Technische Universiteit Delft - Porsche AG Weissach

Quantifying Take-over Quality

A Novel Approach Utilizing a Scenario-Specific Optimized Reference Trajectory

E. Loosveld

TU Delft

PORSCHE

Quantifying Take-over Quality

A Novel Approach Utilizing a Scenario-Specific Optimized Reference Trajectory

by

E. Loosveld

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on June 14th, 2019

Student number: 4115708
Project duration: October 1^{st}, 2018 – May 31^{st}, 2019
Thesis committee: Dr. Ir. R. Happee, *TU Delft*
Dr. Ir. J. de Winter, *TU Delft*
Ir. F. Doubek, *Porsche AG*
Ir. J. Stapel, *TU Delft*

This thesis is confidential and cannot be made public until 06-2024

TU Delft PORSCHE

Acknowledgements

I would like to take this opportunity to sincerely thank my TUDelft supervisors, Riender Happee, and Joost de Winter. Your feedback has always been extremely thorough, and has often challenged me to explore new perspectives. Of even greater importance during this project, has been the day-to-day mentorship I received from Fabian Doubek. Your light-hearted yet constructive guidance has been of immense value to me during my 6-month period at Porsche, and I would like to take this opportunity to show my deepest appreciation for all your support, both on a professional and a personal level.

I would also like to thank the VFP2 driving simulator staff - Ingo Krems, Claus-Heiko Winkler, Andreas Ronellenfitsch, and Niklas Bohne, for their excellent support throughout the entire project. Your expertise has been essential, and it has been a pleasure working with all of you. I would also like to show my gratitude to Andrian Riedel and Dennis Bogen, for all their support regarding the driving simulator HMI implementations. Niko Pärsch, thank you for always taking the time to answer my questions, and for cycling up a mountain just to drop off a usb stick. Memorable indeed. Additionally, I would like to thank Kai Homeier and Kathrin Symkenberg, from the Volkswagen Group Research Facility, for allowing me to use their state-of-the-art trajectory planner for the purpose of this thesis. Your support and knowledge has been of vital importance to the project.

On a more personal note, I would like to thank both my parents for their invaluable personal support and patience throughout my entire studies. Especially, for having allowed me the freedom to develop new passions and to travel the world. Last, but not least, I would like to thank my girlfriend, Steph. You have been there through thick and thin, and you have supported me throughout.

E. Loosveld
The Hague, May 2019

Abstract

Due to recent advancements in automated driving systems, drivers are able to withdraw themselves from the control loop of the vehicle in certain driving situations. However, system limits occur relatively frequently, which results in the driver having to take back control. It is argued most currently used take-over quality quantification methods, such as identifying the minimum time-to-collision or maximum lateral acceleration, neglect inherent criticality of the driving scenario. This can lead to poor overall quality assessment, especially in challenging take-over scenarios. This thesis proposes a novel method of quantifying how well the driver performed in the task of executing an emergency automation-to-manual control takeover, by means of a scenario-specific quality reference in the form of an optimized trajectory. Firstly, a human-in-the-loop driving simulator study was carried out using a high fidelity 6-DOF driving simulator (n=25). The simulated environment consisted of a two-lane highway setting. During the drive, six different automation-to-manual takeover scenarios occurred. Optimized trajectories were also generated for these scenarios, using a State-of-the-Art trajectory planner developed at Volkswagen Group Research Facility, Wolfsburg. The independent variables were the time budget (*5s, 7s, and 20s*), and the traffic density (*5 veh/km, and 10 veh/km*). A variety of dependent variables were determined for both the human driver trajectories and the optimized trajectories: *minimum time-to-collision, minimum distance-to-lane-boundary, minimum distance-to-overtaking-vehicle, maximum forced braking of the overtaking vehicle, maximum and standard deviations of accelerations (lateral and longitudinal), maximum and standard deviations of jerk (lateral and longitudinal), and lateral quickness*. Based on the findings from this human-in-the-loop driving simulator study and the generated optimized trajectories, a quantification framework was developed. The framework subdivides take-over quality assessment into two separate parameters. One with regard to safety, and one with regard to comfort. The safety parameter quantifies criticality of the human driver take-over based on the sum of normalized weighted quantification scores. These scores are computed for the metrics *minimum time-to-collision, minimum distance-to-lane boundary, minimum distance-to-overtaking vehicle, and maximum forced braking of the overtaking vehicle*, and are based on the difference to the metric values corresponding to the optimized trajectory. The comfort parameter quantifies comfort in a similar manner based on the metrics *lateral quickness, maximum lateral and longitudinal jerk, and standard deviation of lateral and longitudinal jerk*. The final quantification scores are comprised of a value between 0 and 1 for both the safety and comfort parameters. A perfect score of 0 would indicate all metrics were identical, or better than the optimized reference trajectory. A value of 1 indicates a crash for the safety parameter, and highly uncomfortable driving for the comfort parameter. The resulting quantification framework is scenario-specific as the quantification is performed relative to an optimized reference trajectory.

Nomenclature

Acronym	Meaning
ADAS	Advanced Driver Automation System
ACC	Automated Cruise Control
BAS	Brake Assist System
CPU	Central Processing Unit
DOF	Degrees Of Freedom
GPU	Graphical Processing Unit
LKA	Lane Keeping Assist
MDLB	Minimum Distance to Lane Boundary
MDOV	Minimum Distance to Overtaking Vehicle
PID	Proportional Integral Derivative
SA	Situational Awareness
SAE	Society of Automotive Engineers
TB	Time Budget
TD	Traffic Density
TTC	Time-to-Collision
TOC	Take Over Controllabilty
TOR	Take Over Request
TOPS	Take Over Performance Score
RDB	Relational Database

Symbol	Meaning	Units
Y	Quantification Score	[-]
k	Exponential tuning parameter	[-]

Subscript	Meaning
Linear	indicates a linear metric quantification
Piecewise	indicates a piece-wise metric quantification
Exponential	indicates a exponential metric quantification
Comfort	Overall comfort quantification
Safety	Overall safety quantification

List of Figures

List of Tables

Contents

1 Introduction

Over the past decade, significant improvements in the field of sensor technology and feedback-controlled systems have led to a substantial increase in vehicle automation becoming available to the public. These developments have attracted a great deal of attention as the main short-term focal points of automating privately-owned vehicles are primarily found in the domains of driver comfort and safety. *ADAS (Advanced Driver Assistance Systems)* such as *Automated Cruise Control (ACC)* and *Brake Assist (BAS)*, which control longitudinal motion of the vehicle, are broadly dispersed throughout the automotive industry and have the potential to increase both driver comfort and safety respectively [18][5]. These systems however, still require the driver to maintain lateral control over the vehicle, which is one of the primary driving tasks as described by [9], as well as according to the definition stated by the *SAE (Society of Automotive Engineers)*. The arrival of assistance systems that can acquire lateral control such as Lane Keeping Assist (*LKA*) has led to a situation where the driver is presented the option to fully extract him-or herself from the control loop of the vehicle. Although in current and near-future levels of automation (*SAE level 2/3*), the human driver still fulfills the role of fallback-system as the automated driving systems face strict limitations, both with regard to their operational domain as well as with handling certain unforeseen road situations.

It has been shown that being out of the control loop can lead to a degradation of situational awareness (*SA*), mental underload, complacency, over-reliance, and a mental overload in case the automation reaches a system boundary and prompts the driver to take back control [14]. This paradoxical situation where the operator faces rapid workload changes has been described by Bainbridge [1], in a paper appropriately named the Ironies of automation. The combination of having an out-of-the-loop human supervisor subject to all the before mentioned accompanying side-effects, that is still responsible for taking over control when needed to potentially avoid a dangerous collision in a time-critical situation, has sparked human factors research in the field of vehicle automation. Studies have revealed that the manner in which the take-over request (*TOR*) is issued influences the quality of the take-over itself [5] [13]. However, different studies often use different take-over quality quantification methods. Mean longitudinal acceleration has been used as a takeover quality metric by [16], in a study investigating the influence of varying traffic conditions on take-over quality. In [22], maximum lateral acceleration and center-line deviation were used when studying the impact of cognitive-visual load on take-over quality. Additionally, steering wheel reversal rate, hands-on time, gaze reaction time, and many other metrics have been used to describe take-over quality. A comprehensive list of take-over quality metrics found throughout literature is presented in [17].

In most cases however, take-over quality is quantified using temporal metrics, most frequently used metric being *Minimum Time-to-Collision (MTTC)*. Although an indicator for criticality of the take-over situation, it is argued that temporal metrics alone do not fully capture how well the driver performed from a vehicle control perspective. On the other hand, purely looking at vehicle dynamics-related metrics might be just as inadequate as vehicle stability in itself does not guarantee a safe and obstacle-free trajectory. Recently, two different so-called quantification frameworks have been published. These frameworks combine a variety of different metrics into a single overall take-over quality score. Firstly,

the *Take-over Controllability Rating (TOC)* quantifies take-over quality through a standardized coding sheet, which trained expert use to grade human driver takeover performance by close inspection of video footage [12]. Secondly, The *Take-over Performance Score (TOPS)* combines metrics chosen through expert assessment into three separate parameters related to vehicle guidance, mental processing, and subjective rating. These three parameters can be combined into a single quality score [17].

It is argued these frameworks are an improvement to traditional methods based on conclusions drawn from standalone metrics. However, even in the recently published methods mentioned above, the scenario itself is neglected in the quantification process. More time-critical take-over scenarios will generally result in 'worse' metric values (lower *MTTC*, higher accelerations and so on and so forth...), but often this is unavoidable due to the inherent criticality of the scenario, and thus fails to reflect accurately on the driver's performance. It is clear the scenario thus has large influence on the metrics themselves, but this tends to be neglected in quantification. Therefore, take-over quality quantification could benefit from an approach that is tailored specifically to the scenario that is being studied. In this thesis, a novel quantification method, from now on referred to as a quantification framework, will be proposed. The framework shares similarities to the Take-over Performance Score, as it combines multiple normalized metrics into seperate parameters. However, inherent scenario criticality and difficulty will be taken into account in the quantification framework by taking the included metric values relative to an optimized, scenario-specific, quality baseline value. This results in an arguably fairer driver take-over quality quantification, as high take-over quality scores can be achieved even with unfavourable values for traditional 'unreferenced' quality metrics, in critical situations. The reference, or quality baseline, will be determined through a state-of-the-art trajectory optimization algorithm.

2 Methods

2.1. Study Design

In order to quantify human driver take-over quality whilst taking the inherent criticality of the situation into account, a scenario-specific quality reference is used in the form of an optimized trajectory. A State-of-the-art trajectory planner developed at Volkswagen Group Research, Wolfsburg, was used to generate the optimized trajectories. The quantification framework proposed in this paper is based on differences between the human driver trajectory, or metrics belonging to the trajectory, and those corresponding to the optimized reference trajectory. The quantification is performed offline. However, theoretically it could also be performed in real time – as the trajectory planner is designed for real-time application as integral part of an autonomous driving system. In order to design this framework, a human-in-the-loop driving simulator study has been carried out using multiple different driving scenarios. Additionally, an optimized trajectory was generated for all driving scenarios. Background information on the trajectory planner used to generate the optimized trajectories will be given in section 2.4.

2.2. Participant Sample

A total of 31 Porsche AG employees were recruited to participate in this study. They were required to have a valid driver's license, as well as normal or corrected-to-normal vision. A total of 6 participants had to be excluded due to incomplete data recordings. Therefore, the analysis was performed on data sets from 25 participants ($n = 13$ females). The majority of the sample fell in the age category of $21 - 30$ years old (44%), the second largest category was $31 - 40$ years old (36%). The remainder was divided between $41 - 50$ and $51 - 60$ years old (16% and 4%, respectively). It was not permitted to ask participants' age directly due to Porsche AG privacy protection regulations, making it impossible to accurately determine mean age and standard deviation of the sample. Two participants indicated having participated in a simulator study regarding automated driving before.

2.3. Human-in-the-loop Driving Simulator Study

The drive-setting was a two-lane motorway with a speed-limit of 130km/h, which was exceeded by 10km/h for simulated left-lane overtaking vehicles. There was no hard-shoulder. At the start of the drive, the participants were asked to engage the automated driving function while driving in the right lane. The automation could be activated using a button located on the steering wheel. Upon activation, a message was shown in the combi-display indicating that the driver is now allowed to take his hands off the steering wheel. Activation was confirmed by the grey steering wheel icon (automation available) turning green (automation active). See figure 2.1. The participants were asked to then enjoy a secondary activity in the form of watching a leisurely video (*Brooklyn Nine-Nine S1E1* & *S1E2*) which automatically started playing on the 10.9 inch center-display when the automated driving function is enabled. This secondary activity was chosen as it fits the use-case of SAE level 3 automated driving, and no cognitive measures were required. During the approximately 30 minute-long drive, multiple take-over scenarios occurred. During which the driver is requested to take over by means of a one-stage auditory signal (details confidential) and a visual warning on the combi-display. The steering wheel icon

switches to red along with two hands holding it, as seen in figure 2.1. Additionally, the center-display turns black and shows a disclaimer stating 'Limited functionality while driving manually'. The driver is required to take-over longitudinal and lateral control of the vehicle and perform an evasive manoeuvre due to road-works (length 361m) on the right lane. The driver is asked to return to the right lane when possible, and manually re-engage the automated driving function.

Figure 2.1: Automated driving function icons. Left to right; Available - Active - Take-over!

A 3x2 multi factorial within-subject design was used in the human-in-the-loop driving simulator study. A total of six different take-over scenarios occurred, comprised of each possible combination for the conditions *time budget* and *traffic density* as shown in table 2.1. All take-overs took place on a straight stretch of road. Both conditions time budget and traffic density have been shown to affect take-over quality in previous studies based on traditional quality metrics [16] [5]. The three time budgets were 5 seconds, 7 seconds, and 20 seconds. 5 seconds is generally seen as highly critical. 7s has been determined as the minimum threshold for a safe take-over after being engaged in a secondary task [5]. The 20 second condition has been included to investigate human satisficing effects on the proposed quantification framework. The above-mentioned time budgets were combined with the condition 'traffic density low' and 'traffic density medium'. Defined in literature as 5 veh/km and 10 veh/km respectively [7]. For the medium traffic density scenarios, a left-lane overtaking vehicle, which was driving at a constant speed of 140km/h, was positioned directly behind the ego-vehicle at the time of the take-over request– making an immediate lane-change highly unsafe. As can be seen in figure 2.2, which shows the exact scenario layout for scenario 1.

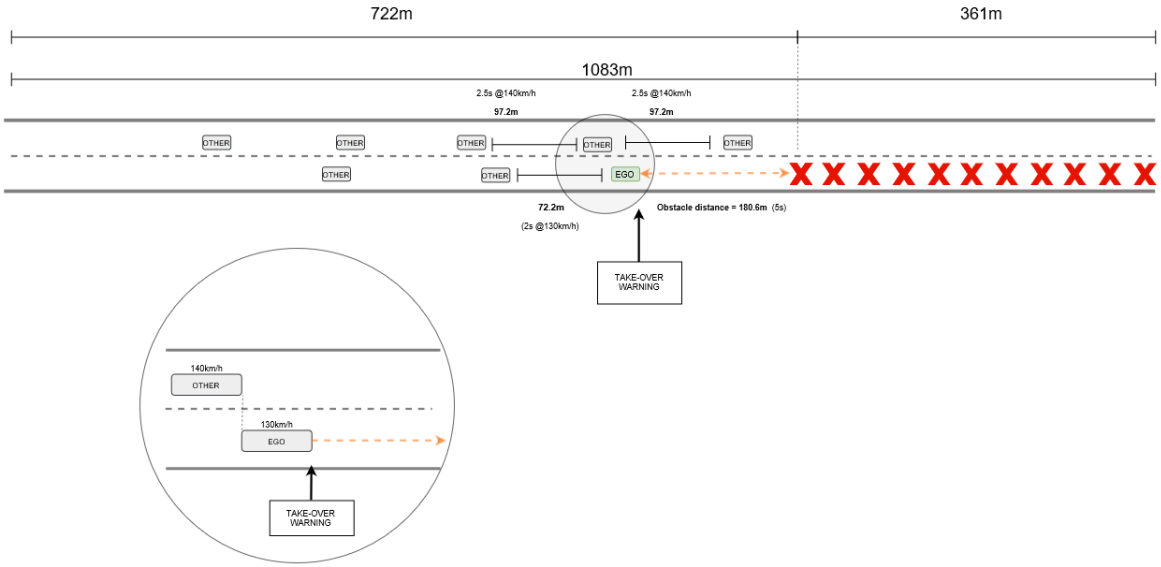


Figure 2.2: The road layout including dimensions and traffic conditions for take-over scenario 1 [TB = 5s, TD = med.]. Detailed visualizations of scenario 2,3,4,5, and 6 can be reviewed in Appendix A.1.

In the low traffic density condition, an immediate lane-change was possible due to the absence of nearby vehicles at the time of the take-over request. The exact road layout and positioning of other vehicles, similar to figure 2.2, for scenario 2,3,4,5 and 6 can be reviewed in Appendix A.1. The order in which the scenarios appeared was randomized in Latin-square design to counterbalance for order-related effects. The automated driving time in between take-overs was varied between 3 and 5.5

minutes in 30 second increments to counteract expectation. However, all automated driving times were identical for all participants.

Scenario	Time budget [s]	Traffic Density [veh/km]
1	5	10 (medium)
2	5	5 (low)
3	7	10 (medium)
4	7	5 (low)
5	20	10 (medium)
6	20	5 (low)

Table 2.1: Scenario composition of both conditions 'Time Budget' and 'Traffic Density'.

2.3.1. Dependent Variables

To ensure a fair comparison with the optimized trajectory generated by the trajectory planner, all dependent variables are either temporal or vehicle dynamics-related metrics. Driver-related metrics such as take-over-time and time to eyes-on-road are therefore not of interest in this study. Table 2.2 shows an overview of all objective dependent variables. The way in which they are incorporated in the quantification framework is discussed in section 2.6.

Metric	Vehicle	Description	Units
Minimum Time-to-Collision (MTTC)	Ego	TTC was determined as the time until a collision with the roadworks occurs. Minimum value is computed in accordance with [6]	$[s]$
Lateral Quickness	Ego	Lateral Quickness is defined as V_{lat}/d_{offset}. Where d_{offset} is the maximum lateral displacement during the lane change and V_{lat} is the maximum lateral velocity during said lane change [2]	$[1/s]$
Maximum Absolute Acceleration (Lateral and Longitudinal)	Ego	Maximum absolute acceleration value determined between the take-over request and the road-works end	$[m/s^2]$
Maximum Absolute Jerk (Lateral and Longitudinal)	Ego	Maximum absolute jerk value determined between the take-over request and the road-works end	$[m/s^3[$
STD of Acceleration (Lateral and Longitudinal)	Ego	Standard deviation of acceleration determined between the take-over request and the road-works end	$[m/s^2]$
STD of Jerk (Lateral and Longitudinal)	Ego	Standard deviation of jerk determined between the take-over request and the road-works end	$[m/s^3]$
Minimum distance to lane boundary	Ego	Minimum distance to the left lane outer boundary. Linear inversely related to lane change overshoot. Determined between the take-over request and the road-works end	$[m]$
Minimum distance to overtaking vehicle	Ego	Computed as the Euclidean distance between the ego-vehicle and the overtaking vehicle. Determined between the take-over request and the road-works end	$[m]$
Maximum deceleration of overtaking vehicle	Overtaking	The deceleration of the overtaking vehicle indicates the level of critical interaction with the ego-vehicle. Determined between the take-over request and the road-works end	$[m/s^2]$

Table 2.2: Overview of all dependent variables, which vehicle they relate to, a brief description, and the corresponding units of each variable.

Additionally, four subjective dependent variables were included. For each take-over scenario, a 7-point Likert-scale rating on *criticality [Sehr Unkritisch - Sehr kritisch], complexity [Sehr unkompliziert - Sehr kompliziert], comfort [Sehr komfortabel - Sehr unkomfortabel], and subjective time budget [Mehr als ausreichend - Viel zu Wenig]* was asked. The subjective rating questionnaire as used in the human-in-the-loop driving simulator study can be reviewed in Appendix A.1.

Dikablis eye-tracking glasses in conjunction with Dlab behavioural research software (Version 3.5) was used to confirm the driver's gaze was directed at the center-display at the time of each take-over request. The dataset was excluded if this was proven false. Eye-tracking was also used in order to be able to determine the gaze-related metrics required to determine the Take-over Performance Score [17]. This was done to allow for future method comparison.

2.3.2. Apparatus

The high-fidelity hexapod driving simulator at the Porsche Research and Development Facility in Weissach, Germany was used for this study. The hexapod was fitted with a fully functional mock-up of a Porsche Macan. The 6-DOF moving base platform (eMove eM6-640-1800) has an actuator stroke of 640mm. It's specifications can be reviewed in Table 2.3. A classical washout algorithm was used during the study.

	Excursions				**Velocity**		**Acceleration**	
	Single DOF		Non-single DOF					
Surge	-0.48	0.60 [m]	-0.64	0.63 [m]	0.8	[m/s]	7	[m/s2]
Sway	-0.50	0.50 [m]	-0.66	0.66 [m]	0.8	[m/s]	7	[m/s2]
Heave	-0.41	0.41 [m]	-0.41	0.41 [m]	0.6	[m/s]	10	[m/s2]
Roll	23.8	23.8 [deg]	-29.2	29.2 [deg]	35	[deg/s]	250	[deg/s2]
Pitch	-23.7	26.0 [m]	-28.2	32.9 [m]	35	[deg/s]	250	[deg/s2]
Yaw	-25.4	25.4 [m]	-28.7	28.7 [m]	40	[deg/s]	500	[deg/s2]

Table 2.3: eMove eM6-640-1800 moving base platform system specifications.

The visualized field of view is 180°, achieved by projectors displaying 3.840 x 2.160 pixels on all three sides, as well as overhead. The visualization is refreshed at a frequency of 60Hz. The simulator is shown in figure 2.3, during one of the take-over scenarios.

Figure 2.3: The hexapod simulator during take-over scenario 1. The participant has regained manual control and executed an emergency lane change to the left lane. Roadworks can be seen projected on the right lane.

The integrated vehicle dynamics model was a complex multi-DOF model corresponding to the high-performance electrically powered Porsche Taycan. Technical details on this model are confidential. The tyre-road friction coefficient was set to 1 for simplicity.

2.3.3. Procedure

All participants were scheduled into one hour time-slots. After welcoming the participant and thanking him or her for participating, the participant was asked to read the instruction form (shown in Appendix A.1). After having read the instruction form, there was the possibility to ask questions. However, it was made sure that no extra information was given about the purpose of the study. The demographic questionnaire was filled out along with a consent form. When finished with the paperwork, which took approximately 10-15 minutes on average, the participant was asked to take place inside the driving simulator. The assistant took place in the passenger seat and while stationary explained how to activate the automated driving function, how the subjective ratings could be filled out on the tablet, and went over the general instructions. When finished, the eye-tracking glasses were handed over to the

participant and calibrated using DLab behavioral research software (Version 3.5). A 5 minute test drive followed during which the participant could get used to the manual controls, automation, display icons, and tablet. After the test drive the moving base platform was brought down and the assistant got out of the simulator. Once the moving base platform was re-activated, the main study began. The participant engaged the automated driving function as soon as it became available. A requirement was that the vehicle was driving in the right lane. There was no speed requirement for activation, although it was advised to activate the automated driving function when driving approximately 130km/h as this is set as the automated driving function target velocity. When activated at a different speed, the automated driving function would accelerate or decelerate to the target speed of 130km/h. When the automated driving function is engaged, a video (Brooklyn nine-nine) is automatically played on the center-display. The participants' gaze was monitored using the front-view eye tracking cameras. In case the gaze was frequently directed at the road instead of the center-display, the intercom was used to remind the participant to trust the automated driving function and enjoy the video displayed. At a certain moment, the automated driving function prompts a take-over request. The driver is required to regain manual control and avoid a static obstacle in the form of roadworks, while being aware of any potential left lane overtaking vehicles.

After the roadworks, which is 361m long, the participant returns to the right lane and re-engages the automated driving function. The participant fills out the subjective rating questions with regards to the take-over scenario. From the control room, it was checked if the form was received properly. If this was not the case, the participants was kindly asked through the intercom to re-submit the subjective rating form. When finished with the subjective rating questions, the participant continues watching the video displayed on the center-screen. This order of events is repeated for a total of six times, with different take-over scenarios. After the 6^{th} take-over scenario, the intercom is used to ask the participant to bring the car to a standstill. The moving base platform is lowered and the participant exits the simulator.

A brief post-drive questionnaire with regards to simulator sickness and simulator fidelity is filled out. On this questionnaire, the participant also has the opportunity to give open feedback regarding the study. The participant is offered something to drink (non-alcoholic), as well as a sweet. After any potential questions the participant has regarding the study are answered, the participant is once again kindly thanked for their time.

2.4. Trajectory Optimization

2.4.1. CudaDP Trajectory Planner

The trajectory planner used to generate the quality reference trajectories for all six scenarios is officially called the *CudaDP planner*, after the C/C++ derived programming language Cuda in which it is developed, and the term *'Dynamic Programming'*.
Dynamic Programming can be defined as *'a method for solving a complex problem by breaking it down into a collection of simpler subproblems, solving each of those subproblems just once, and storing their solutions using a memory-based data structure'*. For dynamic programming to be applicable to a problem, it must exhibit the following characteristics [3].

- *Bellman's Principle of Optimality – a globally optimal solution constructed by locally optimal ones.*
- *The problem can be broken down into recursion.*

When, perhaps slightly oversimplified, defining the problem as getting from the vehicle's initial state to a goal state – defined here as the end of the take-over section, these characteristics are clearly met. The global optimal solution is constructed by locally optimizing the vehicles' motion to a next state. Where each state is defined by the set $[x_i, y_i, v_i, a_i, wheelangle_i, yaw_i]$. This can be performed in a recursive manner until the goal-state is reached. In order to achieve this, the CudaDP planner not only harnesses the computational power of the CPU (Central Processing unit) – but also of the GPU (Graphics Processing unit). The CudaDP planner relies on high-end GPU's manufactured by Nvidia Corporation. Nvidia Corporation states: *'Architecturally, the CPU is composed of just few cores with lots of cache memory that can handle a few software threads at a time. In contrast, a GPU is composed of hundreds of cores that can handle thousands of threads simultaneously. The ability of a GPU with*

100+ cores to process thousands of threads can accelerate some software by 100x over a CPU alone. What's more, the GPU achieves this acceleration while being more power- and cost-efficient than a CPU.' This cooperation between CPU and GPU makes it possible for the CudaDP planner to compute and assess all available states simultaneously. The interaction between the CPU and GPU during one step from an initial state to a subsequent state is shown in pseudo-code below:

1. *Upload valid initial state (CPU > GPU)*
2. *Generate motion transitions to available states (GPU)*
3. *Integrate vehicle dynamics for all possible transitions (GPU)*
4. *Set up motion costs for all possible transitions during integration (GPU)*
5. *Determine new state-space discretization (GPU)*
6. *Select the best candidate from the available states*
7. *Download new solution (GPU > CPU)*

Figure 2.5 shows an illustration of the available states (approximately 15.000) the CudaDP planner generates in a single timestep (40ms). The motion costs for each transition to an available state are based on optimizing progress to the goal-state, while at the same time minimizing the variables acceleration, jerk, wheel angle, wheel angle change, and object proximity. A PID-controller is used to follow the best-candidate trajectory determined by the optimization process. This is represented schematically in figure 2.4.

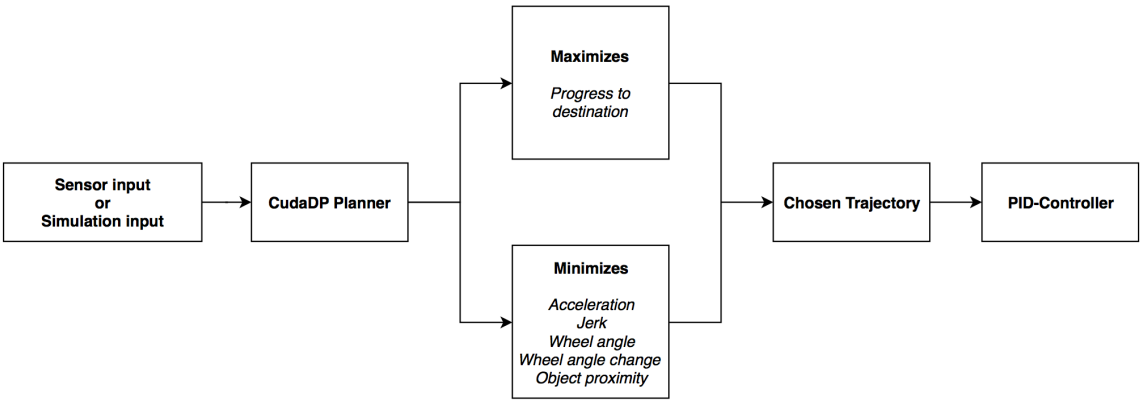


Figure 2.4: Simplified schematic representation of the optimization process executed by the CudaDP planner during trajectory planning.

In order to do so more efficiently, so-called 'masks' are created. These masks are placed on 'no-go' zones, such as sidewalks and lanes of opposite direction traffic, and unsurprisingly correspond to large motion cost offsets. Each of the before mentioned variables has an accompanying weight. These can be changed to alter the driving style of the algorithm. It is for instance possible to lower weights for acceleration (which is minimized) and increase the weight for progress (which is maximized). This makes the algorithm choose best-candidate states that lead to a more dynamic driving style. In some cases, such as for distance to obstacles, exponential variable weights are used. This results in exponentially increasing motion costs for states that linearly close in on an object. The weights generally used in the CudaDP planner have been determined by Volkswagen Group Research by a process similar to backtracking, where vehicles were driven manually and for each time step and corresponding state within the world, the 'human' weights were computed. Averaging out the human

weights over a plentitude of driving data led to the weights used in the CudaDP planner, although generally ‘safer’ weights are used in real-life autonomous driving tests.

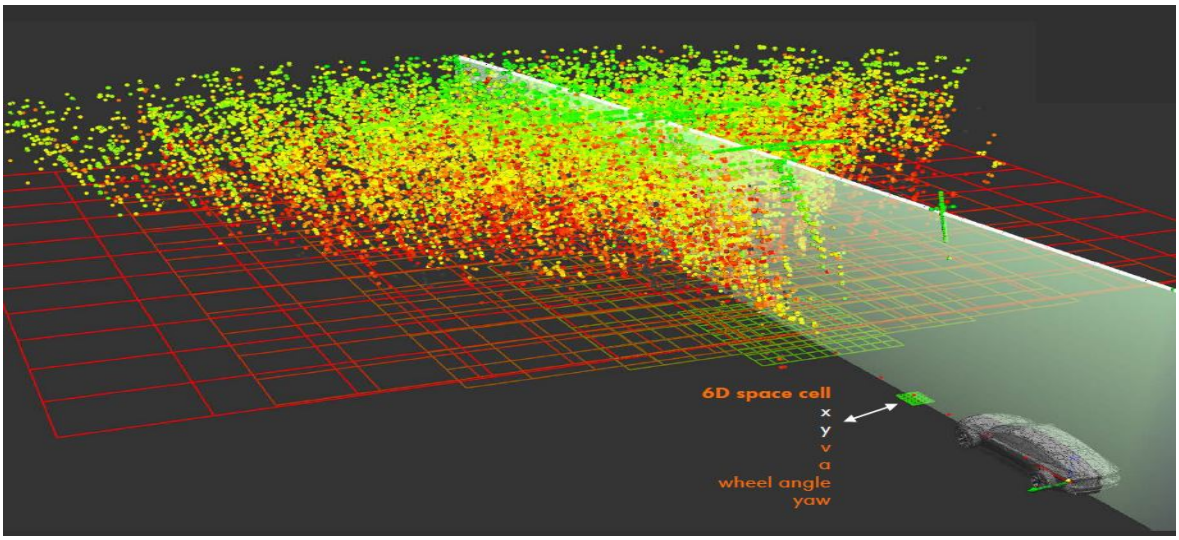


Figure 2.5: CudaDP planner available states: A state is defined as the set $[x_i, y_i, v_i, a_i, wheelangle_i, yaw_i]$. Visualized in the image is x and y position in the horizontal plane, and velocity in the vertical plane. Green are comfortable states, yellow/orange are non-comfortable states, red are collision states. The distance in time corresponds to the variable look-ahead time in simulations, or sensor range in real-life application. *Image courtesy of Volkswagen Group Research*.

To decrease computational effort, all states are divided into three categories. Comfortable (depicted as green), uncomfortable (depicted as yellow-orange), and collision (depicted as red). When available, the best-candidate selection algorithm only looks at comfortable states. An example of a transition to a non-comfortable state would be if high deceleration is required, or if the vehicle is forced to divert onto the sidewalk to avoid a collision. In case neither comfortable nor non-comfortable states exists, the selection criteria are based on minimizing impact energy during collision.

2.4.2. Take-over Simulations

In order to obtain an accurate comparison between the human trajectories and those generated by the CudaDP planner, the same files *(openDRIVE extensions .XODR and .XML)* describing the simulation environment including all simulated vehicles were used. However, some modifications had to be made in order for it to function in conjunction with the CudaDP planner. Figure 2.6 below shows the original road layout as used in the human-in-the-loop driving simulator study. Instead of using a single circuit or stretch, the road layout was designed as to facilitate take-over scenario randomization. The six green sections are the automated driving sections, whereas the blue sections are the take-over sections. A simulation technique called ‘jumping’ is utilized where the egovehicle ‘jumps’ from one location in the world to another, without the driver noticing. These jumps occurred between the ends of each automated driving section to the start of the required take-over section. This technique also made it possible to reset all other vehicle positions at the start of each take-over sections. This guaranteed scenario consistency over all participants. At the end of each take-over sections, the ego vehicle position automatically jumps back to the next automated driving section.

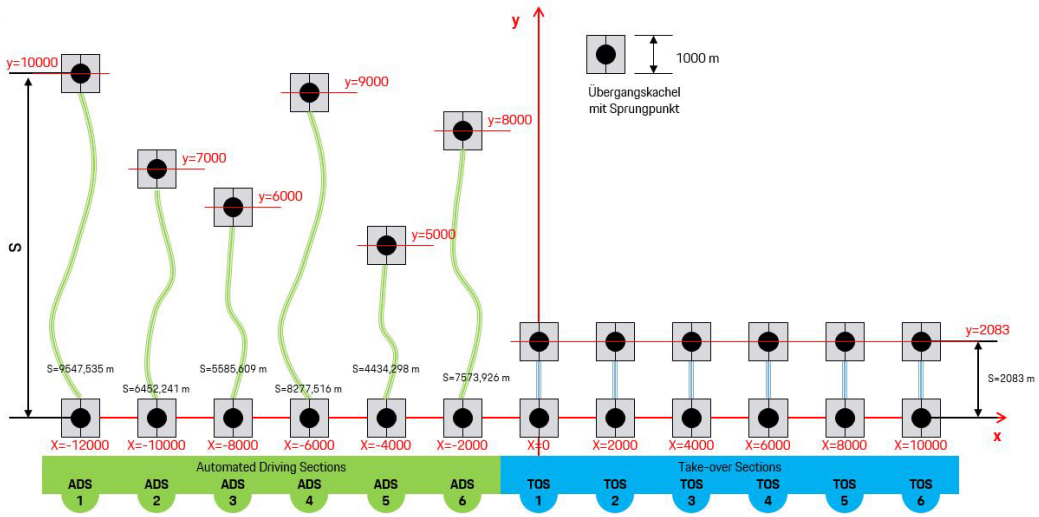


Figure 2.6: Road layout as used to create the simulation. Green sections are automated driving sections, whereas the equally sized red sections are sections where the take-over scenarios take place. During the human-in-the-loop driving simulator study, the ego-vehicle jumps between automated sections and take-over sections unbeknownst to the driver.

As it was only required for the CudaDP planner to generate a trajectory through the take-over sections, the automated driving sections were left out. Each take-over section was isolated and separate files *(OpenDRIVE extensions .XODR and .XML)* were created describing each individual take-over scenario. As the CudaDP planner was in continuous control of the ego-vehicle, the look-ahead time of the CudaDP planner was changed as to correspond with the time budget for each specific scenario. This recreated an identical urgency as for the participants of the human-in-the-loop driving simulator study.
The software and GPU-related hardware used to run the CudaDP planner simulations for this study is described in table 2.4

Name	Manufacturer	Type	Function
VTD 2.1	Vires	Software	Builds and simulates the openDrive environment described in the .XODR and .XML files
Virtual Studio IDE 2010	Microsoft	Software	Compiles the CudaDP planner code
Cuda Toolkit 8.0	Nvidia	Software	Development environment for creating high performance GPU-accelerated applications
ADTF 2.14	Elektrobit	Software	Visualizes CudaDP planner trajectories
ADTF-VTD Toolbox 4.1.0	Sedenius Engineering	Software	Connection toolbox for ADTF visualization within VTD constructed simulations
Quadro P5000	Nvidia	Hardware	High performance GPU

Table 2.4: An overview of the software and GPU-related hardware used to run the CudaDP planner simulations.

The ego-vehicle dimensions were those of a 2012-model Volkswagen Golf, although a simpler bicycle-model was used to describe the vehicle dynamics. A PID-controller controlled the vehicle control inputs in order to minimize deviations from the trajectory computed by the cudaDP planner [20]. Data logging was done using the RDB-datastream generated within the VTD software environment, which contains timestamped position, velocity, and acceleration data of the both the ego-vehicle and other simulated vehicles.

2.5. Analysis

The statistical analysis of the dependent variables focuses on main effects for the conditions *time budgets* and *traffic density*, as this is gives insight into the usefulness for application in quality quantification. Furthermore, additional effort was put into investigating which metrics are suitable for comfort quantification. This was done through a linear regression analysis for the dependent variable *subjective*

comfort, as well as a *Principal Component Analysis (PCA)*. General differences between the human driver trajectories and the optimized trajectories were also studied.

Based on these analyses combined with learnings from literature, a detailed quantification framework will be proposed in chapter 4, which can be used to quantify human driver take-over quality relative to an optimized quality reference. The general form of the quantification framework will be described in section 2.6.

2.6. General Framework

The framework as proposed in this thesis subdivides the matter of human driver take-over quantification in two separate parameters - one being the *safety parameter*, second being the *comfort parameter*. Similar to the TOPS framework, each parameter will be constructed through weighted addition of a selection of metrics. However, the novelty is introduced in the application of the metrics used. Instead of normalizing the original metric value, e.g. MaxAcc$_{driver}$, the difference to the optimized value MaxAcc$_{driver}$ - MaxAcc$_{optimized}$ is computed and normalized. This normalized value is then used to determine the metric score through a metric-specific quantification function. This leads to a quantification score between 0 (Identical to optimized) and 1 (Worst quality) for that particular metric. Where worst quality equals a crash when looking at the *safety parameter*, and highly uncomfortable driving in the *comfort parameter*. Thus, external factors such as the criticality of the scenario are taken into account as the optimized trajectory could also exhibit e.g. large maximum lateral accelerations if the situation so requires. This 'referenced normalization' process is performed on all metrics, for both the safety parameter and comfort parameter. A schematic representation of the quantification framework is shown in figure 2.7. Which metrics are included, as well as which normalization values and quantification functions are chosen is partly depended on the results from the analysis described in chapter 3. A detailed application of the quantification framework on the dataset collected in the human-in-the-loop driving simulator study is described in chapter 4.

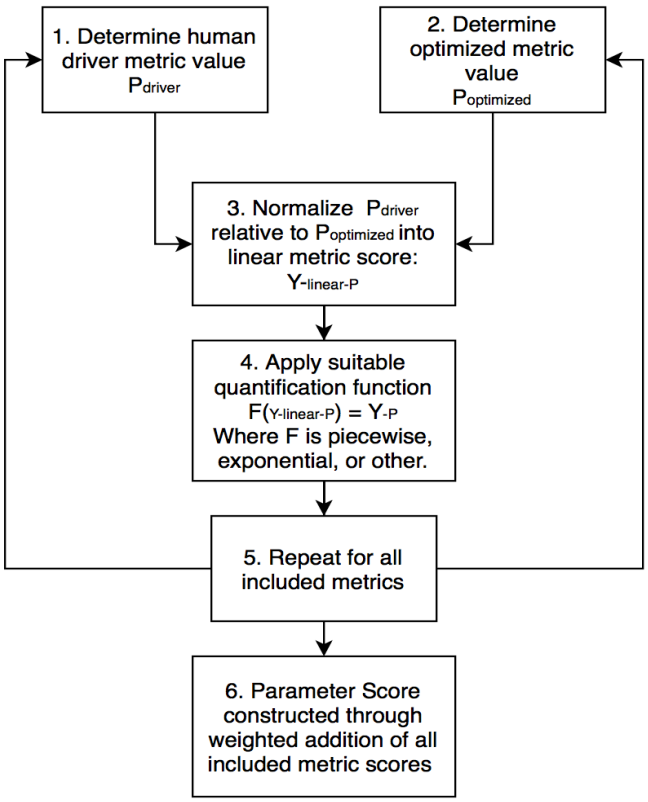


Figure 2.7: Schematic representation of the quantification framework. The metric P is a fictive metric for illustration purposes. Detailed descriptions of steps 3,4,5, and 6 are provided in chapter 4.

3

Results I: Statistical Analysis

3.1. Pre-processing

Multiple steps were performed in the pre-processing of the data. As mentioned earlier, six participants' data were excluded from analysis due to incomplete recordings. After reviewing the eye-tracking data, no further participants had to be excluded due to their gaze not being directed at the center screen during the TOR.

Jerk was not logged during data collection and was therefore algebraically computed using Matlab R2018b as the temporal derivative of acceleration. Before doing so, the data was filtered using a Butterworth Filter (bandpass [0.1 – 0.4Hz] for the CudaDP data – bandpass [0.1 – 0.7Hz] for the driving simulator data). These bandwidths were based on a spectral analysis of the signals. *MTTC* was also computed with Matlab R2018b in accordance with the *MTTC* definition as described in [6]. In case of leaving the road boundaries, negative values for the metric 'distance to lane boundary', were set to 0 to prevent outliers. The entire Matlab script used for data visualization and pre-processing can be reviewed in Appendix A.3.

3.2. Trajectories

Figure 3.1 shows all 25 participants' trajectories for all six scenarios. Additionally, the optimized reference trajectory as driven by the CudaDP planner is shown. The scenario numbers are related to the conditions time budget and traffic density as clarified in Table 2.1. A crash-event is defined as;

1. *A collision with the road-works. Identified as* $MTTC = 0$.
2. *Collision with the overtaking vehicle (only for scenarios 1,3,and 5). Identified as* $'MDOV - 2m = 0$. The -2m subtraction is due to vehicle dimensions, as explained in section 4.1.3.
3. *Leaving the road boundaries. Identified for the outer left lane boundary as* $'MDLB = 0$*, or otherwise through visual inspection.*

Six crashes were identified in scenario 1. Two of which due to a collision with the roadworks, three due to loss of control and leaving the road, and one due to a collision with the overtaking vehicle. Six crashes were also identified in scenario 2. However, only one was due to contact with the roadworks. The other five crashes in scenario 2 were caused by crossing the road boundaries. One crash was identified in scenario 3, due to leaving the road. This is not immediately visible from figure 3.1, but can be explained by the sideways offsets due to the vehicle width. No crashes occured in scenario 4,5, and 6.

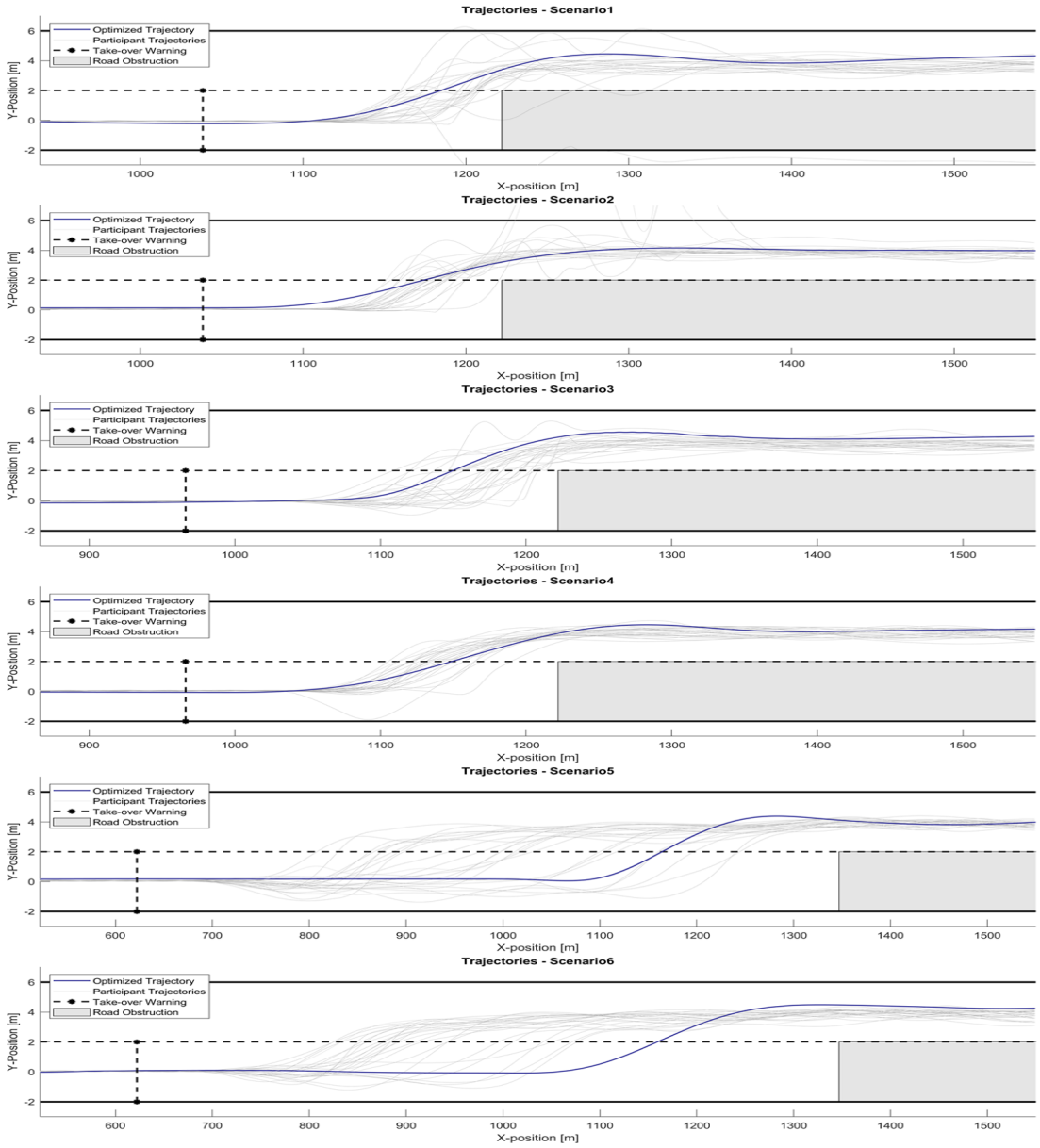


Figure 3.1: Overview of all participant trajectories as well as the optimized trajectory for all six scenarios. Scenario 1 [TB = 5s, TD = med.], Scenario 2 [TB = 5s, TD = low], Scenario 3 [TB = 7s, TD = med.], Scenario 4 [TB = 7s, TD = low], Scenario 5 [TB = 20s, TD = med.], Scenario 6 [TB = 20s, TD = low].

3.3. Main Effects

Acceleration-related metrics

The effects of the two conditions time-budget and traffic density were analysed using SPSS Statistics (Version 25). A two-way repeated measures Anova on maximum absolute lateral acceleration yielded a main effect for Time budget, $F(1.4, 30.8) = 27.3$, $p < .01$ (Greenhouse-Geisser corrected as Mauchly's sphericity test was significant). Pairwise comparison (Bonferroni corrected, $p < 0.01$) show significant differences for maximum absolute lateral acceleration between all time budgets, such that an inverse relationship between maximum lateral acceleration and time budget is determined. No main effects were determined for the traffic density condition. Additionally, the within subject means for maximum absolute longitudinal acceleration yielded a main effect for both time budget $F(2, 44) = 11.5$, $p < .01$ and traffic density $F(1, 22) = 13.9$, $p < .01$. Pairwise comparisons (Bonferroni corrected, $p < 0.05$) show that maximum absolute longitudinal acceleration is lower for the 5 and 7 second conditions compared to the 20s condition. The maximum absolute longitudinal acceleration is also shown to be higher in the

medium traffic density condition compared to the low traffic density condition.

A Two way repeated measures Anova performed on the standard deviations of both lateral and longitudinal acceleration showed similar results. For the standard deviation of lateral acceleration metric - a main effect was only found for time budget, $F(1.2, 26.9) = 29.3$, $p < .01$ (Greenhouse-Geisser corrected). For the standard deviation of longitudinal acceleration, a main effect was determined for time budget, $F(2, 44) = 13.6$, $p < .01$, as well as traffic density, $F(1,22) = 13.5$, $p < .01$. Pairwise comparisons (Bonferroni corrected, $p < 0.05$) were performed on both lateral and longitudinal standard deviations of acceleration. The standard deviations of lateral acceleration decreased significantly with increases in time budgets - both from 5 seconds to 7 seconds, as well as for an increase from 7 seconds to 20 seconds. The standard deviation of longitudinal acceleration showed a decrease for both 5 seconds and 7 seconds as compared to the 20 second time budget. All within subject means of all four acceleration-related metrics are shown for the different scenarios in figure 3.2. The error bars indicate the metrics' respective standard errors.

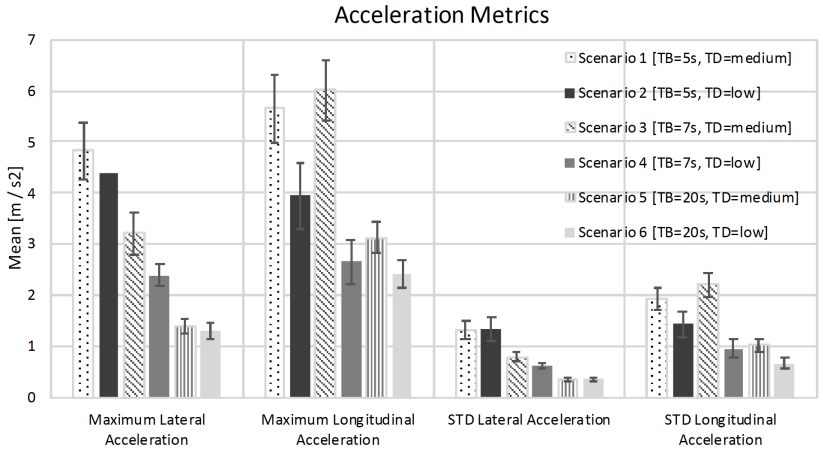


Figure 3.2: Overview of scenario-averaged results for the metrics Maximum Absolute Lateral Acceleration, Maximum Absolute Longitudinal Acceleration, STD Lateral Acceleration and STD Longitudinal Acceleration. The error bars indicate the standard error computed as $\frac{STD(X)}{\sqrt{numberofobservations}}$. Where X represents the respective metric.

Jerks-related metrics

Unsurprisingly, highly similar results were found when analyzing the temporal derivatives of the above mentioned acceleration-related metrics. Both maximum absolute lateral jerk and the standard deviation of lateral jerk show a significant effect for time budget - ($F(1.4, 31.3) = 36.5$, $p < .01$ (Greenhouse-Geisser corrected) and $F(1.3, 29.3) = 32.9$, $p < .01$ (Greenhouse-Geisser corrected), respectively. Maximum absolute longitudinal jerk and standard deviation of longitudinal jerk also show a main effect for time budget - $F(2, 44) = 12.3$, $p < .01$ and $F(2, 44) = 18.5$, $p < .01$, respectively. Additionally, both metrics also show a signifcant difference between the low and medium density traffic density - $F(1,22) = 13$, $p < .01$ and $F(1,22) = 15.4$, $p < .01$ respectively. Pairwise comparisons for all above mentioned jerk-related metrics yield identical results as for their acceleration-based counterparts (Bonferroni corrected , $p < 0.05$). Figure 3.3 shows the within subject means for four jerk-related metrics, across all six scenarios. The error bars indicate the metrics' respective standard errors.

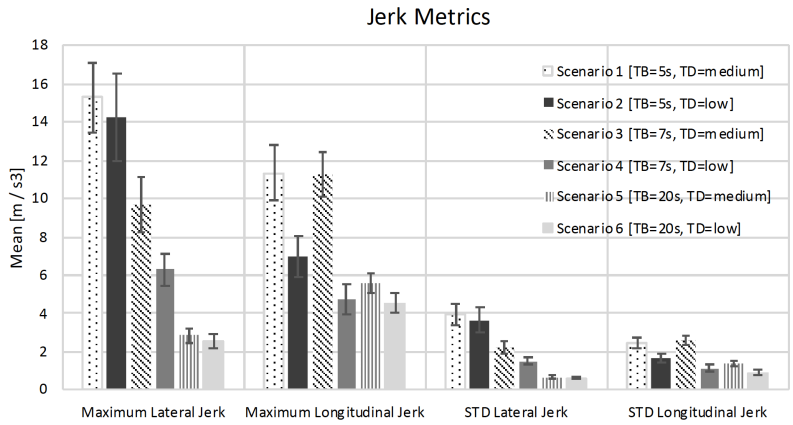


Figure 3.3: Overview of scenario-averaged results for the metrics Maximum Absolute Lateral Jerk, Maximum Absolute Longitudinal Jerk, STD Lateral Jerk and STD Longitudinal Jerk. The error bars indicate the standard error computed as $\frac{STD(X)}{\sqrt{numberofobservations}}$. Where X represents the respective metric.

Table 3.1 shows a comparison of the within-subject means for all acceleration-and jerk related metrics discussed above and the value corresponding to the CudaDP planner simulation, for each scenario.

Scenario		Max Acceleration $[m/s^2]$		STD Acceleration $[m/s^2]$		Max Jerk $[m/s^3]$		STD Jerk $[m/s^3]$	
		Lat.	Long.	Lat.	Long.	Lat.	Long.	Lat.	Long.
1	Within Subjects Mean	4.83	5.65	1.32	1.94	15.25	11.32	3.93	2.44
	CudaDP Planner	0.37	3.48	0.16	1.39	1.85	1.46	0.54	0.27
2	Within Subjects Mean	4.39	3.94	1.33	1.43	14.25	6.94	3.65	1.68
	CudaDP Planner	0.39	0.47	0.18	0.22	0.38	0.58	0.14	0.18
3	Within Subjects Mean	3.21	6.01	0.80	2.21	9.69	11.25	2.20	2.60
	CudaDP Planner	0.63	3.35	0.24	1.65	1.47	1.16	0.34	0.24
4	Within Subjects Mean	2.40	2.66	0.63	0.96	6.29	4.78	1.50	1.13
	CudaDP Planner	0.40	0.76	0.18	0.35	0.32	0.71	0.12	0.28
5	Within Subjects Mean	1.40	3.12	0.35	1.02	2.83	5.57	0.67	1.36
	CudaDP Planner	0.56	0.80	0.20	0.25	0.49	0.59	0.16	0.19
6	Within Subjects Mean	1.30	2.40	0.36	0.67	2.53	4.52	0.62	0.90
	CudaDP Planner	0.33	0.51	0.13	0.25	0.26	0.38	0.08	0.14

Table 3.1: Comparison of CudaDP Planner values and averaged human driver values for all acceleration and jerk-related metrics, for all scenarios.

MTTC

Analysis of the minimum time-to-collision revealed a main effect for time budget $F(1.19, 26.2) = 533.4$, $p < .01$ (Greenhouse-Geisser corrected). Pairwise comparison (Bonferroni corrected, $p < 0.05$) show significantly lower MTTC for the 5 second time budget compared to both the 7 seconand 20s time budget. There was no significant effect for traffic density. The within subject means for each scenario are shown in figure 3.4, along with the values corresponding to the CudaDP Planner's trajectories. The errors bars depict the standard error.

Lateral Quickness

A two-way repeated measures Anova on lateral quickness yielded main effects for Time budget, $F(2, 32) = 21.9$, $p < .01$ and Traffic Density $F(1,16) = 4.6$, $p < 0.05$. Pairwise comparison (Bonferroni corrected, $p < 0.05$) show significant differences for lateral quickness between all time budgets, such that an inverse relationship between lateral quickness and time budget is determined. The low traffic density condition resulted in lower values for lateral quickness compared to the medium traffic density condition. Figure 3.5 shows the within subject means along with their respective standard errors. Moreover, the lateral quickness belonging to the CudaDP Planner trajectories are added for visual comparison.

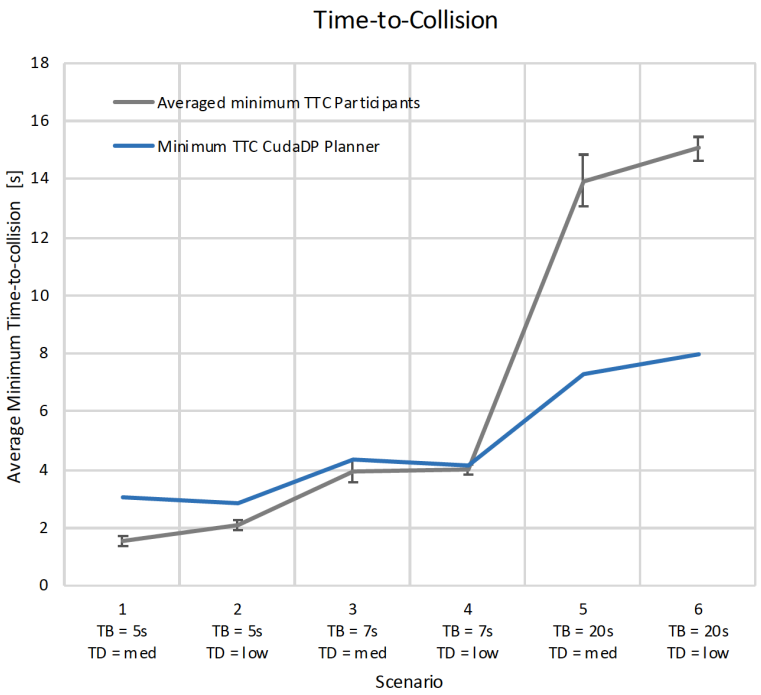


Figure 3.4: Comparison of the scenario-averaged human driver Minimum Time-to-Collision and the CudaDP planner MTTC. The error bars indicate the standard error computed as $\frac{STD(MTTC)}{\sqrt{numberofobservations}}$.

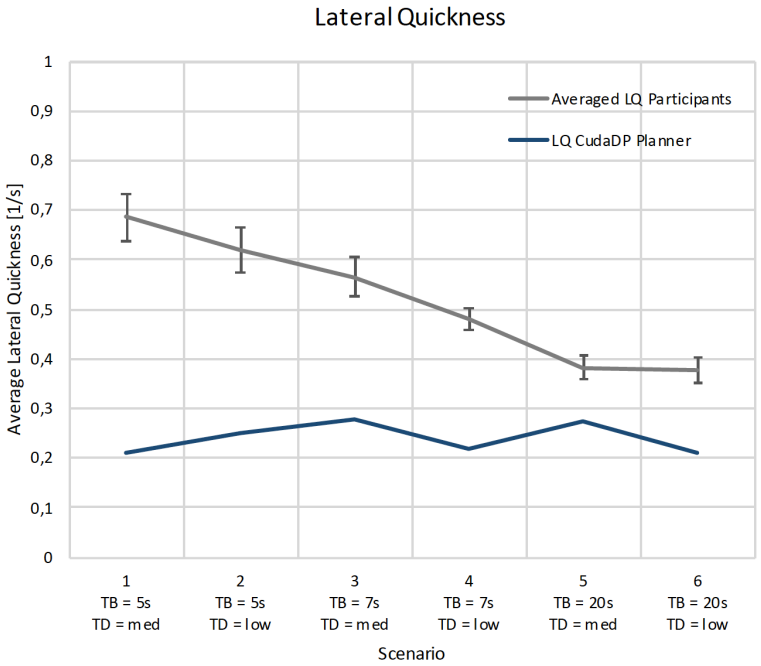


Figure 3.5: Comparison of the scenario-averaged human driver Lateral Quickness and the CudaDP planner Lateral Quickness. The error bars indicate the standard error computed as $\frac{STD(LQ)}{\sqrt{numberofobservations}}$.

Minimum distance to lane boundary

Due to the violation of sphericity and the resulting Greenhouse-geisser correction, a significant effect of time budget on distance to lane boundary was not proven. ($p = 0.059$, Greenhouse-Geisser corrected). There was no significant effect for traffic density.

Minimum distance to overtaking vehicle
A one-way Anova (for time budget only, overtaking vehicle was only present in the medium density condition) resulted in a significant effect $F(2,48) = 8.7$, $p<0.01$ for time budget on the distance to the overtaking vehicle. Pairwise comparison (Bonferroni corrected, $p < 0.05$) showed that the minimum distance to the overtaking vehicle was lower for the 5 second time budget compared to the 7 and 20 second time budgets.

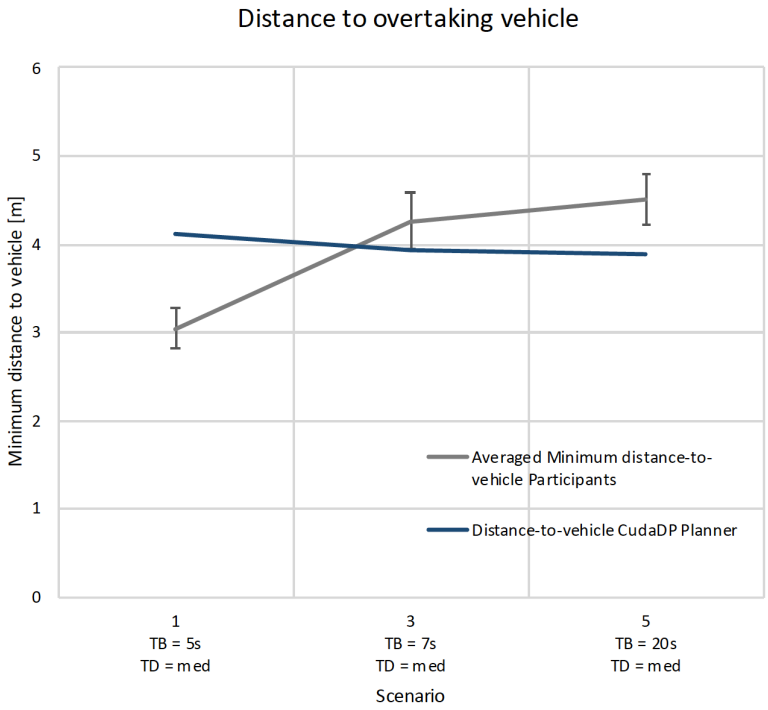


Figure 3.6: Comparison of the scenario-averaged human driver MDOV and the CudaDP planner MDOV. The error bars indicate the standard error computed as $\frac{STD(MDOV)}{\sqrt{number of observations}}$.

Maximum deceleration of overtaking vehicle
In this case, the normality assumption for data distribution was strongly violated. A non-parametric Friedman test of differences among repeated measures was conducted and rendered a Chi-square value of 6.46 which was significant ($P<0.05$).

Subjective Ratings
For use in the quantification framework as described in Chapter 4, the subjective criticality rating and comfort rating play an important role. The complexity and subjective time budget ratings were merely included as to allow for future comparison with the TOPS method [17], and are therefore not currently analyzed for statistical significance. A two-way repeated measures Anova on the subjective comfort rating yielded main effects for both time budget $F(2, 44) = 45.5$ $p < .01$ as well as traffic density $F(1,22) = 47.8$, $p < .01$ Pairwise comparison (Bonferroni corrected, $p<0.05$) show significant differences between all time budgets, as such that lower comfort ratings corresponds to the higher time budgets. Additionally, the low traffic density condition resulted in lower subjective comfort ratings. Keep in mind that a high comfort rating refers to an uncomfortable manoeuvre, and vice versa. This was done for consistency with the *positive [0] tot negative [6]* Likert scales belonging to the criticality, complexity and subjective time budget ratings. For criticality, main effects were found for both time budget $F(2, 44) = 110.1$ $p < .01$ and traffic density $F(1, 22) = 24.7$ $p < .01$. Where lower time budgets as well as the higher traffic density condition were rated more critically. Figure 3.7 shows the within subject means for all four subjective measures, over all six scenarios. The errors bars indicate the standard errors.

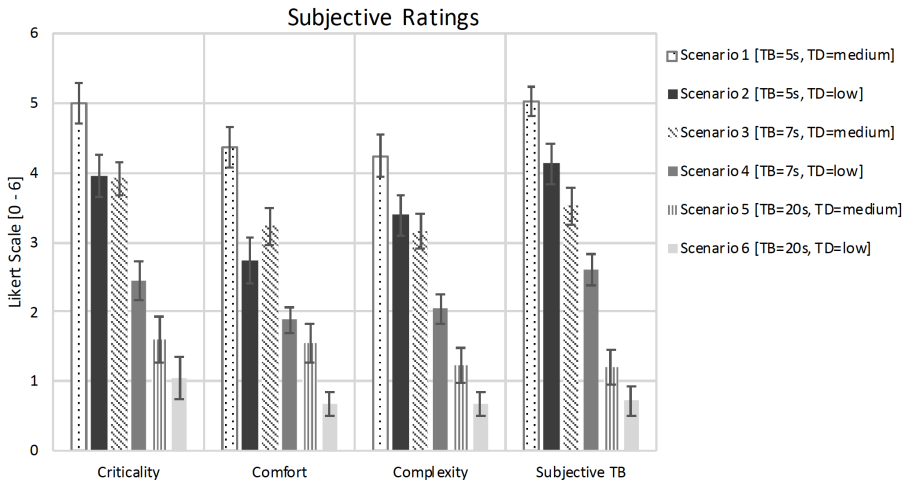


Figure 3.7: Overview of scenario-averaged results for the 7-point Likert scale subjective ratings on criticality [Sehr Unkritisch - Sehr kritisch], comfort [Sehr komfortabel - Sehr unkomfortabel], complexity [Sehr unkompliziert - Sehr kompliziert], and subjective time budget [Mehr als ausreichend - Viel zu Wenig]. The error bars indicate the standard error computed as $\frac{STD(X)}{\sqrt{numberofobservations}}$. Where X represents the respective subjective rating.

3.4. Comfort Analysis

In order to determine which objective metrics are suitable to quantify a phenomenon as subjective and susceptible to individual perception as comfort, theoretical and statistical approaches were combined. The purely statistical approach consisted of a linear regression analysis for the dependent variable *subjective comfort rating*. The theoretical part consisted of a literature analysis on comfort metrics, combined with a somewhat unorthodox use of principal component analysis. Lastly, Pearson correlations between all metrics of interest and the subjective comfort rating were studied.

3.4.1. Linear Regression Analysis

A linear regression analysis was performed to give insight into which of the objective metrics *maximum absolute lateral acceleration, maximum absolute longitudinal acceleration, std of lateral acceleration, std of longitudinal acceleration, maximum lateral jerk, maximum longitudinal jerk, std of lateral jerk, std of longitudinal jerk, and lateral quickness*, were significant predictors for the dependent variable *subjective comfort rating*. The data was combined at the level of trials in order to capture between-scenario variability, resulting in a sample-size of n = 140 participants. A stepwise method was used for the regression analysis. The Durbin-Watson statistic of the final model (4th) was 2.15. Two statistically significant ($p < 0.01$) predictors were included in this model, being *std of longitudinal jerk* and *lateral quickness*. This can be reviewed in figure 3.8, along with the step-wise model construction, coefficients, t-values, and collinearity statistics.

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Collinearity Statistics	
		B	Std. Error	Beta			Tolerance	VIF
1	(Constant)	1,461	,180		8,135	,000		
	MaxJerk_lat	,110	,017	,493	6,654	,000	1,000	1,000
2	(Constant)	,872	,215		4,047	,000		
	MaxJerk_lat	,088	,016	,394	5,398	,000	,906	1,103
	STDjerk_long	,472	,107	,322	4,403	,000	,906	1,103
3	(Constant)	-,099	,482		-,205	,838		
	MaxJerk_lat	,024	,033	,105	,713	,477	,215	4,640
	STDjerk_long	,521	,108	,355	4,830	,000	,869	1,151
	Quickness_lat	2,725	1,216	,320	2,241	,027	,231	4,334
4	(Constant)	-,343	,339		-1,010	,314		
	STDjerk_long	,546	,102	,372	5,359	,000	,971	1,030
	Quickness_lat	3,482	,592	,409	5,885	,000	,971	1,030

a. Dependent Variable: Comfort_likert

Figure 3.8: Linear regression model summary as outputted by SPSS; showing the stepwise model construction as well as the coefficients, std errors, t-values, significance levels, and collinearity statistics on each model.

3.4.2. Principal Component Analysis

To obtain further information about the variance distribution within the data set, a principal component analysis was performed on the same data selection as for the regression analysis. Although PCA is a data-reduction method, it is used here as to examine which metrics account for similar variances within the data. This gives added insight into which metrics should or should not be included in the comfort quantification, which will be described in Chapter 4. The kaiser-Meyer-Olkin Measure of Sampling yielded 0.762, and Bartlett's test of Sphericity was significant ($p < 0.01$, approximated Chi-Square = 1412.3). Two components were constructed based on visual inspection of the spree plot as shown in figure 3.9.

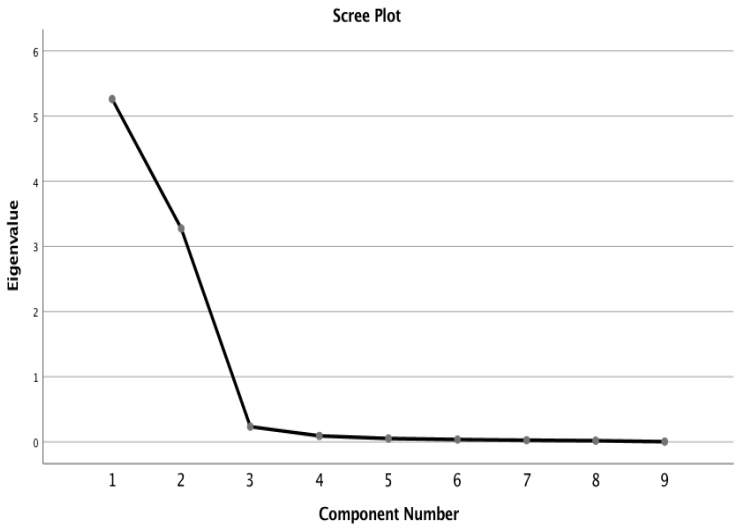


Figure 3.9: Spree plot used for PCA component number determination. Construction of 2 components was selected in the analysis.

The *Oblimin with Kaiser Normalization* rotation method was used to construct the components. The two components account for a cumulative percentage of explained variance of 94.86%. The pattern matrix is shown is table 3.2, where component factors that load less than 0.3 on the respective component are not displayed.

	Component	
	1	2
MaxAccLateral	0.985	
StdAccLat	0.981	
StdJerkLat	0.977	
MaxJerkLat	0.970	
QuicknessLat	0.922	
MaxAccLong		0.989
MaxJerkLong		0.984
StdJerkLong		0.976
StdAccLong		0.969

Table 3.2: The Pattern matrix showing the component loadings for the two constructed components. Loadings less than 0.3 are not displayed.

As logic would suggest, lateral dynamics-related metrics load strongly on one component (1), whereas longitudinal dynamics-related metrics load strongly on another (2). This indicates that in order to capture most variety in variance, metrics from both components must be included into the comfort quantification framework. However, due to the high intercorrelations between the second and third temporal derivates of position, it is argued that only acceleration or jerk be used for comfort quantification. This decision is made based on the Pearson correlations as shown in figure 3.10, and will be dealt with in detail in section 4.2.

		TTC	Quickness_lat	MaxAcc_lat	MaxAcc_long	STDacc_lat	STDacc_long	MaxJerk_lat	MaxJerk_long	STDjerk_lat	STDjerk_long	Criticality_likert	Comfort_likert
TTC	Pearson Correlation	1	-,495**	-,494**	-,248**	-,476**	-,301**	-,524**	-,256**	-,499**	-,290**	-,625**	-,566**
	Sig. (2-tailed)		,000	,000	,002	,000	,000	,000	,002	,000	,000	,000	,000
	N	148	140	148	148	148	148	148	148	148	148	148	148
Quickness_lat	Pearson Correlation	-,495**	1	,890**	,104	,823**	,136	,871**	,101	,819**	,171*	,476**	,473**
	Sig. (2-tailed)	,000		,000	,220	,000	,109	,000	,233	,000	,043	,000	,000
	N	140	140	140	140	140	140	140	140	140	140	140	140
MaxAcc_lat	Pearson Correlation	-,494**	,890**	1	,220**	,959**	,277**	,981**	,261**	,960**	,321**	,488**	,488**
	Sig. (2-tailed)	,000	,000		,007	,000	,001	,000	,001	,000	,000	,000	,000
	N	148	140	148	148	148	148	148	148	148	148	148	148
MaxAcc_long	Pearson Correlation	-,248**	,104	,220**	1	,166*	,942**	,264**	,952**	,201*	,936**	,367**	,436**
	Sig. (2-tailed)	,002	,220	,007		,044	,000	,001	,000	,014	,000	,000	,000
	N	148	140	148	148	148	148	148	148	148	148	148	148
STDacc_lat	Pearson Correlation	-,476**	,823**	,959**	,166*	1	,226**	,945**	,217**	,973**	,276**	,465**	,436**
	Sig. (2-tailed)	,000	,000	,000	,044		,006	,000	,008	,000	,001	,000	,000
	N	148	140	148	148	148	148	148	148	148	148	148	148
STDacc_long	Pearson Correlation	-,301**	,136	,277**	,942**	,226**	1	,313**	,919**	,244**	,932**	,375**	,448**
	Sig. (2-tailed)	,000	,109	,001	,000	,006		,000	,000	,003	,000	,000	,000
	N	148	140	148	148	148	148	148	148	148	148	148	148
MaxJerk_lat	Pearson Correlation	-,524**	,871**	,981**	,264**	,945**	,313**	1	,299**	,971**	,359**	,519**	,511**
	Sig. (2-tailed)	,000	,000	,000	,001	,000	,000		,000	,000	,000	,000	,000
	N	148	140	148	148	148	148	148	148	148	148	148	148
MaxJerk_long	Pearson Correlation	-,256**	,101	,261**	,952**	,217**	,919**	,299**	1	,240**	,965**	,376**	,454**
	Sig. (2-tailed)	,002	,233	,001	,000	,008	,000	,000		,003	,000	,000	,000
	N	148	140	148	148	148	148	148	148	148	148	148	148
STDjerk_lat	Pearson Correlation	-,499**	,819**	,960**	,201*	,973**	,244**	,971**	,240**	1	,296**	,486**	,460**
	Sig. (2-tailed)	,000	,000	,000	,014	,000	,003	,000	,003		,000	,000	,000
	N	148	140	148	148	148	148	148	148	148	148	148	148
STDjerk_long	Pearson Correlation	-,290**	,171*	,321**	,936**	,276**	,932**	,359**	,965**	,296**	1	,417**	,487**
	Sig. (2-tailed)	,000	,043	,000	,000	,001	,000	,000	,000	,000		,000	,000
	N	148	140	148	148	148	148	148	148	148	148	148	148
Criticality_likert	Pearson Correlation	-,625**	,476**	,488**	,367**	,465**	,375**	,519**	,376**	,486**	,417**	1	,772**
	Sig. (2-tailed)	,000	,000	,000	,000	,000	,000	,000	,000	,000	,000		,000
	N	148	140	148	148	148	148	148	148	148	148	148	148
Comfort_likert	Pearson Correlation	-,566**	,473**	,488**	,436**	,436**	,448**	,511**	,454**	,460**	,487**	,772**	1
	Sig. (2-tailed)	,000	,000	,000	,000	,000	,000	,000	,000	,000	,000	,000	
	N	148	140	148	148	148	148	148	148	148	148	148	148

Figure 3.10: Overview of Pearson correlations (including significance levels) for all objective dependent variables related to acceleration or jerk, lateral quickness, MTTC, and the subjective comfort and criticality rating.
**. Correlation is significant at the 0.01 level (2-tailed).
*. Correlation is significant at the 0.01 level (2-tailed)

.

4

Results II: Quantification Framework

4.1. Safety Parameter

The safety parameter describes how well a human driver performed a take-over manoeuvre from a purely safety-critical perspective. Currently, due to a lack of a situation-specific benchmark, absolute values of traditional safety metrics are used. However, this generalizes quantification for all take-over scenarios without taking the actual specifics of the scenario into consideration.

To properly describe the quality from a safety standpoint of an executed take-over manoeuvre, distance to static obstacles is important, as shown by the significant effect of time budget on subjective criticality. The *MTTC* also yielded significant differences for the condition *time budget*. Moreover, interaction with other road users should be taken into account. A significant effect of traffic density on subjective criticality was found. This indicates the presence of an overtaking vehicle - and thus the interaction with said vehicle - effects perceived criticality. Furthermore, the crash-events as described in chapter 3 relate to three types of incidents; a collision with a static obstacle, collision with another road user, and a loss of control/leaving the road. All of these crashes, or nearness of such crashes, must therefore be taken into account when quantifying safety. For the scenario-specific safety quantification proposed in this paper, the following metrics were included:

1. *Minimum time-to-collision [s]*
2. *Minimum distance to lane-boundary [m]*
3. *Minimum distance to overtaking vehicle [m]*
4. *Maximum forced braking overtaking vehicle [m/s^2]*

Metrics *time-to-collision, minimum distance to lane boundary, and minimum distance to overtaking vehicle* lend themselves perfectly for quantification due to the nature of their scale. A value of 0 for either of these metrics, is by the definition given in chapter 3 defined as a crash event. When taking the respective metric value related to the optimized trajectory, a scale between 0 [optimum] - 1 [crash] can be constructed using normalization. The metric *Maximum forced braking overtaking vehicle* is included as it portrays the level of interaction with the overtaking vehicle. In case the driver cuts off the overtaking vehicle, the amount of deceleration required by the overtaking vehicle to prevent a collision gives important insight into the criticality of the situation. This differs from distance to overtaking vehicle, as this deals with the ego-vehicle crashing into the overtaking vehicle from either the side or rear, which will not induce braking action by the overtaking vehicle. Additionally, minimum distance to overtaking vehicle can often be relatively small without the situation being highly critical. A cut-ff manoeuvre executed e.g. 5 meters in front of the overtaking vehicle already has the potential to be highly critical due to the possible velocity differences between both vehicles, whereas the minimum distance to the overtaking vehicle is often much smaller than 5 meters in uncritical scenarios. However, the maximum deceleration of the overtaking vehicle does not range between *0 [optimum] and 1 [crash]* as the other

included safety metrics do. Therefore, a value of $10m/s^2$ is taken as indicating an emergency braking action [17]. The metric *minimum distance to lane-boundary* can be rewritten as $1 - LaneOvershoot$, however due to the direct link with a crash event (*minimum distance to lane-boundary = 0*), this metric was deemed more suitable in safety quantification.

4.1.1. Minimum Time-to-Collision

The quantification score Y_{MTTC} for MTTC can be constructed by normalizing the human driver MTTC with respect to the optimized MTTC as shown in equation 4.1.

$$Y_{linear-MTTC} = \begin{cases} 0 & MTTC_{driver} \geq MTTC_{optimized} \\ 1 - \dfrac{MTTC_{driver}}{MTTC_{optimized}} & MTTC_{driver} < MTTC_{optimized} \end{cases} \tag{4.1}$$

However, this results in a straight-line quantification between the optimized minimum time-to-collision and a static-object collision defined by $MTTC_{driver} = 0$. According to [19], only encounters with a $MTTC_{driver} < 1.5s$ must be considered critical. Therefore, an approximated non-linearity is incorporated in the quantification based on a piece-wise function with $MTTC_{driver} = 1.5s$ as threshold value for the steeper line segment. This threshold is normalized with regard to the optimized trajectories' MTTC as seen in equation 4.2. This is done as to place the cut-off point for the steep segment of the piece-wise function at the appropriate location in relation to the optimized MTTC value. Placing the y-coordinate for the cut-off point at half the value of the normalized x-coordinate allows for an approximated non-linearity. The slope and constant for the steep line segment are determined based on the x-and y cutoff points as shown in equations 4.4 and 4.5.

$$X_{cutoff-mttc} = 1 - \frac{1.5}{MTTC_{optimized}} \tag{4.2}$$

$$Y_{cutoff-mttc} = 0.5 * X_{cutoff-mttc} \tag{4.3}$$

$$a_{mttc} = \frac{1 - Y_{cutoff-mttc}}{1 - X_{cutoff-mttc}} \tag{4.4}$$

$$b_{mttc} = 1 - a_{mttc} \tag{4.5}$$

The piecewise function can now be formulated as shown in equation 4.6.

$$Y_{piecewise-MTTC} = \begin{cases} 0.5 * Y_{linear} & MTTC_{driver} \geq 1.5s \\ a_{mttc} * Y_{linear} + b_{mttc} & MTTC_{driver} < 1.5s \end{cases} \tag{4.6}$$

The piece-wise function, which is dependent on the optimized time-to-collision, is shown in figure 4.1 for all six scenarios. It is clearly visible that the cut-off points for the steeper line segments related to $MTTC_{driver} < 1.5s$ are shifted further to the right as $MTTC_{optimized} - MTTC_{critical}$ increases. All optimized and average participant minimum time-to-collisions can be reviewed in figure 3.4. Hence, the time-to-collision piece-wise quantification function take into account if the driver's minimum time-to-collision is in the critical region, which leads to rapidly decreasing quality values for MTTC metric.

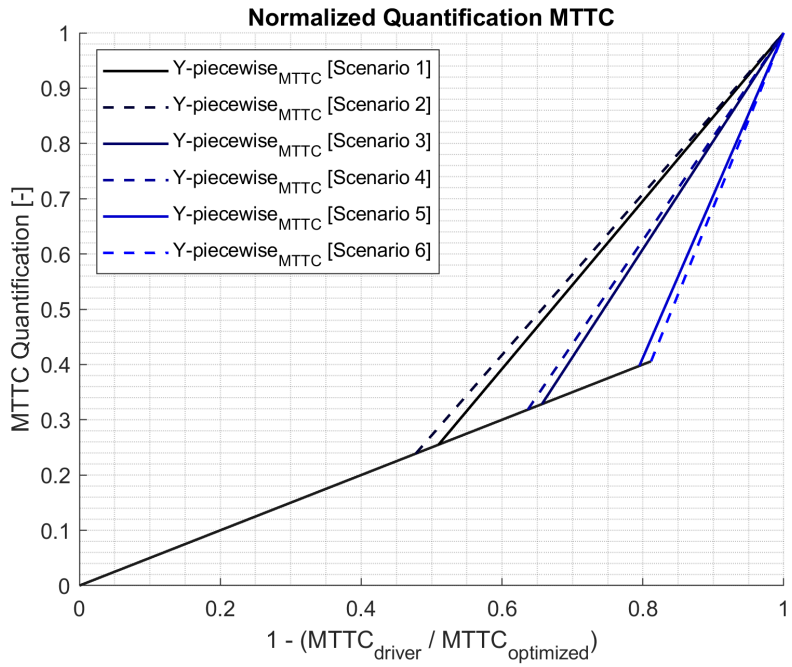


Figure 4.1: Piecewise quantification functions for all six scenarios.

4.1.2. Minimum Distance to Lane-boundary

The minimum distances to lane-boundary (MDLB) over the entire study range between 0m and 1.36m. Where 0 indicates leaving the road and 1m can be interpreted as a 'zero-overshoot / zero-undershoot' manoeuvre. A similar approach to quantify the MDLB-related quality is adopted as for time-to-collision. As drivers have been shown to usually keep a MDLB of 0.4m, a critical threshold of 0.4m distance to lane-boundary is used for the steeper section of the quantification curve [8]. Minimum-distance-to-lane boundaries smaller than 0.4m will therefore reduce the manoeuvre quality more drastically compared to values that are within the range $1 < MDLB < 0.4$. Due to unexplained differences in the optimized MDLB when examined over all six scenarios, the cut-off point will not be determined relative to each scenario-specific optimized MDLB, as done with the time-to-collision metric. Instead, the scenario-average optimized MDLB will be used. Once again, the driver's minimum distance to-lane boundary will be normalized with respect to the (averaged) optimized value as shown in equation 4.7.

$$Y_{linear-MDLB} = \begin{cases} 0 & MDLB_{driver} \geq MDLB_{optimized} \\ 1 - \dfrac{MDLB_{driver}}{MDLB_{optimized}} & MDLB_{driver} < MDLB_{optimized} \end{cases} \tag{4.7}$$

The piecewise quantification function is determined based on the averaged optimized *MDLB* as shown in equations 4.14 through 4.11

$$X_{cutoff-MDLB} = 1 - \frac{0.4}{MDLB_{averaged-optimized}} \tag{4.8}$$

$$Y_{cutoff-MDLB} = 0.5 * X_{cutoff-MDLB} \tag{4.9}$$

$$a_{MDLB} = \frac{1 - Y_{cutoff-MDLB}}{1 - X_{cutoff-MDLB}} \tag{4.10}$$

$$b_{MDLB} = 1 - a_{MDLB} \tag{4.11}$$

The resulting piecewise quantification function with regards to the *MDLB* metric is shown in 4.12 below.

$$Y_{piecewise-MDLB} = \begin{cases} 0.5 * Y_{linear-MDLB} & MDLB_{driver} \geq 0.4m \\ a_{MDLB} * Y_{linear-MDLB} + b_{MDLB} & MDLB_{driver} < 0.4m \end{cases} \tag{4.12}$$

These piece-wise line segments that quantify the driver's *MDLB* are visualized in figure 4.2.

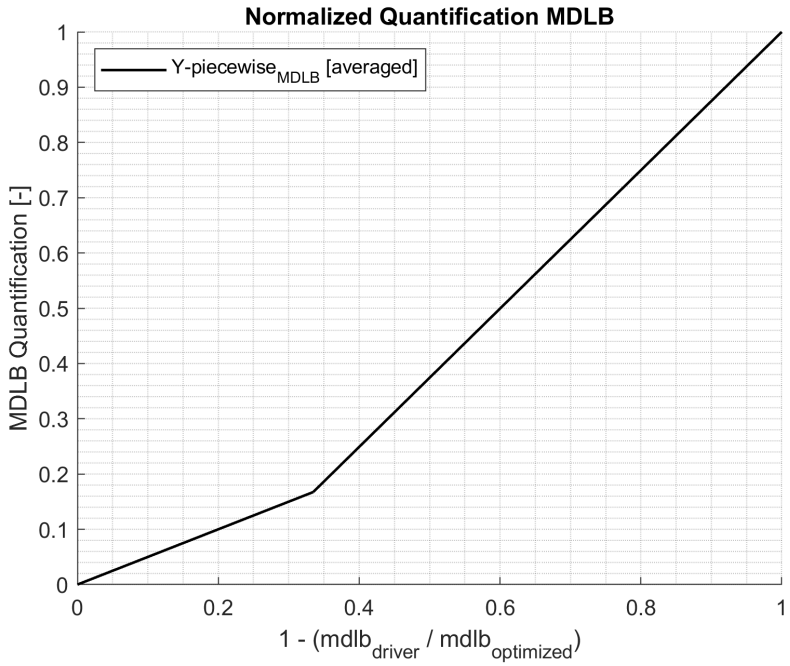


Figure 4.2: The piece-wise quantification function for the MDLB metric, valid for all six scenarios

4.1.3. Minimum Distance to Overtaking Vehicle

As seen in figure 3.6, the minimum distance-to-overtaking-vehicle (*MDOV*) for the optimized trajectory is close to 4m for all three medium-density scenarios. This is logically explained due to this being the lane-width, and also the distance between both vehicle's reference points when exactly next to each other while in their own respective lane. Due to non-equal vehicle dimensions around the reference point, the distances between the actual reference points were used in this analysis. Nonetheless, a $MDOV < 2m$ indicates a collision due to the smallest possible distance from the reference point to non-occupied space of 1m for both vehicles (approximated due to varying simulated overtaking vehicles). During highway driving, it is not uncommon for vehicles to get relatively close to each other. However, this is mainly the case when no collision-intersect is present [6]. This means that low *MDOV* values are generally seen during overtaking - when both vehicles are parallel to each other. As seen with the optimized trajectory, the lowest ideal *MDOV* in a scenario that includes a completed overtaking manoeuvre is 4 meters (for the used road layout). Values slightly smaller than 4m are to be expected as minor centerline-deviations are very frequent in human driving [22] [10]. These small deviations will affect *MDOV-related* quality less, when compared to *MDOV* values below a certain cut-off point. Therefore, piecewise line segments are used to quantify the driver's *MDOV*. The cut-off point is set to a *MDOV* of 0.5m. However, this value may need to be adjusted based on new research efforts. Additional options are proposed in the discussion of this document in chapter 5. Firstly, the normalized linear value is determined according to equation 4.13. The before mentioned 2 meter minimum-collision threshold is subtracted from the *MDOV* values.

$$Y_{linear-MDOV} = \begin{cases} 0 & MDOV_{driver} \geq MDOV_{optimized} \\ 1 - \dfrac{MDOV_{driver} - 2}{MDOV_{optimized} - 2} & MDOV_{driver} < MDOV_{optimized} \end{cases} \quad (4.13)$$

As the optimized trajectories' *MDOV* are very similar for all medium-density scenarios ($MDOV_{optimized} \approx 4m$), a single piecewise function will be used for quality quantification with regards to the driver's *MDOV*.

$$X_{cutoff-MDOV} = 1 - \frac{0.5}{MDOV_{averaged-optimized}} \quad (4.14)$$

$$Y_{cutoff-MDOV} = 0.5 * X_{cutoff-MDOV} \quad (4.15)$$

$$a_{MDOV} = \frac{1 - Y_{cutoff-MDOV}}{1 - X_{cutoff-MDOV}} \quad (4.16)$$

$$b_{MDOV} = 1 - a_{MDOV} \quad (4.17)$$

The resulting piecewise quantification function with regards to the *MDOV* metric is shown in 4.18 below.

$$Y_{piecewise-MDOV} = \begin{cases} 0.5 * Y_{linear-MDOV} & MDOV_{driver} \geq 0.5m \\ a_{MDOV} * Y_{linear-MDOV} + b_{MDOV} & MDOV_{driver} < 0.5m \end{cases} \quad (4.18)$$

The line segments used for quantification are shown in figure 4.3.

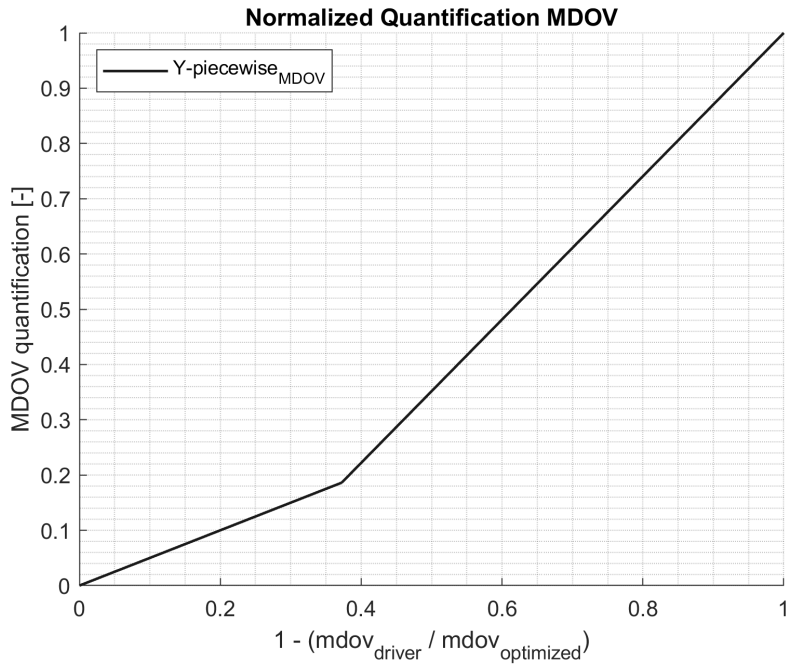


Figure 4.3: The piece-wise quantification function for the MDOV metric, valid for all six scenarios

4.1.4. Maximum Forced Braking Overtaking Vehicle

It is argued that any non-zero forced braking by the overtaking vehicle deteriorates quality strongly. The amount of deceleration required to prevent an accident is a measure of the severity of the overtaking vehicle cut-off. A maximum value of $10m/s^2$ is used, as this can be seen as a full emergency braking action. The value itself is based on the limits of friction forces between the tyre and road-surface [17]. As no overtaking vehicle deceleration occurred for any of the optimized trajectories, the baseline value is simply zero. However, this must not be generalized to all possible cases. Therefore, the linear normalization is defined as seen in equation 4.19.

$$Y_{linear-ForcedBraking} = \begin{cases} 0 & ForcedBraking_{driver} \leq ForcedBraking_{optimized} \\ \Delta ForcedBraking/10 & ForcedBraking_{driver} > ForcedBraking_{optimized} \end{cases} \tag{4.19}$$

Where $\Delta ForcedBraking = ForcedBraking_{driver} - ForcedBraking_{optimized}$.

Due to notion that interaction with other road users involving forced braking manoeuvres should be avoided - an exponential quantification scale is applied. This exponential scale is defined by equation 4.20.

$$Y_{exponential-ForcedBraking} = \frac{e^{k*Y_{linear-ForcedBraking}} - 1}{e^k - 1} \tag{4.20}$$

The variable k is a tuning parameter for the exponential quantification curve. Multiple curves ranging between $-5 < k < -1$ are displayed in Figure 4.20. Higher values for k will result in a harsher quantification for nonzero forced braking of the overtaking vehicle.

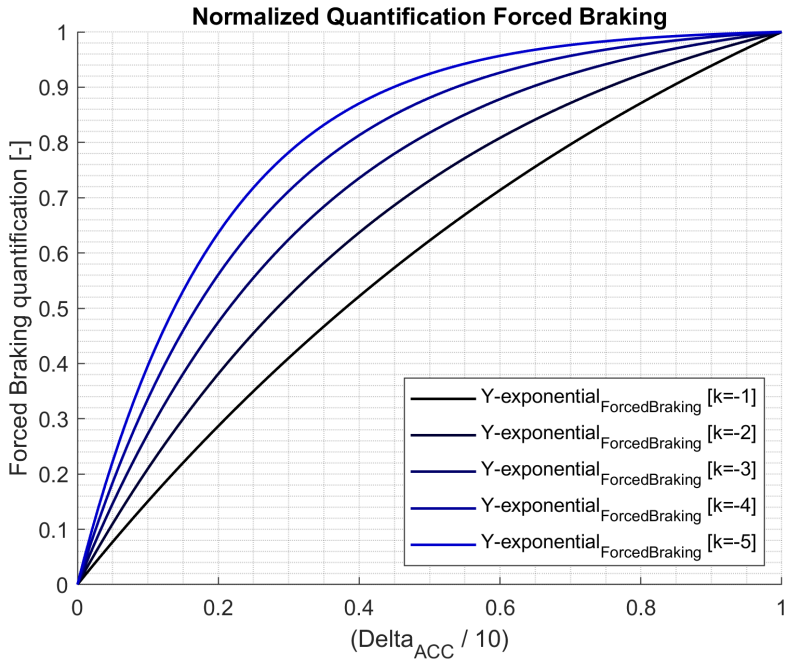


Figure 4.4: Multiple different quantification curves for the forced braking metric, with varying tuning parameter k

4.1.5. Safety Parameter Construction

The overall safety quantification value can now be constructed as a function of the parameter values described in sections 4.1.1, 4.1.2, 4.1.3, and 4.1.4. As all values are normalized to the range of 0 to 1, simple weighted addition can be used to generate an overall quantification value for safety. However, if one of the parameters *MTTC, MDLB, or MDOV* is given a value of 1, this indicates a crash-event. Therefore, a worst safety score of 1 is given for those cases. Mathematically this can be described as shown in equation 4.21.

$$Y_{Safety} = \begin{cases} 0 & Y_{TTC} \vee Y_{MDLB} \vee Y_{MDOV} = 1 \\ \dfrac{w1 * Y_{TTC} + w2 * Y_{MDLB} + w3 * Y_{MDOV} + w4 * Y_{FB}}{\sum w_i} & Y_{TTC} \,\&\, Y_{MDLB} \,\&\, Y_{MDOV} \neq 1 \end{cases} \tag{4.21}$$

As all four parameters have already been determined on their own specifically tailored functions (piece-wise or exponential), the weights (*w1, w2, w3, and w4*) will be set to 1. However, this can be adjusted based on future validation efforts of the quantification framework. The safety parameter yields a value between 0 and 1, which ranges from the optimized reference safety value (0), to a crash-event (1). This scale is displayed in figure 4.5.

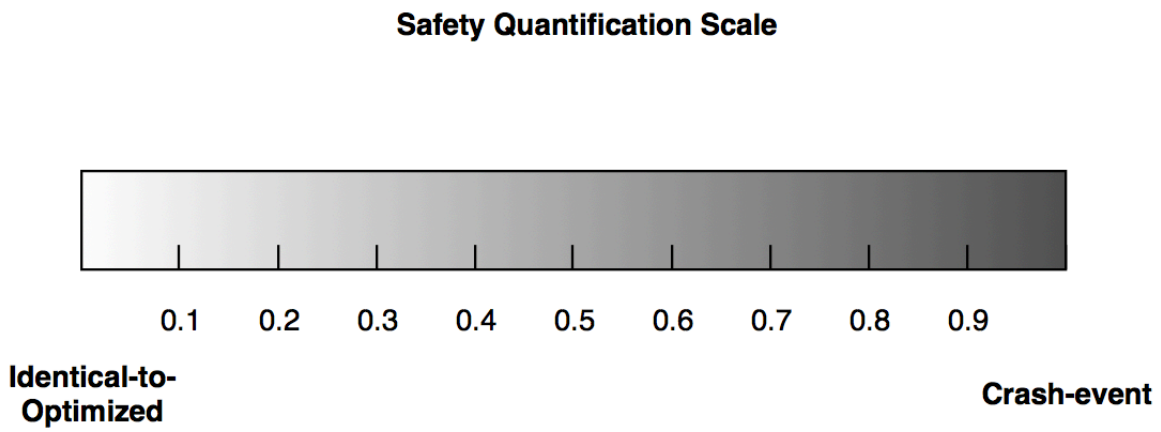


Figure 4.5: The final safety parameter score will be a value on the scale shown above, ranging from identical-to-optimized to a crash-event.

4.2. Comfort Parameter

Similarly to the safety parameter, a comfort parameter is constructed based on multiple metrics. These metrics are based not only on literature, but also partly on the findings obtained during this study. It was decided to include the jerk-related metrics (longitudinal and lateral maximum absolute jerk and longitudinal and lateral standard deviation of jerk) instead of their acceleration-related counterparts. This decision was based on the following findings as described in chapter 3.

1. Lateral acceleration-related metrics and lateral jerk-related metrics have a high intercorrelation as seen in table 3.10. The same goes for their longitudinal counterparts. This result is backed by the Principal Component Analysis, which placed all lateral metrics (max/STD cceleration and max/STD jerk) and longitudinal metrics (max/STD acceleration and max/STD jerk) in the same strongly-loaded components. Including both acceleration-related and jerk-related metrics would add only limited information. More instinctively, this is no surprise as jerk is the temporal derivative of acceleration so no 'new' information is added.

2. Based on literature, jerk is seen as a valuable objective comfort metric. Arguably even more so than acceleration, as rapid changes in acceleration are paired to higher discomfort than high acceleration in itself [11][4].

3. The Pearson correlations between the subjective comfort rating and acceleration-related metrics / jerk-related metrics were compared. It was found that the jerk-related metrics correlate slightly higher to the subjective comfort rating compared to their acceleration-related counterparts as seen in table 3.10.

The way in which the jerk-related metrics are included in the comfort parameter is explained in detail in section 4.2.2 and 4.2.3. Additionally, the lateral quickness metric is included in the comfort parameter.

4.2.1. Lateral Quickness

The linear regression analysis of which the results were shown in chapter 3 indicated lateral quickness to be a significant ($p < 0.01$) predictor for the dependent variable *subjective comfort rating*. This result corresponds to those found in [2], where lateral quickness was used to differentiate between driving styles. The exact definition of lateral quickness was clarified in section 2.3.1. As described in chapter 3, significant differences in lateral quickness were discovered for both time budget and traffic density. The pairwise comparisons for time budget yielded an inverse relationship between lateral quickness and time budget. This, along with the findings in literature and results from the regression analysis described in section 3.4.1 make lateral quickness a valid addition to the comfort quantification parameter. However, the upper-bound of the comfort scale is not as easily defined by physical constraints as for the safety parameter, which is limited either by a physical crash-event or by frictional forces during emergency braking. Therefore, the difference $\Delta_{LateralQuickness}$ between the optimized trajectory and human driver trajectory will be normalized with respect to the average lateral quickness during scenario 1. This scenario was considered most critical (TB = 5s, TD=medium), which showed in the subjective comfort rating. The human drivers rated their take-over manoeuvres most uncomfortable ($subjectiveComfortAveraged = 4.36$ on a scale of 0 to 6 [very comfortable - very uncomfortable]) when compared to the other scenarios ($F(2, 44) = 45.5$ $p < .01$ for time budget, $F(1,22) = 47.8$, $p < .01$ for traffic density, as can be seen in figure 3.7. Additionally, the highest average lateral quickness ($LateralQuicknessAveraged = 0.685$ was recorded for this scenario, as seen in figure 3.5. Therefore, the highest scenario-averaged lateral quickness value of 0.685 is rounded of to 0.7 and applied as the value used in normalization. This results in equation 4.22 defining the quantification line.

$$Y_{linear-LQ} = \begin{cases} 0 & LateralQuickness_{driver} \leq LateralQuickness_{optimized} \\ \dfrac{\Delta_{LQ}}{0.7} & LateralQuickness_{driver} > LateralQuickness_{optimized} \end{cases} \tag{4.22}$$

Where $\Delta_{LQ} = LateralQuickness_{driver} - LateralQuickness_{optimimized}$

Additionally, from figure 3.5 it is visible that the lateral quickness for the optimized trajectory is relatively constant over all scenarios (min = 0.21, max = 0.27). When looking at the raw data for all participants, over 95% of values for lateral quickness fall within $LateralQuickness_{optimized}$ and 1. This means that for this range, a $LateralQuickness_{driver}$ = $LateralQuickness_{optimized}$ will result in the highest possible comfort score, $Y_{linear-LQ} = 0$. On the other extreme, a $\Delta_{LQ} = 0.7$ is only possible with $LateralQuickness_{driver}$ values on the very upper side of the range, more specifically $LateralQuickness_{driver} > 0.92$ for the lowest $LateralQuickness_{optimized}(0.22)$. This means that this quantification scale normalizes more than 95% of all values to within the range of 0 to 1. The few outliers that obtain a $Y_{linear-LQ} > 1$ will be cut off to a value of 1, as this corresponds to worst possible lateral quickness-related comfort on this scale. Contrary to the safety metrics, a simple linear quantification line is used.

4.2.2. Maximum Jerk

Maximum absolute longitudinal and maximum absolute lateral jerk are both included in the comfort quantification parameter, as they describe the largest momentary peaks in longitudinal and lateral jerk, respectively. As jerk is often used as a comfort metric, these peaks can be viewed as peaks in discomfort [2]. The statistical analysis performed in chapter 3 yield significant differences in both absolute lateral and longitudinal maximum jerk for all three time budgets. More specifically, their relation is inverse, indicating higher jerk values are to be expected in more time-critical conditions. Additionally, a significant effect of traffic density was also determined for maximum absolute longitudinal jerk. This can be explained by the driver decelerating in order for overtaking vehicles to pass before executing an emergency lane change. These statistical findings support the use of maximum jerk values in a quantification framework. As normalization value the scenario-average for both absolute lateral and longitudinal maximum jerk belonging to scenario 1 [TB = 5s, TD = medium] are used. This can be seen as a suitable value is it results in relatively uncomfortable maximum jerk-related quantification values for scenario 1, which is deemed most critical. This also corresponds with the subjective comfort rating, which showed most drivers perceived their executed manoeuvre for this scenario as uncomfortable (average comfort subjective rating = 4.36, std error = 0.28). The maximum absolute lateral jerk normalization value is therefore set to $15 m/s^3$, whereas the maximum absolute longitudinal jerk normalization value is set to $11 m/s^3$. Besides this difference, the quantification lines describing lateral and longitudinal maximum jerk are constructed in similar fashion as shown in 4.23 and 4.24. Both will be kept linear.

$$Y_{linear-MaxJerkLat} = \begin{cases} 0 & MaxJerkLat_{driver} \leq MAxJerkLat_{optimized} \\ \dfrac{\Delta_{MaxJerkLat}}{15} & MaxJerkLat_{driver} > MaxJerkLat_{optimized} \\ 1 & \Delta_{MaxJerkLat} \geq 15 \end{cases} \tag{4.23}$$

$$Y_{linear-MaxJerkLong} = \begin{cases} 0 & MaxJerkLong_{driver} \leq MAxJerkLong_{optimized} \\ \dfrac{\Delta_{MaxJerkLong}}{11} & MaxJerkLong_{driver} > MaxJerkLong_{optimized} \\ 1 & \Delta_{MaxJerkLong} \geq 11 \end{cases} \tag{4.24}$$

Where $\Delta_{MaxJerk} = MaxJerk_{driver} - MaxJerk_{optimimized}$ for both equations.

4.2.3. Standard Deviation of Jerk

As opposed to the maximum longitudinal and lateral jerk metrics, the standard deviation of jerk sheds light on the spread of jerk over time, instead of a momentary peak. A higher standard deviation of jerk indicates an overall 'jerky' driving style, often perceived as uncomfortable. [2] Additionally, significant differences in standard deviation of both lateral and longitudinal jerk were determined for the different

time budgets. Traffic density only influenced the standard deviation of longitudinal jerk. Standard deviation of longitudinal jerk was also found to be a significant predictor ($p < 0.01$) for *subjective comfort rating*. The quantification line can be constructed in similar fashion as for the maximum jerk metrics. As the standard deviation of jerk depends on the time-window inspected, and the type of manoeuvre carried out inside that time-window, it is difficult to obtain a suitable value which can be used for normalization. Therefore, similar to the maximum jerk quantification,the scenario-average standard deviations of lateral and longitudinal jerk from scenario 1 [TB = 5s, TD = medium] are used. Using these averages will result in relatively uncomfortable standard deviation of jerk-related quantification values for scenario 1. However, as with the maximum jerk quantification, this corresponds well to the perceived subjective comfort ratings for this highly critical scenario. The normalization value for standard deviation of lateral jerk will therefore be set to $4m/s^3$. The normalization value for standard deviation of longitudinal jerk will be set to $2.5m/s^3$. Both linear quantification lines are defined as shown in equation 4.25 and 4.26.

$$Y_{linear-STDJerkLat} = \begin{cases} 0 & STDJerkLat_{driver} \leq STDJerkLat_{optimized} \\ \frac{\Delta_{STDJerkLat}}{4} & STDJerkLat_{driver} > STDJerkLat_{optimized} \\ 1 & \Delta_{STDJerkLat} \geq 15 \end{cases} \tag{4.25}$$

$$Y_{linear-STDJerkLong} = \begin{cases} 0 & STDJerkLong_{driver} \leq STDJerkLong_{optimized} \\ \frac{\Delta_{STDJerkLong}}{2.5} & STDJerkLong_{driver} > STDJerkLong_{optimized} \\ 1 & \Delta_{STDJerkLong} \geq 11 \end{cases} \tag{4.26}$$

Where $\Delta_{STDJerk} = STDJerk_{driver} - STDJerk_{optimimized}$ for both equations.

4.2.4. Comfort Parameter Construction

The overall comfort quantification score can now be constructed using the metric-specific quantification values as computed in sections 4.2.1, 4.2.2, and 4.2.3. Similar to the construction of the safety parameter, a weighted sum is used. The overall comfort quantification score can be displayed mathematically as shown in equation 4.27.

$$Y_{Comfort} = \frac{w1 * Y_{LQ} + w2 * Y_{MaxJerkLat} + w3 * Y_{MaxJerkLong} + w4 * Y_{STDJerkLat} + w5 * Y_{STDJerkLong}}{\sum w_i} \tag{4.27}$$

According to [17], lateral dynamics influence comfort perception more than longitudinal dynamics by an approximated factor 2. Therefore the weights for lateral quickness, maximum lateral jerk, and standard deviation of lateral jerk are weighed a factor 2. Meaning w1,w2 and w4 are set to 1, whereas w3 and w5 are set to 0.5. This does demand future validation and/or adjustment.

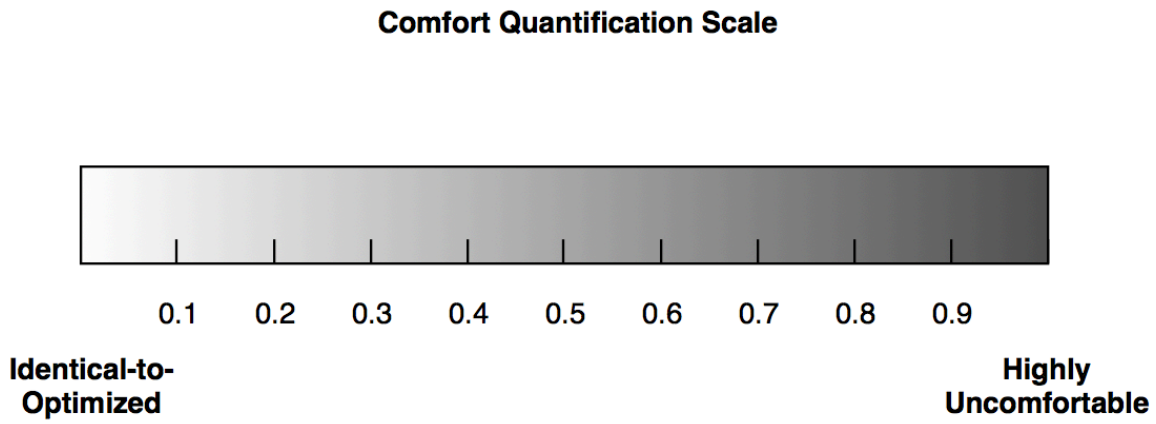


Figure 4.6: The final comfort parameter score will be a value on the scale as shown, ranging from identical-to-optimized to highly uncomfortable.

4.3. Applied Results

All take-over manoeuvres performed during the human-in-the-loop driving simulator study were quantified according to the framework as described in the previous sections. The raw results visualizing each participants' score for all included metrics can be reviewed in Appendix A.2. Table 4.1 below displays the scenario-averaged scores for both the safety parameter and comfort parameter. All thresholds were set to the values as described previously. For quantification of forced braking, the exponential tuning parameter k was set to -3.

Scenario	Average Safety Score [-]	Average Comfort Score [-]
1 (TB = 5s, TD = med)	0.40 (0.39)	0.67 (0.21)
2 (TB = 5s, TD = low)	0.32 (0.43)	0.53 (0.19)
3 (TB = 7s, TD = med)	0.16 (0.20)	0.45 (0.25)
4 (TB = 7s, TD = low)	0.08 (0.10)	0.36 (0.19)
5 (TB = 20s, TD = med)	0.06 (0.08)	0.26 (0.10)
6 (TB = 20s, TD = low)	0.01 (0.03)	0.22 (0.12)

Table 4.1: Scenario-averaged human driver take-over quality scores and their standard deviation (shown in parentheses) as computed with the proposed method.

5

Discussion

This discussion will elaborate on strengths and flaws of the quantification framework as described in chapter 4. But before doing so, the methods and results from both the human-in-the-loop simulator study and CudaDP planner simulations will be discussed, as they were vital instruments in the development of this framework. Therefore, they are inexplicably connected to the discussion about the quantification framework itself.

One thing that became instantly clear during this research, was the difficulty involved in creating a fair comparison between two completely separate simulations, something that strongly decreases applicability of this method. Effectively, the sensor inputs (environment, static objects, and dynamic objects) normally used as input for the CudaDP planner were replaced by the simulated environment as described by the same files (see section 2.4.2 for details) used to run the human-in-the-loop driving simulator study. This created the problem that the CudaDP planner 'knew' everything from *t=0*, which would result in the scenario specific time-budgets not being respected. To counteract this, the look-ahead time of the CudaDP planner was adjusted to fit each time-budget. However, this was not as exact as hoped, which meant average human reaction times could not be included in the CudaDP planner simulations. It could also be argued that this was not necesary to begin with, as the goal was to find a near-optimum trajectory, and human cognitive limitations are inherently sub-optimum. It must also be noted that it is not claimed the trajectories created by the CudaDP planner are in fact, optimum. Rather, they are trajectories optimized for a specific set of variables as described in section 2.4.2. The CudaDP planner is a non-deterministic system, meaning it might drive slightly differently on identical settings, simply due to uncertainties and slight variations in the simulated environment. Due to this phenomenon, it could also not exactly be explained why the MDLB varied as much over scenarios.

Nonetheless, the results it produces are generally extremely good when compared to human drivers. This holds mostly for the time-critical scenarios with time budgets of 5 and 7 seconds combined with the medium traffic density. Whereas many drivers were startled by the upcoming roadworks whilst not being able to make a direct lane change, the CudaDP planner gently decelerates and lets the vehicle pass before executing a smooth lane change. In doing so, it keeps lateral accelerations to an absolute minimum, and safe collision-margins to static and dynamic obstacles. Table 4.1 shows that both safety and comfort scores generally improve when criticality of the scenario decreases. This indicates that the differences between a human driver take-over manoeuvre and that of the CudaDP planner are larger for the critical scenarios, which can most likely be attributed to limitations in human reaction times and cognitive overload [21][22]. In scenario 5 and 6, participants scored very good on safety ($Mean_5 = 0.06(0.08), mean_6 = 0.01(0.03)$) as well as comfort ($mean_5 = 0.26(0.10), mean_6 = 0.22(0.12)$). This suggests obtaining a 'perfect' safety score is much more achievable than a perfect comfort score. On the other hand, the average subjective comfort scores for scenario 5 and 6, being 1.56 and 0.68 respectively, indicate the human drivers perceived these manoeuvres as relatively comfortable. Therefore, values between 0 and 0.2 for the comfort parameter may not be easily reached. This could be due to limitations in the human as a feedback-controlled operator, but could also simply be a matter of satisficing.

A flaw in the comparison however, is in the vehicle dynamic models used. Where the CudaDP planner is linked to a vehicle dynamic model of an ordinary Volkswagen Golf, the human drivers were 'in control' of a high performance electric sportscar capable of much greater lateral and longitudinal accelerations than a Golf. Additionally, they might not have fully noticed high accelerations due to the inherent limitations of simulator fidelity [15]. This was made clear with one participant, who accelerated to over 180km/h (in mere seconds), without even noticing it. Multiple participants have also stated in the open remark of the post-drive questionnaire that high accelerations did not feel realistic. This could potentially also have lead to perceived comfort being dependent more on visual cues than haptic feedback, compared to real driving situations. On the topic of visual cues, it is worth noting that there was a small blind-spot in the left mirror. This was not noticed until a participant pointed it out. Due to consistency for all participants it was decided not to seek a solution.

The 20 second time budgets were included to look for satisficing effects in human driving, and to compare these to an optimized trajectory. As expected, an optimized trajectory for a non-critical takeover scenario is not clearly 'better' in any way. In fact, most acceleration/jerk related metrics are very similar to the more time-critical 5 and 7 seconds time budgets. This indicates that the CudaDP planner tends to want to finish the manoeuvre more quickly, rather than decrease lateral accelerations below a certain value. As the CudaDP planner does not seem to exhibit large differences in lateral quickness and maximum lateral jerk, and all well within dynamic vehicle constraints, it could be assumed that even more time-critical scenarios could quite easily be tackled successfully by such driving algorithms. In non-critical situations, it appears to be much more difficult to give useful information regarding quality. Even though the exact same metrics can be quantified, the values can hardly be linked to what is 'good' and what is not. In critical scenarios, the safety metrics are directly linked to nearness to a crash, based on the critical thresholds. The comfort metrics vary significantly for each time budget and have been shown to be correlated to the subjective comfort rating. For non-critical scenarios however, it can hardly be said a *MTTC* of 17 seconds is safer than a 10 second *MTTC*. This can most likely be purely bestowed upon personal preference. This touches upon the next important point of this discussion; the quantification framework is less suitable to quantify take-over quality in non-critical scenarios. Up to which criticality-threshold (e.g. time budget) useful results can be interpreted would need to be examined. For the 5 and 7 second time budgets however, the quantification framework looks promising. It is the first method proposed that objectively quantifies take-over quality in a situation-specific manner. As an example; A high safety score could be comprised of critical *MTTC*, and low *MDLB* - but only in case the optimization algorithm yields a similar trajectory (which indicates an extremely critical scenario). The differentiation must clearly be made that the quality of the human driver take-over manoeuvre execution is quantified, not safety itself.

An improvement that could be made to the framework is to replace the minimum distance-to-overtaking vehicle by a time-to-collision type metric. As explained, the smallest MDOV values tend to occur when both vehicles are driving parallel (during overtaking), and small values do not necessarily indicate criticality. However, this would require a time-to-collision based on lateral velocity of the vehicle, rather than longitudinal velocity. Additionally, the collision intersect computation needs to be altered for such an application of the TTC metric. Another point of interest are the values used for normalization in the comfort parameter. These values are based on results from this particular driving simulator, and can not be generalized to other datasets. It is expected that different driving simulator, especially with different vehicle dynamics models, produce different acceleration and jerk-related values. This effect may be strengthened by the less-than-realistic force-feedback provided by driving simulators [15]. Therefore, it is advised to newly determine these normalization values in similar manner, but based on the dataset at hand, when using the framework for quantification. This issue would be of lesser importance when applied in conjunction with real vehicles instead of a simulator. Something that is highly recommended for the comfort parameter.

Admittedly, the framework needs validation with different data. Also, it would be higly interesting to compare quality quantification with other frameworks recently proposed, such as *TOPS* and *TOC-rating* [17][12]. The quantification values generated by the proposed framework are heavily dependent on the chosen quantification functions (linear, piecewise, or exponential) and accompanying thresholds.

These were not only based on literature, but also an attempt to include realistic human behaviour. Comparison to different take-over quality quantification frameworks could help improve and validate the quantification functions used in this method.

6

Conclusion

Many conclusions could be drawn purely based on the results from the human-in-the-loop study. Although the metrics themselves were not a focal point of this research, it is worth mentioning some of the outcomes here. *MTTC* was confirmed to have a strong negative correlation to subjective criticality ($p<0.01$), and a significant effect for the time budget condition was statistically shown. This corresponds to most recent literature regarding this topic. Another confirmation of previous studies can be found in the maximum absolute accelerations and maximum absolute jerks. For both lateral maximum acceleration and lateral maximum jerk, significant differences were found for the condition time budget. As lateral maximum acceleration and jerk can be related directly to vehicle stability through the tyre forces, the time budget is confirmed to be one of the main contributors to scenario criticality. This is strengthened by the medium and large correlation ($p<0.01$) to the subjective criticality rating for the beforementioned acceleration and jerk-metric, respectively. Their longitudinal counterparts also showed a significant difference for the traffic density condition, which can be explained due to the required braking action, especially in the 5s and 7s time budget scenarios. Due to sphericity and normality violations, no strong statistical conclusions can be drawn regarding the MDLB and Forced Braking metrics. However, their importance in driving safety can be related directly to crash events, as the largest number of occurred crashes was due to leaving the lane boundary ($N_{Crash-MDLB} = 8$), and one crash occurred due to a collision with the overtaking vehicle.

Regarding the comfort-related metrics, traditional views on acceleration and jerks were confirmed. Both the maximum and standard deviations of lateral and longitudinal acceleration and jerk exhibit medium to large correlations ($p<0.01$) to the subjective comfort rating. More surprisingly however, was lateral quickness, as it was one of two significant predictors ($p<0.01$) for subjective comfort based on a step-wise regression analysis. This indicates that it could be beneficial to research this metric in greater detail in the future, as it is still relatively new to the automotive sector.

In general, it can be concluded that first steps have been taken regarding a novel approach that utilizes an optimized quality baseline trajectory for take-over quality quantification purposes. Additionally, it can be said that the quantification framework is situation-specific in the sense that it takes into account the inherent criticality of a takeover scenario, as the criticality is represented in the optimized reference trajectory created by the CudaDP planner. However, it must be noted that this method is highly time-consuming due to the effort currently needed to prepare and run simulations using the CudaDP planner that are identical to those performed in a driving simulator study. Although the method itself is not limited to use in conjunction with the CudaDP planner alone. Use of other, more user-friendly, trajectory planners could make this method easier to use and thus more widely applicable.

In the future, method validation using different data is recommended. Additionally, improvements can be booked by implementing a time-to-collision metric with respect to other road users. In it's current form, the euclidean distance to an overtaking vehicle is used for the *MDOV* metric. This is difficult to relate directly to situation criticality as there is not necessarily a collision intersect, and velocities with which the two vehicles are 'closing in' on each other are neglected. Also, the normalization values

used in the comfort parameter may not be directly applicable for use with a different driving simulator. Real time applications, however far-away, are highly interesting. Theoretically, the CudaDP planner could run real-time. This implies the framework could also be used to quantify human driver takeover performance in real time. This information could potentially be used to detect driver fatigue, among other things. It is recommended to compare the take-over quantification scores of this method with the Take-over Performance Score (TOPS), as the methods share many similarities.

Bibliography

[1] L. Bainbridge. Ironies of Automation. 1982.

[2] H. Bellem, T. Schönenberg, Josef F. Krems, and M. Schrauf. Objective metrics of comfort: Developing a driving style for highly automated vehicles. *Transportation Research Part F: Traffic Psychology and Behaviour*, 41:45–54, 2016.

[3] R. Bellman and M. Kalaba. Dynamic Programming and Statistical Communication Theory. *Proceedings of the National Academy of Sciences of the United States of America*, 43:749–751, 1957.

[4] E. Dovgan, T. Tušar, M. Javorski, and B. Filipič. Discovering comfortable driving strategies using simulation-based multiobjective optimization. *Informatica (Slovenia)*, 36(3):319–326, 2012.

[5] C. Gold, D. Damböck, L. Lorenz, and K. Bengler. Take over! How long does it take to get the driver back into the loop? In *Proceedings of the Human Factors and Ergonomics Society*, 2013.

[6] R. Happee, C. Gold, J. Radlmayr, S. Hergeth, and K. Bengler. Take-over performance in evasive manoeuvres. *Accident Analysis and Prevention*, 2017.

[7] Mm Körber, C. Gold, D. Lechner, and K. Bengler. The influence of age on the take-over of vehicle control in highly automated driving. *Transportation Research Part F: Traffic Psychology and Behaviour*, 2016.

[8] J. Lee, J. Choi, K. Yi, M. Shin, and B. Ko. Lane-keeping assistance control algorithm using differential braking to prevent unintended lane departures. *Control Engineering Practice*, 23(1):1–13, 2 2014.

[9] Z. Lu, R. Happee, C. Cabrall, M. Kyriakidis, and JCF de Winter. Human factors of transitions in automated driving: A general framework and literature survey. *Transportation Research Part F: Traffic Psychology and Behaviour*, 2016.

[10] N. Merat, A. Jamson, F.C.H Lai, M. Daly, and O.M.J Carsten. Transition to manual: Driver behaviour when resuming control from a highly automated vehicle. *Transportation Research Part F: Traffic Psychology and Behaviour*, 2014.

[11] A.K. Nandi, D. Chakraborty, and W. Vaz. Design of a comfortable optimal driving strategy for electric vehicles using multi-objective optimization. *Journal of Power Sources*, 283:1–18, 6 2015.

[12] F. Naujoks, K. Wiedemann, N. Schömig, O. Jarosch, and C. Gold. Expert-based controllability assessment of control transitions from automated to manual driving. *Elsevier*, 5:579–592, 2018.

[13] A. Neukum, F. Naujoks, and C. Mai. The Effect of Urgency of Take-Over Requests During Highly Automated Driving Under Distraction Conditions. In *5th International Conference on Applied Human Factors and Ergonomics*, Krakow, 2014.

[14] R. Parasuraman, T. B. Sheridan, and C. D. Wickens. A Model for Types and Levels of Human Interaction with Automation. *SYSTEMS AND HUMANS*, 30(3), 2000.

[15] G. D. Park, R. W. Allen, T. J. Rosenthal, and D. Fiorentino. Training Effectiveness: How Does Driving Simulator Fidelity Influence Driver Performance? In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 49, pages 2201–2205. SAGE Publications, 1 2012.

[16] J. Radlmayr, C. Gold, L. Lorenz, M. Farid, and K. Bengler. How traffic situations and non-driving related tasks affect the take-over quality in highly automated driving. In *Proceedings of the Human Factors and Ergonomics Society*, 2014.

[17] J. Radlmayr, M. Ratter, A. Feldhütter, M. Körber, L. Prasch, J. Schmidtler, Y. Yang, and K. Bengler. Take-overs in level 3 automated driving – Proposal of the take-over performance score (TOPS). In *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018)*, pages 436–446, 2019.

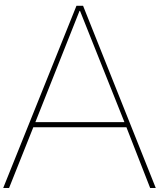
[18] A. Vahidi and A. Eskandarian. Research advances in intelligent collision avoidance and adaptive cruise control. *IEEE Transactions on Intelligent Transportation Systems*, 2003.

[19] R. Van Der Horst and J. Hogema. Time-to-Collision and Collision Avoidance Systems. In *6th ICTCT Workshop Salzburg Proceedings*, 1993.

[20] Tim Wescott. *Embedded Systems Programming*. 2000.

[21] K. Zeeb, A. Buchner, and M. Schrauf. What determines the take-over time? An integrated model approach of driver take-over after automated driving. *Accident Analysis and Prevention*, 2015.

[22] K. Zeeb, A. Buchner, and M. Schrauf. Is take-over time all that matters? the impact of visual-cognitive load on driver take-over quality after conditionally automated driving. *Accident Analysis and Prevention*, 2016.



Appendices

A.1. Simulator-study Related Documents

PORSCHE

INSTRUKTIONEN

Willkommen zu unserer Studie zum hochautomatisierten Fahren. Wir möchten uns herzlich bei Ihnen bedanken, dass Sie sich die Zeit genommen haben, uns zu unterstützen. Zu Beginn einige Informationen zum Ablauf und der Vorgehensweise. Wir bitten Sie, die Instruktionen sorgfältig durchzulesen. Bei Fragen wenden Sie sich jeder Zeit an den Versuchsleiter.

Zu Beginn ist ein demographischer Fragebogen auszufüllen, welcher grundlegende Fragen zu Ihnen enthält. Ferner ist eine Einverständniserklärung angehängt - wir bitten Sie diese zu unterschreiben.

Eine 5-minütige Trainingsfahrt dient zur Gewöhnung an die Fahrdynamik des Fahrsimulators. Das Fahrzeug ist mit einer hochautomatisierten Fahrfunktion ausgestattet. Das heißt, das Fahrzeug kann selbstständig bremsen, beschleunigen und die Spur halten. Sobald die Automation verfügbar ist, können Sie diese über einen Knopf am Lenkrad aktivieren. Bitte beachten Sie, dass die hochautomatisierte Funktion nur auf der rechten Fahrspur verfügbar ist. Der aktuelle Status der Automation wird durch ein Icon im oberen Bereich des Kombiinstruments dargestellt. Mögliche Ausprägungen und die entsprechenden Bedeutungen der Icons sind wie folgt:

Nicht verfügbar *Verfügbar* *Aktiv* *Übernahmeaufforderung*

Nach der Trainingsfahrt können noch offene Fragen geklärt werden. Sind alle Fragen ausgeräumt, folgt die Versuchsfahrt, welche auf einer 2-spurigen Autobahn sein wird. Zu Beginn ist der Autopilot noch nicht verfügbar. Bitte beschleunigen Sie auf eine Geschwindigkeit von 130 km/h und fahren Sie auf der rechten Fahrspur. Sobald der Autopilot verfügbar ist, bitten wir Sie, die Automation zu aktivieren. Nach der Aktivierung bewegt sich das Fahrzeug im automatisierten Modus mit einer Geschwindigkeit von 130 km/h. Während der hochautomatisierten Fahrt können Sie sich voll und ganz der Nebenaufgabe widmen. Als Nebenaufgabe steht eine Folge der US-Sitcom „Brooklyn-nine-nine auf dem Center Display zur Verfügung

Seien Sie sich jedoch bewusst, dass Sie stets der verantwortliche Fahrer bleiben. Die Automation kann nicht jede Situation meistern. Sobald sie an eine Systemgrenze stoßen, werden Sie visuell und auditiv aufgefordert die Fahrzeugführung zu übernehmen. Bitte beachten Sie, dass es Ihre oberste Priorität sein sollte, das Fahrzeug sicher und gemäß den Regeln der Straßenverkehrsordnung zu steuern.

Im Falle einer Übernahmeaufforderung, bitten wir Sie, die Automation wieder zu aktivieren sobald dies risikofrei möglich ist. Sollte eine derartige Situation auftreten, würden wir Sie bitten, nach der Reaktivierung der Automation, 4 Fragen auf dem Tablet rechts neben Ihnen zu beantworten. Nachdem Sie die Fragen beantwortet haben, können Sie sich wieder voll und ganz der Serie widmen. Im Anschluss an die ca. 30-minütige Versuchsfahrt, bitten wir Sie noch 3 weitere Fragen zu beantworten.

Vielen Dank

Figure A.1: The instruction form participants were handed out in preparation for the simulator study.

Figure A.2: The subjective rating questionnaire as filled out on the tablet, by the participants of the human-in-the-loop driving simulator study after each take-over scenario.

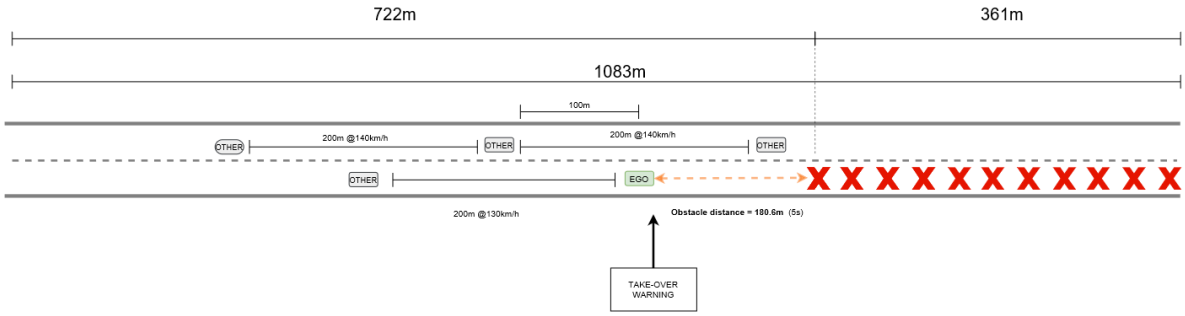


Figure A.3: The road layout including dimensions and traffic conditions for take-over scenario 2 [TB = 5s, TD = low].

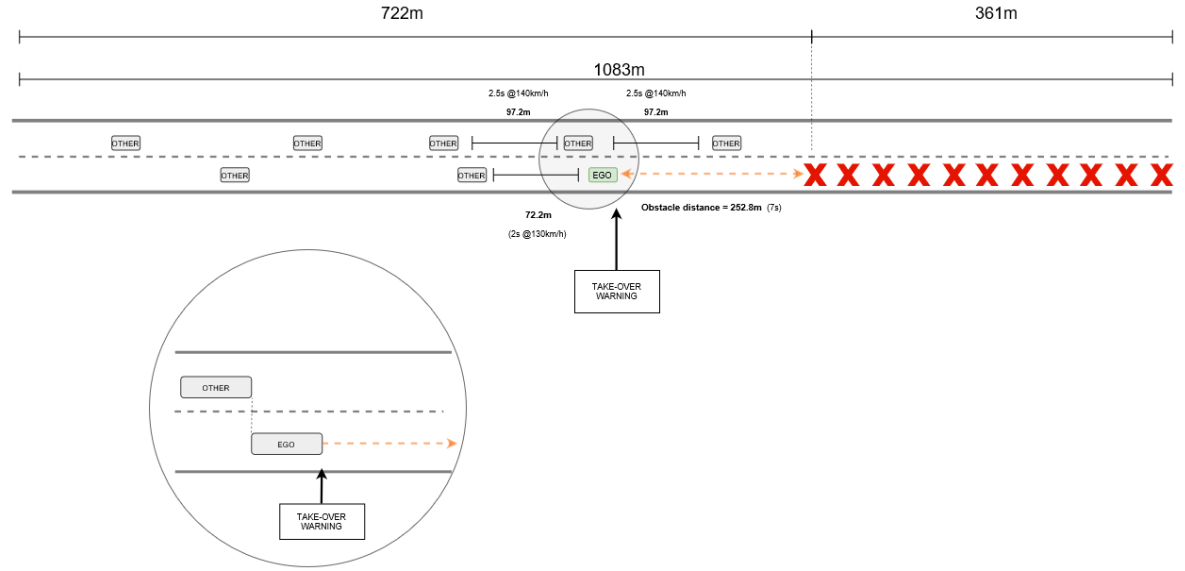


Figure A.4: The road layout including dimensions and traffic conditions for take-over scenario 3 [TB = 7s, TD = med.].

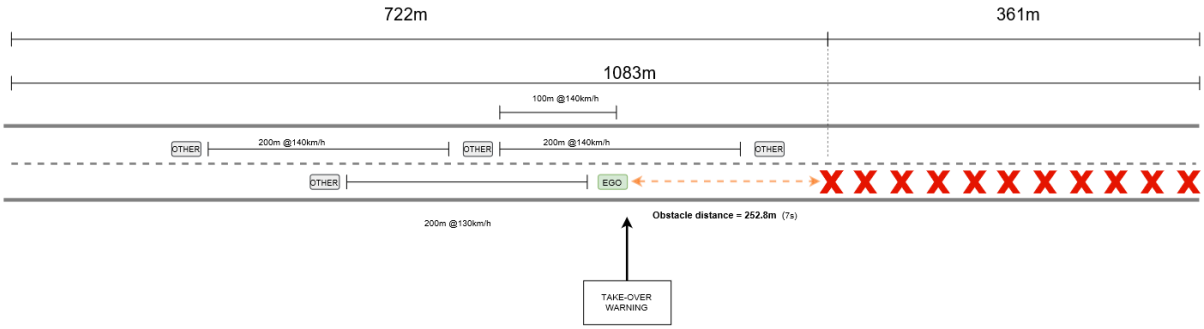


Figure A.5: The road layout including dimensions and traffic conditions for take-over scenario 4 [TB = 7s, TD = low].

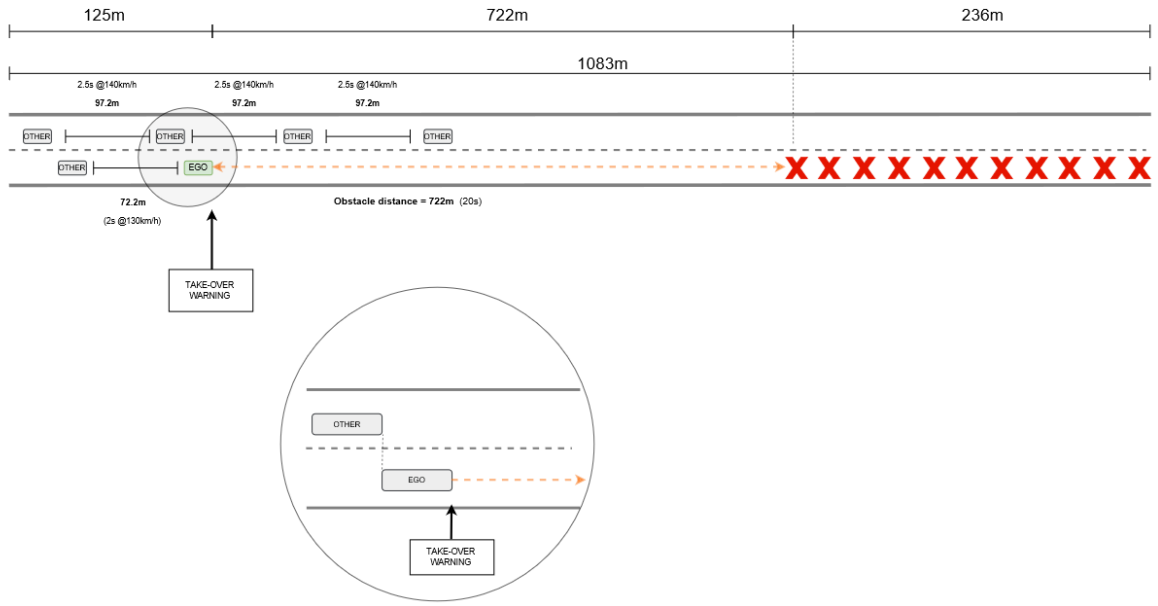


Figure A.6: The road layout including dimensions and traffic conditions for take-over scenario 5 [TB = 20s, TD = med.].

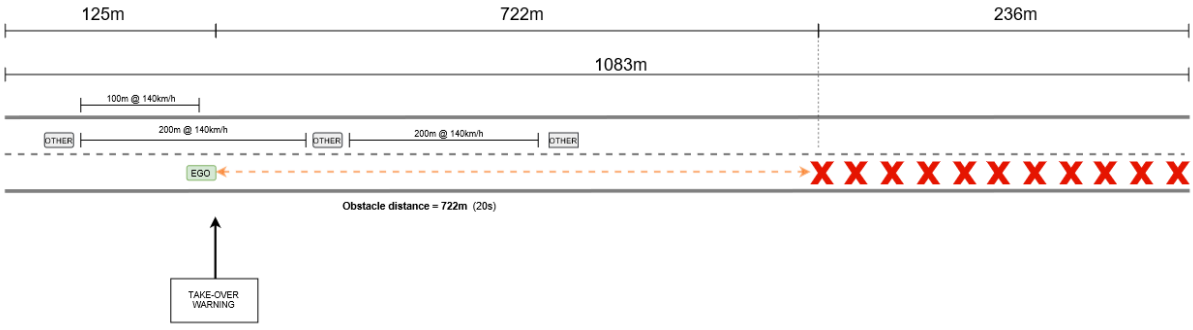


Figure A.7: The road layout including dimensions and traffic conditions for take-over scenario 6 [TB = 20s, TD = low].

A.2. Quantification Results

The following figures show the participant quantification scores for each metric included in the framework, both for the safety parameter as well as the comfort parameter. The scenario-averaged parameter scores can be seen in table 4.1.

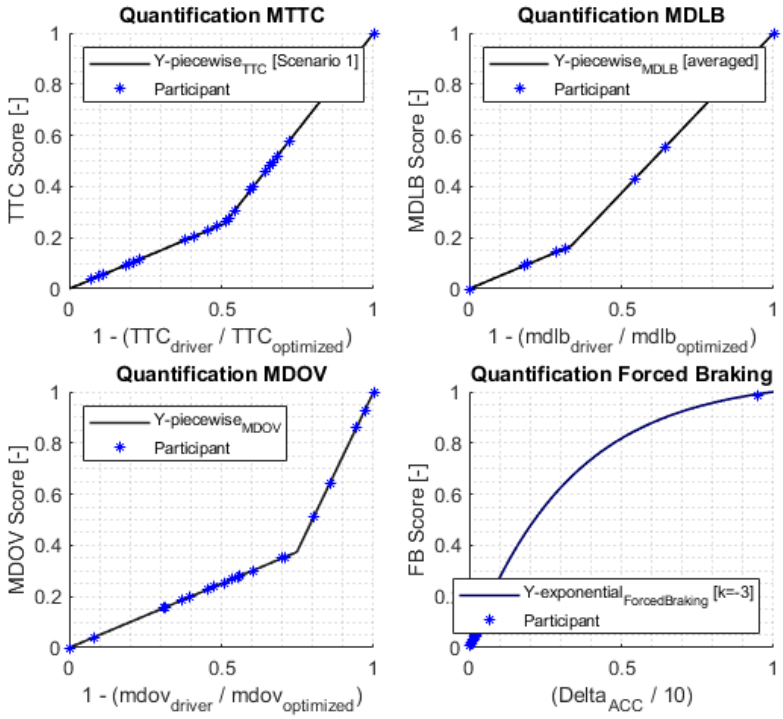


Figure A.8: Scenario 1: Safety metrics quantification for all participants (n=25)

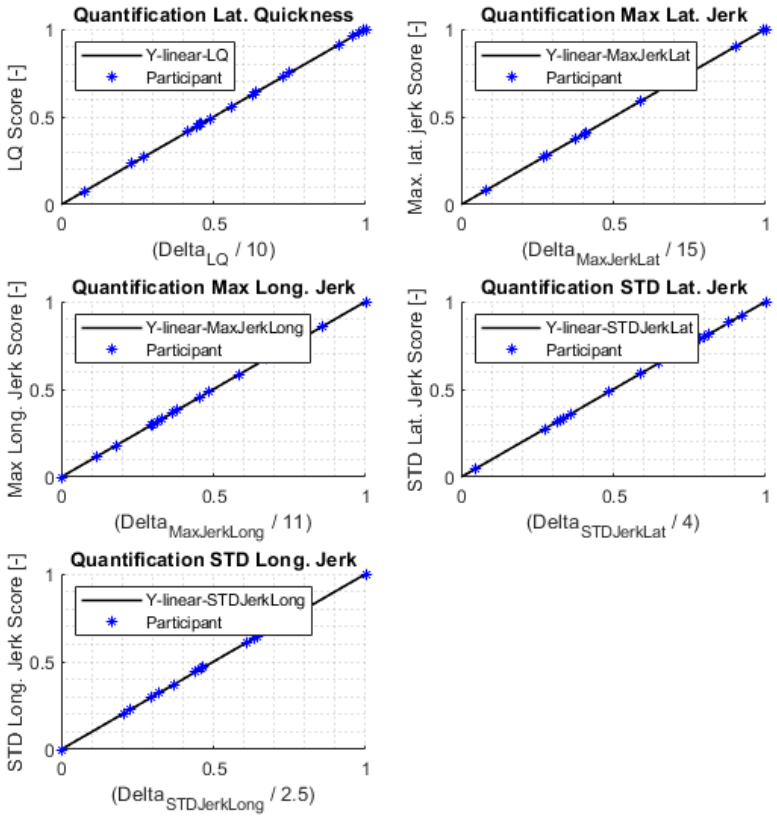


Figure A.9: Scenario 1:Comfort metrics quantification for all participants (n=25)

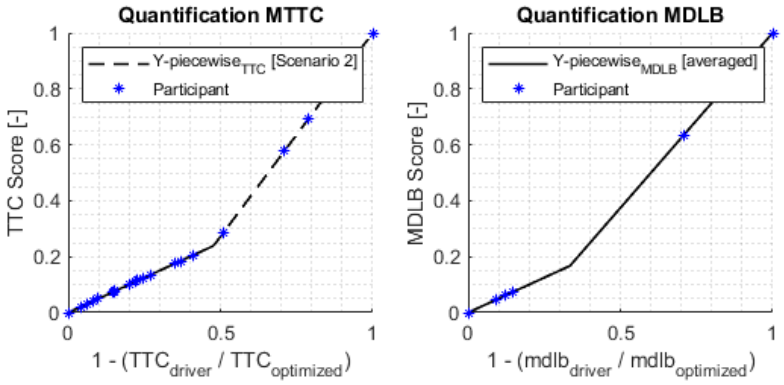


Figure A.10: Scenario 2: Safety metrics quantification for all participants (n=25)

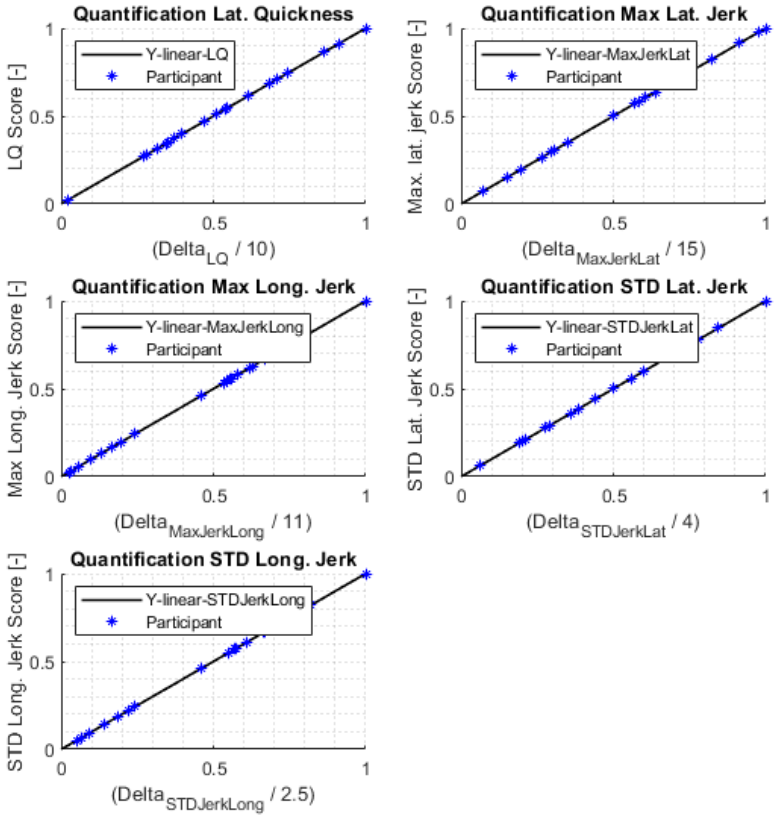


Figure A.11: Scenario 2: Comfort metrics quantification for all participants (n=25)

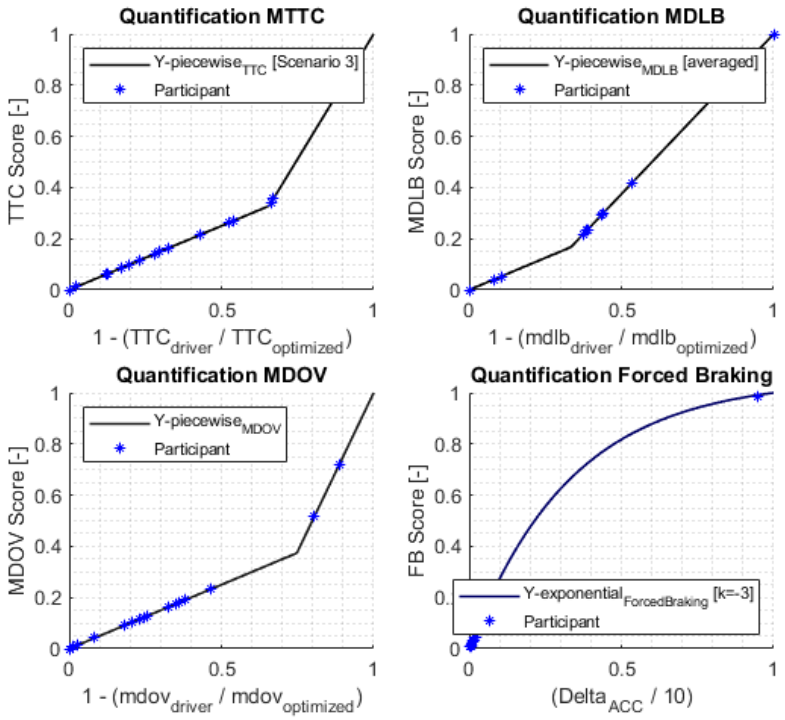


Figure A.12: Scenario 3: Safety metrics quantification for all participants (n=25)

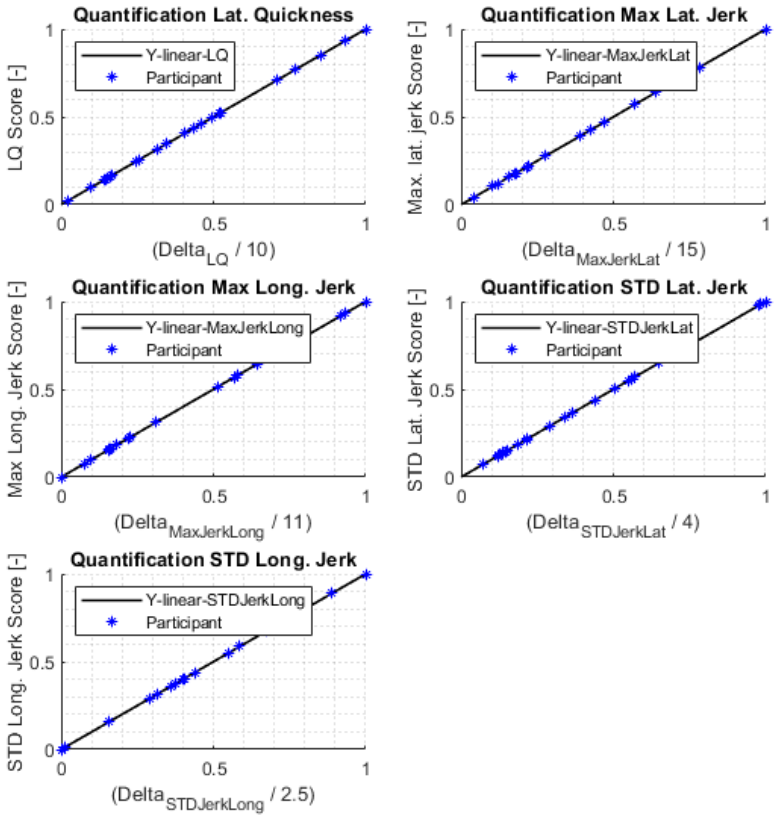


Figure A.13: Scenario 3: Comfort metrics quantification for all participants (n=25)

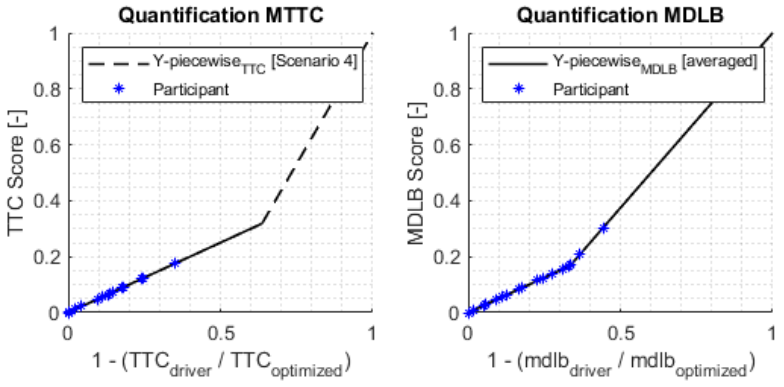


Figure A.14: Scenario 4: Safety metrics quantification for all participants (n=25)

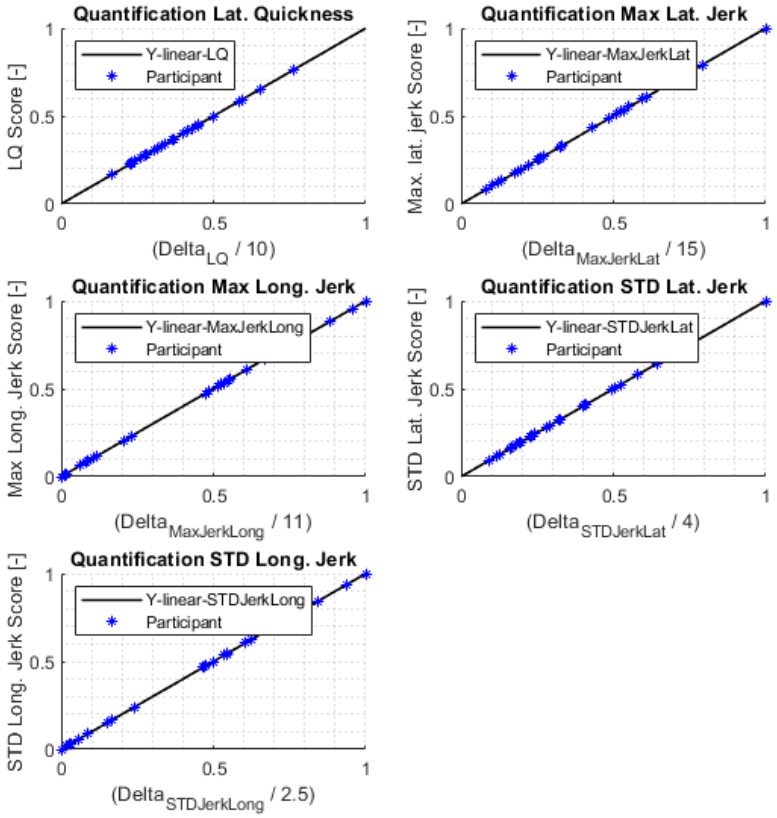


Figure A.15: Scenario 4: Comfort metrics quantification for all participants (n=25)

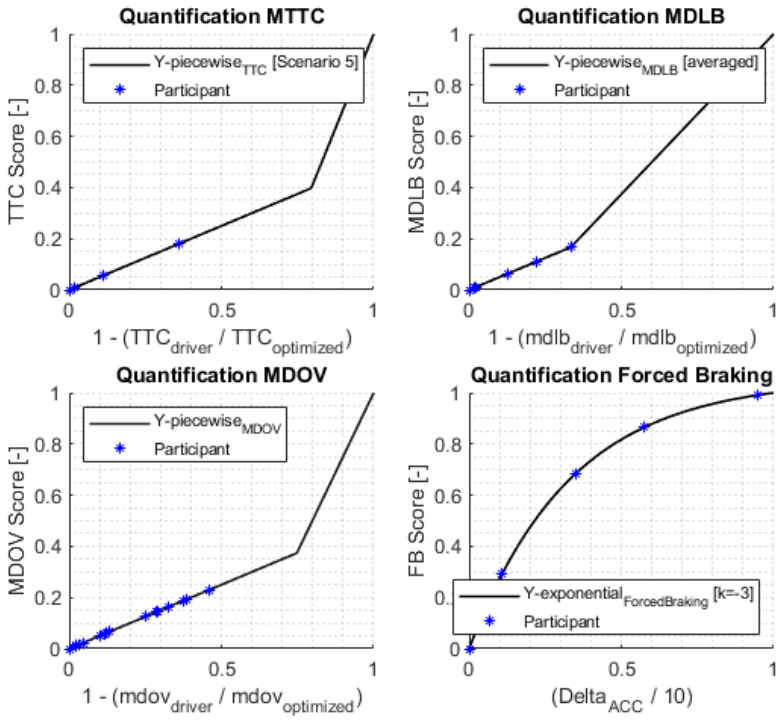


Figure A.16: Scenario 5: Safety metrics quantification for all participants (n=25)

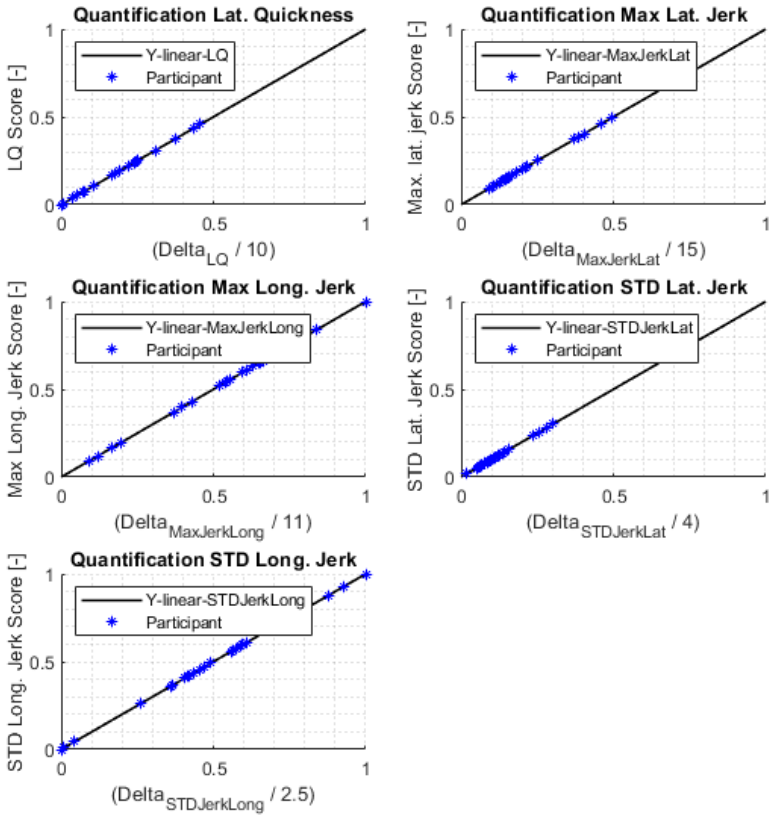


Figure A.17: Scenario 5: Comfort metrics quantification for all participants (n=25)

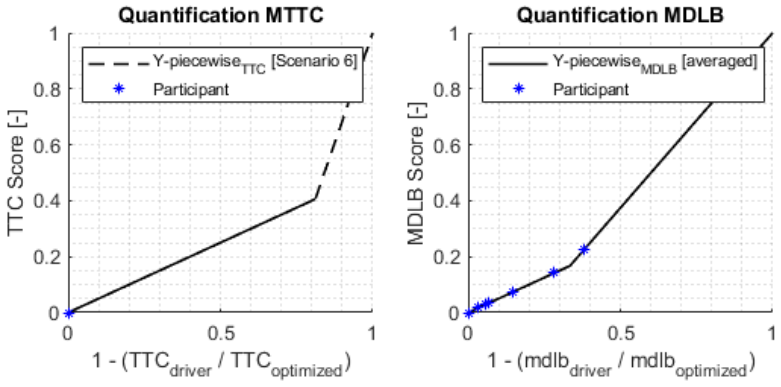


Figure A.18: Scenario 6: Safety metrics quantification for all participants (n=25)

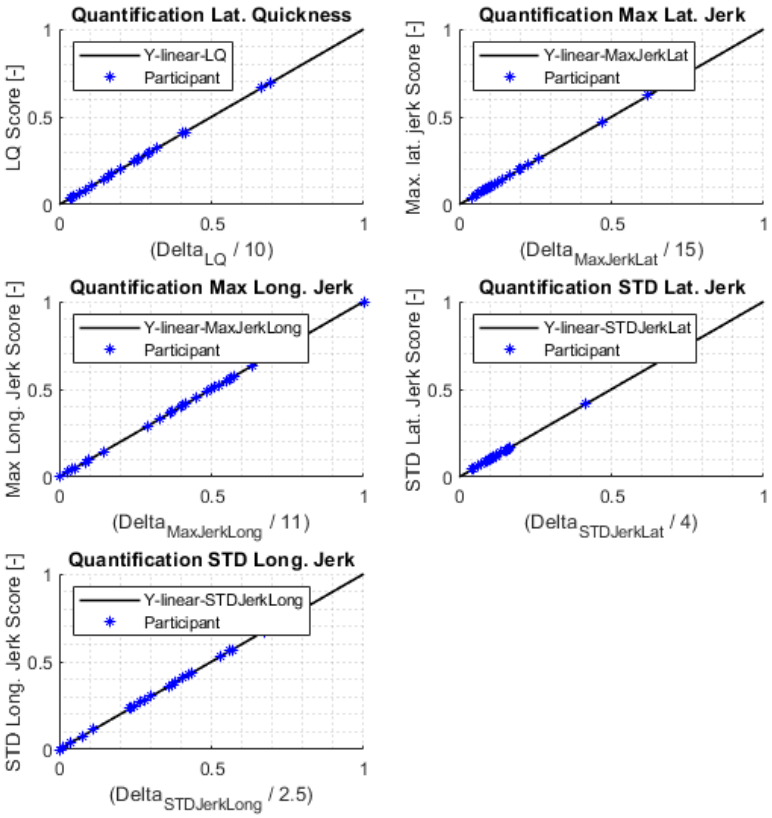


Figure A.19: Scenario 6: Comfort metrics quantification for all participants (n=25)

A.3. Matlab Script

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Erik Loosveld                                                       %%
%%  Porsche AG                                                          %%
%%                                                                      %%
%%  Trajectory optimization - Pre processing & analysis                 %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%% Clear all
clc
clear all
%close all

%% Choose Desired measures and plots
% Settings
scenario = 1;
cuda_id = 7;
participant_id = 1;


% Measures and plots
measures = 0;
cuda_trajectories = 0;
euclidean_plot = 0;
threeD_plot = 0;
TwoD_plot = 0;
all_participants = 0;
TTC_plot = 0;
velocity_gradient = 1;
animation_participant = 0;
animation_cuda = 0;
time_derivatives =0;
quantification = 0;
quickness =0 ;
output_cuda =0;
output = 0;


%% Load Cuda Data

switch scenario

    case 1
disp('Loading Scenario 1...')
[num, txt, raw] = xlsread('Ego_Vehicle_Scenario1');
[num2, txt2, raw2] = xlsread('Overtaker_Vehicle_Scenario1');
[num3, txt3, raw3] = xlsread('Follower_Vehicle_Scenario1');
lane_offset = 8.45;
%Takeover Warning
takeover1y = [1038.4 1038.4];
takeover1x = [-2 2];
%obstacle
obstacley = [1222 1222 1583 1583];
```

```
55 obstaclex = [-2, 2, 2 -2];
56
57 participant_id_vector = [2 6 7 8 9 10 11 12 13 14 15 17 18 20 21 22 23 24
       25 26 27 28 29 30 31];
58 str_id = cell(1,numel(participant_id_vector));
59
60 D = num;              %Ego data
61 O = num2;             %Overtaking vehicle data
62 F = num3;
63 ids_total = numel(find(D(:,1)==1));      % amount of different simulations
       %
64 ids = 1:ids_total;                    % IDS vector
65
66 clear num txt raw num2 txt2 raw2 num3 txt3 raw3
67
68 start_y = 938.4;
69 end_y = 1550;
70
71     case 2
72 disp('Loading Scenario 2...')
73 [num, txt, raw] = xlsread('Ego_Vehicle_Scenario2');
74 lane_offset = 2008.45;
75 %Takeover Warning
76 takeover1y = [1038.4 1038.4];
77 takeover1x = [-2 2];
78 %obstacle
79 obstacley = [1222 1222 1583 1583];
80 obstaclex = [-2, 2, 2 -2];
81
82 participant_id_vector = [2 6 7 8 9 10 11 12 13 14 15 18 20 21 22 23 24 25
       26 27 28 30 31];
83 %TN17 deleted due to lack of data
84 %TN29 deleted due to lack of data
85 str_id = cell(1,numel(participant_id_vector));
86
87 D = num;
88 ids_total = numel(find(D(:,1)==1));      % amount of different simulations
       %
89 ids = 1:ids_total;                    % IDS vector
90
91 clear num txt raw
92
93 start_y = 938.4;
94 end_y = 1550;
95
96     case 3
97 disp('Loading Scenario 3...')
98 [num, txt, raw] = xlsread('Ego_Vehicle_Scenario3');
99 [num2, txt2, raw2] = xlsread('Overtaker_Vehicle_Scenario3');
100 [num3, txt3, raw3] = xlsread('Follower_Vehicle_Scenario3');
101 lane_offset = 4008.45;
102 %Takeover Warning
103 takeover1y = [966.2 966.2];
104 takeover1x = [-2 2];
105 %obstacle
106 obstacley = [1222 1222 1583 1583];
```

```
107 obstaclex = [-2, 2, 2 -2];
108
109 participant_id_vector = [2 6 7 8 9 10 11 12 13 14 15 17 18 20 21 22 23 24
        25 26 27 28 29 30 31];
110 str_id = cell(1,numel(participant_id_vector));
111
112 D = num;              %Ego data
113 O = num2;             %Overtaking vehicle data
114 F = num3;
115 ids_total = numel(find(D(:,1)==1));      % amount of different simulations
        %
116 ids = 1:ids_total;                      % IDS vector
117
118 clear num txt raw num2 txt2 raw2 num3 txt3 raw3
119
120 start_y = 866.2;
121 end_y = 1550;
122
123     case 4
124 disp('Loading Scenario 4...')
125 [num, txt, raw] = xlsread('Ego_Vehicle_Scenario4');
126 lane_offset = 6008.45;
127 %Takeover Warning
128 takeover1y = [966.2 966.2];
129 takeover1x = [-2 2];
130 %obstacle
131 obstacley = [1222 1222 1583 1583];
132 obstaclex = [-2, 2, 2 -2];
133
134 participant_id_vector = [2 6 7 8 9 10 11 12 13 14 15 17 18 20 21 22 23 24
        25 26 27 28 29 30 31];
135 str_id = cell(1,numel(participant_id_vector));
136
137 D = num;
138 ids_total = numel(find(D(:,1)==1));      % amount of different simulations
        %
139 ids = 1:ids_total;                      % IDS vector
140
141 clear num txt raw
142
143 start_y = 866.2;
144 end_y = 1550;
145
146     case 5
147 disp('Loading Scenario 5...')
148 [num, txt, raw] = xlsread('Ego_Vehicle_Scenario5');
149 [num2, txt2, raw2] = xlsread('Overtaker_Vehicle_Scenario5');
150 [num3, txt3, raw3] = xlsread('Follower_Vehicle_Scenario5');
151 lane_offset = 8008.45;
152 %Takeover Warning
153 takeover1y = [622 622];
154 takeover1x = [-2 2];
155 %obstacle
156 obstacley = [1347 1347 1583 1583];
157 obstaclex = [-2, 2, 2 -2];
158
```

```
159 participant_id_vector = [2 6 7 8 9 10 11 12 13 14 15 17 18 20 21 22 23 24
        25 26 27 28 29 30 31];
160 str_id = cell(1,numel(participant_id_vector));
161
162 D = num;              %Ego data
163 O = num2;             %Overtaking vehicle data
164 F = num3;
165 ids_total = numel(find(D(:,1)==1));      % amount of different simulations
        %
166 ids = 1:ids_total;                      % IDS vector
167
168 clear num txt raw num2 txt2 raw2 num3 txt3 raw3
169
170 start_y = 522;
171 end_y = 1550;
172
173     case 6
174 disp('Loading Scenario 6...')
175 [num, txt, raw] = xlsread('Ego_Vehicle_Scenario6');
176 lane_offset = 10008.45;
177 %Takeover Warning
178 takeover1y = [622 622];
179 takeover1x = [-2 2];
180 %obstacle
181 obstacley = [1347 1347 1583 1583];
182 obstaclex = [-2, 2, 2 -2];
183
184 participant_id_vector = [2 6 7 8 9 10 11 12 13 14 15 17 18 20 21 22 23 24
        25 26 27 28 29 30 31];
185 str_id = cell(1,numel(participant_id_vector));
186
187 D = num;
188 ids_total = numel(find(D(:,1)==1));      % amount of different simulations
        %
189 ids = 1:ids_total;                      % IDS vector
190
191 clear num txt raw
192
193 start_y = 522;
194 end_y = 1550;
195
196     otherwise
197         disp('Non-existent scenario input')
198 end
199
200
201 %% Cuda Butterworth initialization
202
203 Fs_cuda = 25;
204 o_cuda = 3;
205 wn_cuda = [0.01 .4]*2/Fs_cuda;
206 [b_cuda,a_cuda] = butter(o_cuda,wn_cuda,'bandpass');
207
208
209 %% Creating Data struct for data storage
210 %Starting points for each simulation
```

```
211 starting_points = find(D(:,1)==0);
212 end_point = length(D(:,1));
213 starting_points = [starting_points' end_point];
214
215 switch scenario
216     case {1,3,5}
217
218 %Assigning specific simulation vehicle data to struct
219 for k = 1:(ids_total)
220
221     Data(k).cuda.raw = D(starting_points(k):starting_points(k+1)-2,:);
222     Data(k).cuda.raw_idstart = find(abs(Data(k).cuda.raw(:,5) - start_y) <
          1,1);
223     Data(k).cuda.raw_idend = find(abs(Data(k).cuda.raw(:,5) - end_y) <
        1,1);
224
225     Data(k).cuda.overtakerraw = O(starting_points(k):starting_points(k+1)
        -2,:);
226     Data(k).cuda.followerraw = F(starting_points(k):starting_points(k+1)
        -2,:);
227
228     Data(k).cuda.ego.posx = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
        (k).cuda.raw_idend,4)-lane_offset;
229     Data(k).cuda.ego.posx = - Data(k).cuda.ego.posx;
230     Data(k).cuda.ego.posy = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
        (k).cuda.raw_idend,5);
231     Data(k).cuda.ego.velx = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
        (k).cuda.raw_idend,13);
232     Data(k).cuda.ego.velx = - Data(k).cuda.ego.velx;
233     Data(k).cuda.ego.vely = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
        (k).cuda.raw_idend,14);
234     Data(k).cuda.ego.accx = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
        (k).cuda.raw_idend,22);
235     Data(k).cuda.ego.accx = - Data(k).cuda.ego.accx;
236     Data(k).cuda.ego.accx = filter(b_cuda, a_cuda,Data(k).cuda.ego.accx);
237     Data(k).cuda.ego.accy = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
        (k).cuda.raw_idend,23);
238     Data(k).cuda.ego.accy = filter(b_cuda,a_cuda,Data(k).cuda.ego.accy);
239     Data(k).cuda.ego.time = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
        (k).cuda.raw_idend,2);
240
241     Data(k).cuda.overtaker.posx = Data(k).cuda.overtakerraw(Data(k).cuda.
        raw_idstart:Data(k).cuda.raw_idend,4)-lane_offset;
242     Data(k).cuda.overtaker.posx = - Data(k).cuda.overtaker.posx;
243     Data(k).cuda.overtaker.posy = Data(k).cuda.overtakerraw(Data(k).cuda.
        raw_idstart:Data(k).cuda.raw_idend,5);
244     Data(k).cuda.overtaker.velx = Data(k).cuda.overtakerraw(Data(k).cuda.
        raw_idstart:Data(k).cuda.raw_idend,13);
245     Data(k).cuda.overtaker.velx = - Data(k).cuda.overtaker.velx;
246     Data(k).cuda.overtaker.vely = Data(k).cuda.overtakerraw(Data(k).cuda.
        raw_idstart:Data(k).cuda.raw_idend,14);
247     Data(k).cuda.overtaker.accx = Data(k).cuda.overtakerraw(Data(k).cuda.
        raw_idstart:Data(k).cuda.raw_idend,22);
248     Data(k).cuda.overtaker.accx = - Data(k).cuda.overtaker.accx;
249     Data(k).cuda.overtaker.accy = Data(k).cuda.overtakerraw(Data(k).cuda.
        raw_idstart:Data(k).cuda.raw_idend,23);
```

```
250     Data(k).cuda.overtaker.time = Data(k).cuda.overtakerraw(Data(k).cuda.
            raw_idstart:Data(k).cuda.raw_idend,2);
251
252     Data(k).cuda.follower.posx = Data(k).cuda.followerraw(Data(k).cuda.
            raw_idstart:Data(k).cuda.raw_idend,4)-lane_offset;
253     Data(k).cuda.follower.posx = - Data(k).cuda.follower.posx;
254     Data(k).cuda.follower.posy = Data(k).cuda.followerraw(Data(k).cuda.
            raw_idstart:Data(k).cuda.raw_idend,5);
255     Data(k).cuda.follower.velx = Data(k).cuda.followerraw(Data(k).cuda.
            raw_idstart:Data(k).cuda.raw_idend,13);
256     Data(k).cuda.follower.velx = - Data(k).cuda.follower.velx;
257     Data(k).cuda.follower.vely = Data(k).cuda.followerraw(Data(k).cuda.
            raw_idstart:Data(k).cuda.raw_idend,14);
258     Data(k).cuda.follower.accx = Data(k).cuda.followerraw(Data(k).cuda.
            raw_idstart:Data(k).cuda.raw_idend,22);
259     Data(k).cuda.follower.accx = - Data(k).cuda.follower.accx;
260     Data(k).cuda.follower.accy = Data(k).cuda.followerraw(Data(k).cuda.
            raw_idstart:Data(k).cuda.raw_idend,23);
261     Data(k).cuda.follower.time = Data(k).cuda.followerraw(Data(k).cuda.
            raw_idstart:Data(k).cuda.raw_idend,2);
262 end
263
264
265     case {2,4,6}
266         for k = 1:(ids_total)
267
268     Data(k).cuda.raw = D(starting_points(k):starting_points(k+1)-2,:);
269     Data(k).cuda.raw_idstart = find(abs(Data(k).cuda.raw(:,5) - start_y) <
            1,1);
270     Data(k).cuda.raw_idend = find(abs(Data(k).cuda.raw(:,5) - end_y) <
            1,1);
271
272     Data(k).cuda.ego.posx = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
            (k).cuda.raw_idend,4)-lane_offset;
273     Data(k).cuda.ego.posx = - Data(k).cuda.ego.posx;
274     Data(k).cuda.ego.posy = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
            (k).cuda.raw_idend,5);
275     Data(k).cuda.ego.velx = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
            (k).cuda.raw_idend,13);
276     Data(k).cuda.ego.velx = - Data(k).cuda.ego.velx;
277     Data(k).cuda.ego.vely = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
            (k).cuda.raw_idend,14);
278     Data(k).cuda.ego.accx = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
            (k).cuda.raw_idend,22);
279     Data(k).cuda.ego.accx = - Data(k).cuda.ego.accx;
280     Data(k).cuda.ego.accx = filter(b_cuda,a_cuda,Data(k).cuda.ego.accx);
281     Data(k).cuda.ego.accy = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
            (k).cuda.raw_idend,23);
282     Data(k).cuda.ego.accy = filter(b_cuda,a_cuda,Data(k).cuda.ego.accy);
283     Data(k).cuda.ego.time = Data(k).cuda.raw(Data(k).cuda.raw_idstart:Data
            (k).cuda.raw_idend,2);
284
285         end
286 end
287
288 % Cuda Standard deviation/mean/max of acceleration
```

```
289
290     Data(cuda_id).cuda.ego.STDaccy = std(Data(cuda_id).cuda.ego.accy);
291     Data(cuda_id).cuda.ego.STDaccx = std(Data(cuda_id).cuda.ego.accx);
292     Data(cuda_id).cuda.ego.mean_accy = mean(Data(cuda_id).cuda.ego.accy);
293     Data(cuda_id).cuda.ego.mean_accx = mean(Data(cuda_id).cuda.ego.accx);
294     [~,Data(cuda_id).cuda.ego.max_accy_id] = max(abs(Data(cuda_id).cuda.ego
            .accy));
295     Data(cuda_id).cuda.ego.max_accy = Data(cuda_id).cuda.ego.accy(Data(
            cuda_id).cuda.ego.max_accy_id);
296     [~,Data(cuda_id).cuda.ego.max_accx_id] = max(abs(Data(cuda_id).cuda.ego
            .accx));
297     Data(cuda_id).cuda.ego.max_accx = Data(cuda_id).cuda.ego.accx(Data(
            cuda_id).cuda.ego.max_accx_id);
298
299
300 %% Cuda butterworth frequency analysis
301 % DeltaT_cuda = 0.04;
302 %
303 % Fs_cuda = 25;
304 % length_signal = length(Data(cuda_id).cuda.ego.accy);
305 % NEFT = 2^nextpow2(length_signal);
306 % accy_fft = abs(fft(Data(cuda_id).cuda.ego.accy, NEFT));
307 % freq = Fs/2*linspace(0,1,NEFT/2+1);
308 % figure(1)
309 % subplot(2,2,1)
310 % plot(Data(cuda_id).cuda.ego.posy,Data(cuda_id).cuda.ego.accy)
311 % subplot(2,2,2);
312 % plot(freq,accy_fft(1:length(freq)))
313 %
314 % o_cuda = 3;
315 % wn_cuda = [0.01 .4]*2/Fs_cuda;
316 % [b_cuda,a_cuda] = butter(o_cuda,wn_cuda,'bandpass');
317 %
318 % accy_filtered = filter(b_cuda,a_cuda,Data(cuda_id).cuda.ego.accy);
319 % figure(2)
320 % subplot(2,2,1)
321 % plot(Data(cuda_id).cuda.ego.posy, accy_filtered)
322 % subplot(2,2,2)
323 % plot(Data(cuda_id).cuda.ego.posy,Data(cuda_id).cuda.ego.accy)
324 %
325 %
326 %    for z =2:length(Data(cuda_id).cuda.ego.accx)
327 %
328 %     cuda_jerky(z) = (accy_filtered(z) - accy_filtered(z-1))/(DeltaT_cuda
        );
329 %    end
330 % subplot(2,2,3)
331 % plot(Data(cuda_id).cuda.ego.posy,Data(cuda_id).cuda.ego.accy)
332 % subplot(2,2,4)
333 % plot(Data(cuda_id).cuda.ego.posy, cuda_jerky)
334
335
336 %% Load Simulator Data for specific scenario over all participants
337
338 switch scenario
339
```

```
340     case{1}
341  for i = participant_id_vector
342 filename = sprintf('Scenario1_TN(%d).xlsx',i);                %Change excel
    names here!
343 [num txt raw] = xlsread(filename);
344 %starting_point_participant = find(num(:,15) == 7, 1 );
345 starting_point_participant = find(abs(num(:,4) - start_y) < 1,1);
346 end_point_participant =  find(abs(num(:,4) - end_y) < 1,1);
347 num(:,44) = 1:length(num);  %add frames
348 %Determine end points of spreadsheets
349 %end_point_participant =  find(num(starting_point_participant:end,15) ~=
    7,1) + (starting_point_participant -2);
350 TF = isempty(end_point_participant);
351 if TF == 1
352     end_point_participant = max(find(num(starting_point_participant:end
        ,15))) + (starting_point_participant-1 );
353 else
354     %Do nothing
355 end
356
357 %load egovehicle data
358 time = datestr(num(starting_point_participant:end_point_participant,1),'MM
    :SS.FFF');
359 Data(i).participant.ego.time = time;
360 Data(i).participant.ego.posx = num(starting_point_participant:
    end_point_participant,3)-lane_offset;
361 Data(i).participant.ego.posx = - Data(i).participant.ego.posx;
362 Data(i).participant.ego.posy = num(starting_point_participant:
    end_point_participant,4);
363
364 Data(i).participant.ego.velx = num(starting_point_participant:
    end_point_participant,6);
365 Data(i).participant.ego.velx = - Data(i).participant.ego.velx;
366 Data(i).participant.ego.vely = num(starting_point_participant:
    end_point_participant,7);
367
368 Data(i).participant.ego.SWangle =  num(starting_point_participant:
    end_point_participant,12);
369 Data(i).participant.ego.BPpos =  num(starting_point_participant:
    end_point_participant,11);
370
371 %Aceleration was logged incorrectly --> Algebraic calculation follows
372 % Data(i).participant.ego.accx = num(starting_point_participant:
    end_point_participant,8);
373 % Data(i).participant.ego.accx = - Data(i).participant.ego.accx;
374 % Data(i).participant.ego.accy = num(starting_point_participant:
    end_point_participant,9);
375
376 %Load Other vehicles
377 CV_id1 = nonzeros(unique(num(starting_point_participant:
    end_point_participant,20)));
378 CV_id2 = nonzeros(unique(num(starting_point_participant:
    end_point_participant,28)));
379 CV_id3 = nonzeros(unique(num(starting_point_participant:
    end_point_participant,36)));
380 CV_ids = unique([CV_id1' CV_id2' CV_id3]');
```

```
381
382 for j = 1:length(CV_ids)
383 CV1 = num(num(:,20)==CV_ids(j),[20:27 44]);
384 CV2 = num(num(:,28)==CV_ids(j),[28:35 44]);
385 CV3 = num(num(:,36)==CV_ids(j),[36:43 44]);
386 CV = [CV1; CV2 ;CV3];
387 CV = sortrows(CV,9);
388
389 Data(i).participant.CV(j).posx = CV(:,[2 9]);
390 Data(i).participant.CV(j).posx(:,1) = - Data(i).participant.CV(j).posx
    (:,1)+lane_offset;
391 Data(i).participant.CV(j).posy = CV(:,[3 9]);
392 Data(i).participant.CV(j).vely = CV(:,[5 9]);
393 Data(i).participant.CV(j).distance = CV(:,[8 9]);
394 Data(i).participant.CV(j).accy = CV(:,[7 9]);
395 Data(i).participant.CV(j).starting_point_cv = find(Data(i).participant.CV(
    j).posy(:,2) == starting_point_participant);
396 Data(i).participant.CV(j).end_point_cv = find(Data(i).participant.CV(j).
    posy(:,2) == end_point_participant);
397
398         if ~isempty(Data(i).participant.CV(j).starting_point_cv) && ~
               isempty(Data(i).participant.CV(j).end_point_cv)
399
400
401          disp('Additional vehicle Data correctly imported')
402          Data(i).participant.CV(j).posy = Data(i).participant.CV(j).posy(
                Data(i).participant.CV(j).starting_point_cv:Data(i).
                participant.CV(j).end_point_cv,1);
403          Data(i).participant.CV(j).posx = Data(i).participant.CV(j).posx(
                Data(i).participant.CV(j).starting_point_cv:Data(i).
                participant.CV(j).end_point_cv,1);
404          Data(i).participant.CV(j).vely = Data(i).participant.CV(j).vely(
                Data(i).participant.CV(j).starting_point_cv:Data(i).
                participant.CV(j).end_point_cv,1);
405          Data(i).participant.CV(j).distance = Data(i).participant.CV(j).
                distance(Data(i).participant.CV(j).starting_point_cv:Data(i).
                participant.CV(j).end_point_cv,1);
406          Data(i).participant.CV(j).accy = Data(i).participant.CV(j).accy(
                Data(i).participant.CV(j).starting_point_cv:Data(i).
                participant.CV(j).end_point_cv,1);
407
408         elseif ~isempty(Data(i).participant.CV(j).starting_point_cv) &&
               isempty(Data(i).participant.CV(j).end_point_cv)
409
410          max_id = max(Data(i).participant.CV(j).posy(:,2));
411          Data(i).participant.CV(j).end_point_cv = find(Data(i).participant
                .CV(j).posy(:,2) ==  max_id);
412
413
414
415         str_id{i} = ['End points were corrected for participant ID ',
               num2str(i)];
416         disp(str_id{i})
417
418
```

```
419             Data(i).participant.CV(j).posy = Data(i).participant.CV(j).posy(
                    Data(i).participant.CV(j).starting_point_cv:Data(i).
                    participant.CV(j).end_point_cv,1);
420             Data(i).participant.CV(j).posx = Data(i).participant.CV(j).posx(
                    Data(i).participant.CV(j).starting_point_cv:Data(i).
                    participant.CV(j).end_point_cv,1);
421             Data(i).participant.CV(j).vely = Data(i).participant.CV(j).vely(
                    Data(i).participant.CV(j).starting_point_cv:Data(i).
                    participant.CV(j).end_point_cv,1);
422             Data(i).participant.CV(j).distance = Data(i).participant.CV(j).
                    distance(Data(i).participant.CV(j).starting_point_cv:Data(i).
                    participant.CV(j).end_point_cv,1);
423             Data(i).participant.CV(j).accy = Data(i).participant.CV(j).accy(
                    Data(i).participant.CV(j).starting_point_cv:Data(i).
                    participant.CV(j).end_point_cv,1);
424         else
425
426         str_id{i} = ['No starting point detected for participant ID ',
                num2str(i)];
427         disp(str_id{i})
428         Data(i).participant.CV(j).posx = [];
429         Data(i).participant.CV(j).posy = [];
430         Data(i).participant.CV(j).vely =[];
431         Data(i).participant.CV(j).distance=[];
432         Data(i).participant.CV(j).accy = [];
433         end
434
435 end
436
437     if size(Data(i).participant.CV) == [1 1]
438     Data(i).participant.overtaker.posx = Data(i).participant.CV.posx;
439     Data(i).participant.overtaker.posy = Data(i).participant.CV.posy;
440     Data(i).participant.overtaker.vely = Data(i).participant.CV.vely;
441     Data(i).participant.overtaker.distance = Data(i).participant.CV.
            distance;
442     Data(i).participant.overtaker.accy = Data(i).participant.CV.accy;
443     else
444     Data(i).participant.overtaker.posx = Data(i).participant.CV(2).posx;
445     Data(i).participant.overtaker.posy = Data(i).participant.CV(2).posy;
446     Data(i).participant.overtaker.vely = Data(i).participant.CV(2).vely;
447     Data(i).participant.overtaker.distance = Data(i).participant.CV(2).
            distance;
448     Data(i).participant.overtaker.accy = Data(i).participant.CV(2).accy;
449
450     end
451
452  end
453
454
455
456
457  case{2}
458       for i = participant_id_vector
459
460 filename = sprintf('Scenario2_TN(%d).xlsx',i);              %Change excel
    names here!
```

```
461 [num txt raw] = xlsread(filename);
462
463
464 starting_point_participant = find(abs(num(:,4) - start_y) < 1,1);
465 end_point_participant =  find(abs(num(:,4) - end_y) < 1,1);
466
467
468 TF = isempty(end_point_participant);
469 if TF == 1
470     end_point_participant = max(find(num(starting_point_participant:end
        ,15))) + (starting_point_participant -1);
471 else
472     %Do nothing
473 end
474
475 time = datestr(num(starting_point_participant:end_point_participant,1),'MM
    :SS.FFF');
476 Data(i).participant.ego.time = time;
477 Data(i).participant.ego.posx = num(starting_point_participant:
    end_point_participant,3)-lane_offset;
478 Data(i).participant.ego.posx = - Data(i).participant.ego.posx;
479 Data(i).participant.ego.posy = num(starting_point_participant:
    end_point_participant,4);
480
481 Data(i).participant.ego.velx = num(starting_point_participant:
    end_point_participant,6);
482 Data(i).participant.ego.velx = - Data(i).participant.ego.velx;
483 Data(i).participant.ego.vely = num(starting_point_participant:
    end_point_participant,7);
484
485 Data(i).participant.ego.SWangle =  num(starting_point_participant:
    end_point_participant,12);
486 Data(i).participant.ego.BPpos =  num(starting_point_participant:
    end_point_participant,11);
487 % Acceleration was logged incorrectly --> Algebraic calculation follows
488 % Data(i).participant.ego.accx = num(starting_point_participant:
    end_point_participant,8);
489 % Data(i).participant.ego.accx = - Data(i).participant.ego.accx;
490 % Data(i).participant.ego.accy = num(starting_point_participant:
    end_point_participant,9);
491
492
493       end
494
495
496  case{3}
497       for i = participant_id_vector
498    filename = sprintf('Scenario3_TN(%d).xlsx',i);              %Change
        excel names here!
499 [num txt raw] = xlsread(filename);
500
501 starting_point_participant = find(abs(num(:,4) - start_y) < 1,1);
502 end_point_participant =  find(abs(num(:,4) - end_y) < 1,1);
503 num(:,44) = 1:length(num);  %add frames
504 TF = isempty(end_point_participant);
505 if TF == 1
```

```
506     end_point_participant = max(find(num(starting_point_participant:end
            ,15))) + (starting_point_participant-1);
507 else
508     %Do nothing
509 end
510 time = datestr(num(starting_point_participant:end_point_participant,1),'MM
        :SS.FFF');
511 Data(i).participant.ego.time = time;
512 Data(i).participant.ego.posx = num(starting_point_participant:
        end_point_participant,3)-lane_offset;
513 Data(i).participant.ego.posx = - Data(i).participant.ego.posx;
514 Data(i).participant.ego.posy = num(starting_point_participant:
        end_point_participant,4);
515
516 Data(i).participant.ego.velx = num(starting_point_participant:
        end_point_participant,6);
517 Data(i).participant.ego.velx = - Data(i).participant.ego.velx;
518 Data(i).participant.ego.vely = num(starting_point_participant:
        end_point_participant,7);
519
520 Data(i).participant.ego.SWangle =  num(starting_point_participant:
        end_point_participant,12);
521 Data(i).participant.ego.BPpos =  num(starting_point_participant:
        end_point_participant,11);
522 %Acceleration was logged incorrectly --> Algebraic calculation follows
523 % Data(i).participant.ego.accx = num(starting_point_participant:
        end_point_participant,8);
524 % Data(i).participant.ego.accx = - Data(i).participant.ego.accx;
525 % Data(i).participant.ego.accy = num(starting_point_participant:
        end_point_participant,9);
526
527
528 %Load Other vehicles
529 CV_id1 = nonzeros(unique(num(starting_point_participant:
        end_point_participant,20)));
530 CV_id2 = nonzeros(unique(num(starting_point_participant:
        end_point_participant,28)));
531 CV_id3 = nonzeros(unique(num(starting_point_participant:
        end_point_participant,36)));
532 CV_ids = unique([CV_id1' CV_id2' CV_id3']');
533
534 for j = 1:length(CV_ids)
535 CV1 = num(num(:,20)==CV_ids(j),[20:27 44]);
536 CV2 = num(num(:,28)==CV_ids(j),[28:35 44]);
537 CV3 = num(num(:,36)==CV_ids(j),[36:43 44]);
538 CV = [CV1; CV2 ;CV3];
539 CV = sortrows(CV,9);
540
541 Data(i).participant.CV(j).posx = CV(:,[2 9]);
542 Data(i).participant.CV(j).posx(:,1) = - Data(i).participant.CV(j).posx
        (:,1)+lane_offset;
543 Data(i).participant.CV(j).posy = CV(:,[3 9]);
544 Data(i).participant.CV(j).vely = CV(:,[5 9]);
545 Data(i).participant.CV(j).accy = CV(:,[7 9]);
546 Data(i).participant.CV(j).distance = CV(:,[8 9]);
```

```
547 Data(i).participant.CV(j).starting_point_cv = find(Data(i).participant.CV(
        j).posy(:,2) == starting_point_participant);
548 Data(i).participant.CV(j).end_point_cv = find(Data(i).participant.CV(j).
        posy(:,2) == end_point_participant);
549
550         if ~isempty(Data(i).participant.CV(j).starting_point_cv) && ~
                isempty(Data(i).participant.CV(j).end_point_cv)
551          disp('Additional vehicle Data correctly imported')
552          Data(i).participant.CV(j).posy = Data(i).participant.CV(j).posy(
                 Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
553          Data(i).participant.CV(j).posx = Data(i).participant.CV(j).posx(
                 Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
554          Data(i).participant.CV(j).accy = Data(i).participant.CV(j).accy(
                 Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
555          Data(i).participant.CV(j).vely = Data(i).participant.CV(j).vely(
                 Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
556             Data(i).participant.CV(j).distance = Data(i).participant.CV(j)
                    .distance(Data(i).participant.CV(j).starting_point_cv:Data(
                    i).participant.CV(j).end_point_cv,1);
557
558         elseif ~isempty(Data(i).participant.CV(j).starting_point_cv) &&
                isempty(Data(i).participant.CV(j).end_point_cv)
559
560          max_id = max(Data(i).participant.CV(j).posy(:,2));
561          Data(i).participant.CV(j).end_point_cv = find(Data(i).participant
                 .CV(j).posy(:,2) ==  max_id);
562
563          str_id{i} = ['End points were corrected for participant ID ',
                 num2str(i)];
564           disp(str_id{i})
565
566          Data(i).participant.CV(j).posy = Data(i).participant.CV(j).posy(
                 Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
567          Data(i).participant.CV(j).posx = Data(i).participant.CV(j).posx(
                 Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
568          Data(i).participant.CV(j).vely = Data(i).participant.CV(j).vely(
                 Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
569          Data(i).participant.CV(j).distance = Data(i).participant.CV(j).
                 distance(Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
570         Data(i).participant.CV(j).accy = Data(i).participant.CV(j).accy(
                Data(i).participant.CV(j).starting_point_cv:Data(i).participant
                .CV(j).end_point_cv,1);
571         else
572         str_id{i} = ['No starting point detected for participant ID ',
                num2str(i)];
573         disp(str_id{i})
574
575         Data(i).participant.CV(j).posx = [];
```

```
576         Data(i).participant.CV(j).posy = [];
577         Data(i).participant.CV(j).vely =[];
578         Data(i).participant.CV(j).distance =[];
579         Data(i).participant.CV(j).accy = [];
580         end
581
582 end
583
584     if size(Data(i).participant.CV) == [1 1]
585     Data(i).participant.overtaker.posx = Data(i).participant.CV.posx;
586     Data(i).participant.overtaker.posy = Data(i).participant.CV.posy;
587     Data(i).participant.overtaker.vely = Data(i).participant.CV.vely;
588     Data(i).participant.overtaker.distance = Data(i).participant.CV.
            distance;
589     Data(i).participant.overtaker.accy = Data(i).participant.CV.accy;
590
591     else
592     Data(i).participant.overtaker.posx = Data(i).participant.CV(2).posx;
593     Data(i).participant.overtaker.posy = Data(i).participant.CV(2).posy;
594     Data(i).participant.overtaker.vely = Data(i).participant.CV(2).vely;
595     Data(i).participant.overtaker.distance = Data(i).participant.CV(2).
            distance;
596     Data(i).participant.overtaker.accy = Data(i).participant.CV(2).accy;
597
598     end
599
600
601       end
602
603 case{4}
604       for i = participant_id_vector
605    filename = sprintf('Scenario4_TN(%d).xlsx',i);              %Change
          excel names here!
606 [num txt raw] = xlsread(filename);
607
608 starting_point_participant = find(abs(num(:,4) - start_y) < 1,1);
609 end_point_participant =  find(abs(num(:,4) - end_y) < 1,1);
610
611
612 TF = isempty(end_point_participant);
613 if TF == 1
614     end_point_participant = max(find(num(starting_point_participant:end
            ,15))) + (starting_point_participant-1);
615 else
616     %Do nothing
617 end
618 time = datestr(num(starting_point_participant:end_point_participant,1),'MM
        :SS.FFF');
619 Data(i).participant.ego.time = time;
620 Data(i).participant.ego.posx = num(starting_point_participant:
        end_point_participant,3)-lane_offset;
621 Data(i).participant.ego.posx = - Data(i).participant.ego.posx;
622 Data(i).participant.ego.posy = num(starting_point_participant:
        end_point_participant,4);
623
```

```
624 Data(i).participant.ego.velx = num(starting_point_participant:
    end_point_participant,6);
625 Data(i).participant.ego.velx = - Data(i).participant.ego.velx;
626 Data(i).participant.ego.vely = num(starting_point_participant:
    end_point_participant,7);
627 Data(i).participant.ego.SWangle =  num(starting_point_participant:
    end_point_participant,12);
628 Data(i).participant.ego.BPpos =  num(starting_point_participant:
    end_point_participant,11);
629 %Lateral acceleration was logged incorrectly --> Algebraic calculation
    follows
630 % Data(i).participant.ego.accx = num(starting_point_participant:
    end_point_participant,8);
631 % Data(i).participant.ego.accx = - Data(i).participant.ego.accx;
632 % Data(i).participant.ego.accy = num(starting_point_participant:
    end_point_participant,9);
633
634
635       end
636
637 case{5}
638       for i = participant_id_vector
639    filename = sprintf('Scenario5_TN(%d).xlsx',i);              %Change
       excel names here!
640 [num txt raw] = xlsread(filename);
641
642 starting_point_participant = find(abs(num(:,4) - start_y) < 1,1);
643 end_point_participant =  find(abs(num(:,4) - end_y) < 1,1);
644 num(:,44) = 1:length(num);  %add frames
645
646 TF = isempty(end_point_participant);
647 if TF == 1
648     end_point_participant = max(find(num(starting_point_participant:end
       ,15))) + (starting_point_participant-1);
649 else
650     %Do nothing
651 end
652 time = datestr(num(starting_point_participant:end_point_participant,1),'MM
    :SS.FFF');
653 Data(i).participant.ego.time = time;
654 Data(i).participant.ego.posx = num(starting_point_participant:
    end_point_participant,3)-lane_offset;
655 Data(i).participant.ego.posx = - Data(i).participant.ego.posx;
656 Data(i).participant.ego.posy = num(starting_point_participant:
    end_point_participant,4);
657
658 Data(i).participant.ego.velx = num(starting_point_participant:
    end_point_participant,6);
659 Data(i).participant.ego.velx = - Data(i).participant.ego.velx;
660 Data(i).participant.ego.vely = num(starting_point_participant:
    end_point_participant,7);
661 Data(i).participant.ego.SWangle =  num(starting_point_participant:
    end_point_participant,12);
662 Data(i).participant.ego.BPpos =  num(starting_point_participant:
    end_point_participant,11);
663 % Acceleration was logged incorrectly --> Algebraic calculation follows
```

```
664 % Data(i).participant.ego.accx = num(starting_point_participant:
        end_point_participant,8);
665 % Data(i).participant.ego.accx = - Data(i).participant.ego.accx;
666 % Data(i).participant.ego.accy = num(starting_point_participant:
        end_point_participant,9);
667
668
669 %Load Other vehicles
670 CV_id1 = nonzeros(unique(num(starting_point_participant:
        end_point_participant,20)));
671 CV_id2 = nonzeros(unique(num(starting_point_participant:
        end_point_participant,28)));
672 CV_id3 = nonzeros(unique(num(starting_point_participant:
        end_point_participant,36)));
673 CV_ids = unique([CV_id1' CV_id2' CV_id3']');
674
675 for j = 1:length(CV_ids)
676 CV1 = num(num(:,20)==CV_ids(j),[20:27 44]);
677 CV2 = num(num(:,28)==CV_ids(j),[28:35 44]);
678 CV3 = num(num(:,36)==CV_ids(j),[36:43 44]);
679 CV = [CV1; CV2 ;CV3];
680 CV = sortrows(CV,9);
681
682 Data(i).participant.CV(j).posx = CV(:,[2 9]);
683 Data(i).participant.CV(j).posx(:,1) = - Data(i).participant.CV(j).posx
        (:,1)+lane_offset;
684 Data(i).participant.CV(j).posy = CV(:,[3 9]);
685 Data(i).participant.CV(j).vely = CV(:,[5 9]);
686 Data(i).participant.CV(j).accy = CV(:,[7 9]);
687 Data(i).participant.CV(j).distance = CV(:,[8 9]);
688 Data(i).participant.CV(j).starting_point_cv = find(Data(i).participant.CV(
        j).posy(:,2) == starting_point_participant);
689 Data(i).participant.CV(j).end_point_cv = find(Data(i).participant.CV(j).
        posy(:,2) == end_point_participant);
690
691         if ~isempty(Data(i).participant.CV(j).starting_point_cv) && ~
              isempty(Data(i).participant.CV(j).end_point_cv)
692          disp('Additional vehicle Data correctly imported')
693          Data(i).participant.CV(j).posy = Data(i).participant.CV(j).posy(
                 Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
694          Data(i).participant.CV(j).posx = Data(i).participant.CV(j).posx(
                 Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
695          Data(i).participant.CV(j).vely = Data(i).participant.CV(j).vely(
                 Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
696          Data(i).participant.CV(j).accy = Data(i).participant.CV(j).accy(
                 Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
697          Data(i).participant.CV(j).distance = Data(i).participant.CV(j).
                 distance(Data(i).participant.CV(j).starting_point_cv:Data(i).
                 participant.CV(j).end_point_cv,1);
698
699         elseif ~isempty(Data(i).participant.CV(j).starting_point_cv) &&
              isempty(Data(i).participant.CV(j).end_point_cv)
```

```
700
701             max_id = max(Data(i).participant.CV(j).posy(:,2));
702             Data(i).participant.CV(j).end_point_cv = find(Data(i).participant
                    .CV(j).posy(:,2) ==  max_id);
703             str_id{i} = ['End points were corrected for participant ID ',
                    num2str(i)];
704             disp(str_id{i})
705
706             Data(i).participant.CV(j).posy = Data(i).participant.CV(j).posy(
                    Data(i).participant.CV(j).starting_point_cv:Data(i).
                    participant.CV(j).end_point_cv,1);
707             Data(i).participant.CV(j).posx = Data(i).participant.CV(j).posx(
                    Data(i).participant.CV(j).starting_point_cv:Data(i).
                    participant.CV(j).end_point_cv,1);
708             Data(i).participant.CV(j).vely = Data(i).participant.CV(j).vely(
                    Data(i).participant.CV(j).starting_point_cv:Data(i).
                    participant.CV(j).end_point_cv,1);
709             Data(i).participant.CV(j).distance = Data(i).participant.CV(j).
                    distance(Data(i).participant.CV(j).starting_point_cv:Data(i).
                    participant.CV(j).end_point_cv,1);
710           Data(i).participant.CV(j).accy = Data(i).participant.CV(j).accy(
                  Data(i).participant.CV(j).starting_point_cv:Data(i).participant
                  .CV(j).end_point_cv,1);
711           else
712           str_id{i} = ['No starting point detected for participant ID ',
                  num2str(i)];
713           disp(str_id{i})
714           Data(i).participant.CV(j).posx = [];
715           Data(i).participant.CV(j).posy = [];
716           Data(i).participant.CV(j).vely =[];
717           Data(i).participant.CV(j).distance =[];
718           Data(i).participant.CV(j).accy = [];
719           end
720
721 end
722
723       if size(Data(i).participant.CV) == [1 1]
724       Data(i).participant.overtaker.posx = Data(i).participant.CV.posx;
725       Data(i).participant.overtaker.posy = Data(i).participant.CV.posy;
726       Data(i).participant.overtaker.vely = Data(i).participant.CV.vely;
727       Data(i).participant.overtaker.distance = Data(i).participant.CV.
              distance;
728       Data(i).participant.overtaker.accy = Data(i).participant.CV.accy;
729
730       else
731       Data(i).participant.overtaker.posx = Data(i).participant.CV(2).posx;
732       Data(i).participant.overtaker.posy = Data(i).participant.CV(2).posy;
733       Data(i).participant.overtaker.vely = Data(i).participant.CV(2).vely;
734       Data(i).participant.overtaker.distance = Data(i).participant.CV(2).
              distance;
735       Data(i).participant.overtaker.accy = Data(i).participant.CV(2).accy;
736
737       end
738         end
739
740 case{6}
```

```
741         for i = participant_id_vector
742     filename = sprintf('Scenario6_TN(%d).xlsx',i);                %Change
            excel names here!
743 [num txt raw] = xlsread(filename);
744
745
746 starting_point_participant = find(abs(num(:,4) - start_y) < 1,1);
747 end_point_participant =  find(abs(num(:,4) - end_y) < 1,1);
748
749 TF = isempty(end_point_participant);
750 if TF == 1
751     end_point_participant = max(find(num(starting_point_participant:end
            ,15))) + (starting_point_participant-1);
752 else
753     %Do nothing
754 end
755 time = datestr(num(starting_point_participant:end_point_participant,1),'MM
    :SS.FFF');
756 Data(i).participant.ego.time = time;
757 Data(i).participant.ego.posx = num(starting_point_participant:
    end_point_participant,3)-lane_offset;
758 Data(i).participant.ego.posx = - Data(i).participant.ego.posx;
759 Data(i).participant.ego.posy = num(starting_point_participant:
    end_point_participant,4);
760
761 Data(i).participant.ego.velx = num(starting_point_participant:
    end_point_participant,6);
762 Data(i).participant.ego.velx = - Data(i).participant.ego.velx;
763 Data(i).participant.ego.vely = num(starting_point_participant:
    end_point_participant,7);
764 Data(i).participant.ego.SWangle =  num(starting_point_participant:
    end_point_participant,12);
765 Data(i).participant.ego.BPpos =  num(starting_point_participant:
    end_point_participant,11);
766 %Acceleration was logged incorrectly --> Algebraic calculation follows
767 % Data(i).participant.ego.accx = num(starting_point_participant:
    end_point_participant,8);
768 % Data(i).participant.ego.accx = - Data(i).participant.ego.accx;
769 % Data(i).participant.ego.accy = num(starting_point_participant:
    end_point_participant,9);
770
771         end
772 end
773 %% Butterworth Filter initialization
774 Fs = 100;
775 o = 3;
776 wn = [0.01  .7]*2/Fs;
777 [b,a] = butter(o,wn,'bandpass');
778
779
780 %% Calculate Ego-Acceleration analytically
781 DeltaT_participant = 0.01;
782
783 for i = participant_id_vector
784
785     ii = find(Data(i).participant.ego.posy(:,1 ) - takeover1y(1) > 0,1);
```

```
786
787     for z =2:length(Data(i).participant.ego.velx)
788
789     Data(i).participant.ego.accx(z) = (Data(i).participant.ego.velx(z) -
            Data(i).participant.ego.velx(z-1))/(DeltaT_participant);
790     Data(i).participant.ego.accy(z) = (Data(i).participant.ego.vely(z) -
            Data(i).participant.ego.vely(z-1))/(DeltaT_participant);
791
792     end
793
794      Data(i).participant.ego.accx = filter(b,a,Data(i).participant.ego.
            accx);
795      Data(i).participant.ego.accy = filter(b,a,Data(i).participant.ego.
            accy);
796
797      Data(i).participant.ego.STDaccx = std(Data(i).participant.ego.accx(ii
            :end));
798      Data(i).participant.ego.STDaccy = std(Data(i).participant.ego.accy(ii
            :end));
799      Data(i).participant.ego.mean_accx = mean(Data(i).participant.ego.accx
            (ii:end));
800      Data(i).participant.ego.mean_accy = mean(Data(i).participant.ego.accy
            (ii:end));
801
802      [~,Data(i).participant.ego.max_accx_id] = max(abs(Data(i).participant
            .ego.accx));
803      Data(i).participant.ego.max_accx = Data(i).participant.ego.accx(Data(
            i).participant.ego.max_accx_id);
804      [~,Data(i).participant.ego.max_accy_id] = max(abs(Data(i).participant
            .ego.accy));
805      Data(i).participant.ego.max_accy = Data(i).participant.ego.accy(Data(
            i).participant.ego.max_accy_id);
806
807
808 end
809
810 clear z ii
811
812 %% Filter Design (use for troubleshooting purposes)
813
814 % Fs = 100;
815 % length_signal = length(Data(participant_id).participant.ego.accy);
816 % NEFT = 2^nextpow2(length_signal);
817 % accy_fft = abs(fft(Data(participant_id).participant.ego.accy, NEFT));
818 % freq = Fs/2*linspace(0,1,NEFT/2+1);
819 % figure(1)
820 % subplot(2,2,1)
821 % plot(Data(participant_id).participant.ego.posy,Data(participant_id).
      participant.ego.accy)
822 % subplot(2,2,2);
823 % plot(freq,accy_fft(1:length(freq)))
824 %
825 % o = 3;
826 % wn = [0.01 .7]*2/Fs;
827 % [b,a] = butter(o,wn,'bandpass');
828 %
```

```
% accy_filtered = filter(b,a,Data(participant_id).participant.ego.accy);
% figure(2)
% subplot(2,2,1)
% plot(Data(participant_id).participant.ego.posy, accy_filtered)
% subplot(2,2,2)
% plot(Data(participant_id).participant.ego.posy,Data(participant_id).
    participant.ego.accy)

%% Calculate Jerk analytically
%Participants

DeltaT_participant = 0.01;
DeltaT_cuda = 0.04;

for i = participant_id_vector
    ii = find(Data(i).participant.ego.posy(:,1 ) - takeover1y(1) > 0,1);
    for z =2:length(Data(i).participant.ego.accx)

    Data(i).participant.ego.jerk_lat(z) = (Data(i).participant.ego.accx(z)
        - Data(i).participant.ego.accx(z-1))/(DeltaT_participant);
    Data(i).participant.ego.jerk_long(z) = (Data(i).participant.ego.accy(z
        ) - Data(i).participant.ego.accy(z-1))/(DeltaT_participant);
    end
    Data(i).participant.ego.STDjerk_lat = std(Data(i).participant.ego.
        jerk_lat(ii:end));
    Data(i).participant.ego.STDjerk_long = std(Data(i).participant.ego.
        jerk_long(ii:end));
    Data(i).participant.ego.meanjerk_lat = mean(Data(i).participant.ego.
        jerk_lat(ii:end));
    Data(i).participant.ego.meanjerk_long = mean(Data(i).participant.ego.
        jerk_long(ii:end));

     [~,Data(i).participant.ego.max_jerk_long_id] = max(abs(Data(i).
         participant.ego.jerk_long));
     Data(i).participant.ego.max_jerk_long = Data(i).participant.ego.
         jerk_long(Data(i).participant.ego.max_jerk_long_id);
     [~,Data(i).participant.ego.max_jerk_lat_id] = max(abs(Data(i).
         participant.ego.jerk_lat));
     Data(i).participant.ego.max_jerk_lat = Data(i).participant.ego.
         jerk_lat(Data(i).participant.ego.max_jerk_lat_id);

end

clear z ii

   for z =2:length(Data(cuda_id).cuda.ego.accx)
    Data(cuda_id).cuda.ego.jerk_lat(z) = (Data(cuda_id).cuda.ego.accx(z) -
        Data(cuda_id).cuda.ego.accx(z-1))/(DeltaT_cuda);
    Data(cuda_id).cuda.ego.jerk_long(z) = (Data(cuda_id).cuda.ego.accy(z)
       - Data(cuda_id).cuda.ego.accy(z-1))/(DeltaT_cuda);
   end

   Data(cuda_id).cuda.ego.STDjerk_long = std(Data(cuda_id).cuda.ego.
       jerk_long);
   Data(cuda_id).cuda.ego.STDjerk_lat = std(Data(cuda_id).cuda.ego.
       jerk_lat);
```

```
870     Data(cuda_id).cuda.ego.mean_jerk_lat = mean(Data(cuda_id).cuda.ego.
            jerk_lat);
871     Data(cuda_id).cuda.ego.mean_jerk_long = mean(Data(cuda_id).cuda.ego.
            jerk_long);
872     [~,Data(cuda_id).cuda.ego.max_jerk_long_id] = max(abs(Data(cuda_id).
            cuda.ego.jerk_long));
873     Data(cuda_id).cuda.ego.max_jerk_long = Data(cuda_id).cuda.ego.jerk_long
            (Data(cuda_id).cuda.ego.max_jerk_long_id);
874     [~,Data(cuda_id).cuda.ego.max_jerk_lat_id] = max(abs(Data(cuda_id).cuda
            .ego.jerk_lat));
875     Data(cuda_id).cuda.ego.max_jerk_lat = Data(cuda_id).cuda.ego.jerk_lat(
            Data(cuda_id).cuda.ego.max_jerk_lat_id);
876
877 clear z
878
879
880 %% Define Roadlanes
881 %road
882 roady1 = [0 3000];
883 roadx1 = [-2 -2];
884 roady2 = [0 3000];
885 roadx2 = [6 6];
886 centerliney = [0 3000];
887 centerlinex = [2 2];
888
889 %% Calculate Quickness metric
890
891 cuda_velocity_threshold = 35.8;
892 cuda_pushback = 0;
893
894 % Cuda
895 Data(cuda_id).cuda.ego.measures.quickness =    max(Data(cuda_id).cuda.ego.
        velx) / max(Data(cuda_id).cuda.ego.posx);
896 Data(cuda_id).cuda.ego.measures.Long_quickness_start = find(Data(cuda_id).
        cuda.ego.vely(:,1 ) - cuda_velocity_threshold < 0,1);
897 Data(cuda_id).cuda.ego.measures.Long_quickness_start = Data(cuda_id).cuda.
        ego.measures.Long_quickness_start - cuda_pushback;
898 Data(cuda_id).cuda.ego.measures.Long_quickness_end = find(Data(cuda_id).
        cuda.ego.vely == min(Data(cuda_id).cuda.ego.vely));
899 Data(cuda_id).cuda.ego.measures.Long_quickness_DeltaV = Data(cuda_id).cuda
        .ego.vely(Data(cuda_id).cuda.ego.measures.Long_quickness_start) - Data(
        cuda_id).cuda.ego.vely(Data(cuda_id).cuda.ego.measures.
        Long_quickness_end);
900 Data(cuda_id).cuda.ego.measures.Long_quickness_MeanAcc = mean(Data(cuda_id
        ).cuda.ego.accy(Data(cuda_id).cuda.ego.measures.Long_quickness_start:
        Data(cuda_id).cuda.ego.measures.Long_quickness_end));
901 Data(cuda_id).cuda.ego.measures.Long_quickness = Data(cuda_id).cuda.ego.
        measures.Long_quickness_MeanAcc / Data(cuda_id).cuda.ego.measures.
        Long_quickness_DeltaV;
902
903 % Participant
904 %Lateral
905 start_id_quickness = 1;
906 search_distance =500;
907 angle_threshold = 0.05;
908 pushback_distance = 0 ;
```

```
909
910 %Longitudinal
911 BP_threshold = 0.1;
912 search_distance2 = 400;
913 pushback_distance2 = 0;
914 for i = participant_id_vector
915
916     %Lateral
917     Data(i).participant.ego.measures.Lat_quickness_start = find(Data(i).
            participant.ego.SWangle(start_id_quickness:end,1 ) -
            angle_threshold > 0,1);
918     Data(i).participant.ego.measures.Lat_quickness_start = Data(i).
            participant.ego.measures.Lat_quickness_start - pushback_distance +
            start_id_quickness;
919     Data(i).participant.ego.measures.Lat_quickness_end = find(Data(i).
            participant.ego.posx == max(Data(i).participant.ego.posx(Data(i).
            participant.ego.measures.Lat_quickness_start: Data(i).participant.
            ego.measures.Lat_quickness_start+search_distance)),1);
920
921     Data(i).participant.ego.measures.quickness_dlat = (Data(i).participant
            .ego.posx(Data(i).participant.ego.measures.Lat_quickness_end) -
            Data(i).participant.ego.posx(Data(i).participant.ego.measures.
            Lat_quickness_start));
922     [Data(i).participant.ego.measures.Lat_quickness_velocity ,Data(i).
            participant.ego.measures.Lat_quickness_velocity_id] = max(Data(i).
            participant.ego.velx(Data(i).participant.ego.measures.
            Lat_quickness_start:Data(i).participant.ego.measures.
            Lat_quickness_end));
923     Data(i).participant.ego.measures.Lat_quickness_velocity_id = Data(i).
            participant.ego.measures.Lat_quickness_velocity_id + Data(i).
            participant.ego.measures.Lat_quickness_start;
924     Data(i).participant.ego.measures.Lat_quickness =  Data(i).participant.
            ego.measures.Lat_quickness_velocity / Data(i).participant.ego.
            measures.quickness_dlat;
925
926     %Longitudinal
927     Data(i).participant.ego.measures.Long_quickness_start = find(Data(i).
            participant.ego.BPpos(start_id_quickness:end,1 ) - BP_threshold >
            0,1);
928     Data(i).participant.ego.measures.Long_quickness_start = Data(i).
            participant.ego.measures.Long_quickness_start - pushback_distance2
            + start_id_quickness;
929     Data(i).participant.ego.measures.Long_quickness_end = find(Data(i).
            participant.ego.vely == min(Data(i).participant.ego.vely(Data(i).
            participant.ego.measures.Lat_quickness_start: Data(i).participant.
            ego.measures.Lat_quickness_start+search_distance2)),1);
930
931     Data(i).participant.ego.measures.quickness_deltaV = Data(i).
            participant.ego.vely(Data(i).participant.ego.measures.
            Long_quickness_start) - Data(i).participant.ego.vely(Data(i).
            participant.ego.measures.Long_quickness_end);
932     Data(i).participant.ego.measures.quickness_long_MeanAcc = mean(Data(i)
            .participant.ego.accy(Data(i).participant.ego.measures.
            Long_quickness_start:Data(i).participant.ego.measures.
            Long_quickness_end));
933
```

```
934     Data(i).participant.ego.measures.Long_quickness = Data(i).participant.
            ego.measures.quickness_long_MeanAcc / Data(i).participant.ego.
            measures.quickness_deltaV;
935
936 end
937
938
939 switch quickness
940     case 1
941 disp(['The Quickness D-lat for this participant is ' num2str(Data(
        participant_id).participant.ego.measures.quickness_dlat) ' m'])
942 disp(['The Quickness D-lat for the Cuda is ' num2str(max(Data(cuda_id).
        cuda.ego.posx)) ' m'])
943 disp(['The Lateral Quickness Velocity for this participant is ' num2str(
        Data(participant_id).participant.ego.measures.Lat_quickness_velocity) '
         m/s'])
944 disp(['The Lateral Quickness Velocity for the cuda is ' num2str(max(Data(
        cuda_id).cuda.ego.velx)) ' m/s'])
945 disp(['The Lateral Quickness for this participant is ' num2str(Data(
        participant_id).participant.ego.measures.Lat_quickness) ' [1/s]'])
946 disp(['The Lateral Quickness for the cuda is ' num2str(Data(cuda_id).cuda
        .ego.measures.quickness) ' [1/s]'])
947
948 disp(['The Delta Velocity longitduinal for this participant is ' num2str(
        Data(participant_id).participant.ego.measures.quickness_deltaV) ' m/s'
        ])
949 disp(['The Delta Velocity longitduinal for the Cuda is ' num2str(Data(
        cuda_id).cuda.ego.measures.Long_quickness_DeltaV) ' m/s'])
950 disp(['The Mean acceleration the participant is ' num2str(Data(
        participant_id).participant.ego.measures.quickness_long_MeanAcc) ' m/s2
        '])
951 disp(['The Mean acceleration the Cuda is ' num2str(Data(cuda_id).cuda.ego.
        measures.Long_quickness_MeanAcc) ' m/s2'])
952 disp(['The Longitudinal Quickness for this participant is ' num2str(Data(
        participant_id).participant.ego.measures.Long_quickness) ' [1/s]'])
953 disp(['The Longitudinal Quickness for the Cuda is ' num2str(Data(cuda_id)
        .cuda.ego.measures.Long_quickness) ' [1/s]'])
954
955 figure
956 set(gcf, 'Position', [0, 400, 800, 400])
957 subplot(2,1,1)
958 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
        participant.ego.velx);
959 hold on
960 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.velx);
961 plot(Data(participant_id).participant.ego.posy(Data(participant_id).
        participant.ego.measures.Lat_quickness_velocity_id),Data(participant_id
        ).participant.ego.measures.Lat_quickness_velocity,'k*','markersize',5)
962 grid minor
963 legend('Participant','Cuda','Participant Maximum')
964 xlim([start_y end_y]);
965 ylabel('Lateral velocity [m]')
966 xlabel('Y-pos [m]')
967 title('Lateral Velocity for Lateral Quickness Computation')
968
969 subplot(2,1,2)
```

```
970 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
        participant.ego.posx);
971 hold on
972 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.posx);
973 grid minor
974 h=fill(obstacley,obstaclex,'black','HandleVisibility','off');
975 h.FaceAlpha=0.05;
976 plot(Data(participant_id).participant.ego.posy(Data(participant_id).
        participant.ego.measures.Lat_quickness_start), Data(participant_id).
        participant.ego.posx(Data(participant_id).participant.ego.measures.
        Lat_quickness_start),'k*','markersize',5)
977 plot(Data(participant_id).participant.ego.posy(Data(participant_id).
        participant.ego.measures.Lat_quickness_end), Data(participant_id).
        participant.ego.posx(Data(participant_id).participant.ego.measures.
        Lat_quickness_end),'k*','markersize',5)
978 line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
        ,2)
979 line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
980 line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
        HandleVisibility','off','linewidth',1.5)
981 line(takeover1y,takeover1x, 'color', 'red','LineStyle','--','linewidth'
        ,1.5)
982 title('Posititon')
983 legend('Participant', 'Cuda','Quickness cut-off points')
984 xlim([start_y end_y]);
985 ylabel('X-pos [m]')
986 xlabel('Y-pos [m]')
987 title('Lateral Offset Cut-off points for Lateral Quickness')
988
989 % Longitudinal
990 figure
991 set(gcf, 'Position', [800, 400, 800, 400])
992 subplot(2,1,1)
993 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
        participant.ego.vely);
994 hold on
995 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.vely);
996 plot(Data(participant_id).participant.ego.posy(Data(participant_id).
        participant.ego.measures.Long_quickness_start),Data(participant_id).
        participant.ego.vely(Data(participant_id).participant.ego.measures.
        Long_quickness_start),'k*','markersize',5);
997 plot(Data(participant_id).participant.ego.posy(Data(participant_id).
        participant.ego.measures.Long_quickness_end),Data(participant_id).
        participant.ego.vely(Data(participant_id).participant.ego.measures.
        Long_quickness_end),'k*','markersize',5);
998
999 plot(Data(cuda_id).cuda.ego.posy(Data(cuda_id).cuda.ego.measures.
        Long_quickness_start),Data(cuda_id).cuda.ego.vely(Data(cuda_id).cuda.
        ego.measures.Long_quickness_start),'k*','markersize',5);
1000 plot(Data(cuda_id).cuda.ego.posy(Data(cuda_id).cuda.ego.measures.
        Long_quickness_end),Data(cuda_id).cuda.ego.vely(Data(cuda_id).cuda.ego.
        measures.Long_quickness_end),'k*','markersize',5);
1001
1002
1003 grid minor
1004 legend('Participant','Cuda','Velocity cutoff points')
```

```
1005 xlim([start_y end_y]);
1006 ylabel('Longitudinal velocity [m]')
1007 xlabel('Y-pos [m]')
1008 title('Delta Velocity for Longitudinal Quickness')
1009
1010
1011 subplot(2,1,2)
1012 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
     participant.ego.accy);
1013 hold on
1014 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.accy);
1015 grid minor
1016
1017 plot(Data(participant_id).participant.ego.posy(Data(participant_id).
     participant.ego.measures.Long_quickness_start),Data(participant_id).
     participant.ego.accy(Data(participant_id).participant.ego.measures.
     Long_quickness_start),'k*','markersize',5);
1018 plot(Data(participant_id).participant.ego.posy(Data(participant_id).
     participant.ego.measures.Long_quickness_end),Data(participant_id).
     participant.ego.accy(Data(participant_id).participant.ego.measures.
     Long_quickness_end),'k*','markersize',5);
1019
1020 plot(Data(cuda_id).cuda.ego.posy(Data(cuda_id).cuda.ego.measures.
     Long_quickness_start),Data(cuda_id).cuda.ego.accy(Data(cuda_id).cuda.
     ego.measures.Long_quickness_start),'k*','markersize',5);
1021 plot(Data(cuda_id).cuda.ego.posy(Data(cuda_id).cuda.ego.measures.
     Long_quickness_end),Data(cuda_id).cuda.ego.accy(Data(cuda_id).cuda.ego.
     measures.Long_quickness_end),'k*','markersize',5);
1022
1023 title('Cutoff-Points Acceleration for Longitudinal Quickness')
1024 legend('Participant', 'Cuda','Acceleration cut-off points')
1025 xlim([start_y end_y]);
1026 ylabel('Longitudinal Acceleration [m/s2]')
1027 xlabel('Y-pos [m]')
1028
1029     case 0
1030         %Do Nothing
1031 end
1032 %% Plot time derivatives and statistical values
1033
1034 switch time_derivatives
1035
1036     case 1
1037 %longitudinal
1038 figure
1039 subplot(4,1,1)
1040 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
     participant.ego.posx);
1041 hold on
1042 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.posx);
1043 grid minor
1044 h=fill(obstacley,obstaclex,'black');
1045 h.FaceAlpha=0.05;
1046 line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
     ,2)
1047 line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
```

```
1048 line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
     HandleVisibility','off','linewidth',1.5)
1049 line(takeover1y,takeover1x, 'color', 'red','LineStyle','--','linewidth'
     ,1.5)
1050 title('Posititon')
1051 legend('Participant', 'Cuda')
1052 xlim([start_y end_y]);
1053 ylabel('X-pos [m]')
1054 xlabel('Y-pos [m]')
1055
1056 subplot(4,1,2)
1057 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
     participant.ego.vely);
1058 hold on
1059 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.vely);
1060 grid minor
1061 title('longitudinal Velocity')
1062 legend('Participant', 'Cuda')
1063 xlim([start_y end_y]);
1064 ylabel('X-vel [m/s]')
1065 xlabel('Y-pos [m]')
1066
1067 subplot(4,1,3)
1068 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
     participant.ego.accy);
1069 hold on
1070 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.accy);
1071 plot([start_y end_y],[Data(participant_id).participant.ego.STDaccy Data(
     participant_id).participant.ego.STDaccy],'k','linewidth',0.8)
1072 plot([start_y end_y],[Data(cuda_id).cuda.ego.STDaccy Data(cuda_id).cuda.
     ego.STDaccy],'k:','linewidth',0.8)
1073
1074 plot(Data(cuda_id).cuda.ego.posy(Data(cuda_id).cuda.ego.max_accy_id),Data(
     cuda_id).cuda.ego.max_accy,'k*','markersize',5)
1075 plot(Data(participant_id).participant.ego.posy(Data(participant_id).
     participant.ego.max_accy_id), Data(participant_id).participant.ego.
     max_accy,'k*','markersize',5)
1076
1077 plot([start_y end_y],[-Data(participant_id).participant.ego.STDaccy -Data(
     participant_id).participant.ego.STDaccy],'k','linewidth',0.8)
1078 plot([start_y end_y],[-Data(cuda_id).cuda.ego.STDaccy -Data(cuda_id).cuda.
     ego.STDaccy],'k:','linewidth',0.8)
1079 grid minor
1080 title('longitudinal acceleration')
1081 legend('Participant', 'Cuda','Standard deviation Participant','Standard
     deviation Cuda','Maximum Values')
1082
1083 xlim([start_y end_y]);
1084 ylabel('X-acc [m/s2]')
1085 xlabel('Y-pos [m]')
1086
1087 subplot(4,1,4)
1088 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
     participant.ego.jerk_long);
1089 hold on
1090 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.jerk_long);
```

```
1091 plot([start_y end_y],[Data(participant_id).participant.ego.STDjerk_long
         Data(participant_id).participant.ego.STDjerk_long],'k','linewidth',0.8)
1092 plot([start_y end_y],[Data(cuda_id).cuda.ego.STDjerk_long Data(cuda_id).
         cuda.ego.STDjerk_long],'k:','linewidth',0.8)
1093
1094 plot(Data(cuda_id).cuda.ego.posy(Data(cuda_id).cuda.ego.max_jerk_long_id),
         Data(cuda_id).cuda.ego.max_jerk_long,'k*','markersize',5)
1095 plot(Data(participant_id).participant.ego.posy(Data(participant_id).
         participant.ego.max_jerk_long_id), Data(participant_id).participant.ego
         .max_jerk_long,'k*','markersize',5)
1096
1097
1098 plot([start_y end_y],[-Data(participant_id).participant.ego.STDjerk_long -
         Data(participant_id).participant.ego.STDjerk_long],'k','linewidth',0.8)
1099 plot([start_y end_y],[-Data(cuda_id).cuda.ego.STDjerk_long -Data(cuda_id).
         cuda.ego.STDjerk_long],'k:','linewidth',0.8)
1100 grid minor
1101 legend('Participant', 'Cuda','Standard deviation Participant','Standard
         deviation Cuda','Maximum Values')
1102 title('Longitudinal jerk')
1103 xlim([start_y end_y]);
1104 ylabel('Y-jerk [m/s3]')
1105 xlabel('Y-pos [m]')
1106
1107 % Lateral
1108 figure
1109 subplot(4,1,1)
1110 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
         participant.ego.posx);
1111 hold on
1112 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.posx);
1113 h=fill(obstacley,obstaclex,'black');
1114 h.FaceAlpha=0.05;
1115 line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
         ,2)
1116 line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
1117 line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
         HandleVisibility','off','linewidth',1.5)
1118 line(takeover1y,takeover1x, 'color', 'red','LineStyle','--','linewidth'
         ,1.5)
1119 grid minor
1120 title('Posititon')
1121 legend('Participant', 'Cuda')
1122 xlim([start_y end_y]);
1123 ylabel('X-pos [m]')
1124 xlabel('Y-pos [m]')
1125
1126 subplot(4,1,2)
1127 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
         participant.ego.velx);
1128 hold on
1129 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.velx);
1130 grid minor
1131 title('Lateral Velocity')
1132 legend('Participant', 'Cuda')
1133 xlim([start_y end_y]);
```

```
1134 ylabel('X-vel [m/s]')
1135 xlabel('Y-pos [m]')
1136
1137 subplot(4,1,3)
1138 hold on
1139 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
     participant.ego.accx);
1140 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.accx);
1141 plot([start_y end_y],[Data(participant_id).participant.ego.STDaccx Data(
     participant_id).participant.ego.STDaccx],'k','linewidth',0.8)
1142 plot([start_y end_y],[Data(cuda_id).cuda.ego.STDaccx Data(cuda_id).cuda.
     ego.STDaccx],'k:','linewidth',0.8)
1143
1144 plot(Data(cuda_id).cuda.ego.posy(Data(cuda_id).cuda.ego.max_accx_id),Data(
     cuda_id).cuda.ego.max_accx,'k*','markersize',5)
1145 plot(Data(participant_id).participant.ego.posy(Data(participant_id).
     participant.ego.max_accx_id), Data(participant_id).participant.ego.
     max_accx,'k*','markersize',5)
1146
1147
1148 plot([start_y end_y],[-Data(participant_id).participant.ego.STDaccx -Data(
     participant_id).participant.ego.STDaccx],'k','linewidth',0.8)
1149 plot([start_y end_y],[-Data(cuda_id).cuda.ego.STDaccx -Data(cuda_id).cuda.
     ego.STDaccx],'k:','linewidth',0.8)
1150 grid minor
1151 title('lateral acceleration')
1152
1153 legend('Participant', 'Cuda','Standard deviation Participant','Standard
     deviation Cuda','Maximum Values')
1154 xlim([start_y end_y]);
1155 ylabel('X-acc [m/s2]')
1156 xlabel('Y-pos [m]')
1157
1158 subplot(4,1,4)
1159 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
     participant.ego.jerk_lat);
1160 hold on
1161 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.jerk_lat);
1162 plot([start_y end_y],[Data(participant_id).participant.ego.STDjerk_lat
     Data(participant_id).participant.ego.STDjerk_lat],'k','linewidth',0.8)
1163 plot([start_y end_y],[Data(cuda_id).cuda.ego.STDjerk_lat Data(cuda_id).
     cuda.ego.STDjerk_lat],'k:','linewidth',0.8)
1164
1165 plot(Data(participant_id).participant.ego.posy(Data(participant_id).
     participant.ego.max_jerk_lat_id), Data(participant_id).participant.ego.
     max_jerk_lat,'k*','markersize',5)
1166 plot(Data(cuda_id).cuda.ego.posy(Data(cuda_id).cuda.ego.max_jerk_lat_id),
     Data(cuda_id).cuda.ego.max_jerk_lat,'k*','markersize',5)
1167
1168
1169 plot([start_y end_y],[-Data(participant_id).participant.ego.STDjerk_lat -
     Data(participant_id).participant.ego.STDjerk_lat],'k','linewidth',0.8)
1170 plot([start_y end_y],[-Data(cuda_id).cuda.ego.STDjerk_lat -Data(cuda_id).
     cuda.ego.STDjerk_lat],'k:','linewidth',0.8)
1171 grid minor
```

```
1172 legend('Participant', 'Cuda','Standard deviation Participant','Standard
     deviation Cuda','Maximum Values')
1173 title('Lateral jerk')
1174 xlim([start_y end_y]);
1175 ylabel('Y-jerk [m/s3]')
1176 xlabel('Y-pos [m]')
1177
1178     case 0
1179         disp('Time derivatives not switched on')
1180
1181 end
1182
1183
1184
1185
1186 %% Plot all cuda ego trajectories
1187 if cuda_trajectories == 1
1188
1189 figure
1190 for i = 1:ids_total;
1191
1192 subplot(ids_total,1,i)
1193 title(strcat('Cuda Ego Trajectories for Scenario  ', num2str(scenario)'))
1194 hold on
1195 plot(Data(i).cuda.ego.posy, Data(i).cuda.ego.posx,'b','linewidth',1.5)
1196 h=fill(obstacley,obstaclex,'black');
1197 h.FaceAlpha=0.05;
1198 line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
     ,2)
1199 line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
1200 line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
     HandleVisibility','off','linewidth',1.5)
1201 line(takeover1y,takeover1x, 'color', 'red','LineStyle','--','linewidth'
     ,1.5)
1202 axis([start_y end_y -2 6]);
1203
1204 end
1205
1206 else
1207     disp(['All cuda ego trajectories plot is switched off'])
1208 end
1209
1210 %% Plot all participant trajectories and Cuda
1211
1212 if all_participants == 1
1213
1214 figure
1215 for i = participant_id_vector(1:end-1)
1216 title(strcat('Trajectories - Scenario  ', num2str(scenario)'))
1217 hold on
1218 plot(Data(i).participant.ego.posy, Data(i).participant.ego.posx,'color'
     ,[0.7 0.7 0.7],'linewidth',0.1,'HandleVisibility','off')
1219 line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
     ,2)
1220 line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
```

```
1221 line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
     HandleVisibility','off','linewidth',1.5)
1222 axis([start_y end_y -3 7]);
1223 xlabel('X-position [m]')
1224 ylabel('Y-Position [m]')
1225
1226
1227 end
1228 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.posx,'color',[0.2
      0.2 0.6],'linewidth',1.2)
1229 plot(Data(participant_id_vector(end)).participant.ego.posy, Data(
     participant_id_vector(end)).participant.ego.posx,'color',[0.8 0.8 0.8],
     'linewidth',0.1)
1230 plot(takeover1y,takeover1x,'k*' ,'LineStyle','--','linewidth',1.5)
1231 h=fill(obstacley,obstaclex,'black');
1232 h.FaceAlpha=0.1;
1233 set(gcf, 'Position', [100, 100, 1400, 300])
1234 legend('Optimized Trajectory','Participant Trajectories','Take-over
     Warning','Road Obstruction','location','Northwest')
1235
1236 else
1237     disp(['All participants vs cuda plot is switched off'])
1238 end
1239
1240 %% Colour gradient velocity trajectory
1241 switch velocity_gradient
1242     case 1
1243
1244 %Create colormap
1245 G = linspace(1, 0.1, 130)';
1246 G2 = zeros(100,1);
1247 green = [G;G2];
1248 R = linspace(0.1, 1, 130)';
1249 R2 = zeros(100,1);
1250 red = [R2 ;R];
1251 blue = zeros(length(red),1);
1252 myColormap = [green red blue];
1253 clear G G2 R R2 red green blue
1254
1255 % Plot velocity gradients
1256 figure
1257 ax1 = subplot(2,1,1);
1258 title(strcat('Participant And Cuda Ego Trajectories for Scenario ',
     num2str(scenario)'))
1259 hold on
1260 h=fill(obstacley,obstaclex,'black','HandleVisibility','off');
1261 h.FaceAlpha=0.05;
1262 line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
     ,2)
1263 line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
1264 line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
     HandleVisibility','off','linewidth',1.5)
1265 line(takeover1y,takeover1x, 'color', 'red','LineStyle','--','linewidth'
     ,1.5,'HandleVisibility','off')
1266 axis([start_y end_y -3 7]);
1267 xlabel('X-position [m]')
```

```
1268 ylabel('Y-Position [m]')
1269
1270
1271 z = Data(participant_id).participant.ego.vely'.*3.6;
1272 y = Data(participant_id).participant.ego.posx';
1273 x = Data(participant_id).participant.ego.posy';
1274 C = z;  % This is the color, vary with x in this case.
1275 surface([x;x],[y;y],[z;z],[C;C],...
1276         'facecol','no',...
1277         'edgecol','interp',...
1278         'linew',2);
1279 colormap(ax1, myColormap)
1280 colorbar
1281
1282 ax2 = subplot(2,1,2);
1283 h=fill(obstacley,obstaclex,'black','HandleVisibility','off');
1284 h.FaceAlpha=0.05;
1285 line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
     ,2)
1286 line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
1287 line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
     HandleVisibility','off','linewidth',1.5)
1288 line(takeover1y,takeover1x, 'color', 'red','LineStyle','--','linewidth'
     ,1.5,'HandleVisibility','off')
1289 axis([start_y end_y -3 7]);
1290 xlabel('X-position [m]')
1291 ylabel('Y-Position [m]')
1292 z2 = Data(cuda_id).cuda.ego.vely'.*3.6;
1293 y2 = Data(cuda_id).cuda.ego.posx';
1294 x2 = Data(cuda_id).cuda.ego.posy';
1295 C2 = z2;  % This is the color, vary with x in this case.
1296 surface([x2;x2],[y2;y2],[z2;z2],[C2;C2],...
1297         'facecol','no',...
1298         'edgecol','interp',...
1299         'linew',2);
1300 colormap(ax2, myColormap)
1301 colorbar
1302 set(gcf, 'Position', [100, 100, 1400, 300])
1303 hold off
1304
1305
1306     case 2
1307         disp(['Velocity gradient plot is switched off'])
1308
1309 end
1310
1311
1312 %% Animation participant
1313 switch animation_participant
1314   case 1
1315
1316 figure
1317 subplot(3,1,1)
1318 title(strcat('Participant Animation for Scenario  ', num2str(scenario)'))
1319 hold on
1320 h=fill(obstacley,obstaclex,'black');
```

```
1321 h.FaceAlpha=0.05;
1322 line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
     ,2)
1323 line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
1324 line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
     HandleVisibility','off','linewidth',1.5)
1325 line(takeover1y,takeover1x, 'color', 'black','LineStyle','--','linewidth'
     ,1.5)
1326 axis([start_y end_y -2 6]);
1327 xlabel('X-Position [m]')
1328 ylabel('Y-Position [m]')
1329 grid minor
1330 set(gcf, 'Position', [100, 100, 1400, 300])
1331 aline1=animatedline('Color','b','linewidth',1.5);
1332 aline2=animatedline('Color','r','linewidth',1);
1333 aline3=animatedline('Color','r','linestyle','--','linewidth',1);
1334 legend('Obstacle [roadworks]', 'Takeover Warning', 'Participant','
     Overtaking vehicle','location','Southeast')
1335 n=1;
1336 for k=1:(length(Data(participant_id).participant.ego.posx))
1337     addpoints(aline1,Data(participant_id).participant.ego.posy(n),Data(
          participant_id).participant.ego.posx(n)) ;
1338     addpoints(aline2,Data(participant_id).participant.overtaker.posy(n),
          Data(participant_id).participant.overtaker.posx(n))  ;
1339     %addpoints(aline7,Data(cuda_id).cuda.follower.posy(n),Data(cuda_id).
          cuda.follower.posx(n))  ;
1340
1341     drawnow
1342     n = n+1;
1343     pause(0.00001);
1344 end
1345
1346 subplot(3,1,2)
1347 %plot(Data(participant_id).participant.overtaker.posy, Data(participant_id
     ).participant.overtaker.vely .* 3.6,'r','linewidth',1.5)
1348 plot(Data(participant_id).participant.overtaker.posy, Data(participant_id)
     .participant.overtaker.accy,'r','linewidth',1.5)
1349 xlim([start_y end_y]);
1350 xlabel('X-Position [m]')
1351 ylabel('Y-Acc [m]')
1352 legend('Overtaking Vehicle')
1353 grid minor
1354
1355 subplot(3,1,3)
1356 plot(Data(participant_id).participant.overtaker.posy, Data(participant_id)
     .participant.overtaker.distance,'r','linewidth',1.5)
1357 xlim([start_y end_y]);
1358 xlabel('X-Position [m]')
1359 ylabel('Proximity distance [m]')
1360 legend('Overtaking Vehicle')
1361 grid minor
1362
1363
1364    case 2
1365    disp(['Participant animation is switched off'])
1366 end
```

```
1367
1368
1369 %% Animation Cuda
1370 switch animation_cuda
1371     case 1
1372
1373 figure
1374 subplot(2,1,1)
1375 title(strcat('Cuda Animation for Scenario  ', num2str(scenario)'))
1376 hold on
1377 h=fill(obstacley,obstaclex,'black');
1378 h.FaceAlpha=0.05;
1379 line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
     ,2)
1380 line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
1381 line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
     HandleVisibility','off','linewidth',1.5)
1382 line(takeover1y,takeover1x, 'color', 'black','LineStyle','--','linewidth'
     ,1.5)
1383 axis([start_y end_y -2 6]);
1384 xlabel('X-Position [m]')
1385 ylabel('Y-Position [m]')
1386 grid minor
1387 set(gcf, 'Position',  [100, 100, 1400, 300])
1388 aline1=animatedline('Color','b','linewidth',1.5);
1389 aline2=animatedline('Color','r','linestyle','--','linewidth',1);
1390 aline7=animatedline('Color','r','linestyle','--','linewidth',1);
1391 legend('Obstacle [roadworks]', 'Takeover Warning', 'Cuda','Overtaking
     vehicle', 'Following vehicle','location','Southeast')
1392 n=1;
1393 for k=1:(length(Data(cuda_id).cuda.ego.posx))
1394     addpoints(aline1,Data(cuda_id).cuda.ego.posy(n),Data(cuda_id).cuda.ego
         .posx(n)) ;
1395     addpoints(aline2,Data(cuda_id).cuda.overtaker.posy(n),Data(cuda_id).
         cuda.overtaker.posx(n))  ;
1396     addpoints(aline7,Data(cuda_id).cuda.follower.posy(n),Data(cuda_id).
         cuda.follower.posx(n))  ;
1397
1398     drawnow
1399     n = n+1;
1400     pause(0.003);
1401 end
1402
1403
1404 subplot(2,1,2)
1405 plot(Data(cuda_id).cuda.overtaker.posy, Data(cuda_id).cuda.overtaker.vely
     .* 3.6,'r','linewidth',1.5)
1406 xlim([start_y end_y]);
1407 xlabel('X-Position [m]')
1408 ylabel('Y-Velocity [m]')
1409 legend('Overtaking Vehicle')
1410 grid minor
1411
1412
1413     case 0
1414     disp(['Cuda Animation is switched off'])
```

```
1415 end
1416
1417
1418 %% Compute all euclidean distances
1419 switch scenario
1420     case {1,3,5}
1421 euclidean_lowest = zeros(ids_total,1);
1422
1423 %Compute all euc distances and their mean value
1424 for i = 1:ids_total
1425 Data(i).cuda.measures.euclidean=sqrt((Data(i).cuda.ego.posx - Data(i).cuda
        .overtaker.posx).^2+(Data(i).cuda.ego.posy - Data(i).cuda.overtaker.
        posy).^2);
1426 %Data(i).measures.euclidean_plotvector = (Data(i).cuda.ego.posy + Data(i).
        cuda.overtaker.posy)/2;
1427 Data(i).cuda.measures.euclidean_lowest = min(Data(i).cuda.measures.
        euclidean(1:end,1));
1428 euclidean_lowest(i) = Data(i).cuda.measures.euclidean_lowest;
1429 end
1430
1431 id_min_euc = find(euclidean_lowest == min(euclidean_lowest));
1432 id_max_euc = find(euclidean_lowest == max(euclidean_lowest));
1433
1434     case {2,4,6}
1435         %Nothing
1436 end
1437
1438 % Euclidean distance animation
1439 switch euclidean_plot
1440     case 1
1441
1442 figure
1443 subplot(2,1,1);
1444 axis([0 3000 0 max(Data(cuda_id).cuda.measures.euclidean)/6]);
1445 aline3=animatedline('Color','r','linewidth',1.5);
1446 grid minor
1447 title('Euclidean Distance Metric')
1448 ylabel('Euclidean Distance [m]')
1449 xlabel('Road length [m]')
1450
1451 subplot(2,1,2)
1452 h=fill(obstacley,obstaclex,'black');
1453 h.FaceAlpha=0.05;
1454 line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
        ,2)
1455 line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
1456 line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
        HandleVisibility','off','linewidth',1.5)
1457 axis([start_y end_y -2 6]);
1458
1459 grid minor
1460 title('Trajectories Animation')
1461 xlabel('X-Position [m]')
1462 ylabel('Y-Position [m]')
1463 aline4=animatedline('Color','b','linewidth',1.5);
1464 aline5=animatedline('Color','r','linewidth',1.5);
```

```
1465 aline6=animatedline('LineWidth',0.01,'color',[0.5 0.5 0.5 0.3]);
1466 legend(strcat('Human Trajectory  ',num2str(cuda_id)),'Cuda Trajectory',':'
     )
1467
1468 n=1;
1469 for k=1:(length(Data(cuda_id).cuda.ego.posy))
1470 %    addpoints(aline3,Data(cuda_id).measures.euclidean_plotvector(n),Data(
     cuda_id).measures.euclidean(n)) ;
1471      addpoints(aline3,Data(cuda_id).cuda.ego.posy(n),Data(cuda_id).cuda.
          measures.euclidean(n)) ;
1472
1473      addpoints(aline4,Data(cuda_id).cuda.ego.posy(n),Data(cuda_id).cuda.ego
          .posx(n)) ;
1474      addpoints(aline5,Data(cuda_id).cuda.overtaker.posy(n),Data(cuda_id).
          cuda.overtaker.posx(n))
1475      addpoints(aline6,[Data(cuda_id).cuda.ego.posy(n) Data(cuda_id).cuda.
          overtaker.posy(n)], [Data(cuda_id).cuda.ego.posx(n) Data(cuda_id).
          cuda.overtaker.posx(n)]);
1476
1477      drawnow
1478      n = n+1;
1479      pause(0.01);
1480 end
1481
1482      case 0
1483      disp('Euclidean plot not wanted or not possible for certain scenarios'
          )
1484 end
1485
1486
1487
1488
1489
1490
1491
1492 %% 3D plot
1493 switch threeD_plot
1494      case 1
1495
1496 figure
1497 plot3(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.posx, Data(
     cuda_id).cuda.ego.vely.*3.6,'b')
1498 hold on
1499 plot3(Data(participant_id).participant.ego.posy, Data(participant_id).
     participant.ego.posx, Data(participant_id).participant.ego.vely.*3.6,'r
     ')
1500 h=fill(obstacley,obstaclex,'black');
1501 h.FaceAlpha=0.05;
1502 line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
     ,2)
1503 line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
1504 line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
     HandleVisibility','off','linewidth',1.5)
1505 line(takeover1y,takeover1x, 'color', 'black','LineStyle','--','linewidth'
     ,1.5)
1506 axis([start_y end_y -2 6]);
```

```
1507 xlabel('X-Position [m]')
1508 ylabel('Y-Position [m]')
1509 zlabel('Y-Velocity [km/h]')
1510 set(gcf, 'Position', [100, 100, 1400, 300])
1511 title(strcat('3-D velocity plot for Scenario ', num2str(scenario)'))
1512 legend('Cuda Trajectory', strcat('Participant ID ', num2str(
     participant_id)'), 'Roadworks', 'Takeover Warning')
1513 grid minor
1514     case 0
1515         disp('3-D plot not wanted')
1516 end
1517
1518
1519 %%
1520 if TwoD_plot == 1
1521
1522 figure
1523 title(strcat('Participant ID And Cuda Ego Trajectories for Scenario ',
     num2str(scenario)'))
1524 hold on
1525
1526 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.posx,'r','
     linewidth',0.7)
1527 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
     participant.ego.posx,'b','linewidth',0.5)
1528 h=fill(obstacley,obstaclex,'black','HandleVisibility','off');
1529 h.FaceAlpha=0.05;
1530 line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
     ,2)
1531 line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
1532 line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
     HandleVisibility','off','linewidth',1.5)
1533 line(takeover1y,takeover1x, 'color', 'red','LineStyle','--','linewidth'
     ,1.5,'HandleVisibility','off')
1534 axis([start_y  end_y -3 7]);
1535 xlabel('X-position [m]')
1536 ylabel('Y-Position [m]')
1537 legend('Cuda', 'Participant')
1538 grid minor
1539 set(gcf, 'Position', [100, 100, 1400, 300])
1540
1541 else
1542     disp(['Two dimensional trajectories plot switched off'])
1543 end
1544
1545
1546 %% Cuda TTC
1547 slope = diff(Data(cuda_id).cuda.ego.posx)./diff(Data(cuda_id).cuda.ego.
     posy);
1548 L_tangent_cuda = 900;
1549 ttc_id_cuda = 1;        % (start)
1550 xi = 1;                 %nonzero start
1551 yi = 1;
1552 margin = 0.5;
1553 disp(['TTC collision margin is ',num2str(margin)])
1554
```

```
1555            while ~isempty(xi)
1556            ttc_cuda_y_last = yi;
1557            ttc_cuda_x_last = xi;
1558            x2 = (slope(ttc_id_cuda) .* ((Data(cuda_id).cuda.ego.posy(
                    ttc_id_cuda)+L_tangent_cuda) - Data(cuda_id).cuda.ego.posy(
                    ttc_id_cuda)) + Data(cuda_id).cuda.ego.posx(ttc_id_cuda));
1559            [xi, yi] = polyxpoly([Data(cuda_id).cuda.ego.posy(ttc_id_cuda) (
                    Data(cuda_id).cuda.ego.posy(ttc_id_cuda)+L_tangent_cuda)],[Data
                    (cuda_id).cuda.ego.posx(ttc_id_cuda) x2],[obstacley(1)
                    obstacley(2)],[obstaclex(1) obstaclex(2)+margin]);
1560
1561            ttc_cuda_x = xi;
1562            ttc_cuda_y = yi;
1563
1564            ttc_id_cuda = ttc_id_cuda + 1;
1565
1566            end
1567
1568    ttc_id_cuda = ttc_id_cuda-1;
1569
1570    Data(cuda_id).cuda.ego.measures.ttc =   (obstacley(1) - Data(cuda_id).cuda
            .ego.posy(ttc_id_cuda)) /  Data(cuda_id).cuda.ego.vely(ttc_id_cuda);
1571    disp(['The minimum TTC for the Cudaplanner is ' num2str(Data(cuda_id).cuda
            .ego.measures.ttc) 's'])
1572
1573    %% Participant TTC
1574    L_tangent = 900;
1575
1576
1577    for ids = participant_id_vector
1578
1579        switch scenario
1580        case {1 2}
1581            ttc_id = 1;
1582        case{3 4}
1583            ttc_id =400;
1584
1585        case{5 6}
1586            ttc_id = 600;
1587        end
1588
1589        Data(ids).participant.ego.measures.ttc.slope = diff(Data(ids).
                participant.ego.posx) ./ diff(Data(ids).participant.ego.posy);
1590        %Data(ids).participant.ego.measures.ttc.slope = movmean(Data(ids).
                participant.ego.measures.ttc.slope,6);
1591        xi = 1;
1592            yi = 1;
1593
1594            %Troubleshoot figure below
1595
1596    %From HERE
1597
1598    % title('TTC Cuda: Collision-free point')
1599    % hold on
1600    % plot(Data(participant_id).participant.ego.posy, Data(participant_id).
            participant.ego.posx,'r','linewidth',1)
```

```
% h=fill(obstacley,obstaclex,'black','HandleVisibility','off');
% h.FaceAlpha=0.05;
% plot([obstacley(1) obstacley(2)],[obstaclex(1) obstaclex(2)], 'color', 'black','Marker','*', 'markersize', 4,'HandleVisibility','off')
% line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth',2)
% line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
% line(centerliney,centerlinex, 'color', 'black','LineStyle','--','HandleVisibility','off','linewidth',1.5)
% axis([start_y end_y -3 7]);
% grid minor
% set(gcf, 'Position', [100, 100, 1400, 300])
%
% Until HERE

        while ~isempty(xi)
        Data(ids).participant.ego.measures.ttc.ttc_x = xi;
        Data(ids).participant.ego.measures.ttc.ttc_y = yi;

        Data(ids).participant.ego.measures.ttc.ttc_id_tangent = Data(ids).participant.ego.posy(ttc_id) + L_tangent;
        x1 = (Data(ids).participant.ego.measures.ttc.slope(ttc_id) .* (Data(ids).participant.ego.measures.ttc.ttc_id_tangent - Data(ids).participant.ego.posy(ttc_id))) + Data(ids).participant.ego.posx(ttc_id);
        Data(ids).participant.ego.measures.ttc.x1 = x1;
        TF = isnan(Data(ids).participant.ego.measures.ttc.x1);
        if TF == 1
            Data(ids).participant.ego.measures.ttc.x1 = 0;
        else
            Data(ids).participant.ego.measures.ttc.x1 = Data(ids).participant.ego.measures.ttc.x1;
        end
        Data(ids).participant.ego.measures.ttc.ttc_id = ttc_id;
        [xi, yi] = polyxpoly([Data(ids).participant.ego.posy(ttc_id) Data(ids).participant.ego.measures.ttc.ttc_id_tangent],[Data(ids).participant.ego.posx(ttc_id) Data(ids).participant.ego.measures.ttc.x1],[obstacley(1) obstacley(2)],[obstaclex(1)-10000 obstaclex(2)+margin]);

        ttc_id = ttc_id + 1;

       %Troubleshoot figure below
       %From HERE
% Data(ids).participant.ego.measures.ttc.ttc_id = ttc_id
% line([Data(ids).participant.ego.posy(Data(ids).participant.ego.measures.ttc.ttc_id), Data(ids).participant.ego.measures.ttc.ttc_id_tangent], [Data(ids).participant.ego.posx(Data(ids).participant.ego.measures.ttc.ttc_id), Data(ids).participant.ego.measures.ttc.x1], 'Color', 'r','linestyle',':')
       %Until HERE

        end
        Data(ids).participant.ego.measures.ttc.ttc_id = ttc_id-1;
```

```
1639            Data(ids).participant.ego.measures.ttc.TTC =    (obstacley(1) -
                    Data(ids).participant.ego.posy(Data(ids).participant.ego.
                    measures.ttc.ttc_id)) /   Data(ids).participant.ego.vely(Data(
                    ids).participant.ego.measures.ttc.ttc_id);
1640    end
1641
1642
1643    disp(['The minimum TTC for the Participant is ' num2str(Data(
            participant_id).participant.ego.measures.ttc.TTC) 's'])
1644
1645    %% TTC Plot
1646
1647    switch TTC_plot
1648        case 1
1649
1650    figure
1651    subplot(3,1,1)
1652    title('Time-to-Collision and Acceleration values')
1653    hold on
1654    plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.posx,'b','
            linewidth',1)
1655    plot(Data(participant_id).participant.ego.posy, Data(participant_id).
            participant.ego.posx,'r','linewidth',1)
1656
1657    h=fill(obstacley,obstaclex,'black','HandleVisibility','off');
1658    h.FaceAlpha=0.05;
1659    line(takeover1y,takeover1x, 'color', 'red','LineStyle','--','linewidth'
            ,1.5,'HandleVisibility','off')
1660    plot([obstacley(1) obstacley(2)],[obstaclex(1) obstaclex(2)], 'color', '
            black','Marker','*', 'markersize', 4,'HandleVisibility','off')
1661    line(roady1,roadx1, 'color', 'black','HandleVisibility','off','linewidth'
            ,2)
1662    line(roady2,roadx2,'Color','black','HandleVisibility','off','linewidth',2)
1663    line(centerliney,centerlinex, 'color', 'black','LineStyle','--','
            HandleVisibility','off','linewidth',1.5)
1664    axis([start_y end_y -3 7]);
1665    grid minor
1666    set(gcf, 'Position', [100, 100, 1400, 300])
1667
1668    line([Data(participant_id).participant.ego.posy(Data(participant_id).
            participant.ego.measures.ttc.ttc_id), Data(participant_id).participant.
            ego.measures.ttc.ttc_id_tangent], [Data(participant_id).participant.ego
            .posx(Data(participant_id).participant.ego.measures.ttc.ttc_id), Data(
            participant_id).participant.ego.measures.ttc.x1], 'Color', 'r','
            linestyle',':','linewidth',1.2)
1669    line([Data(cuda_id).cuda.ego.posy(ttc_id_cuda), Data(cuda_id).cuda.ego.
            posy(ttc_id_cuda)+L_tangent_cuda], [Data(cuda_id).cuda.ego.posx(
            ttc_id_cuda), x2], 'Color', 'b','linestyle',':','linewidth',1.2)
1670    legend('Trajectory Cuda','Trajectory Participant')
1671
1672    subplot(3,1,2)
1673    title('Lateral Acceleration')
1674    plot(Data(participant_id).participant.ego.posy, Data(participant_id).
            participant.ego.accx,'r','linewidth',1)
1675    hold on
```

```
1676 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.accx,'b','
     linewidth',1)
1677 grid minor
1678 xlim([start_y end_y])
1679 xlabel('Position y [m]')
1680 ylabel('Lateral acceleration [m/s2]')
1681
1682 subplot(3,1,3)
1683 title('Longitduinal Acceleration')
1684 plot(Data(participant_id).participant.ego.posy, Data(participant_id).
     participant.ego.accy,'r','linewidth',1)
1685 hold on
1686 plot(Data(cuda_id).cuda.ego.posy, Data(cuda_id).cuda.ego.accy,'b','
     linewidth',1)
1687 grid minor
1688 xlim([start_y end_y])
1689 xlabel('Position y [m]')
1690 ylabel('Longitudinal acceleration [m/s2]')
1691
1692     case 0
1693         disp(['TTC Plot not wanted'])
1694 end
1695
1696
1697 %% Measures
1698
1699 switch measures
1700     case 1
1701     %Maximum lateral acceleration
1702     disp(['The Maximum lateral acceleration for the participant is '
         num2str(Data(participant_id).participant.ego.max_accx) ' [m/s2]'])
1703     %Maximum Longitudinal acceleration
1704     disp(['The Maximum longitudinal acceleration for the participant is '
          num2str(Data(participant_id).participant.ego.max_accy) ' [m/s2]'])
1705     %Mean lateral acceleration
1706     disp(['The mean lateral acceleration for the participant is ' num2str
         (Data(participant_id).participant.ego.mean_accx) ' [m/s2]'])
1707     %Mean longitudinal acceleration
1708      disp(['The mean longitudinal acceleration for the participant is '
          num2str(Data(participant_id).participant.ego.mean_accy) ' [m/s2]'
          ])
1709     %STD Longitudinal acceleration
1710      disp(['The standard deviation of longitudinal acceleration for the
          participant is ' num2str(Data(participant_id).participant.ego.
         STDaccy) ' [m/s2]'])
1711     %STD Lateral acceleration
1712      disp(['The standard deviation of lateral acceleration for the
          participant is ' num2str(Data(participant_id).participant.ego.
         STDaccx) ' [m/s2]'])
1713
1714     %Maximum lateral Jerk
1715     disp(['The Maximum lateral Jerk for the participant is ' num2str(Data
         (participant_id).participant.ego.max_jerk_lat) ' [m/s3]'])
1716     %Maximum Longitudinal Jerk
1717     disp(['The Maximum longitudinal Jerk for the participant is ' num2str
         (Data(participant_id).participant.ego.max_jerk_long) ' [m/s3]'])
```

```
1718    %Mean lateral Jerk
1719    disp(['The Mean lateral Jerk for the participant is ' num2str(Data(
            participant_id).participant.ego.meanjerk_lat) ' [m/s3]'])
1720    %Mean longitudinal Jerk
1721     disp(['The Mean longitudinal Jerk for the participant is ' num2str(
             Data(participant_id).participant.ego.meanjerk_long) ' [m/s3]'])
1722    %STD Lateral Jerk
1723    disp(['The Standard deviation for lateral Jerk for the participant is
            ' num2str(Data(participant_id).participant.ego.STDjerk_lat) ' [m/
            s3]'])
1724    %STD Longitudinal Jerk
1725     disp(['The Standard deviation for longitudinal Jerk for the
             participant is ' num2str(Data(participant_id).participant.ego.
             STDjerk_long) ' [m/s3]'])
1726    %Minimum distance to CV
1727
1728    %disp(['The Smallest distance to the overtaker is ' num2str(Data(
            participant_id).participant.overtaker.distance_lowest) ' [m]'])
1729    %Minimum distance to LB
1730    disp(['The Smallest distance to the lane boundary is ' num2str(Data(
            participant_id).participant.ego.measures.LBdistance) ' [m]'])
1731    %Deceleration other vehicle
1732
1733
1734    %disp(['The largest deceleration of the overtaking vehicle is '
            num2str(Data(participant_id).participant.ego.measures.X_CVbraking *
             10) ' [m/s2]'])
1735
1736
1737
1738    All_ID = zeros(length(participant_id_vector),1);
1739    All_mean_accx = zeros(length(participant_id_vector),1);
1740    All_mean_accy = zeros(length(participant_id_vector),1);
1741    All_max_accy = zeros(length(participant_id_vector),1);
1742    All_max_accx = zeros(length(participant_id_vector),1);
1743    All_STD_accy = zeros(length(participant_id_vector),1);
1744    All_STD_accx = zeros(length(participant_id_vector),1);
1745    All_mean_jerk_long = zeros(length(participant_id_vector),1);
1746    All_mean_jerk_lat = zeros(length(participant_id_vector),1);
1747    All_max_jerk_long = zeros(length(participant_id_vector),1);
1748    All_max_jerk_lat = zeros(length(participant_id_vector),1);
1749    All_STD_jerk_long = zeros(length(participant_id_vector),1);
1750    All_STD_jerk_lat = zeros(length(participant_id_vector),1);
1751   All_distance_CV = zeros(length(participant_id_vector),1);
1752    All_distance_LB = zeros(length(participant_id_vector),1);
1753    All_ttc = zeros(length(participant_id_vector),1);
1754   All_deceleration_CV = zeros(length(participant_id_vector),1);
1755    All_quicknesslong = zeros(length(participant_id_vector),1);
1756    All_quicknesslat = zeros(length(participant_id_vector),1);
1757
1758
1759    switch scenario
1760        case{1,3,5}
1761
1762    %Cuda
1763   Cuda_ttc = Data(cuda_id).cuda.ego.measures.ttc ;
```

```
1764    Cuda_distance_LB = Data(cuda_id).cuda.measures.LBdistance;
1765    Cuda_distance_CV = Data(cuda_id).cuda.measures.euclidean_lowest   ;
1766    Cuda_deceleration_CV = Data(cuda_id).cuda.measures.braking_follower;
1767
1768    Cuda_MeanAccx = Data(cuda_id).cuda.ego.mean_accx;
1769    Cuda_MeanAccy = Data(cuda_id).cuda.ego.mean_accy;
1770    Cuda_MaxAccx = Data(cuda_id).cuda.ego.max_accx;
1771    Cuda_MaxAccy = Data(cuda_id).cuda.ego.max_accy;
1772    Cuda_STDAccx = Data(cuda_id).cuda.ego.STDaccx;
1773    Cuda_STDAccy = Data(cuda_id).cuda.ego.STDaccy;
1774
1775    Cuda_MeanJerkx = Data(cuda_id).cuda.ego.mean_jerk_lat;
1776    Cuda_MeanJerky = Data(cuda_id).cuda.ego.mean_jerk_long;
1777    Cuda_MaxJerkx = Data(cuda_id).cuda.ego.max_jerk_lat;
1778    Cuda_MaxJerky = Data(cuda_id).cuda.ego.max_jerk_long;
1779    Cuda_STDJerkx = Data(cuda_id).cuda.ego.STDjerk_lat;
1780    Cuda_STDJerky = Data(cuda_id).cuda.ego.STDjerk_long;
1781
1782    All_cuda = [Cuda_ttc, Cuda_distance_CV, Cuda_distance_LB,
        Cuda_deceleration_CV , 0, 0, ...
1783      Cuda_MeanAccx, Cuda_MeanAccy, Cuda_MaxAccx, Cuda_MaxAccy ,
            Cuda_STDAccx, Cuda_STDAccy...
1784    Cuda_MeanJerkx, Cuda_MeanJerky, Cuda_MaxJerkx, Cuda_MaxJerky,
          Cuda_STDJerkx, Cuda_STDJerky ];
1785
1786    filename = (['Cuda_measures_scenario' num2str(scenario) '_ID_' num2str(
        cuda_id)]);
1787          xlswrite(filename,All_cuda)
1788
1789
1790      case{2,4,6}
1791
1792    Cuda_ttc = Data(cuda_id).cuda.ego.measures.ttc ;
1793    Cuda_distance_LB = Data(cuda_id).cuda.measures.LBdistance;
1794
1795    Cuda_MeanAccx = Data(cuda_id).cuda.ego.mean_accx;
1796    Cuda_MeanAccy = Data(cuda_id).cuda.ego.mean_accy;
1797    Cuda_MaxAccx = Data(cuda_id).cuda.ego.max_accx;
1798    Cuda_MaxAccy = Data(cuda_id).cuda.ego.max_accy;
1799    Cuda_STDAccx = Data(cuda_id).cuda.ego.STDaccx;
1800    Cuda_STDAccy = Data(cuda_id).cuda.ego.STDaccy;
1801
1802    Cuda_MeanJerkx = Data(cuda_id).cuda.ego.mean_jerk_lat;
1803    Cuda_MeanJerky = Data(cuda_id).cuda.ego.mean_jerk_long;
1804    Cuda_MaxJerkx = Data(cuda_id).cuda.ego.max_jerk_lat;
1805    Cuda_MaxJerky = Data(cuda_id).cuda.ego.max_jerk_long;
1806    Cuda_STDJerkx = Data(cuda_id).cuda.ego.STDjerk_lat;
1807    Cuda_STDJerky = Data(cuda_id).cuda.ego.STDjerk_long;
1808
1809          All_cuda = [Cuda_ttc, 0, Cuda_distance_LB, 0 , 0, 0, ...
1810      Cuda_MeanAccx, Cuda_MeanAccy, Cuda_MaxAccx, Cuda_MaxAccy ,
            Cuda_STDAccx, Cuda_STDAccy...
1811    Cuda_MeanJerkx, Cuda_MeanJerky, Cuda_MaxJerkx, Cuda_MaxJerky,
          Cuda_STDJerkx, Cuda_STDJerky ];
1812
```

```
1813        filename = (['Cuda_measures_scenario' num2str(scenario) '_ID_' num2str
                (cuda_id)]);
1814                xlswrite(filename,All_cuda)
1815
1816
1817
1818    end
1819
1820        case 0
1821            %Nothing
1822        end
1823
1824
1825     %% Write output file
1826
1827        switch output
1828            case 1
1829
1830        for i = participant_id_vector
1831
1832
1833    All_ID(i) = i;
1834    All_ID(All_ID==0) = [];
1835
1836    All_mean_accx(i) = Data(i).participant.ego.mean_accx;
1837    All_mean_accx( All_mean_accx==0) = [];
1838
1839    All_mean_accy(i) = Data(i).participant.ego.mean_accy;
1840    All_mean_accy( All_mean_accy==0) = [];
1841
1842    All_max_accx(i) = Data(i).participant.ego.max_accx;
1843    All_max_accx(All_max_accx==0) = [];
1844
1845    All_max_accy(i) = Data(i).participant.ego.max_accy;
1846    All_max_accy( All_max_accy==0) = [];
1847
1848    All_STD_accx(i) = Data(i).participant.ego.STDaccx;
1849    All_STD_accx( All_STD_accx==0) = [];
1850
1851    All_STD_accy(i) = Data(i).participant.ego.STDaccy;
1852    All_STD_accy( All_STD_accy==0) = [];
1853
1854    All_mean_jerk_long(i) = Data(i).participant.ego.meanjerk_long;
1855    All_mean_jerk_long(All_mean_jerk_long==0) = [];
1856
1857    All_mean_jerk_lat(i) = Data(i).participant.ego.meanjerk_lat;
1858    All_mean_jerk_lat(All_mean_jerk_lat==0) = [];
1859
1860    All_max_jerk_long(i) = Data(i).participant.ego.max_jerk_long;
1861    All_max_jerk_long(All_max_jerk_long==0) = [];
1862
1863    All_max_jerk_lat(i) = Data(i).participant.ego.max_jerk_lat;
1864    All_max_jerk_lat(All_max_jerk_lat==0) = [];
1865
1866    All_STD_jerk_long(i) = Data(i).participant.ego.STDjerk_long;
1867    All_STD_jerk_long(All_STD_jerk_long==0) = [];
```

```
1868
1869    All_STD_jerk_lat(i) = Data(i).participant.ego.STDjerk_lat;
1870    All_STD_jerk_lat(All_STD_jerk_lat==0) = [];
1871
1872  %All_distance_CV(i) = Data(i).participant.overtaker.distance_lowest;
1873  %All_distance_CV(All_distance_CV==0) = [];
1874
1875    All_distance_LB(i) = Data(i).participant.ego.measures.LBdistance;
1876    All_distance_LB(All_distance_LB==0) = [];
1877
1878    All_ttc(i) = Data(i).participant.ego.measures.ttc.TTC;
1879    All_ttc(All_ttc==0) = [];
1880
1881   %All_deceleration_CV(i) = min(Data(i).participant.overtaker.accy(
          overtake_start:overtake_end));
1882  %All_deceleration_CV(i) = min(Data(i).participant.overtaker.accy(
         overtake_start:end));
1883   %All_deceleration_CV(All_deceleration_CV==0) = [];
1884
1885
1886  %   All_quicknesslong(i) = Data(i).participant.ego.measures.Long_quickness;
1887  %   All_quicknesslong(All_quicknesslong==0) = [];
1888  %
1889  %   All_quicknesslat(i) = Data(i).participant.ego.measures.Lat_quickness;
1890  %   All_quicknesslat(All_quicknesslat==0) = [];
1891
1892
1893       end
1894      zerovector = zeros(length(All_ID),1);
1895
1896      switch scenario
1897          case 1
1898
1899                  All_metrics = [All_ID', All_ttc' , All_distance_CV' ,
                        All_distance_LB', All_deceleration_CV' , zerovector ,
                        zerovector ...
1900            All_mean_accx', All_mean_accy' ,All_max_accx' , All_max_accy' ,
                All_STD_accx' , All_STD_accy' , All_mean_jerk_lat',...
1901             All_mean_jerk_long' ,All_max_jerk_lat', All_max_jerk_long' ,
                 All_STD_jerk_lat', All_STD_jerk_long'];
1902
1903
1904                filename = 'All_measures_scenario1';
1905                xlswrite(filename,All_metrics)
1906           case 2
1907
1908                  All_metrics = [All_ID', All_ttc' ,zerovector,
                        All_distance_LB' , zerovector, zerovector ,zerovector ...
1909            All_mean_accx', All_mean_accy' ,All_max_accx' , All_max_accy' ,
                All_STD_accx' , All_STD_accy' , All_mean_jerk_lat',...
1910             All_mean_jerk_long' ,All_max_jerk_lat', All_max_jerk_long' ,
                 All_STD_jerk_lat', All_STD_jerk_long'];
1911
1912
1913                filename = 'All_measures_scenario2';
1914                xlswrite(filename,All_metrics)
```

```
1915        case 3
1916
1917                All_metrics = [All_ID', All_ttc' , All_distance_CV' ,
                       All_distance_LB', All_deceleration_CV' , zerovector ,
                       zerovector ...
1918        All_mean_accx', All_mean_accy' ,All_max_accx' , All_max_accy' ,
               All_STD_accx' , All_STD_accy' , All_mean_jerk_lat',...
1919         All_mean_jerk_long' ,All_max_jerk_lat', All_max_jerk_long' ,
               All_STD_jerk_lat', All_STD_jerk_long'];
1920
1921
1922              filename = 'All_measures_scenario3';
1923              xlswrite(filename,All_metrics)
1924        case 4
1925
1926                  All_metrics = [All_ID', All_ttc'  ,zerovector,
                         All_distance_LB', zerovector, zerovector ,zerovector
                         ...
1927        All_mean_accx', All_mean_accy' ,All_max_accx' , All_max_accy' ,
               All_STD_accx' , All_STD_accy' , All_mean_jerk_lat',...
1928         All_mean_jerk_long' ,All_max_jerk_lat', All_max_jerk_long' ,
               All_STD_jerk_lat', All_STD_jerk_long'];
1929
1930
1931              filename = 'All_measures_scenario4';
1932              xlswrite(filename,All_metrics)
1933        case 5
1934
1935               All_metrics = [All_ID', All_ttc' , All_distance_LB',
                          zerovector ,zerovector ...
1936        All_mean_accx', All_mean_accy' ,All_max_accx' , All_max_accy' ,
               All_STD_accx' , All_STD_accy' , All_mean_jerk_lat',...
1937         All_mean_jerk_long' ,All_max_jerk_lat', All_max_jerk_long' ,
               All_STD_jerk_lat', All_STD_jerk_long'];
1938
1939
1940
1941              filename = 'All_measures_scenario5';
1942              xlswrite(filename,All_metrics)
1943         case 6
1944
1945                     All_metrics = [All_ID', All_ttc' , All_distance_LB' ,
                            zerovector ,zerovector ...
1946        All_mean_accx', All_mean_accy' ,All_max_accx' , All_max_accy' ,
               All_STD_accx' , All_STD_accy' , All_mean_jerk_lat',...
1947         All_mean_jerk_long' ,All_max_jerk_lat', All_max_jerk_long' ,
               All_STD_jerk_lat', All_STD_jerk_long'];
1948
1949
1950              filename = 'All_measures_scenario6';
1951              xlswrite(filename,All_metrics)
1952    end
1953
1954     case 0
1955     disp(['output not wanted'])
1956 end
```

A.4. Reproducibility and Archiving

The exact driving scenario descriptions can be found in Appendix A, these were used as templates for the simulator programming, which was performed by Ingo Krems, Claus-Heiko Winkler, and Andreas Ronellenfitsch, at the Porsche 'Virtueller Fahrerplatz 2' in Weissach, Germany. All data, both from the human-in-the-loop driving simulator study and that belonging to the CudaDP planner simulations carried out at Volkswagen Research in Wolfsburg, are stored and kept by Fabian Doubek, at the Porsche $R\&D$ Facility in Weissach. Besides the raw data, the pre-processed dataset used for statistical analysis, and the used Matlab script (Appendix C) are stored there as well.