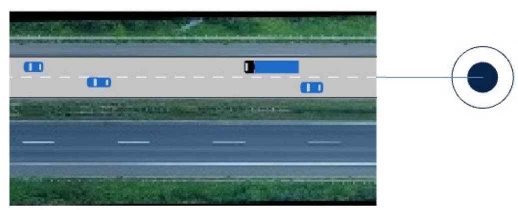


Fig. 3: Segment of simulated road.

Since Aimsun Next has a wide variety of parameters which can be used for calibration of the traffic and safety parameters, only 7 parameters will be changed for simplicity. These parameters are as follows:

- *Max desired speed*: the preferred speed at which the car would travel under normal circumstances.
- *Max acceleration*: the maximal acceleration allowed.
- *Normal deceleration rate*: the deceleration coefficient of which a car/truck would normally brake with.
- *Sensitivity factor*: a measure of how much a vehicle is an effect on how the other vehicles act.
- *Reaction Time*: the speed at which a vehicle would on average react on other vehicles.
- *Vehicle demand*: the number of cars on average that pass through a section in an hour.
- *Gap*: the minimum time on average that a previous car has to be in front (i.e. how much space is left in front of the vehicle).

These parameters will be changed within normal ranges found in the literature [15]. These ranges are shown in Appendix F. Since there are cars and trucks in the simulation, the parameters will be altered for both cars and for trucks,(14 parameters in total), to better match the data in the HighD dataset.

Phase 2: Calibration & Validation

The goal of phase 2 of this project is to make a simulation that resembles real-world traffic as accurately as possible before the insertion of AVs. The easiest way to accomplish this would be to exactly match the trajectories of the real-world vehicles. However, by doing that, the simulation would lose all sense of generality. Instead, it would be much more useful to match the real-world scenario statistically as accurately as possible. Therefore, the aim is to make the distributions of the selected traffic and safety parameters of interest over time match the real-world distributions as closely as possible.

In order to calibrate the simulation to the real-world data, two methods are investigated. The first is using the Aimsun-Next calibration tool, which compares the simulation results with the real-life dataset over a set of intervals. At each interval, the relative difference between the results is given. This could be useful if a detector-based dataset is used which has temporal information on a specific road section. However, the HighD dataset has video-recorded trajectories which include more information. To utilise the trajectories in calibration ideally a different method has to be used. The other method to calibrate to the real-world data is based on a method presented in a paper by D. Sha et al. [16], in which a Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm is used to calibrate the simulation. The latter is chosen since it is based on external software (i.e. a self-written script created in Python), which allows for more flexibility than the Aimsun tool. The procedure is explained in more detail in the *Model calibration algorithm*-section.

Distribution-matching: In order to match the distributions obtained from the HighD dataset with the Aimsun Next simulation, a metric that measures the similarity of the two distributions is needed. To this end, two methods are investigated: the Kolmogorov-Smirnov test (KS test) and the Kullback-Leibler divergence test (KL divergence test). Both tests give a score on the similarity between the two distributions.

At the basis of the KS test lies the hypothesis that two distributions originate from the same distribution. The maximum difference between the two cumulative distribution functions, the statistic, is calculated with:

$$D = \sup_x |F_{data}(x) - F_{sim}(x)|, \tag{5}$$

where $F_{data}(x)$ and $F_{sim}(x)$ are cumulative distributions obtained from the HighD dataset and the Aimsun simulation, respectively. $\sup_x$ denotes the supremum of the difference, which is the maximum absolute value. The p-value is the likelihood of obtaining a certain value of D from the Kolmogorov-Smirnov distribution. It is taken as the goodness-of-fit measure, and if it is under a certain threshold the hypothesis is rejected. This means that the real-world data and simulation would not come from the same distribution and are therefore not similar. To use this metric to measure the similarity of the simulation and the real-world data, the distributions of the traffic and safety parameter values have to be extracted from the HighD dataset and the simulation.

The KL divergence test can also be used to determine the similarity between distributions. This test measures the distance between a distribution $P(x)$ and another distribution $Q(x)$. The (discrete) KL divergence test is defined as:

$$D_{\mathrm{kl}}(P||Q) = \sum_{x\in\mathcal{X}} P(x)\log\left(\frac{P(x)}{Q(x)}\right), \tag{6}$$

where the sum is taken over the bins of (normalised) histograms, with identical bin sizes of the two distributions. Note that the result is non-negative and the KL divergence test is only valid if the bins of histograms have nonzero counts on the domain. The straightforward implementation of histograms in the KL divergence test is a great advantage with respect to the KS test, where a distribution has to be fitted to the data. Furthermore, the KL divergence test incorporates the entirety of the data, while the KS test focuses on the maximum difference. Therefore, the KL divergence test is chosen as a metric to compare the distributions of the safety and traffic parameters.

The loss function used in the calibration is the weighted sum of the KL divergence between all distributions of safety and traffic parameters (car speed, truck speed, density, TTC and DRAC):

$$L = \sum_i w_i\, D_{\mathrm{kl}}(P_i||Q_i), \tag{7}$$

where i indicates the parameter. The weights can be used to additionally penalise a parameter if needed.

Model calibration algorithm: In the SPSA algorithm [16], the best parameter values are found using an update rule, which has the following form:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k\hat{g}_k(\hat{\theta}_k). \tag{8}$$

$\hat{\theta}_k$ is the parameter vector at time step k, a_k is a vector of positive step sizes, and $\hat{g}_k$ is the approximation of the gradient of the loss at the current iteration. This gradient is approximated by perturbing the parameters twice[2], in opposite directions in the loss space:

$$\hat{g}_k\left(\hat{\theta}_k\right) = \begin{bmatrix} \frac{L\left(\hat{\theta}_k + c_k\Delta_k\right) - L\left(\hat{\theta}_k - c_k\Delta_k\right)}{2c_k\Delta_{k1}} \\ \cdots \\ \frac{L\left(\hat{\theta}_k + c_k\Delta_k\right) - L\left(\hat{\theta}_k - c_k\Delta_k\right)}{2c_k\Delta_{kp}} \end{bmatrix} = \frac{L\left(\hat{\theta}_k + c_k\Delta_k\right) - L\left(\hat{\theta}_k - c_k\Delta_k\right)}{2c_k}\begin{bmatrix} \Delta_{k1}^{-1} \\ \Delta_{k2}^{-1} \\ \cdots \\ \Delta_{kp}^{-1} \end{bmatrix} \tag{9}$$

in which c_k is the general perturbation size and Δ_{kj} is the perturbation per parameter.

[2]This perturbation is based on the ranges found in the literature divided by a factor of 10

Validation: Validation steps are performed to verify the correctness of the calibration and the generality of the simulation. Using the calibrated system, other sets of data are investigated to test if the loss retrieved using the distribution matching is similar. Since only one side of the highway is used for calibrating the system, the other side is used for validation. Additionally, other HighD data files are used to validate. The results are assessed on similarity by comparing the KL divergence of the resulting distributions. The results of the calibration and validation based on this method are given in Section III.

Phase 3: Autonomous Vehicle analysis

After calibrating the simulation, the AV model can be introduced. SAE [17] divide autonomous vehicles into six levels of automation:

- 0 - 2: driver support features, e.g., Adaptive Cruise Control, lane centring, and emergency braking.
- 3 - 5: includes automated driving features, e.g., traffic jam chauffeur.
- 4 - 5, the automated system will not request the human to drive.

We only consider levels 4 and 5, since modelling human driver behaviour adds unnecessary complexity to the project. To make the analysis applicable to the rest of the simulation, the used AV model is based on the same car-follower model as the calibrated human-driven vehicles (Gipps driver model), which means that the AVs don't have connective and cooperative capabilities. We distinguish two AV types: assertive and cautious, following previous research [18]. Their car-follower parameters characterise their different behaviour. The previously calibrated car-follower parameters are changed for both AV types, following the trend presented in the same paper. The exact changes are shown in Appendix E.

For both AV types, their effect on the safety and traffic parameters is tested by varying the penetration rate: a percentage of cars in the simulation is exchanged for AVs. Only one type of AV is inserted at a time, and the penetration rate (of the total number of cars) is varied between 0% and 100%. Two types of analyses are done: The first aims to show the difference between the cautious and the assertive AV types. The second focuses on the influence of the AV penetration rate. The results of these simulations are compared to the baseline, with a penetration rate of 0%. The results of the AV analysis are given in Section III.

Method of Execution

Calibration pipeline: Based on the methods explained in previous sections, the calibration code structures are set up. The flowchart visualising the hierarchical structure of the Python codes is shown in Fig.4. Since the necessary parameters have been identified, it is necessary to extract them from the real-world road traffic dataset (the HighD dataset) and the Aimsun Next simulation model. For the case of the HighD road traffic dataset, depending on the parameters used, the data is read from the appropriate CSV file. Once files are read using *Open_files.py*, the necessary parameters are computed individually by *HighD_parameters.py*.

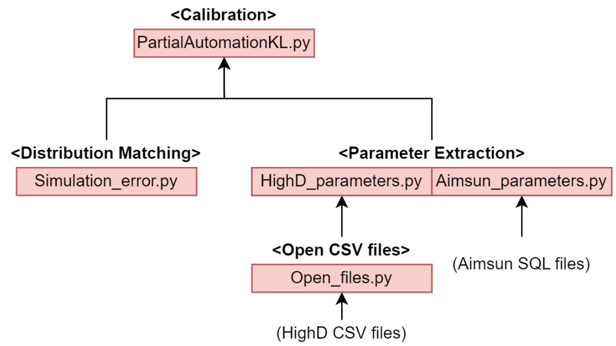


Fig. 4: A flowchart of the main Python code hierarchy.

For the case of the Aimsun Next simulation data, a *.sqlite* database file is automatically output after running the simulation, using the software's UI. After this, the necessary parameters are extracted using a number of fixed SQL queries with *Aimsun_parameters.py*. The KL divergence test and single SPSA step: random perturbation and updating of parameters, are implemented inside *Simulation_error.py*. The distribution matching file and parameter extraction files are called inside *PartialAutomationKL.py* for iterative steps to update parameters and read/write the results. Further information regarding the Python files and functions is given in *documentation.md* on GitHub.

Scripting in Aimsun Next

In order to decrease the number of manual steps in the calibration and validation process, scripting inside Aimsun Next can be used. Scripting involves using Python to perform operations with the subset of UI and kernel classes exposed by the software. Each of the objects defined in Aimsun (such as vehicle types, models, etc) is classified using an identifier. Using the defined identifier, Aimsun classes and their methods can be used to modify certain model values. It then becomes possible to modify the parameters of a specific model using internal scripts in Python and run multiple replications after making these modifications. Further information can be found at [19].

With regards to the AV analysis step, Aimsun Scripting is used to automatically modify the internal parameters of a predefined Cautious and Assertive AV model, run multiple iterations of the simulation, and save the trajectory data in separate files.

III. Results

Phase 1: Data selection & Processing and Simulation setup

Correlation between traffic flow parameters: The results of the correlation analysis are given in this section. The intensity and density are calculated with an interval of one minute and the mean car speed and truck speed are calculated each minute, to match the units. TABLE I shows the resulting coefficients for all HighD datasets. Additionally, Fig. 5 shows

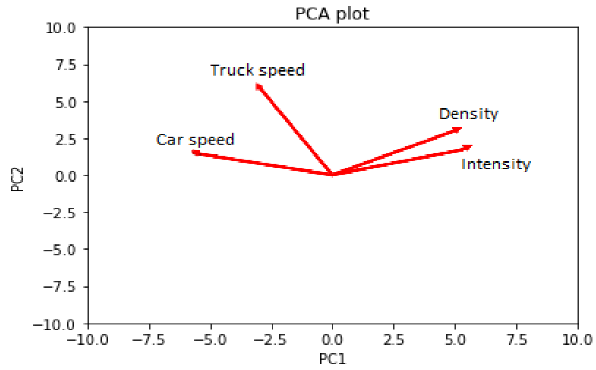


Fig. 5: PCA analysis on traffic flow parameters.

TABLE I: Correlation coefficients of traffic flow parameters.

	Car speed	Truck speed	Intensity	Density
Car speed	1	0.32	-0.47	-0.35
Truck speed	0.32	1	-0.14	-0.06
Intensity	-0.47	-0.14	1	0.49
Density	-0.35	-0.06	0.49	1

the PCA result. It can be seen that there is a negative correlation between the speeds and the intensity and density. An explanation could be that, with a decrease in traffic on the road section (low density and intensity), cars are relieved of their speed limitation as a result of other traffic and are able to drive faster. Trucks are less likely to go faster since their general speed limit is low when compared to the speed limit of cars. Features with perpendicular arrows in the PCA are considered uncorrelated, arrows in the same direction indicate a positive correlation and arrows in an opposite direction indicate a negative correlation. Therefore, the PCA provides similar results as the correlation coefficients.

The Intensity and Density parameters are positively correlated and the speeds both have a similar effect on the Intensity and Density. Based on these indications it is concluded that using both Intensity and Density is not necessary and for the remainder of this research only Density is considered.

Phase 2: Calibration & Validation

Calibration: The simulation is calibrated using the calibration pipeline discussed in Section II. The pipeline allows reading the data from the HighD dataset and the simulation results, using the KL divergence test to compute the loss and iteratively calibrating the simulation using the SPSA algorithm. Only 20 iterations are performed due to the amount of time that it takes to complete a single iteration. This is limited by the procedure of manually filling in the car-follower parameters at each iteration. The results of the baseline compared to the best-calibration loss after 20 iterations for the distributions of interest (Car speed, Truck Speed, Density, TTC, DRAC) are shown in Fig. 6. The same distributions are shown larger in Appendix A. The corresponding losses of the distributions are given in TABLE II.

The calibration simulation performs better than the uncalibrated (baseline) loss. The baseline loss was computed as the median value of ranges found in the literature[15], except for the Aimsun Next parameter for the number of cars and trucks in the simulation. These were taken from the HighD dataset and were assigned ranges. The reason why the number of cars and trucks were taken from HighD is a result of every scenario being different. Trying to calibrate the scenario to the dataset requires the use of the number of cars and trucks in these datasets. The ranges were assigned by adding and subtracting 10% from the median value. 10% was taken for ease of computation. This could be improved in future research. Although TABLE II shows improvement over the number of iterations performed, the loss is showing random behaviour over iterations. This means that the loss is not improving every iteration. Furthermore, since the loss space is non-convex, it is uncertain if the lowest loss value is found in the global optimum. After the 20 iterations, it was found that only a handful had a loss lower than the baseline result. This is a result of the random nature of the SPSA algorithm and the simulation. Results in other studies show that with an increasing number of iterations, the SPSA algorithm converges to a (local) minimum, proof of this can be seen in the study of Hirokami et al.[20]. Because only 20 iterations are performed, it is inconclusive to say that the simulation is fully calibrated. However, it can be stated that the implementation of the algorithm works, since at every iteration it randomly perturbs the car-follower parameters, and finds the lowest loss direction. For this reason, it is expected to find an optimum if more iterations are performed since it is proven that as the number of iterations goes to infinity, the SPSA converges to the global optimum [20].

TABLE II: Calibration loss and Validation loss per emergent parameter using the calibrated input parameters.

Parameter	Car Speed	Truck Speed	Density	TTC	DRAC	Sum of Losses
Baseline Loss	0.664	4.378	0.828	0.126	0.012	6.008
Calibration Loss	0.618	3.292	0.784	0.123	0.002	4.818
Validation Loss (Validation set 1)	0.800	3.546	2.696	0.165	1.157	8.364
Difference $\|\|\text{Val} - \text{Cal}\|\|$	0.182	0.254	1.912	0.042	1.155	3.546
Validation Loss (Validation set 2)	1.652	4.523	1.398	0.144	0.008	7.727
Difference $\|\|\text{Val} - \text{Cal}\|\|$	1.034	1.231	0.614	0.021	0.006	2.909

The distribution of TTC and DRAC of the calibrated simulation visually behaves similarly to the distributions from the HighD dataset. However, the distributions of the car speed, truck speed and density visually differ from the HighD dataset. From TABLE II, it is clear that the truck speed is performing the worst, which can also be seen in Fig. 6(a) and Fig. 6(b). A tail on the left side is visible for the distribution of the truck speed, which shows that trucks drive slowly on a part of the highway for the Aimsun simulation. The results shown in the graphs further emphasize the necessity of conducting additional iterations during the calibration process in future studies.

Validation: The first validation is performed in the right-heading direction, which is not used during the calibration (validation set 1). Validation set 2 is the 15$^{\text{th}}$ road section of the HighD dataset, which also consists of two lanes per driving direction. The resulting distributions are shown in Fig. 7. The loss per parameter is again shown in TABLE II. The values of

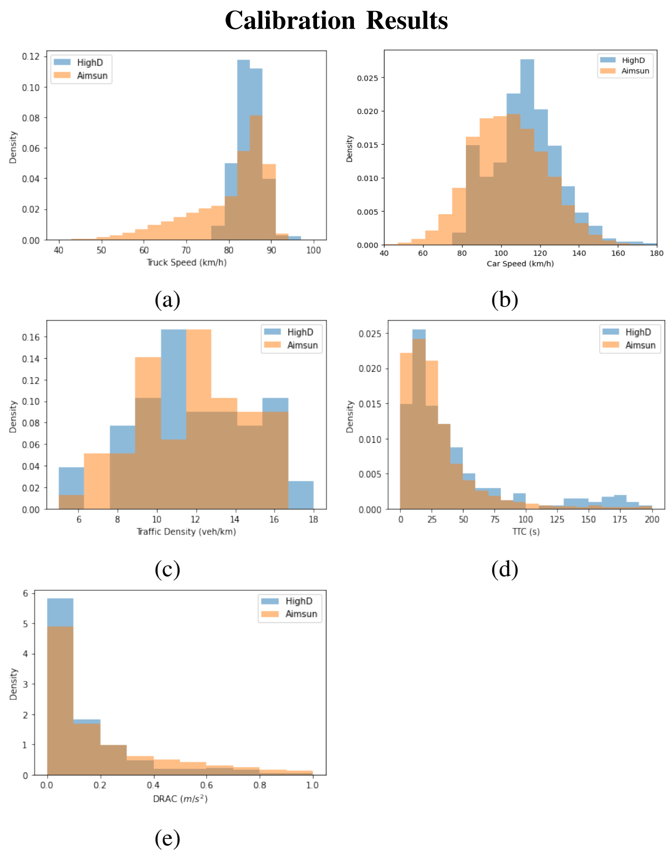


Fig. 6: Resulting distribution of the best loss acquired after calibrating the simulation. In (a) the full range of the truck speeds, the plot in (b) has the distribution for the car speeds while in plot (c) the Traffic Density is given. (d) contains the calibrated TTC value, whereas DRAC is shown in (e). In the plots the normalized distribution of Aimsun is given in Blue, while the normalized distribution of HighD is shown in Orange.

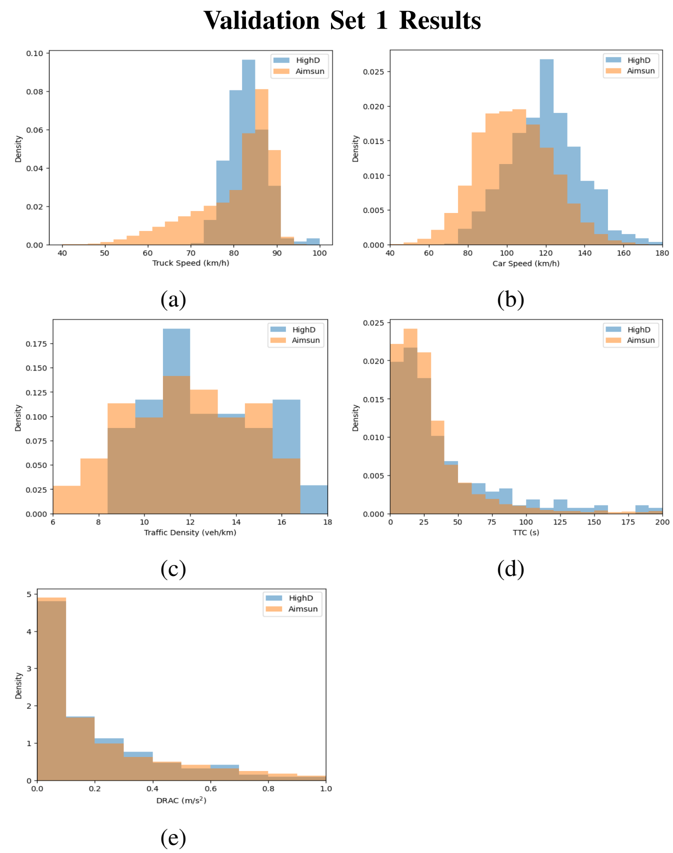


Fig. 7: Resulting loss distribution acquired in validating the simulation model compared to the validation set 1. In (a) has the full distribution for the truck speeds, where (b) has the car speed distribution, (c) has the Traffic Density and (d) the calibrated TTC value. (e) contains the distribution of DRAC. In the plots, the normalised distribution of Aimsun is given in Blue, while the normalised distribution of HighD is shown in Orange.

the KL divergence tests on the two validation sets are higher than the calibration.

Out of all validation losses in TABLE II, it is clear that the loss of TTC has increased the least overall, compared to the calibration. There has been a slight change in the density of low TTC ranges (values below 50sec) of the HighD distribution. When comparing the TTC distribution of validation set 1 (Fig. 7d) and calibration (Fig. 6d), it can be observed that these distributions exhibit similar density shapes compared to the given HighD dataset resulting in comparable losses and small deviation.

The loss of Density and DRAC increases the most. Looking at the distribution of density first (Fig. 7c), it can be seen that there are ranges in which there are no counts of HighD data. This increases the loss of the function. However, as mentioned in the calibration, the density histogram contains very few values. This might result in little overlap of distributions. DRAC, on the other hand, has a small range of values ($0 \leq DRAC \leq 1$), resulting in high-density values. As a result, the DRAC loss is sensitive to small changes when calculating the KL divergence (Eq. 6). The large difference in the first bin o DRAC may have greatly influenced the change in losses.

Comparing calibration and validation set 1, it is observed that the average truck speed was shifted to the left for the HighD dataset, meaning that on average the truck speed is lower than in the calibration. This would explain the increase in loss. The car speed distribution is similar for validation set 1, however, it shows a larger difference compared to validation set 2. This could be explained by the shift in average speeds driven.

From the two validation sets it can be seen that a generalizable optimal calibration is not possible. The calibration has a significantly lower loss than the validation sets. This is to be expected since the validation data sets are not statistically identical to the one that is used for the calibration.

Phase 3: Autonomous Vehicle analysis

Once the simulation is calibrated, a threshold for "human driving" on the road is derived and simulated. With this serving as a baseline, an analysis of the effects of varying levels of penetration of AVs into the road system can be tested and observed. As described in Section II, two types of AVs are used as comparative behaviour indicators for driving: the cautious AV and the assertive AV. Both of these car types follow the Gipps' car-following model, with varying parameter configurations. For every experiment, one vehicle type is inserted into the traffic scenario with varying penetration levels.

The first analysis considers the introduction of Assertive and Cautious AVs into the road network with $10\%, 25\%, 50\%$ and 80% penetration rates. These are then compared to the baseline (the calibrated driver model). Since no significant change from the Baseline is observed at penetration rates of 10% and below, only higher penetration rates were visualised. The results of penetration rates of 50% and 80% can be seen in Fig. 8. Fig. 12 in Appendix C shows a broader range of AV penetration rates.

The second analysis considers the introduction of Cautious and Assertive AVs into the road network in the range of $[0\%, 100\%]$ penetration rates with small increments, and compares each to the baseline. Some results can be found in Fig. 9. Fig. 13 in Appendix 13 contains more detailed results.

Discussion on the AV analysis: For both of these analyses, a focus is made on observing the behaviour of the SSMs, however, the other traffic flow parameters are also visualised for the purpose of evaluating general traffic flow trends. The first observation that can be made is regarding Fig. 8a and Fig. 8b. At lower penetration levels, the impact of the introduction of both assertive and cautious AVs is almost negligible. However, the impact of the introduction of AVs on the TTC becomes noticeable at a 50% and 80% penetration rate. Fig. 8a and Fig. 8b seem to indicate that the density of TTC values is higher than the baseline between 10 and 50 seconds. With regards to DRAC values, both Fig. 8c and Fig. 8d seemingly suggest that at higher DRAC values (0.17 to 0.2), there are fewer cases of the Cautious AV having these values than the baseline, while there are more cases of the Assertive AV having these values than the baseline. While at lower values (approximately less than 0.17), there are generally more cases of the Cautious AV rather than the baseline, and fewer cases of the Assertive AV than the baseline. This is in line with the fact that Cautious AVs have higher reaction times, and can therefore brake much faster than human drivers, which can be seen in Appendix E[21][22]. Results from figures (e) - (j) in Fig. 8 suggest that car speeds are more often slower with AVs than the baseline, however, the traffic density is also more spread out, with close to a normal distribution of traffic density at 80% AV penetration rates for Assertive AVs. This seems to indicate fewer cases of higher traffic density in comparison to the baseline and as a result a better flow of traffic when introducing Assertive AVs. However, the same results are not observed for Cautious AVs, where the density distribution is similar to the baseline at 80% AV penetration rates.

With regards to the second analysis results of Fig. 9, there is evidence that higher penetration rates of AVs (both Cautious and Assertive) result in more cases of lower TTC values in comparison to the baseline (from Fig. 9a and Fig. 9b). This is in line with the previous observation from the first

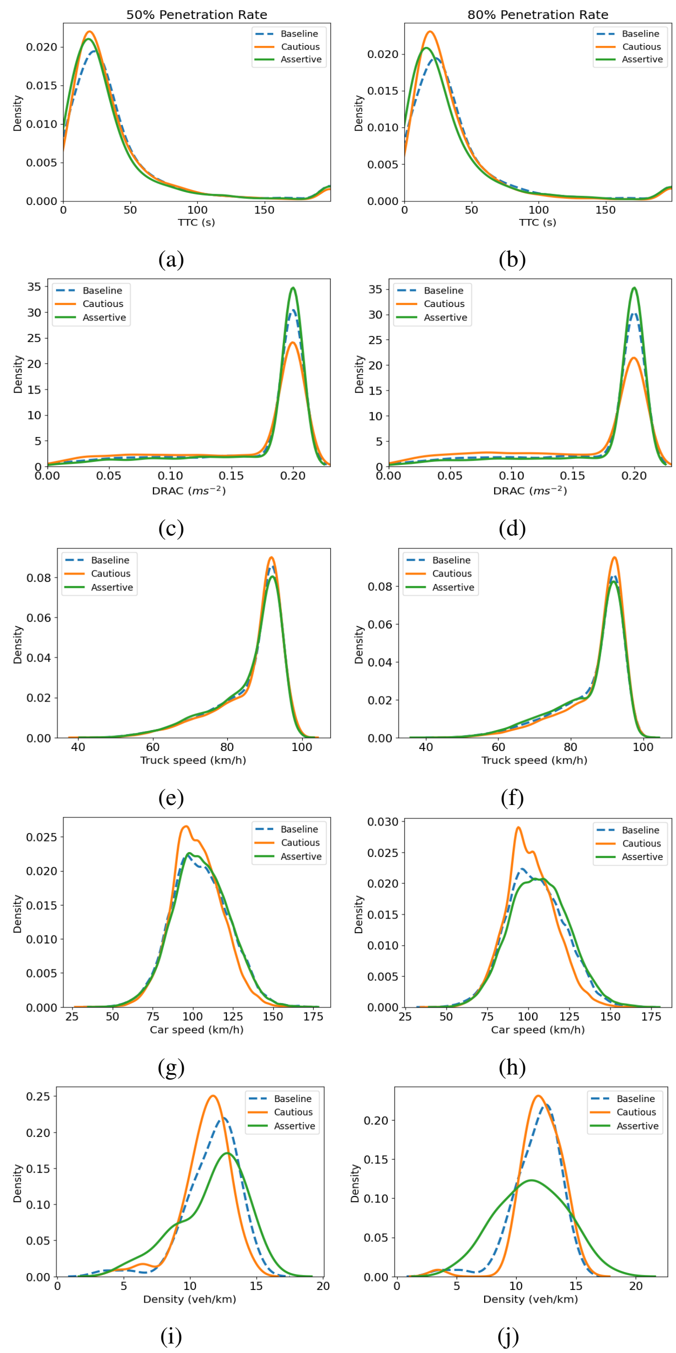


Fig. 8: Resulting TTC, DRAC, Truck speed, Car speed and traffic density values from the AV analysis at 50% and 80% penetration rates. In (a) and (b), the results can be seen for TTC, (c) and (d) are the results for DRAC, (e) and (f) are the results for the Truck speed, (g) and (h) are the results for the car speeds and in (i) and (j) the results are shown for the density. In the plots, the behaviour of the Baseline (human driver) is given in dashed blue, the cautious AV in orange and the assertive AV in green.

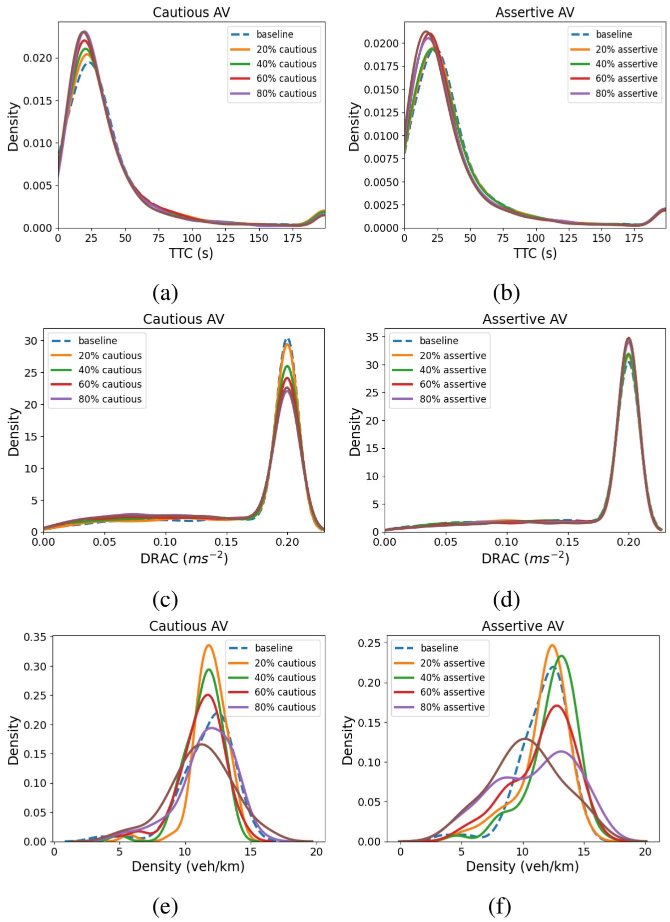


Fig. 9: Resulting TTC, DRAC, Truck speed, Car speed and traffic density values from the AV analysis for penetration rates 20, 40, 60 and 80%. In (a) & (b) the AV analysis results can be seen for the clipped ranges of TTC for Cautious and Assertive AVs, (c) & (d) the AV analysis results for the clipped ranges of DRAC for Cautious and Assertive AVs. Finally in (e) & (f) the AV analysis results can be seen for the clipped ranges of traffic density for Cautious and Assertive AVs respectively. In the plots, the baseline behaviour is given by a dashed Blue line and each of the penetration rates by a different colour.

analysis in Fig. 8. Similar trends observed in the first analysis are found in Fig. 9c & Fig. 9d, with evidence of a higher number of Cautious AV cases at lower values of DRAC than baseline cases. However, this is not as prominent for Assertive AVs. Moreover, most noticeably, there is evidence of lower cases of high traffic density at higher penetration rates of Cautious and Assertive AVs (at around 80 to 90% penetration rates, visible in Fig. D in the Appendix) in comparison to baseline values. This seems to suggest lower traffic congestion at higher penetration rates with Cautious and Assertive AVs, and generally a better flow of traffic. Taken together, these results seem to indicate that Cautious AVs are generally safer than human drivers when considering SSMs such as TTC and DRAC. Moreover, their introduction at around 90% penetration rates has positive effects on overall traffic flow.

IV. Conclusion

This study aimed to develop a system to calibrate a traffic micro-simulation from the perspectives of traffic and safety in relation to a real-world data set. This was done using Aimsun-Next software and afterwards, the impact of implementing AVs was analysed through simulation. The model was calibrated and validated using real-world traffic data, ensuring its reliability and accuracy in capturing the behaviour of vehicles in a traffic environment.

The methods presented in Section II provide a systematic approach to calibrate and validate a traffic simulation model using the HighD dataset and Aimsun Next. By matching the statistical distributions of the traffic flow parameters and SSMs, the simulation model can accurately represent real-world traffic conditions, laying the foundation for the analysis with AVs.

The calibration process of the simulation environment using the pipeline was somewhat successful in adjusting the parameters to match the HighD dataset. The loss was computed between the best simulation results and the HighD dataset, and the best-calibrated parameters were obtained after 20 iterations. The resulting distributions of interest (Car speed, Truck Speed, Density, TTC and DRAC) were shown along with the corresponding losses. However, more iterations of the calibration process need to be performed to converge to a local minimum, and achieve a close match between the simulated and real-world data.

The analysis of the calibrated parameters revealed several observations. The Truck Speed distribution showed lower speeds after the calibration and the Density distribution had a higher loss.

The difference in traffic conditions, particularly the slower truck speeds in the simulation, contributed to the higher traffic density loss. However, the overall car speed distribution appeared to match well with the HighD data. The TTC distribution had a very low loss, indicating a good fit between the simulation and the HighD dataset.

The validation process was performed by using data from two other, independent HighD data sets. The validation results showed higher losses compared to the calibration.

In the AV analysis, different penetration levels of Assertive and Cautious AVs were introduced into the road network. The effects of varying AV penetration levels were compared to the baseline calibrated driver model. The results of the AV analysis revealed interesting observations. At lower penetration levels, the introduction of both assertive and cautious AVs had minimal impact on the Time-to-Collision (TTC). However, as the penetration rate reached 50% and 80%, the impact became apparent. The density of TTC values for AVs was generally higher than the baseline between 10 and 50 seconds, with lower values exhibiting greater differences. These findings suggest that the introduction of AVs can affect TTC, particularly at higher penetration rates.

Overall, this study contributes to the understanding of traffic micro-simulation calibration, validation, and the impacts of AV implementation. In spite of shortcomings during the calibration process, the findings have implications for traffic and safety management and the design of future transportation systems.

V. Recommendations

Sensitivity analysis: During the project, the use of a sensitivity analysis has been investigated to gauge the impact of the car-follower parameters in Aimsun on traffic and safety parameters. This analysis could be used to reduce the number of tunable car-follower parameters and set the weights for the KL divergence loss summation. The analysis was not yet performed fully, but the method could be very useful in future research on the subject. Hence, the methodology of the analysis is shown in Appendix F.

Aimsun API: During the execution of the project, scripting inside Aimsun was used to more efficiently calibrate, validate and perform the AV analysis. On top of that, Aimsun Next has an API which allows an external application to communicate with the simulation. This makes it possible to automate the process and run multiple simulations uninterrupted. This is useful for the calibration and for the sensitivity analysis since both require multiple iterations. More information about the Aimsun API and its application in this research can be seen in Appendix G.

Consider other traffic scenarios: Future research should encompass a broader range of traffic scenarios to account for various real-world conditions and situations. For instance, urban scenarios, construction zones, different weather conditions and different countries could be investigated. While the current study focused on a specific dataset and calibration process, expanding the analysis to include different traffic scenarios will enhance the applicability and generalisability of the findings.

Calibration algorithm improvement: The calibration algorithm can be further optimised in a number of ways. For instance, momentum can be applied to the update rule in the SPSA optimisation. This would help to avoid local minima and converge to a lower minimum. Furthermore, the impact of the random perturbation size could be investigated and optimised further. Lastly, the weights in the loss function (Equation 7) can be changed to further penalise certain parameters and improve the optimisation.

Expanded AV analysis: To gain a comprehensive understanding of AV impacts, future research should expand the analysis to include additional criteria and considerations. This includes assessing traffic flow efficiency, energy consumption and emissions, and pedestrian and cyclist safety. By examining these factors, researchers can provide a more thorough assessment of the effects of AV implementation. These insights will inform decision-making, support policy development, and guide the design of future transportation systems for optimal safety, efficiency, and sustainability in the age of autonomous vehicles.

References

[1] W. H. Organization, *Global status report on road safety 2018*. World Health Organization, 2018.

[2] D. Parekh, N. Poddar, A. Rajpurkar, *et al.*, "A review on autonomous vehicles: Progress, methods and challenges," *Electronics*, vol. 11, no. 14, p. 2162, 2022. DOI: 10.3390/electronics11142162.

[3] T. Mishra, "Human Factors Causing Accidents," no. 12, p. 1, Sep. 2022. [Online]. Available: https://www.safeopedia.com/definition/687/human-factors-causing-accidents.

[4] H. Stout, *What Happens When Self-Driving Cars Crash? The Legal Ramifications of Automation*, Oct. 2022. [Online]. Available: https://www.entrepreneur.com/living/what-happens-when-self-driving-cars-crash-the-rise-of/436942.

[5] *Who's At Fault in a Self-Driving Car Accident?* Sep. 2021. [Online]. Available: https://www.dlawgroup.com/personal-injury/who-at-fault-self-driving-car-accident/.

[6] W. Young, A. Sobhani, M. Lenné, and M. Sarvi, "Simulation of safety: A review of the state of the art in road safety simulation modelling," *Accident; analysis and prevention*, vol. 66C, pp. 89–103, Jan. 2014. DOI: 10.1016/j.aap.2014.01.008.

[7] Aimsun, *Aimsun: simulation and AI for intelligent mobility*, Nov. 2022. [Online]. Available: https://www.aimsun.com/.

[8] Aimsun, *Modeling Vehicle Movement - Aimsun Next Users Manual*. [Online]. Available: https://docs.aimsun.com/next/22.0.1/UsersManual/MicrosimulationModellingVehicleMovement.html#car_following_model.

[9] D. Pitt Van Aerde Zhang, *Microscopic Traffic Simulation*, 2019. [Online]. Available: https://slideplayer.com/slide/15784796/.

[10] *The highD Dataset*. [Online]. Available: https://www.highd-dataset.com/.

[11] F. Hauer, I. Gerostathopoulos, T. Schmidt, and A. Pretschner, "Clustering traffic scenarios using mental models as little as possible," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1007–1012. DOI: 10.1109/IV47402.2020.9304636.

[12] S. Hoogendoorn, *Traffic flow theory and simulation, vk4821*. TU Delft, Delft, 2010, vol. 152.

[13] H. Yang, "Simulation-based evaluation of traffic safety performance using surrogate safety measures," 2012.

[14] K. Adjenughwure, P. Huertas Leyva, F. Prinz, A. Tejada, and X. Wang, "Description metrics for traffic interactions," *proactive SAFEty systems and tools for a constantly UPgrading road environment*, 2021.

[15] M. Figueiredo, Á. Seco, and A. B. Silva, "Calibration of microsimulation models – the effect of calibration parameters errors in the models' performance," *Transportation Research Procedia*, vol. 3, pp. 962–971, 2014, 17th Meeting of the EURO Working Group on Transportation, EWGT2014, 2-4 July 2014, Sevilla, Spain, ISSN: 2352-1465. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352146514002397.

[16] D. Sha, J. Gao, D. Yang, F. Zuo, and K. Ozbay, "Calibrating stochastic traffic simulation models for safety

and operational measures based on vehicle conflict distributions obtained from aerial and traffic camera videos," *Accident Analysis & Prevention*, vol. 179, p. 106 878, 2023, ISSN: 0001-4575. DOI: https://doi.org/10.1016/j.aap.2022.106878. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000145752200313X.

[17] SAE, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," SAE Mobilus, Tech. Rep. J3016GroundVehicleStandard, Jun. 2018. [Online]. Available: https://www.sae.org/standards/content/j3016_201806/.

[18] A. B. Ims and H. Pedersen, "Simulation of automated vehicles in aimsun," M.S. thesis, Norwegian University of Science and Technology, 2021.

[19] *Aimsun Next Scripting*. [Online]. Available: https://docs.aimsun.com/next/22.0.2/UsersManual/ScriptIntro.html.

[20] T. Hirokami, Y. Maeda, and H. Tsukada, "Parameter estimation using simultaneous perturbation stochastic approximation," *Electrical Engineering in Japan*, vol. 154, no. 2, pp. 30–39, 2006. DOI: https://doi.org/10.1002/eej.20239. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/eej.20239. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/eej.20239.

[21] J. Khoury, K. Amine, and R. Saad, "An Initial Investigation of the Effects of a Fully Automated Vehicle Fleet on Geometric Design," *Journal of Advanced Transportation*, vol. 2019, pp. 1–10, May 2019. DOI: 10.1155/2019/6126408. [Online]. Available: https://doi.org/10.1155/2019/6126408.

[22] V. Dixit, S. Chand, and D. Nair, "Autonomous Vehicles: Disengagements, Accidents and Reaction Times," *PLOS ONE*, vol. 11, no. 12, e0168054, Dec. 2016. DOI: 10.1371/journal.pone.0168054. [Online]. Available: https://doi.org/10.1371/journal.pone.0168054.

[23] A. Saltelli, M. Ratto, T. Andres, *et al.*, *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

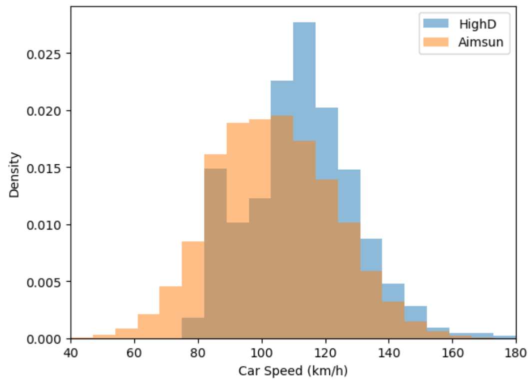
[24] R. Ghanem, D. Higdon, H. Owhadi, *et al.*, *Handbook of uncertainty quantification*. Springer, 2017, vol. 6.

[25] *API Architecture*. [Online]. Available: https://docs.aimsun.com/next/22.0.1/UsersManual/ApiArchitecture.html.

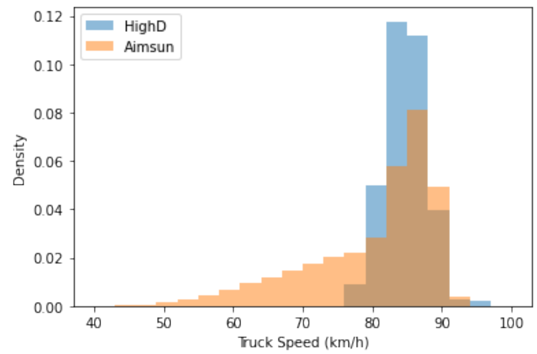
Appendix

A. Calibration results

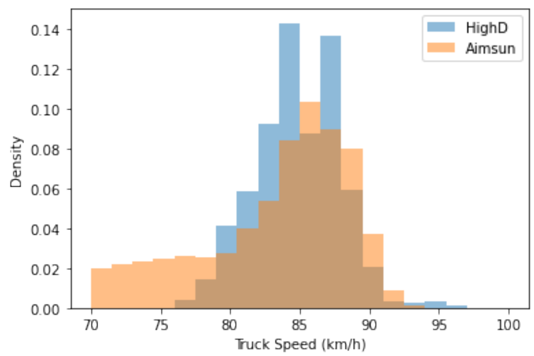
In this section, all the figures are given on a larger scale for easier viewing. Furthermore, some additional figures will be placed here.



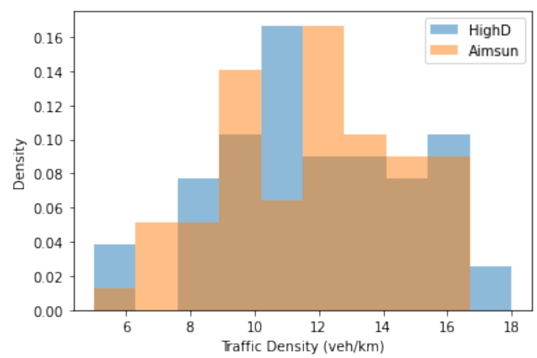
(a) Car speed distribution by using calibrated simulation values



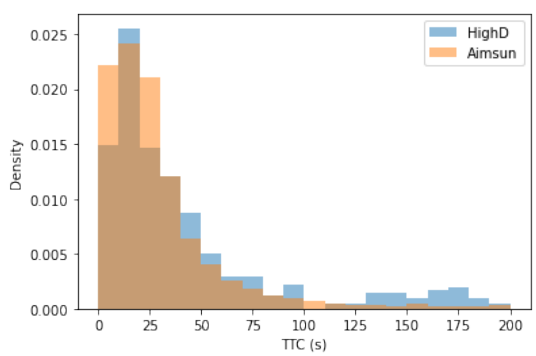
(b) Truck speed distribution by using calibrated simulation values, including the whole tail



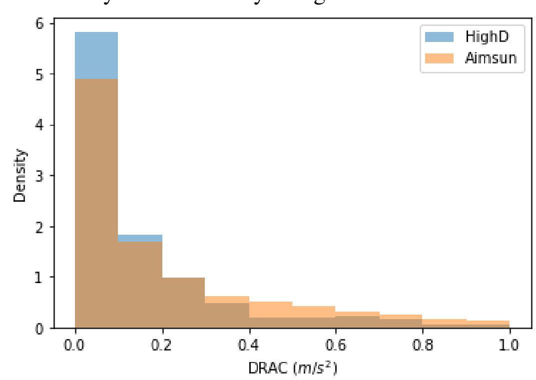
(c) Truck speed distribution by using calibrated simulation values, zoomed in



(d) Traffic density distribution by using calibrated simulation values



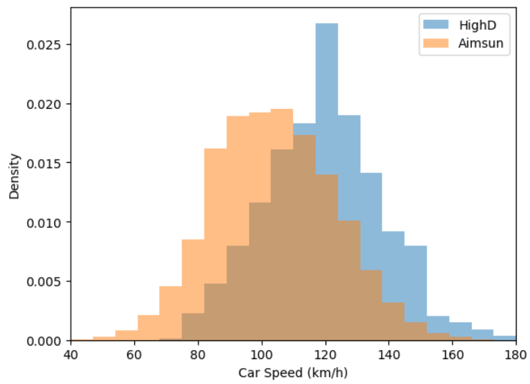
(e) TTC distribution by using calibrated simulation values



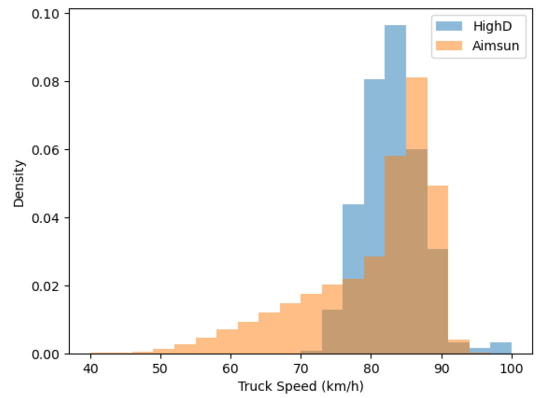
(f) DRAC distribution by using calibrated simulation values

B. Validation results

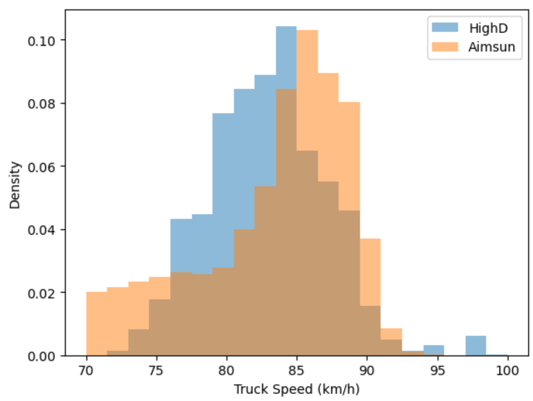
The validation had the following results



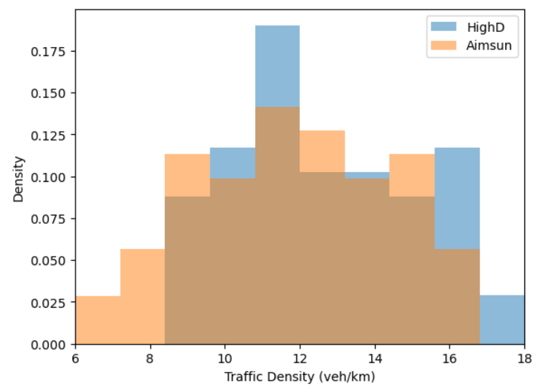
(a) Car speed distribution of the validation



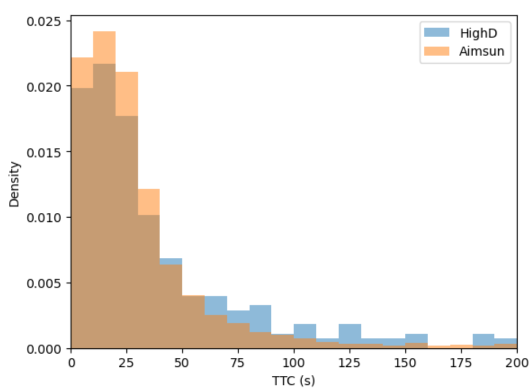
(b) Truck speed distribution of the Validation including the tail



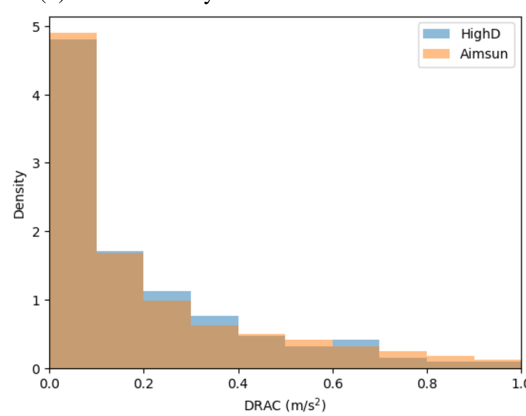
(c) Truck speed distribution of the Validation, zoomed in



(d) Traffic density distribution of the Validation



(e) TTC distribution of Validation



(f) DRAC distribution of the validation

C. First AV analysis results

The first analysis of the AV has the following results:

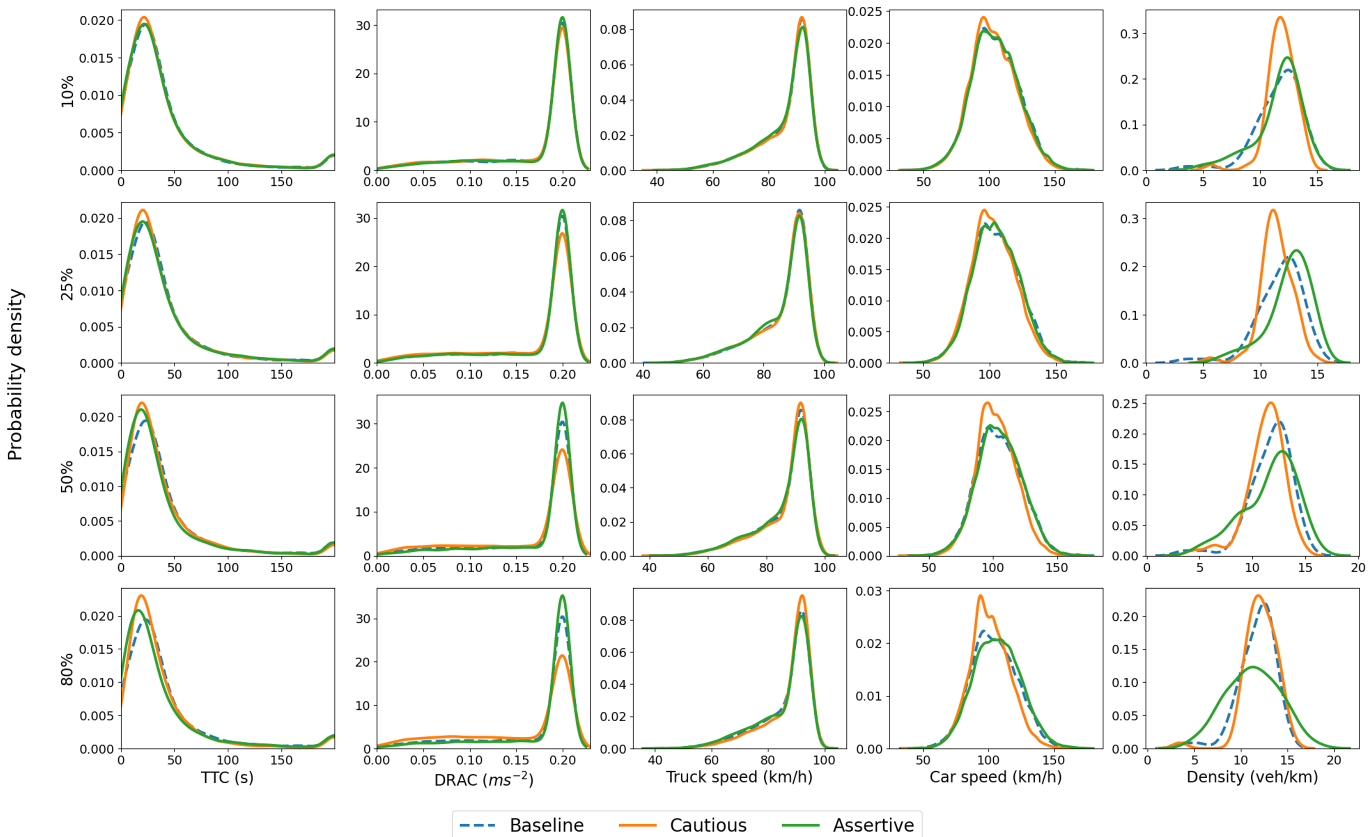


Fig. 12: AV full analysis results with 10, 25, 50 and 80% penetration rate for TTC, DRAC, Truck speed, Car speed, and traffic Density.

D. *Second AV analysis results*

The second analysis of the AV has the following results:

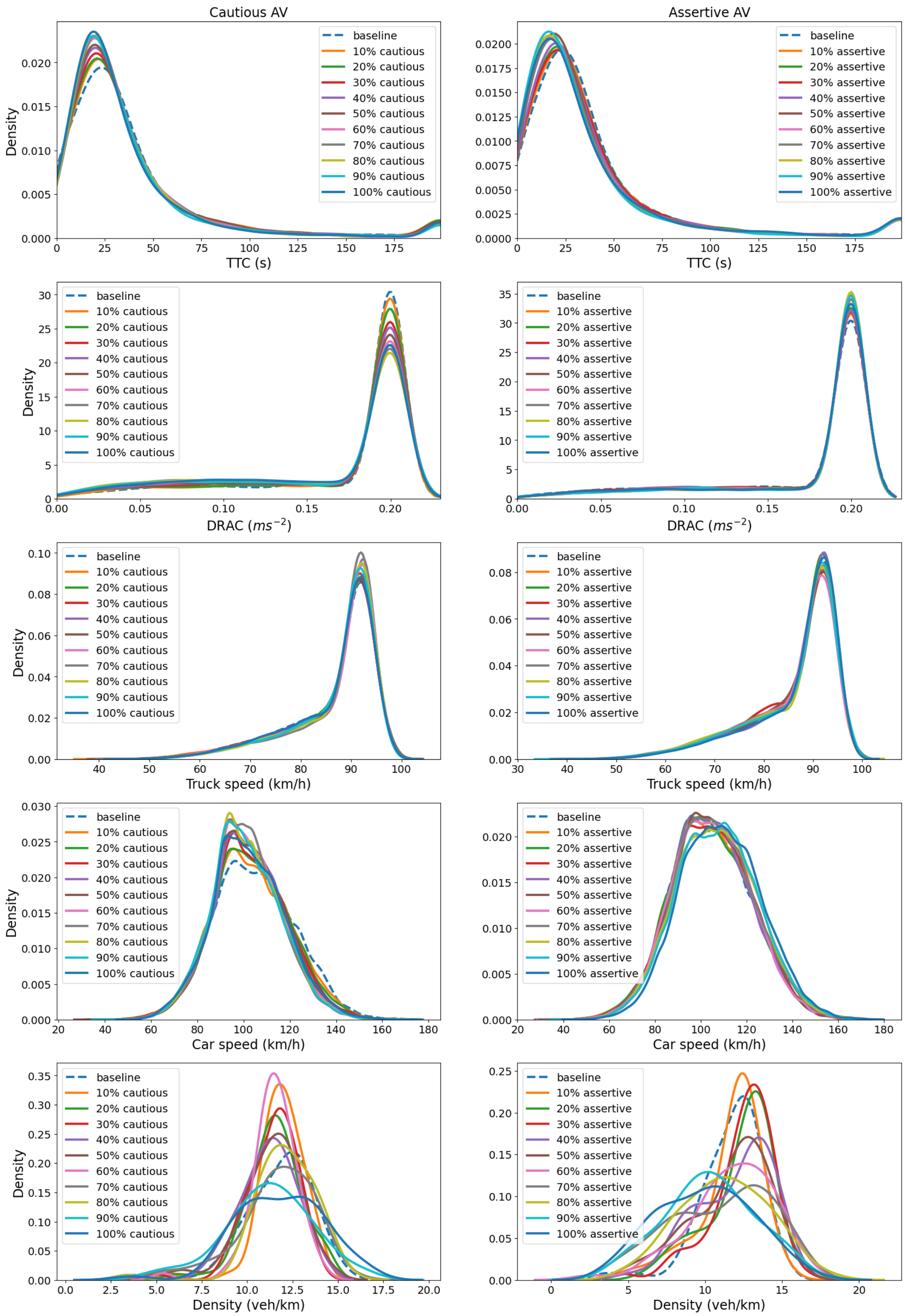


Fig. 13: AV full analysis results with 10, 20, 30, ... 100% penetration rate for TTC, DRAC, Truck speed, Car speed, and traffic Density.

E. Car-follower parameters

TABLE III: Car-follower parameters.

Input Parameters		Cautious AV		Assertive AV	
		Original	Updated	Original	Updated
Max Desired Speed	[km/h]	110.0	133.51	110.0	133.51
Speed Limit Acceptance	[-]	1.0	1.1	1.0	1.1
Max Give Way Time	[s]	12.0	12.0	8.0	8.0
Clearance	[m]	1.0	1.0	1.0	1.0
Reaction Time	[s]	0.1	0.1	0.1	0.1
Reaction Time at Stop	[s]	0.1	0.1	0.1	0.1
Reaction Time Traffic Light	[s]	0.1	0.1	0.1	0.1
Max Acceleration	[m/s^2]	3.0	2.9	3.0	2.9
Max Deceleration	[m/s^2]	6.0	6.0	6.0	6.0
Normal Deceleration	[m/s^2]	2.0	2.0	2.0	2.0
Safety Margin Factor	[-]	2.0	2.0	1.0	1.0
Sensitivity Factor	[-]	1.5	1.0	1.0	0.47
Overtake Speed Threshold	[%]	80.0	80.0	90.0	90.0
Gap	[s]	2.0	1.8	1.0	1.8
Look Ahead Distance Factor	[s]	1.5	1.5	1.25	1.25
Aggressiveness Level	[-]	0.0	0.0	0.0	0.0

Note that the values given in the table are changes with respect to the human driver values in the paper and our calibration, in the original and updated columns, respectively.

F. Sensitivity Analysis

The car-follower model includes many tunable parameters, which all have to be optimised in the calibration procedure. The more parameters to be optimised, the harder it becomes to find a low local minimum in the loss space using the SPSA algorithm. Therefore, a sensitivity analysis can be performed to investigate to what extent changes in the parameters of the car-follower model affect the simulation results, and to potentially disregard one or more parameters entirely.

Previous research [15] shows the usual ranges of the parameters. In TABLE IV, some of these parameters and their ranges are shown. The Aimsun model is run with each combination of input parameters in the corresponding ranges, varied by the step size as denoted in the table. Then, collect the speed (car and truck), Density, TTC and DRAC distributions for each set of input parameters. The deviations in the distributions with respect to the changes in each of the parameters are a measure of sensitivity, which can be used to disregard certain parameters. More details on the procedure can be found in [23] and [24].

TABLE IV: Input Parameters and Parameter Ranges.

	Car's Range	**Truck's Range**	**Step size**
Max desired speed (km/h)	50-200	40-120	1
Max Acceleration (m/s^2)	2.6-3.4	0.6-1.8	0.01
Normal Deceleration (m/s^2)	3.5-4.5	2.5-4.8	0.01
Reaction Time (s)	0.5-1.5	0.5-3	0.01
Vehicle Demand	450-550	80-100	1
Gap (m)	0-5	0-5	0.01
Sensitivity Factor	0-1	0-1	0.01

G. Aimsun API

Aimsun Next also has an API, which comes in the form of an Advanced Telematic Application to be tested using the model as an external application that can communicate with the simulation. Using Aimsun Next API functions, data from the simulated network is transferred to the external application. The application can apply its own algorithms to evaluate the situation in the simulation and respond with appropriate dynamic actions to be implemented in the simulation. Hence, this format is more convenient if in-simulation dynamic modifications are desired. Further information can be found in the Aimsun documentation [25].

On a micro-simulation level, it is possible to use the Aimsun Next API or Scripts to automate a number of steps of the calibration and analysis process and produce more concise results. During the process of this research, most of these steps were completed manually. However, it is entirely possible to automate them. The steps considered for automation were the calibration and validation processes (section II) and the sensitivity analysis. The motivation for the need for such automation is that both of these steps require the Aimsun simulation to run over multiple iterations while continuously modifying the internal parameters of the model initialized using specific values. Hence, during a manual calibration, a bottleneck is present around manually modifying these internal parameters and running the simulation to completion, which can be alleviated using the Aimsun API or Scripting.

In the case of the calibration step, the model calibration algorithm takes as input the values of the traffic parameters, as well as values of the parameters from the HighD dataset, and returns values for each of the parameters that are used to modify the Aimsun parameters. Then, Aimsun Next needs to internally modify these parameters, run the simulation, and return new trajectories that are then through the calibration pipeline used as input to the model calibration algorithm. This is repeated until a desired error or stopping point is reached, as shown in Fig. 14.

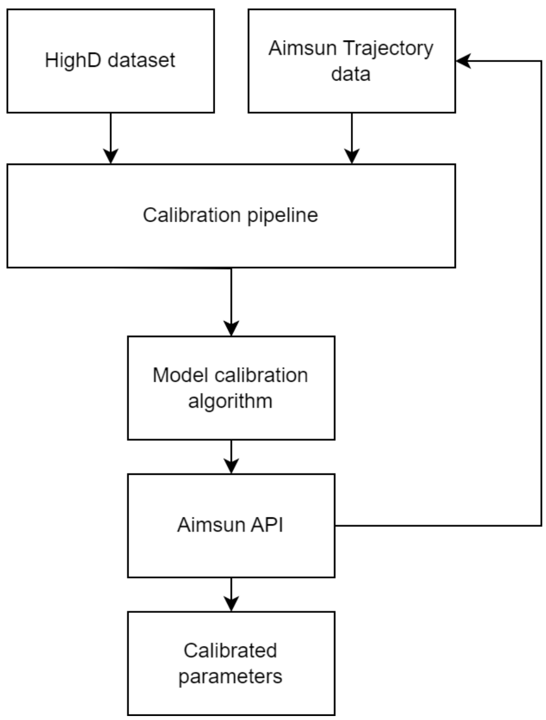


Fig. 14: The proposed structure for automatic calibration using the API.

In the case of the Sensitivity Analysis step, the model needs to internally randomly modify (within a pre-specified range of values) a set of parameters, and for each modification output the resulting trajectory data. This information is then compiled and used separately.

H. Individual Contributions to the Project

Diederik: My contributions in the first quartile were mostly setup tasks. This entails literature research, planning, and creating templates for writing. This resulted in a more administrative workload. Very little engineering came into play as a result, but this is to be expected at the beginning of the project. I did investigate the possibility of altering the model, but this bared not fruit since it required a license which we did not have. Finding out that this did not work took over a week which lost a bit of progress. But at least there was an attempt. The last couple of weeks revolved again around writing the midterm report. This was later altered by the group to better fit their needs.

In the second quartile, I took over the team leader role. This would include scheduling meetings and mostly keeping things running at an optimal level. I still had some leftover tasks with planning but these were relatively minor in this quartile. I met up with the writing coach and ensured that our writing was up to par.

I have spent the most time writing code this quartile. There were many small tasks regarding the code that had to be implemented since there was a change in the calibration method. The KL-divergence had to be implemented and the SPSA. Which I did with Ryo and Teun.

Furthermore, a lot of time was spent on doing the actual calibration. There were 4 separate calibration runs of at least 10 iterations. Which were performed by Ryo and I. Sadly every time we found a small mistake, which cost some more time on fixing the mistake. Luckily since we wrote the code ourselves it was fairly easy to fix. I have spent a lot of time with Ryo calibrating the simulation since a single semi-automation run takes roughly 7.5-15 minutes The biggest problem lay with changing parameters in Aimsun. With some investigation, it is possible to change parameters automatically with our code using the API, but this was again not possible since it is locked behind a required license, it would be possible to change it within the environment but then we had to alter our code such that it works in the environment and that would have taken to much time, this was mostly investigated by Hicham.

After the calibration, I worked on the validation with Ryo. The validation of the model. Again this took some time since some functions had to be altered but this took significantly less time than working on the calibration since the calibrated model was already implemented and the file was created. This took about 1.5-2 weeks. The reason why this took longer than perhaps expected was a result of some functions had to be changed and a new notebook had to be created for creating the figures seen in the report.

The last two weeks mostly consisted of writing the report again. Meeting with the writing coach and reorganizing the report. Where again I made a setup and this was edited by the rest of the group.

I do recon that with the first quartile I had a good balance between tasks where the first part consisted of administrative works and the second of ”hands-on” implementation.

Hicham: With regards to the first quartile, the main tasks which I handled consisted in reading the relevant literature related to said project, getting familiar with the Aimsun Next simulation software (by looking at video tutorials), setting up the data processing pipeline, data visualisation, improving the github workflow, and investigating a number of tools to either apply for the calibration, or extract data from the Aimsun trajectory data (SSAM). On top of this, I investigated an internal calibration method with Aimsun Next, and made contributions to the report.

In the second quartile, my main contributions were related the to usage of the Aimsun Next API, Scripting methods, the Autonomous Vehicle analysis, as well as the Sensitivity Analysis of the input parameters to Aimsun Next simulation.

My main contributions in the first 2 weeks consisted in investigating and matching the computation of the density parameters between the simulation and the pipeline computation such that it could be used in the model, and the investigation of the Aimsun Next API together with Vincent and Diederik, as well as investigating the AV implementation and getting an initial method that could be used to implement it. The first and last tasks were completed successfully, however after arriving at a standstill with the implementation of the 2nd one (due to problems with using external libraries locally in Aimsun), I continued investigating the problems allowing for the possibility of automation of the process later on. In the next 2 weeks, I worked on the sensitivity analysis step, together with Vincent and Xingyi, to figure out which parameters could be left out of the calibration step. Simultaneously, I worked to fix the problems related to using the Aimsun API for automation of the calibration process and others. Results regarding the Aimsun API were that the the official documentation was incorrect regarding Python versioning, and an older version of Python would be required to run the simulation. On the sensitivity analysis side, an initial analysis could be completed, and some results were extracted, however these initial results were not conclusive enough to be generalizable (as doing the sensitivity analysis even once was high in time overhead). Hence in order to complete the Sensitivity Analysis, the API would have to be completed. On top of this, I worked on the AV analysis. Hence, the weeks that followed involved work in the API and the AV analysis (together with Teun). Since the calibration process could be completed manually, it was decided that a focus on the next steps was preferable (namely the AV analysis). In order to still use some form of automation, I used Aimsun Scripting to automatically run multiple replications, modify internal parameters, and save simulation files. These were then used to visualise the results of the AV analysis. I also designed code to be used for automated Sensitivity Analysis for next iterations of the project that could not be used in this case due to time constraints. I then wrote the resulting conclusions and observations in the report regarding the AV analysis. Moreover, I was in charge of structuring the github repository. In summary, my tasks in the second quartile were much more related to using the Aimsun API and Scripting methods, as well as data visualisation work.

Ryo: During the first quarter, my contribution to this project fell into two main categories: further research on traffic simulation and building the pipeline. As part of further investigations, a literature search was carried out to decide on the data set and the calibration methods. In addition, I performed a parameter analysis to better understand the relationship between them. When setting up the pipeline, I created a parameter extraction file for both the data and implemented the initial calibration assessment (KS test) so that the mean velocity could be calibrated. In addition, I investigated how the Aimsun software implements the driver behaviour model, so we could implement a different model as that was one of the problems. I also had some administrative duties like creating/updating the Gantt chart we used to track our schedule and creating the GitHub repository.

In the second quarter, I invested most of my time in calibration. This included the correlation analysis between the parameters, the implementation of the KL divergence/SPSA and the actual calibration.

In the first two weeks, the correlation analysis was performed with Vincent to analyze the relationship between the traffic parameters. As this only applied to the dataset we decided on, further analysis was performed on all datasets with similar conditions to draw conclusions. In addition, further research was conducted to examine what type of distribution is used to represent each parameter. We tried to implement the KS test for all parameters, which resulted in an error because the distribution cannot be fitted to a distribution. Since this failed, Teun and I looked for other ways to implement the scoring method and calibration. In the coming weeks, the alternative method (KL divergence test and SPSA) was implemented in collaboration with Diederik and Teun. However, when the scoring was conducted, the loss due to the outliers was high. We have reviewed the literature on how to remove these outliers or transform data that benefit our methods. In the end, the removal of the outlier was selected and implemented. Additionally, we discussed how to perform the validation step so that we can test the generality of the calibration. In the next week, the device check was carried out and the calibration was carried out. However, since the calibration did not work as planned, a weight loss was applied. Based on the pipeline we have set up, calibration and validation have been done over the past few weeks. I have included the calibration and validation results in the report and the methods of these sections have been added and revised.

Although I mainly focused on the pipeline and calibration for bot quartiles, I was able to work a bit on the administrative tasks as well.

Teun: This section outlines my contributions to the project. My main contributions can be grouped as follows: I have been the group leader in quartile 3, which means that I was in charge of the general overview and flow of the project, I contributed to the data selection and to the method and framework of the data preprocessing pipeline. I was a main contributor to developing and executing the method for a scoring metric for the calibration. I also found the SPSA method and strongly recommended using it and was a main contributor to the implementation. Generally, I was very active during meetings and tried to make sure that there was a clear path to the finished product and that what we were doing made sense. Also after the job of team leader was taken over by Diederik.

At the start of the project, the project description and aims were presented to the group. Several papers were handed to us, which we started reading. I also took charge of making the first meeting requests for the group and teachers. I made an agenda for our first team meeting and made recurring meeting requests for the entire quartile. I was chosen as the team leader during the first team meeting. My next main objective was to investigate the datasets and find the best one to use. I took charge of organising many meetings with group members to discuss subjects like these. After that, I wrote out a section on the used dataset for the midterm report. I also set up the framework for the project proposal presentation (an extra opportunity provided by Arturo) and made the introduction and calibration sections to the project proposal. Next, I investigated how to calibrate the simulation, and extracted the relevant information from the parameters. I also set up the framework for the data processing pipeline. Then, I looked at existing literature to investigate how other researchers previously calibrated their simulations. I also wrote a Python notebook to process the SQL data from Aimsun. In the last part of the first quartile, I worked on a scoring program in Python to assess the accuracy of the simulation. Next, I worked on parts of the midterm presentation and report.

Starting in quartile 4, I set up the program to extract TTC and DRAC from Aimsun. Then, I was involved in setting up the scoring method, which later turned into the loss function. Later, I found the SPSA algorithm, which I strongly recommended we'd use and I was then involved in the implementation of this algorithm. When this was set up, I was involved in calibrating the simulation and I made the SPSA part for a presentation for Arturo. Toward the end of the project, I was involved in setting up the AV in Aimsun and analysing the impact of this implementation.

For the final report, I wrote almost all of the methods of phase 2 and 3. I was also made responsible for the entire structure and styling of the report, together with vincent.

Vincent: In the first part of the project I did similar work as the other team members, this included reading literature and Aimsun tutorials. I also investigated what parameters we should eventually use. For the proposal, I wrote a part about why we use Aimsun Next and the general idea of implementing an autonomous vehicle in the simulation. Afterwards, I worked on the parameters, car-follower and geometry sections of the project proposal. Finally, I set up a highway scenario in Aimsun. I worked on data extraction from Aimsun, the speed, intensity and density were obtained but the TTC and DRAC were deemed more difficult. Afterwards, I updated the Aimsun model to the location and direction which we want to use and changed the width and length of the cars. Hicham and I started looking into the Aimsun tool which allows us to import data. I also extracted the exact data we need for scoring from Aimsun. Finally, I wrote a part about simulation setup in the midterm report. I worked on the calibration of the speed distributions and wrote about it in the midterm report, besides that some other report tasks were done.

For the second quartile, I worked with Ryo to compare the KS test with the Aimsun scoring. Afterwards, we worked on finding the correlation between the traffic flow parameters using correlation coefficients and PCA, the week after this was expanded to feature more data and based on this the intensity parameter was dropped. Next, I worked on the start of the Aimsun API and scripting to try and automate the processes which take a lot of manual steps. I made a beginning to the sensitivity analysis based on literature which was executed by Hicham and Xingyi. Next, the update-presentation of our process was presented to our project manager, based on this feedback we continues with the final steps of the project. In the final stages of the project, I spend all of my time on the report. This consisted of writing a new section, leaving comments and improving the existing sections. At every meeting, there would be feedback on the report which would mean a lot needed to be changed. Together with Teun, we were responsible for the final report. This includes restructuring the report, rewriting sections and finalizing everything.

When comparing my contributions of the first quartile to the contributions of the second quartile it becomes clear that in the first quartile, I was the main group member who worked with Aimsun, while in the second quartile, other group members took this role over for the calibration, validation and the AV analysis. In the second quartile, I had fewer crucial tasks and instead focussed on other tasks such as reports and API. During the project, I was quite present during the meeting, said what I wanted to say and discuss with my team members. Additionally, together with Xingyi and Teun, I volunteered to be chairman during the meetings.

Xingyi: In the first part of the project I did the same tasks as my teammates, which are reading some of the literature related to this project and doing the Aimsun Next tutorials. Then, I discussed our group agreement and found out which traffic datasets we will use and presented the advantages and disadvantages of these datasets in the project proposal and then made the final choice. I also started to find out the intensity distribution based on HighD dataset with different circumstances such as different directions and lanes. Next, I calculated the intensity distribution and plot graphs. I also checked what different images exist and which tracks are occupied by calculating each locations' trajectory and finally decided to use location 2 first. Because we made a little change for our final goal, I also read some extra literature to find out what we will do for autonomous vehicle's part. Afterwards, I gave the data to a python file to use distfit libaray to find out which will fit the data to a distribution for speed, DRAC, density and intensity parameters. Then, I did the data extraction HighD dataset for one direction and 15 seconds interval based on all of the locations. I mainly focused on reading the midterm report and give the comments. I also worked with Vincent and Hicham to visualize the distribution of Aimsun and highD and print each result.

For the second quartile, I checked the density computation and figured out the correct one with Hicham. Then, I did the literature check to find out what parameters to change in the Gipps model and IDM model, and check the calibration function such as MSE, MSN and give my own opinion. Then, I together with Hicham and Vincent did the sensitivity analysis for speed, density, TTC and DRAC. Since the API haven't been finished, we together explore other methods such as absolute error and Pearson's correlation coefficient to gifure out which parameter influences the simulation result most. Then, I implemented ruleset and tested the AV model with Hicham, and created an assertive/cautious AV. Finally, I wrote some part of the final report which are AV model and sensitivity analysis. Then, I mainly focused on reading the whole report and give the comments.

Comparing my contribution in both quartile, we can see my tasks in the first quartile were rather haphazard and in the second quartile were more focused on the sensitivity analysis and AV model, and produced some outcomes. What's more, together with Vincent and Teun, I volunteered to be chairman during the meetings. Looking back to the whole progress, I am delighted to have completed this project with my team.