Optically Motivated Machine Learning Speed Control Model for Curved Roads

MSc Thesis

S. Ceelen

Technische Universiteit Delft

TU Delft

Optically Motivated Machine Learning Speed Control Model for Curved Roads

Stef Ceelen
Control and Simulation, Faculty of Aerospace Engineering,
Delft University of Technology

***Abstract*—During the transition from manual driving to (partially) automated driving, conflicts between human and automation should be minimized. An important factor to reduce the human-machine interaction risk is by increasing the trust drivers have in the system. A system including human-like behaviour tends to be trusted more. Accurate models of human driving behaviour are important in safety studies and the development of human-compatible automated driving. While global models exist showing the relationship between road geometry and speed control, research on models that use observable visual inputs for drivers is scarce. Previous research used a heuristic approach to develop such a model. This work extends on that basis, using machine learning to develop a model for speed control based on angles and time margins that can be extracted from the driver's visual field. The model is developed using two optimization methods with a single hidden layer feed-forward neural network structure and training data obtained from 14 participants in a fixed-base driving simulator. Validation runs have been done by the same participants in order to validate the individual and general (one-size-fits-all) longitudinal driving models. The validation shows that a general model is able to accurately capture human longitudinal acceleration behaviour on single-lane curved high-speed roads (100km/h).**

1. Introduction

Nowadays driving is a very familiar task for most people. In order to enhance safety and driving convenience, many Advanced Driver Assistance Systems (ADAS) are being developed. Some of the newest systems include lane keeping assistance and Adaptive Cruise Control (ACC) [1]. A transition is taking place from manual driving towards autonomous driving by the use of ADAS. During this transition humans and machines have to understand each other and work together to execute the driving task. One of the factors that has an effect on the human-automation interaction is the trust that drivers have in the automated system [10, 21]. When an automated system has human-like behavioral aspects, drivers tend to have more trust in the system [33]. This is also shown by a study in which participants were played multiple trajectories, of which the test subjects preferred their own trajectory [2]. Even though human-like behaviour can be of great importance in order to avoid conflict between automation and driver during curve driving, studies on this subject are still incomplete.

Most studies in the field of cornering speed adaptation directly relate road geometry to vehicle speed [7, 30, 8]. Since curves usually have a constant radius and width, these studies relate the average speed in such a curve to its geometry. Correlations are found between road geometry and average or minimum velocity [25]. However, even though these relations can have a positive impact on driving safety, they do not include the human behavioural aspects. Drivers tend to slow down while entering a curve and accelerate while exiting a curve [23, 31]. A study by Zhang [35] on the development of an ACC system including adaptive speed control in curve driving takes this phenomenon into account. However, this study still relates only road curvature and desired velocity.

Most of the relations between road geometry and vehicle speed can be applied to ADAS to enhance safety by setting upper speed limits for specific curves. These limits do however not provide any information about the desired driver speed, hence do not improve the human-automation interaction by implementing human-like behaviour. Drivers tend to make a trade-off between maximum speed and comfort (often measured as lateral acceleration) [28]. As stated above, for most curves drivers slow down before the curve starts, before any lateral acceleration is felt [7, 4, 19]. This indicates the capability of humans to estimate roughly what the lateral acceleration ahead will be. Reymond et al. [28] shows that a driver is able to estimate lateral accelerations during static test runs, without inertial motion, indicating that this estimation is more strongly determined by cognitive and visual cues rather than motion cues. Even though many studies discussed above relate geometry to speed, people tend to be rather poor judges of road curvature [12]. Therefore, other visual cues provided to drivers must play a role in speed adaptation.

Research from Lehtonen et al. [22] investigates the eye movement when approaching a curve on a rural road. It was found that drivers switch their focus between the on-road Far Point (FP) and Tangent Point (TP) and off-road Occlusion Point (OP). The OP would be used to monitor upcoming traffic emerging from this point [22], the FP for steering purposes [29, 18] and the (Extended) TP (ETP) while driving towards/in curves [18]. Contrary to drivers being rather poor judges of distances and curvature [12],

they seem to be rather good judges of visual angles and time margins to visual points [5]. In the case of speed adaptation, humans tend to use safety margins related to time in order to control the accompanied risk levels. It is argued that the ETP is used in curves [15], the time to collision (TTC) when approaching obstacles [20], the Time to Headway (THW) for car following purposes [32] and the Time to Lane Crossing (TLC) for lane keeping [32].

Time values to specific visual cues are easily perceived by humans. A resulting output of the human neuromuscular system is not the vehicle velocity but the accelerator and brake pedal deflections. Research that relates the TETP value to pedal deflection was done by Gruppelaar [15], who investigated time margin thresholds of the TETP and its derivative to decide on the driving phases before, during and after a curve. Although this model serves as a good initial estimation for the relationship between TETP and pedal deflections on single curves, a few flaws are present. Highest priority for improving this model is by fixing the oscillations in pedal deflection close to turn exit, which are assumed to be due to the hard threshold on the rate of change of the TETP. Furthermore, additional visual cues, such as the TP angle, can be added to the model in order to improve the performance for single curves and larger road distances. In order to remove the hard thresholds on model inputs, a neural network approach will be used for modeling driver speed adaptation. Using this approach, it will be easier to include and exclude certain inputs without changing the entire model architecture. This is the first research in speed adaptation that uses visually observable inputs to humans from the optical world as opposed to the Cartesian world for a machine learning algorithm. Since drivers seem rather good judges of time margins and visual angles, these will be investigated as candidate model inputs.

Section 2 of this document shows the model architecture, including optimization loops for the neural network weights. In Section 4 the experimental goals will be explained, including the methodology in order to achieve these goals. Section 5 exhibits the results found after processing measured data and developing speed control models. The results are analysed and discussed in Section 6.

2. Visual Cues

Most research related to speed adaptation on curved roads focuses on models that relate speed choice to the geometry of the road. These models do not include brake and accelerator pedal deflections, which are direct inputs for the car dynamics. Gruppelaar [15] investigated relations between time margins of a driver's visual fixation point to the pedal control behaviour. Here, the TETP turned out to be an important factor in accelerator and brake pedal control rather than the TLC value, which was argued to be kept constant by trading off speed and steering performance in the work of Godthelp and Winsum [32]. The research carried out by Gruppelaar used an unique approach, hence difficult to compare to the performance of other models [15]. While this work produces good initial results, when analysing the final speed control model performance, it seems that room for improvement is possible. The TETP value will be the central point of focus in the new model while analysing additional visual fixations of humans during a driving task. In order to not be limited to a pre-established model structure, a neural network approach is used, additionally resulting in uncomplicated model input changes.

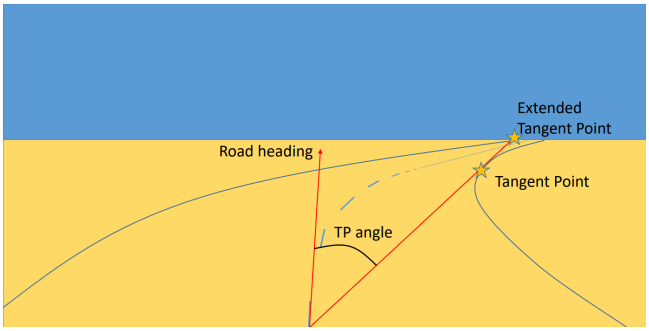


Figure 1. TP, ETP and TP angle representation from a driver's perspective

In contrast to most speed adaptation related research, the visual field of the driver will be used in order to model the vehicle pedal deflections. A representation of the visual cues that are used for the designed model in the perspective of the driver is shown in Fig. 1. The Tangent Point (TP) is the point where the driver's line of sight is parallel to the road edge at the inner part of the road curve. This point is found by computing the angle between the line from driver location to road edge sample point and the road heading line. When the derivative of this angle becomes zero, the TP point is found. The Extended Tangent Point (ETP) is the point where the extended TP line crosses the road edge a second time, usually at the other side of the road. The TP angle is defined angle between road heading and TP point.

2.1. Preliminary Findings

Godthelp and Winsum [32] have shown that drivers trade-off steering performance and speed in order to maintain a contant TLC value. Later studies have shown that for larger curve radii ($> 100m$), the TLC value does not remain constant [7, 25, 28], which is verified by Gruppelaar [15], who points out that the TLC does not appear to be a suitable candidate to use as threshold for speed adaptation.

In a preliminary study it was found that distances to visual fixation points such as the TP and ETP are highly related to the geometry of the road [9]. Field studies have shown that a correlation exists between road geometry (hence distances to visual points) and vehicle velocity [7, 30, 8]. Curves with a small radius are driven at lower speeds and vice versa. A weakness of these models is revealed by considering that road curvature is generally constant in extended curves, while observed velocity profiles tend to vary in the curve. The relationship between curvature and speed predicts one speed value for such a curve, usually the maximum speed in the observations. In reality, a ve-

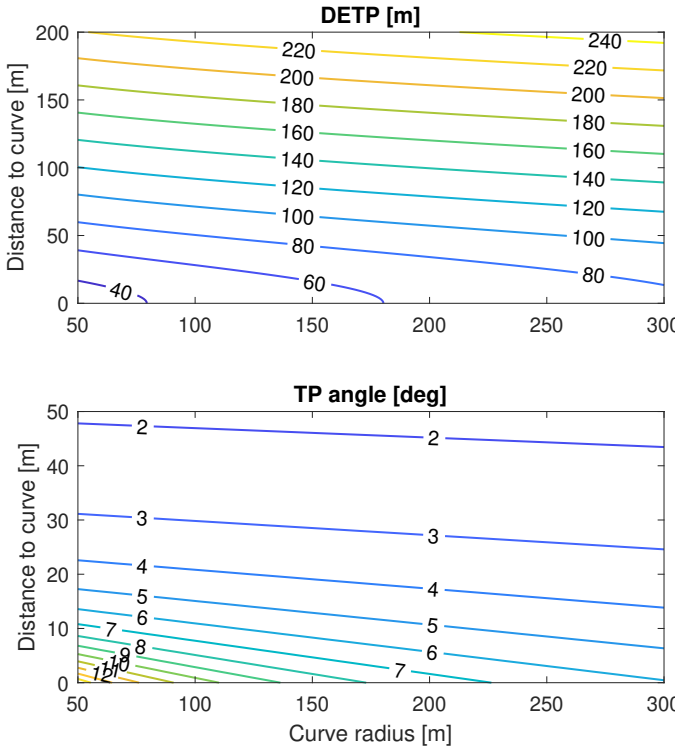


Figure 2. Sensitivity of ETP distance (upper) and TP angle (lower) with respect to changing curve radius and distance (ϕ = 60 degrees, width = 3.4m)

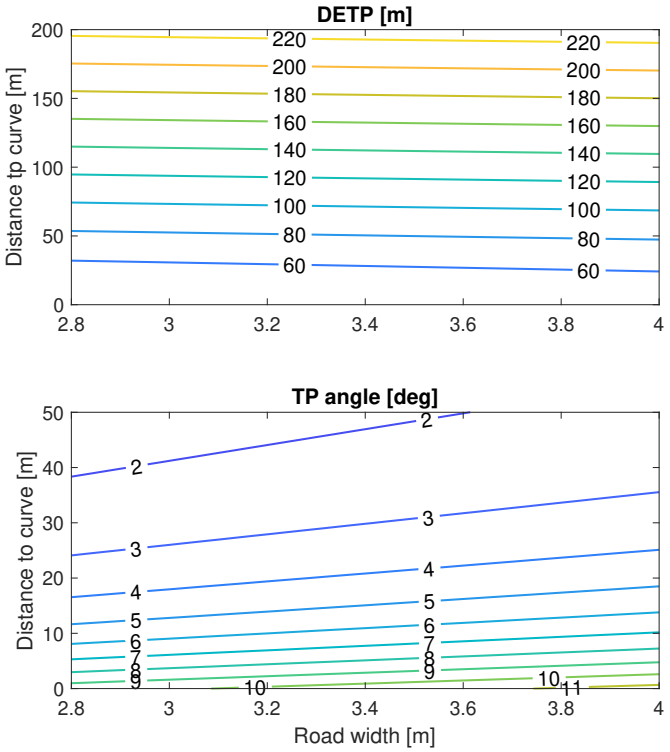


Figure 3. Sensitivity of ETP distance (upper) and TP angle (lower) with respect to changing road width and curve distance (ϕ = 60 degrees, R = 100m)

locity profile can be observed throughout a curve, in which a driver goes through the following phases: acceleration, deceleration, braking, brake release and re-acceleration [15]. In the work of Xie [34] a curve was split into three phases instead of 5. In order to capture these phases in human driving behaviour, road geometry and hence distances to visual fixation points are not good model inputs, as they remain relatively constant throughout a curve, while velocity does not [9].

In the above-mentioned preliminary study a correlation was found between time margins to visual fixation points and accelerator and brake pedal control [9], verifying the findings of Gruppelaar [15]. Furthermore, there are several advantages of using brake and accelerator pedal deflection as model output. First, the pedal deflections are bounded between 0% and 100% eliminating the possibility of unrealistic acceleration values. Secondly, accelerator and brake pedal deflection are direct inputs to the vehicle dynamics. When developing a Speed Control Algorithm, it is not necessary to also develop a (complex) control system computing the pedal deflection input from the model velocity error.

An additional visual cue that shows promising values is the TP angle. In contrast to drivers being poor judges of the Cartesian world (distances, velocities and curvature) [12], they can more accurately judge the optical world (e.g. visual angles). The TP angle is defined as the angle between the road direction at the vehicle location and the TP point. Since field studies have shown that road geometry is correlated with speed [7, 30, 8, 25, 23], it seems that the visual cues that drivers use are geometry sensitive.

2.2. Sensitivity Analysis

A sensitivity analysis of the ETP distance (DETP) and TP angle with respect to road geometry is performed, and is visualized in Fig. 2 and Fig. 3. The sensitivity analysis is done theoretically in a similar way as in the work from Boer and Mulder [6]. Note that the scales for the ETP and TP angle analysis are not the same since the TP angle barely changes at distances above 50m. It is assumed that the driver's view point is in the middle of the road. During this analysis the following findings were discovered. Both the DETP and the TP angle show high sensitivity to curve radius. The closer the vehicle is to the upcoming curve, the larger $\frac{\delta DETP}{\delta R}$ and $\frac{\delta \angle TP}{\delta R}$ are. Additionally, the ETP distance is sensitive with respect to curve distance starting far (200m) away, while the TP angle shows high sensitivity only close (50m) to the curve.

Fig. 3 shows that road width has little to no influence on the ETP distance while approaching a curve while the

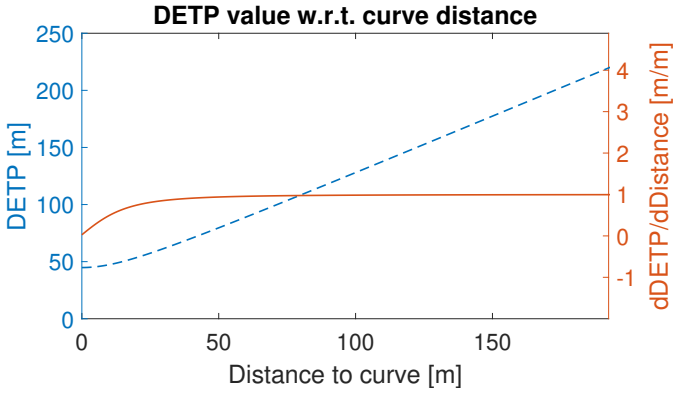


Figure 4. DETP (and derivative) with respect to curve distance

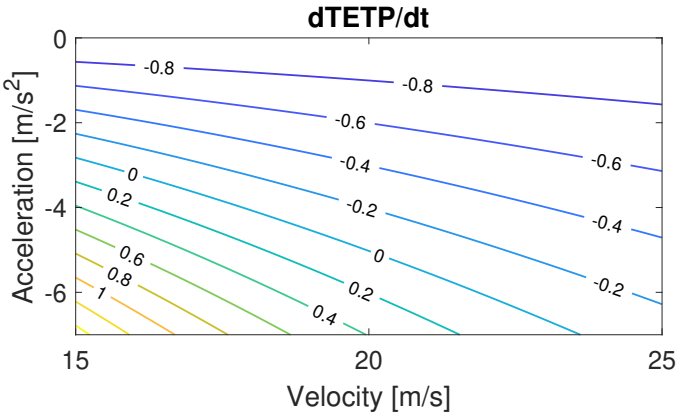


Figure 5. Change in TETP while approaching a curve in the deceleration phase (distance to curve = 50m)

TP angle is a bit more sensitive to changes in road width. In this area the TP angle can be a distinctive input. Both variables show negligible sensitivity towards changing curve deflection (change in road heading) above 20 degrees, which could be a point to improve upon during simulation runs since Gruppelaar [15] has shown variation in average TETP and velocity with changing curve deflection.

It is also important to understand that the sensitivity shown in Figs. 2 and 3 is favorable for a neural network, but not necessarily observable in a similar way by humans. In general humans tend to follow Weber's Law when investigating human perception [11]. This law argues that the perceived change (dS) to humans is proportional with constant K to the magnitude of stimulus S (see Eq. 1). This would mean that when the DETP changes from 40m to 60m, much more change is perceived compared to a change from 200m to 220m, making the DETP more sensitive to humans closer to curves.

$$dS = K \cdot S \tag{1}$$

Although the ETP distance shows high variability with respect to curve radius at close curve distances, this may not necessarily be the case for the TETP as it also depends on vehicle velocity. The change in TETP with respect to time can be calculated using Eq. 2a, in which both the DETP and the velocity are changing in time. In Fig. 4 it can be seen that for distances larger than 50m, the derivative of the DETP with respect to distance is close to 1. Therefore the assumption is made that $\frac{\partial DETP}{t}$ is equal to -V. Also, $\frac{\partial V}{t}$ is the vehicle acceleration (α). Eq. 2a therefore simplifies to Eq. 2b. A contour plot (Fig. 5) is made for a velocity range from 15 to 25 m/s and an acceleration range from -7 to 0 m/s^2 (based on measured data before curves). In this plot a zero-line can be seen, indicating the possibility of a non-changing TETP value with respect to time (upper part of fraction in Eq. 2b equal to 0). This phenomenon often occurs before and during curves (see Fig. 6). Research into speed adaptation in car following and lane keeping has shown that speed control is likely governed by keeping time margins to salient visual points above constant minimum values [32]. This is also found in the experimental data. When approaching a curve, the TETP value drops rapidly (see Fig. 6). At some point (around 50 meters before curve) the driver starts to brake (drop in $\delta_a - \delta_b$), causing the TETP value to become relatively constant initially and then to rise slightly since the change (decrease) in distance to the ETP becomes smaller than the change (decrease) in velocity (below zero-line in Fig. 5). This phase is the braking phase defined by Gruppelaar [15]. In the next phases of the curve (brake-release and re-acceleration in [15]), the driver accelerates to a velocity in which it is comfortable. During this phase (inside the curve), the distance to the ETP remains relatively constant while the velocity increases. This causes the TETP to decrease slightly a second time. At the end of the curve the distance to the ETP increases rapidly, hence also the TETP. This results in a time series where the TETP follows the same trend (slight drop) twice while the output value ($\delta_a - \delta_b$) is negative (braking) in the first drop and positive during the second drop.

$$\frac{\partial TETP}{\partial t} = \frac{\frac{\partial DETP}{\partial t} \cdot V - \frac{\partial V}{\partial t} \cdot DETP}{V^2} \tag{2a}$$

$$\left.\frac{\partial TETP}{\partial t}\right|_{d \geq 50} \approx \frac{-V^2 - \alpha \cdot DETP}{V^2} \tag{2b}$$

Due to the sensitivity of the TP angle with respect to width, radius and curve distance during the braking phase, it is a visual cue that potentially can identify the braking peak magnitude and distinguish between the braking and re-acceleration phase while the TETP (which is fairly constant below certain time margin) identifies when deceleration starts (see Fig. 6).

In mid-curve, after the braking phase (and brake-release), the re-acceleration phase is active. When the velocity increases significantly during this phase, the TETP value will become smaller, resulting in an input change to which the model can react (to stay above a minimum TETP value). If not, the input values will remain relatively constant and the pedal deflection output will too. At curve exit, the TETP

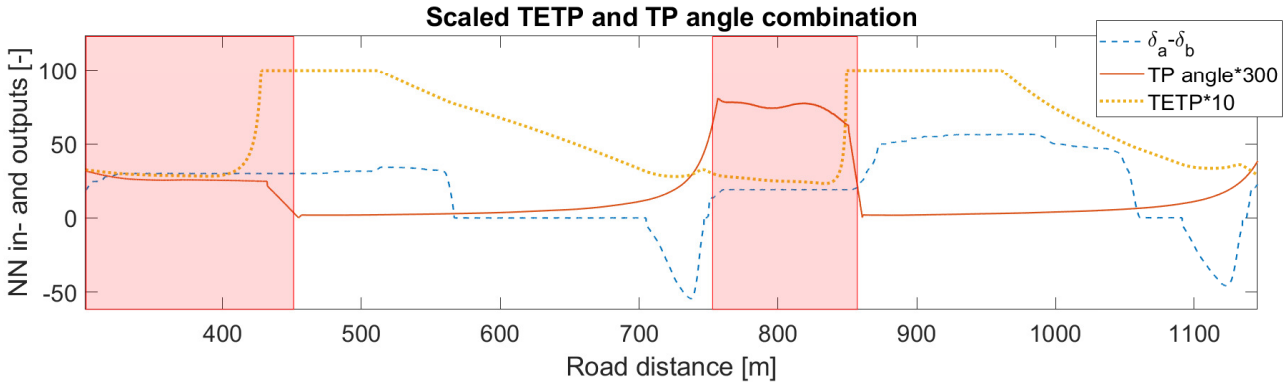


Figure 6. TETP and TP angle behaviour with respect to road distance. Red marked areas are inside a curve.

increases to its maximum value, and the TP angle will drop to close to zero. During this transition, the model can switch from the re-acceleration phase to the acceleration phase. Therefore, during mid-curve and curve exit, it is expected that the model is able to respond accordingly.

3. Model Design

3.1. Model Optimization

The general idea of the design is a model with neural network structure to which visual inputs (TETP and TP angle) are given. Using these inputs, the model will output the desired vehicle pedal deflections. Such a model can be integrated in a speed control algorithm. The complete model architecture is shown in Fig. 7, and consists of multiple components. With the experiments presented in Section 4, the vehicle states were collected during test runs. Together with the road coordinates these states remain unchanged during optimization. To start the optimization of the driving model, the fixed variables are used as input to the visual cues algorithm in which the time, distance and angle to the ETP and TP are generated. The methods for deriving these values are explained and illustrated in Appendix A.

The relevant visual cues (TETP and TP angle) are then used as inputs to a single hidden layer, feed-forward neural network with a sigmoidal activation function. A combination of more inputs was found to more accurately model the test data, but reduced performance on validation data as the dimensionality of the problem increases. This is known as the 'Curse of Dimensionality' [3]. Due to the exponential increase in volume with increasing dimensions, the available data rapidly becomes sparse.

The neural network weights are optimized and updated in the 'First Optimization Loop' of Fig. 7. For this, the second order Levenberg-Marquardt (LM) method is used which uses the Jacobian matrix. A more detailed representation of the neural network and the LM method are found in Appendix B. The neural network uses the Mean Squared Error (MSE) between the measured accelerator and brake pedal deflection (δ_a, δ_b) and the generated pedal deflections

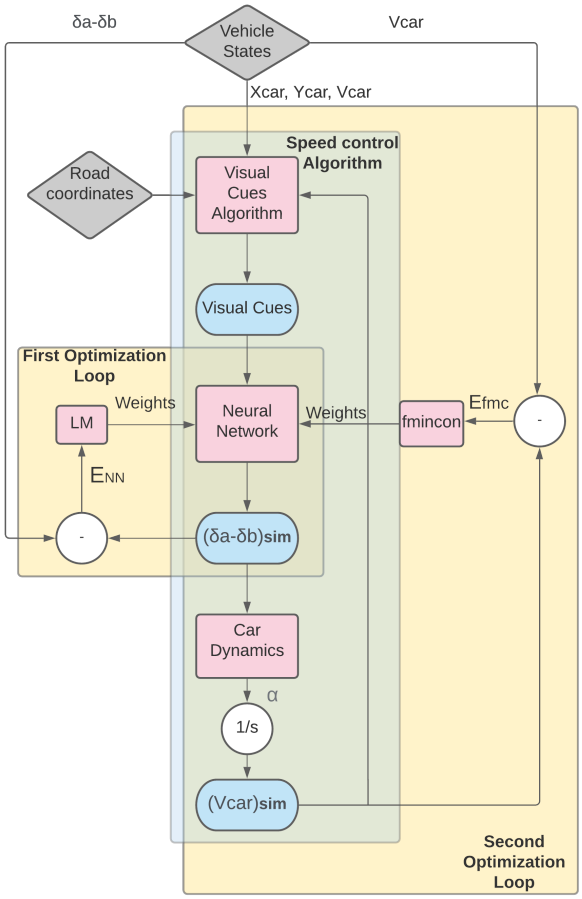


Figure 7. Optimization Schematic (blue/oval = generated variables, pink/square = algorithms, grey/diamond = fixed/measured data, white/round = operators)

of the network itself. Note that the accelerator and brake pedal are never used simultaneous as only one leg is used for both accelerating and braking. Therefore only one output is generated which equals $\delta_a - \delta_b$. When the output is negative, a positive brake pedal deflection is represented.

Some problems may arise when optimizing the neural network weights based on the pedal deflection MSE. A positive brake pedal deflection only occurs for a small period of time relative to the periods of positive acceleration pedal deflection during an experimental run. The optimized model often neglects large braking peaks and focuses more on general acceleration trends as this lowers the MSE with higher significance. Even though braking happens relatively fast, it is considered an important factor in human driving behaviour. In the iterative process of obtaining the velocity profile of a simulation run (Subsection 3.2), large errors between measured and modeled data are accumulated when braking is not taking into account with high enough significance (see Fig. 8). In order to re-direct the focus of the neural network weights on specific periods of pedal deflection which influence the accumulated error of the velocity profile, a second optimization loop is introduced. The second optimization loop from Fig. 7 uses the optimized weights from the neural network as initial conditions. It also uses the speed control algorithm (see Subsection 3.2) to generate a velocity profile from the obtained model. Instead of minimizing the pedal deflection MSE, this optimization has the objective of minimizing the MSE between the measured and simulated velocity on a test track by changing the neural network weights. This is done by a constrained, nonlinear optimization of the objective function (using Matlab's fmincon algorithm). The active-set (medium-scale) method that is used by the fmincon algorithm is the Sequential Quadratic Programming (SQP) method. The implementation of this method consist of three stages. At each iteration an approximation is made of the Hessian (second-order partial derivative square matrix) of the Lagrangian function, which in this case (unconstrained) is the objective function. This is used to generate a Quadratic Programming (QP) subproblem of which the solution is used to generate a search direction for the line search procedure. An overview of SQP is found in Fletcher [13], Gill et al. [16], Powell [27], and Schittkowski [16].

3.2. Speed Control Algorithm

The speed control algorithm is the loop segment (blue marked area) in the optimization schematic (see Fig. 7) in which a velocity profile is produced by the 'self-accelerating' model. Instead of using the generated visual cues from the experimental measurements, the speed control algorithm generates its own visual cues. With the initial vehicle states at t=0 ($V_{car} = 0$, $X_{car} = 0$ and $Y_{car} = 0$), the speed control algorithm obtains a modeled pedal deflection ($\delta_a - \delta_b$) from the neural network. The pedal deflection is used as input to the vehicle dynamical model which returns an acceleration (see Appendix C). Integrating the acceleration over time results a vehicle velocity at the next

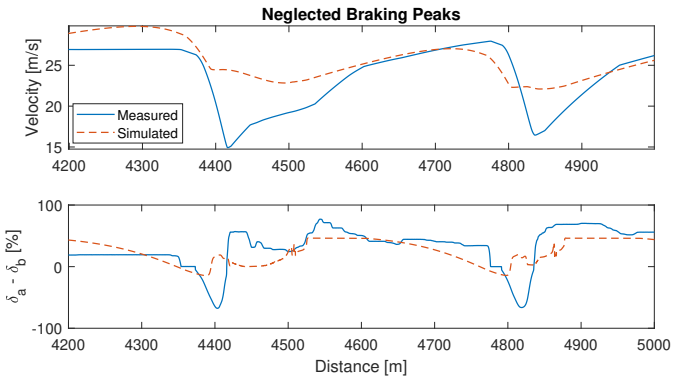


Figure 8. Error in velocity profile resulting from neglected braking in simulation

time interval ($t + \Delta t$). Now instead of using the measured vehicle states to obtain the visual cues at time $t + \Delta t$, the simulated velocity at this time interval is used to generate the next visual cue as after t=0 the simulation is diverging from the measured vehicle states. The represented speed control algorithm loop in Fig. 7 is repeated for each time interval until the end of the test track is reached.

There are two ways in which the speed control loop is used. First, inside the second optimization loop in order to calculate the objective function. For each iteration of the SQP algorithm, the weights of the neural network change resulting in a completely new velocity profile as the velocity is obtained iteratively since velocity at time t is depending on its own value at time $t - \Delta t$ (see Eq. 4). The process of completely going through the speed control loop including the visual cue generator for each SQP iteration can be quite time consuming. In order to reduce the run time in the second optimization loop, a variable Δt is used as opposed to a variable distance to the ETP. With a variable Δt it is possible to keep the sample points of the vehicle location fixed to those of the measurements, hence preserving the distances to the ETP for each sample. The TETP value can in this case be calculated by dividing the distance to the ETP by the vehicle velocity instead of using the visual cue algorithm. The difference in time between two samples is calculated using Eq. 3, which assumes a constant acceleration (α) between time t and $t+\Delta t$. Δs is the (fixed) difference in distance between two sample points.

$$\Delta t = \frac{-V_{car} + \sqrt{V_{car}^2 + 4 \cdot \frac{1}{2}\alpha \cdot \Delta s}}{2 \cdot \frac{1}{2}\alpha} \tag{3}$$

The second objective for which the speed control algorithm is used is to validate the model. In reality, the algorithm does not know the future location and is forced to use the visual cue generator in order to obtain the neural network inputs. A fixed sampling rate is used to gather the vehicle states, hence a fixed Δt is used. The speed control algorithm also assumes constant acceleration between two

Figure 9. Static driving simulator in HMI laboratory (TU Delft)

sample points hence uses the linear approximation from Eq. 4 to update the velocity with fixed Δt.

$$V_1 = V_0 + \alpha \cdot \Delta t \tag{4}$$

The acceleration between two sample points is obtained from the vehicle dynamical model. The inputs of this model are the current velocity and gear, and the pedal deflections δ_a and δ_b. A detailed representation of the longitudinal vehicle model is given in Appendix C.

4. Experimental Method

All data were collected with the fixed-base driving simulator in the HMI Laboratory at the faculty of Aerospace Engineering of Delft University of Technology. The experiments were done in order to satisfy the following four goals:

1) Show that the TETP in combination with the TP angle can capture general trends of human cornering speed adaptation.
2) Analyse the inconsistency of speed adaptation between participants
3) Provide validation data to measure the individual and one-size-fits-all model performance.
4) Deliver a comparison between the classical method of speed adaptation modeling [15] and the machine learning method (neural network).

4.1. Apparatus

The fixed-base simulator, shown in Fig. 9, has an actuated steering wheel, an actuated accelerator pedal and a passive spring loaded brake pedal [17]. Instruments displays (speedometer) are visualized on a 12 inch LCD panel. Three DLP projectors with HD resolution project their image on a three-sided projection screen, and cover a field of view of over 180 degrees [26]. Furthermore, realistic engine sound is played through speakers to aid in speed perception.

4.2. Participants

A total of 15 Participants (13 male, 2 female) took part in the experiment. During one experiment the simulator failed, and therefore no training data could be collected, and is therefore not taken into further analysis. Most participants were young males (24±4 years), with a driving experience of 6±4 years. The average annual kilometers driven varied between 15,000 and 500 km, with a median of 8750 km. There are two outliers in age (58 and 64 years) with driving experience of 40 and 46 years respectively. One participant drives an average of 80,000 km annually.

4.3. Road Design

The test subjects drove on 8 different roads. Two of the 8 roads for model development and 6 for validation purposes. The two roads for training are relatively long (16-17 km) compared to the validation roads (3-4 km). One of the two training roads has a width of 3.6m and the other a width of 3.2m. To avoid mental fatigue, the training roads were split into two and breaks could be held when the participant experienced tiredness and/or concentration problems. Both training roads were randomly generated such that all linear combinations of the following curvature parameters were present:

- Radius = [50 100 200 300] (meters)
- Deflection = [30 60 90 120] (degrees)
- Direction = [left right]

Each turn is 300m apart from the next one, such that the influence of one turn on the velocity in the next curve is low. As opposed to the research from Gruppelaar [15], a more complete set of data was chosen to be collected to provide data with sufficient variation for training the neural network. An incomplete training set may lead to the neural network extrapolating outside or interpolating in large gaps of validation road data, resulting in a poorly predicted output.

The six validation roads are used to measure the performance of the developed speed control algorithm. Each validation road except Road 1 (from 2 to 6) deviates in terms of geometry parameters from the training roads. Validation Road 1 is similar to the training roads in terms of radius, deflection and distance between turns. The width of this road is similar to one of the training roads (3.2m). Validation Roads 2, 3, 4 and 5 all change one road geometry parameter compared to the training roads in order to check whether the model is still valid. Road 2, 3, 4 and 5 have, respectively, a different distance between curves (0, 100 or 200m), a different curve radius set (75, 125, 175 and 250m), a different curve deflection set (15, 45, 75 and 105 degrees), and a different road width (3.4m). The final validation road (6), has all of the above-mentioned modifications implemented in the track, making it the most different from the training road. Validation Road 6 is shown in Fig. 10.

4.4. Experimental Task

All participants were informed of the experimental procedure before the experiment. The subjects were told to

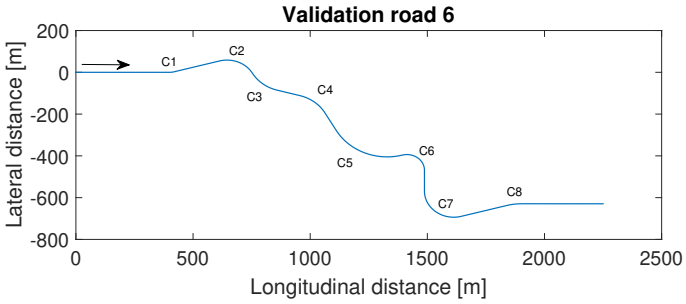


Figure 10. Example of validation road 6

drive as they would in a real life scenario, adapting their speed in order to stay (as much as possible) within the boundaries of the road. The roads were one-way, such that the participants were able to use the entire width of the road. Before the data were collected, the participants had a familiarization run in which they could get familiar with the dynamics of the simulator. Once each participant felt comfortable, the training test tracks were started. Participants were not told what kind of road (training or validation) they were driving and also were not told what type of corners they could expect. This way the subject's speed adaptation was mainly based on visual cues instead of biased ideas of what is ahead on the roads.

5. Results

This section describes how the four goals from Section 4 are satisfied. First results will be shown which demonstrate the ability of the TETP in combination with the TP angle as model input to capture general trends of human cornering speed adaptation on the training data of subjects. These models, trained on individual participants, will be tested on other participants in order to capture the inconsistency between subjects. Some participants show more average behaviour than others. The models of the participants that show more average behaviour are used in order to get a one-size-fits-all model. This model will be compared with the individual models to investigate whether it is possible to generalize human speed adaptation. This will be done using the validation roads which will also indicate what influences the road geometry adjustments have on model performance. Finally, the new one-size-fits-all model will be compared to the classical on-size-fits-all model from Gruppelaar [15]. Model performance is evaluated using the MSE and Variance Accounted For (VAF) between measured and simulated velocity profiles during a specific track. The VAF (also called Explained Variance), is the part of the total variance that is actually present, and not due to error- or residual variance. The VAF indicates how well the model output is related to the actual output without taking the impact of extraneous factors into account. The MSE does not account for whether the error is due to extraneous factors (such as measurement imperfections / noise) or model imperfections. As explained in Section 2 (Fig. 8), some periods of pedal deflections (e.g. braking) have a larger influence on the velocity during curves than others. Therefore, in the analysis of the results, mainly the velocity profiles will be compared.

5.1. Training Performace

As explained in Section 2, the model inputs are the TETP and TP angle. Model optimization is done on the two training roads from Section 4. In Table 1 and 2 the average MSE and VAF values between measured and modeled velocity and pedal deflection are given for each participant. For these results, the speed control algorithm model is compared to the training data of each participant. The performance on the wide road is only slightly better on average. Furthermore, the VAF values for velocity are higher than for pedal deflections (76% and 20% respectively). A reason for this is the human variability/randomness in pedal deflection where the model output remains constant. When for example increasing in velocity with a fluctuating or constant pedal deflection, the deflection shows large differences while the measured velocity only fluctuates slightly around the steadily increasing modeled velocity resulting in lower variance for the error in velocity.

The model from Participant 5 shows a clear difference in performance with respect to the other participant models, as the MSE for the velocity is on average (narrow and wide road combined) 2.22 standard deviations above the average value of 5.21 $(m/s)^2$. In Fig. 11 a comparison between measured and modeled data from Participant 1 is visualized over a 7.5km road (part 1 of test track 1). The performance of this model is a little above average in terms of velocity MSE and close to average in terms of velocity VAF. Interesting to notice in the model behaviour is that there is only a small variability in the acceleration pedal deflection maximums. This can be explained by the fact that the TETP does never exceed 10s, and that there is only a small variability in the TP angle when far away from a curve. Small variations in input lead to only small variations in output. At far distances from curves, pedal deflection behaviour is most likely not entirely governed by outside visual cues as the visual cues TETP and TP angle have little sensitivity at that stage. A cue that is used in the work of Gruppelaar [15] at large curve distances is the velocity. More on the topic of driving phases will be discussed in Section 6.

5.2. Consistency of Participants

In order to come up with a one-size-fits-all model, it is important to investigate the variability between and within participants. The optimized models of each participant will be used as speed control algorithm on the driving path of every other participant. Subjects that have similar driving behaviour will show a lower MSE and higher VAF when a comparison is made between measured velocity data of Participant A and the speed control algorithm optimized

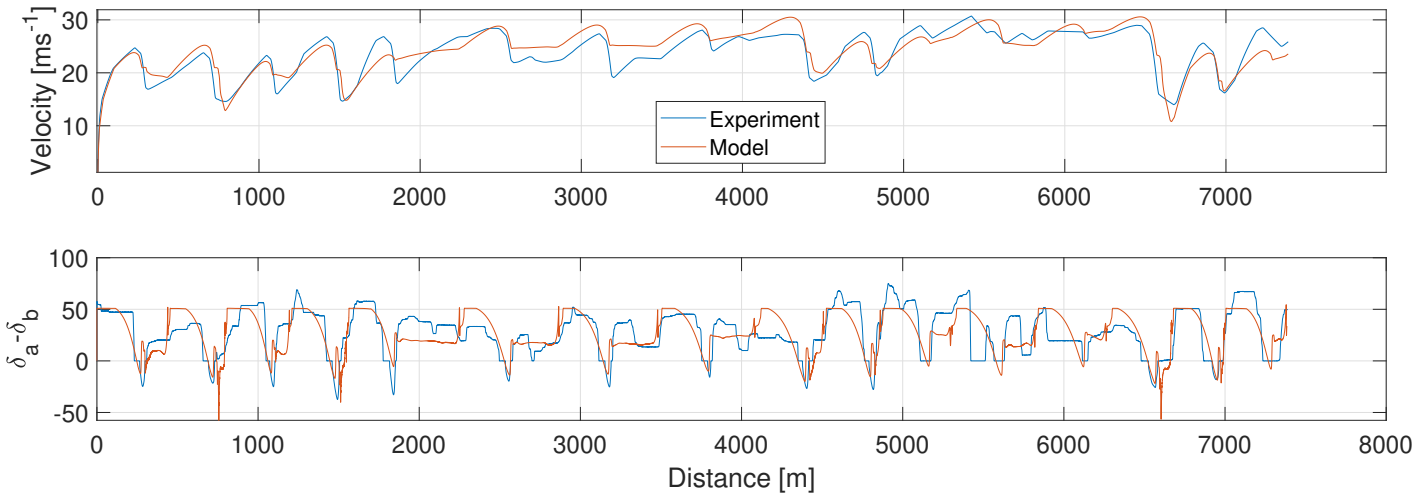


Figure 11. Participant 1, training road 1a, comparison between optimized model and measured data

TABLE 1. TRAINING PERFORMANCE (MSE AND VAF) ON WIDE ROAD (3.6M) FOR EACH INDIVIDUAL PARTICIPANT MODEL

	MSE (v)	VAF (v)	MSE ($\delta_a - \delta_b$)	VAF ($\delta_a - \delta_b$)
S01	4.54	80.60	240.01	37.11
S02	4.33	78.01	541.00	0.00
S03	2.48	87.72	197.35	8.48
S04	3.48	76.88	301.69	50.86
S05	9.72	53.71	263.63	0.00
S06	3.78	80.68	435.49	23.53
S07	5.49	71.39	173.33	6.52
S08	3.89	83.00	171.62	0.00
S09	7.09	66.57	314.86	3.97
S10	5.91	75.42	286.88	0.00
S11	2.86	85.88	173.15	53.16
S12	3.42	85.69	272.54	43.86
S13	4.58	84.23	222.73	18.18
S14	5.38	81.49	348.84	8.38
Mean	4.78	77.95	281.65	18.15
StdDev	1.89	9.11	105.72	19.98

TABLE 2. TRAINING PERFORMANCE (MSE AND VAF) ON NARROW ROAD (3.2M) FOR EACH INDIVIDUAL PARTICIPANT MODEL

	MSE (v)	VAF (v)	MSE ($\delta_a - \delta_b$)	VAF ($\delta_a - \delta_b$)
S01	4.67	77.25	221.39	41.53
S02	2.79	85.04	426.58	5.81
S03	2.87	86.39	260.35	0.00
S04	4.03	70.34	292.67	54.74
S05	11.65	31.55	257.13	0.00
S06	4.53	80.40	267.39	46.65
S07	9.08	58.38	227.90	12.04
S08	8.37	69.38	287.08	0.00
S09	3.31	85.58	168.47	43.95
S10	9.11	61.05	374.43	0.00
S11	1.71	90.89	138.93	63.07
S12	5.75	74.37	743.12	13.38
S13	4.34	83.67	195.33	30.63
S14	6.86	75.65	417.59	1.79
Mean	5.65	73.57	305.60	22.40
StdDev	2.94	15.41	152.31	23.34

with data from Participant B. This analysis is done using the relatively short validation roads (6 in total).

The results for Validation Road 1 are shown in Fig. 12. In this figure a vertical change indicates a model from another participant, and a horizontal change indicates a change in participant validation data. Similar to the results from Tables 1 and 2, the model optimized with data from Participant 5 results in the highest MSE and lowest VAF. The model performance is not only low when used on other participants, but also on Participant 5 itself. Additionally, when data from Participant 5 is used for validation, there exist some models which are a relatively good fit. Therefore, it is safe to conclude that optimization problems (such as being stuck in an undesired local minimum) have occurred when optimizing the model with data from Participant 5. Furthermore, there is not a single participant that shows relatively bad performance on every model for each validation road. This would indicate that such a participant is an outlier with respect to the other participants. The best model fit was found to be optimized with data from Participant 12. Almost every other test subject shows relatively similar speed control behaviour with a few validation road exceptions.

The inconsistency within participants themselves is investigated comparing the validation roads of the participants with the model trained with data of that same test subject (diagonal values in Fig. 12). These values are checked with the values of Table 1 and 2, since for an inconsistent participant it is most likely also more difficult to precisely model their speed adaptation behaviour. The results (velocity MSE and VAF) of this analysis are shown in Fig. 13. First, it is interesting to see the increase in performance for individual models compared to the average performance found in Fig. 12. On average the VAF has increased from approximately 65% to 75% and the MSE has dropped from around 11.7 to 7.5 (when all validation roads are considered). Since the model from Participant 5 has not been optimized properly, this model has not been taken

		VRoad1 - MSE														
participant	test	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
model	1	4.17	6.97	4.66	7.69	11.40	8.41	16.30	8.85	8.13	13.40	18.38	5.57	5.26	4.68	8.85
	2	2.86	3.18	8.84	5.40	11.03	9.27	14.93	10.99	8.59	12.40	23.24	4.63	3.91	7.38	9.05
	3	15.01	16.04	4.76	6.65	8.31	12.78	6.46	1.31	16.99	10.74	8.63	12.49	12.98	7.45	10.04
	4	12.12	11.81	13.16	3.65	20.41	13.34	8.32	11.54	19.79	19.25	11.41	11.36	14.41	17.48	13.43
	5	35.90	31.44	24.73	20.56	44.45	35.38	18.70	18.80	38.44	30.60	22.33	32.65	33.88	29.36	29.80
	6	2.99	5.39	8.11	8.91	11.56	6.33	21.30	14.14	10.77	10.38	22.42	7.94	6.87	6.44	10.25
	7	21.00	20.51	12.53	7.12	28.66	18.12	5.31	3.13	23.03	25.36	7.14	17.57	17.34	14.65	15.82
	8	20.79	23.56	6.64	14.28	11.92	15.27	11.10	3.30	20.63	14.32	8.80	16.47	17.60	10.02	13.91
	9	10.68	9.95	4.76	6.81	16.58	10.44	9.57	2.20	11.00	10.92	11.56	8.23	6.88	5.39	8.93
	10	23.29	22.68	8.77	9.73	28.55	18.95	5.77	3.08	24.95	16.81	5.31	18.89	20.77	13.28	15.77
	11	16.93	17.29	13.17	7.30	13.97	12.80	5.06	5.31	20.40	19.56	4.14	12.89	15.24	12.81	12.63
	12	4.28	6.16	4.85	5.01	3.18	7.23	13.87	6.60	8.90	10.13	15.83	5.23	5.57	4.92	7.27
	13	6.97	8.85	9.42	9.93	12.02	13.15	17.28	6.33	7.83	14.73	20.64	7.19	4.99	3.92	10.23
	14	6.26	9.99	3.53	10.67	12.64	7.51	17.13	4.89	6.45	5.83	20.74	6.17	4.57	2.31	8.48
Mean		11.33	12.49	7.94	7.93	14.63	11.82	11.72	6.28	14.42	14.14	13.71	10.36	10.49	8.52	11.13

	VRoad1 - VAF														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
1	84.39	57.80	73.23	47.70	38.67	59.04	76.38	76.78	77.10	43.23	87.81	72.79	81.32	80.38	8.33
2	89.46	81.98	45.18	76.23	37.18	53.29	81.54	66.31	72.83	48.45	79.79	75.85	83.78	67.61	68.53
3	60.57	38.90	79.90	42.59	68.72	55.24	75.49	93.44	71.56	69.22	79.06	58.31	76.25	87.07	68.31
4	89.19	75.06	26.90	68.12	24.88	59.08	73.93	43.06	64.41	40.63	61.02	70.27	70.04	44.36	57.93
5	0.00	0.00	0.00	0.00	0.00	0.00	5.77	8.28	18.07	1.97	0.00	0.00	12.88	16.17	4.51
6	85.75	65.01	58.84	58.31	32.78	67.89	70.83	69.74	64.23	54.09	83.06	58.73	70.99	72.39	65.19
7	60.19	41.44	60.22	42.59	24.52	48.98	77.05	86.63	69.93	50.26	71.19	55.36	74.15	76.96	59.96
8	43.48	2.28	74.19	0.00	55.00	55.72	47.99	84.39	71.76	60.96	64.98	49.66	69.63	87.10	54.80
9	69.26	55.19	74.61	41.50	32.61	56.66	79.14	91.38	78.54	64.57	88.83	66.75	84.67	85.69	69.24
10	55.98	41.75	78.13	35.81	36.21	51.60	70.69	90.55	70.31	62.60	75.85	54.03	73.72	86.67	63.14
11	85.71	63.03	61.83	46.63	75.84	77.31	76.82	75.05	80.65	60.13	89.59	78.12	85.42	80.66	74.06
12	88.43	68.60	69.91	64.74	81.80	67.03	78.30	82.13	76.31	57.46	88.77	75.41	81.80	80.48	75.80
13	66.23	43.93	44.38	38.63	36.27	34.91	80.65	91.98	75.73	42.14	81.19	62.52	80.20	83.77	61.61
14	69.83	36.13	79.29	22.96	29.85	62.32	68.57	91.98	81.05	74.91	81.52	67.96	82.54	90.50	67.10
Mean	72.96	51.62	63.59	45.06	44.18	57.62	73.65	80.26	73.42	56.05	79.44	65.06	78.04	78.74	65.69

Figure 12. Variability table for validation road 1 (MSE above and VAF below for velocity). up → down: the model is trained with data from Participants 1-14. left → right: the model is tested with data from Participants 1-14. Model of Participant 5 not taken into the mean values

into account in both cases. As predicted beforehand, the most inconsistent speed adaptation behaviour is found for Participants 7, 8 and 10. When looking in Table 1 and 2, these participants also show the highest training error with respect to the other participants (ignoring Participant 5). The most consistent participant is found to be number 11. Again this can be validated by the training error from Table 1 and 2, as this participant has the lowest training error and consistent speed adaptation behaviour data are easier to model with higher accuracy.

5.3. Validation

From Fig. 12 and similar tables from other validation roads it becomes clear that the model trained with data from Participant 12 has the most identical speed adaptation behaviour with respect to the other participants since it has the highest average VAF and lowest average MSE when applied to other participants. Initially, this model would be used as initial condition after which the model would be further optimized using data from each participant. However, due to the excessive computational power required by the optimization algorithm, this optimization step is abandoned. Instead of being an initial model, this model is used as a one-size-fits-all model as high average and general performance is already obtained. The one-size-fits-all speed control algorithm has been used on the routes driven by all participants and the velocity profiles between participants and model have been compared in Fig. 14. Interestingly, the model achieves on average almost identical performance values in terms of MSE and VAF compared to the individual models. For some participants, this one-size-fits-all model performs

	Validation Road						
MSE	1	2	3	4	5	6	**Mean**
S01	4.17	3.05	7.22	4.07	5.79	2.65	4.49
S02	3.18	4.41	1.76	3.77	2.82	7.39	3.89
S03	4.76	10.99	2.08	6.37	12.29	5.60	7.02
S04	3.65	3.37	1.33	3.70	3.32	7.41	3.80
S05	44.45	33.99	8.91	20.35	16.98	14.05	23.12
S06	6.33	5.63	10.58	4.70	6.26	9.27	7.13
S07	5.31	13.09	8.44	13.33	16.26	15.84	12.04
S08	3.30	8.81	5.02	10.11	9.74	13.43	8.40
S09	11.00	20.62	5.43	12.57	3.96	4.78	9.73
S10	16.81	12.57	12.33	16.12	15.79	17.72	15.22
S11	4.14	2.11	3.11	1.54	2.46	2.99	2.73
S12	5.23	8.35	5.06	7.65	13.66	7.93	7.98
S13	4.99	16.68	2.45	5.28	5.49	3.58	6.41
S14	2.31	28.40	4.97	5.91	3.44	14.06	9.85
Mean	5.78	10.62	5.37	7.32	7.79	8.66	7.59
stdDev	3.94	7.73	3.45	4.40	5.10	5.07	3.56
VAF							
S01	84.39	90.70	62.58	85.77	84.85	90.88	83.20
S02	81.98	83.60	88.16	84.02	90.28	84.38	85.40
S03	79.90	91.07	90.51	82.83	79.50	88.17	85.33
S04	68.12	84.57	89.28	87.15	88.51	83.61	83.54
S05	0.00	0.00	41.75	13.47	45.11	4.30	17.44
S06	67.89	63.41	24.19	78.09	70.21	69.08	62.14
S07	77.05	36.79	61.17	59.70	60.36	20.96	52.67
S08	84.39	30.55	73.85	53.33	31.87	0.00	45.67
S09	78.54	80.17	86.82	75.36	92.05	86.49	83.24
S10	62.60	57.46	55.65	66.67	61.48	55.47	59.89
S11	89.59	94.15	92.76	94.00	91.95	79.17	90.27
S12	75.41	80.69	83.92	76.27	76.80	77.31	78.40
S13	80.20	70.51	89.43	83.63	82.74	84.75	81.88
S14	90.50	36.25	79.15	84.68	91.33	63.54	74.24
Mean	78.51	69.22	75.19	77.81	77.07	67.98	74.30
stdDev	8.357	22.48	19.75	11.61	17.48	27.83	14.32

Figure 13. Performance (MSE and VAF for velocity) of individual models on all validation roads

even better than their own individual model (e.g. Participants 5, 6, 7, 8, 10, 13 and 14 in terms of MSE). This indicates that there is a large chance that those individual models got stuck in an less-desired local minimum during the second optimization loop. This shows that the optimization is sensitive to the given initial conditions that resulted from the neural network optimization.

The participant with the least identical behaviour to the model is Participant 11 as it is 2.3 standard deviations from the average velocity MSE value. It is interesting to see that that same participant is also the most consistent driver, with lowest training error and validation error on its own optimized model. The changing characteristics for validation roads 2-5 are curve distances, radii, deflection and road width respectively. Validation Road 6 has all parameters changed. What is also seen in the results is that worst performance is achieved in Validation Road 2, when looking

	Validation Road						
MSE	1	2	3	4	5	6	**Mean**
S01	4.28	2.98	5.80	5.20	8.32	2.65	4.87
S02	6.16	4.72	3.81	5.59	3.27	8.89	5.41
S03	4.85	10.95	2.55	5.54	12.76	5.64	7.05
S04	5.01	3.32	6.06	5.76	4.15	2.66	4.49
S05	3.18	3.21	2.70	4.70	2.49	3.28	3.26
S06	7.23	3.06	10.04	3.96	6.62	10.46	6.89
S07	13.87	6.33	4.63	3.76	7.25	3.64	6.58
S08	6.60	10.46	12.77	7.77	10.74	5.59	8.99
S09	8.90	16.62	5.59	14.31	6.85	6.58	9.81
S10	10.13	5.38	7.76	9.62	16.44	5.94	9.21
S11	15.83	13.67	18.03	10.13	11.13	11.34	13.36
S12	5.23	8.35	5.06	7.65	13.66	7.93	7.98
S13	5.57	7.77	3.75	5.96	7.12	5.17	5.89
S14	4.92	17.93	7.42	6.11	7.25	8.56	8.70
Mean	7.27	8.20	6.86	6.86	8.43	6.31	7.32
stdDev	3.71	5.07	4.27	2.87	4.05	2.81	2.61
VAF							
S01	88.43	92.23	68.86	80.71	83.71	90.45	84.06
S02	68.60	84.56	78.05	80.03	87.33	82.08	80.11
S03	69.91	71.39	89.54	74.94	61.61	61.98	71.56
S04	64.74	81.05	62.42	66.04	78.95	79.28	72.08
S05	81.80	89.57	86.22	82.72	90.84	78.21	84.89
S06	67.03	79.56	35.99	81.69	72.26	82.24	69.80
S07	78.30	76.99	76.19	82.54	72.94	79.48	77.74
S08	82.13	23.23	70.22	63.26	48.63	46.23	55.62
S09	76.31	63.14	78.61	59.72	79.99	72.19	71.66
S10	57.46	67.52	61.78	61.40	41.68	74.77	60.77
S11	88.77	89.07	88.43	89.27	91.53	76.08	87.19
S12	75.41	80.69	83.92	76.27	76.80	77.31	78.40
S13	81.80	81.25	83.20	81.78	78.38	78.35	80.79
S14	80.48	49.36	67.17	80.83	84.13	69.88	71.98
Mean	75.80	73.54	73.62	75.80	74.91	74.89	74.76
stdDev	9.159	18.5	14.24	9.331	14.9	10.52	8.998

Figure 14. Performance (MSE and VAF for velocity) of the one-size-fits-all model on all validation roads

	Validation Road					
	1	2	3	4	5	6
S01	0	0	0	0	0.95	0
S02	0	0	0	0	0	0
S03	0	0	0	0	0	0
S04	0	0	0	0.55	0	0
S05	2.65	0	0	3.38	0	0
S06	0	0	2.19	0	0	0
S07	0	0	0	0.74	0	0
S08	0	0	0	0	0	0
S09	0	4.88	0	0.97	2.35	0
S10	0	0	0	0	0	0
S11	0	0	0	0	0	0
S12	0	0	0	0	2.93	0
S13	0	2.39	0	0	0.3	0
S14	0	4.33	0	0	0	0

Figure 15. Time (seconds) of being off-road on the validation roads for each participant

at the individual and the one-size-fits-all model. Validation road 2 has different distances between curves with respect to the training data. However, it is not possible to conclude that changing the distances between curves leads to worse performance. Validation road 6 also has different distances between curves, yet the performance maintains relatively high. Most low performing validation roads are caused by either the driver being off-road for a significant amount of time or having a different average speed while the behaviour is still very similar. When the driver is off-road, the visual cue algorithm is not able to generate the distances and times to visual cues and is set to a constant value. If the optimized neural network weights are set in such a way that heavy braking occurs for this set constant value, the algorithm tends to brake to zero velocity, after which it accelerates again. Being off-road for a prolonged amount of time causes this phenomenon to repeat itself, causing oscillations in the velocity profile at low speeds. An example of this is shown in Fig. 16. This is also the case for Participants 6 (Road 3), 9 (Road 2 and 5) and 12 (Road 5), which is also shown in the additional results (Part II of report, Section B.1). Note that being off-road does not always lead to large errors. For example when it occurs at a moment in which the driver is braking in a similar way as the model has learned, or when the braking is not sufficient enough to bring the vehicle to zero velocity (e.g. Participant 1, Road 5). An example of a large error due to a different average velocity is shown in Fig. 17. This is also the participant (11) with the least identical behaviour compared to the one-size-fits-all model (MSE), yet the highest VAF value. As seen in this figure, and the other validation roads in Part II, Section B.1, driving is more slowly compared to the model. The final example shown is an example where the model shows very comparable driving behaviour as the participant to which it is compared. This is shown in Fig. 18.

5.4. Comparison to the Gruppelaar Model

The TETP model input for the speed control algorithm is used as it turned out to be an important cue from the work of Gruppelaar [15]. The developed model has changed from a classical approach to a Machine Learning (neural network) approach such that inputs can be easily added or changed without having to change the complete model architecture and to remove the hard thresholds on driving phases.

The results of the heuristic model by Gruppelaar [15], that is applied to the data from all participants on the various validation roads, are shown in Fig. 20. When using the machine learning approach, in general, the performance in terms of VAF and MSE is increased. When comparing the Gruppelaar model to the Machine Learning (one-size-fits-all) model on the six validation roads from Section 4. The average VAF for the Machine Learning approach is almost 20% higher (75%-56%) and the average MSE around 4 $(m/s)^2$ (11.5 - 7.3) lower. When looking at the standard deviation of the MSE and VAF (last row in Figs. 14 and 20), the Gruppelaar model has a slightly lower value for

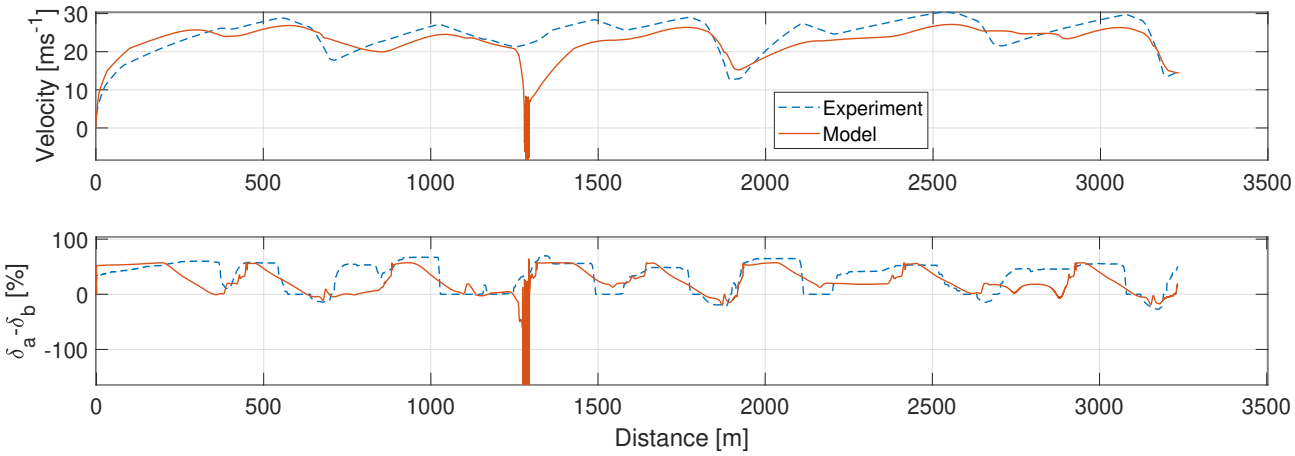


Figure 16. Flawed model example 1 (Participant 9, Road 4): Off-road data leading to oscillations in δ_a, δ_b and V close to zero velocity.

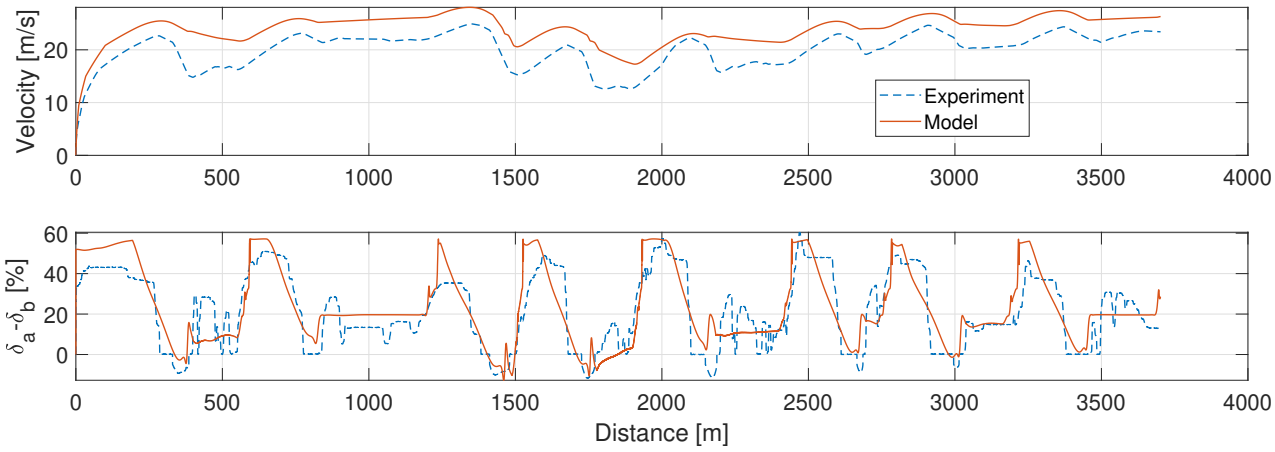


Figure 17. Flawed model example 2 (Participant 11, Road 3): Velocity shift between model and experiment

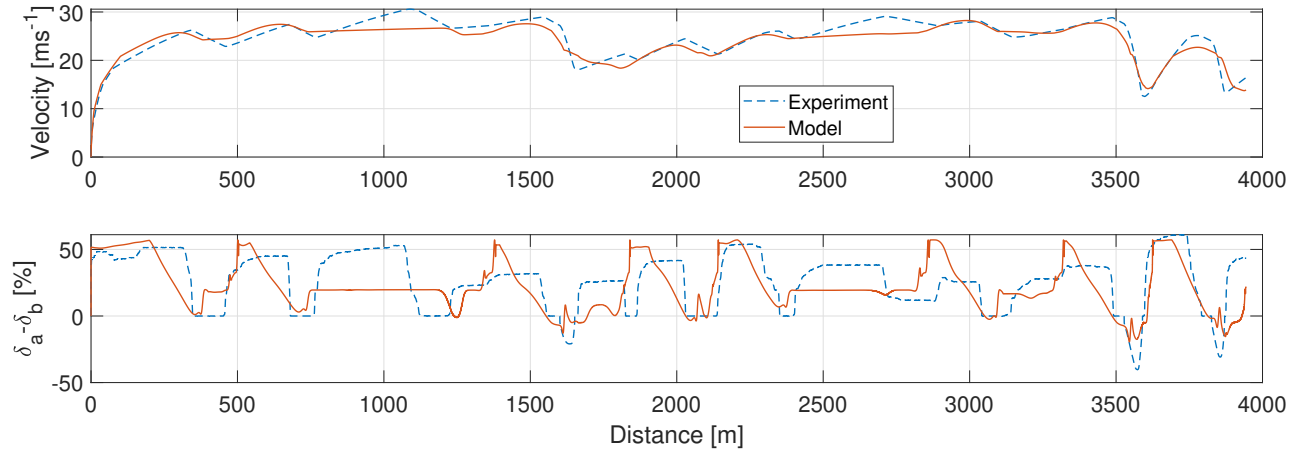


Figure 18. Accurate model example (Participant 4, Road 5)

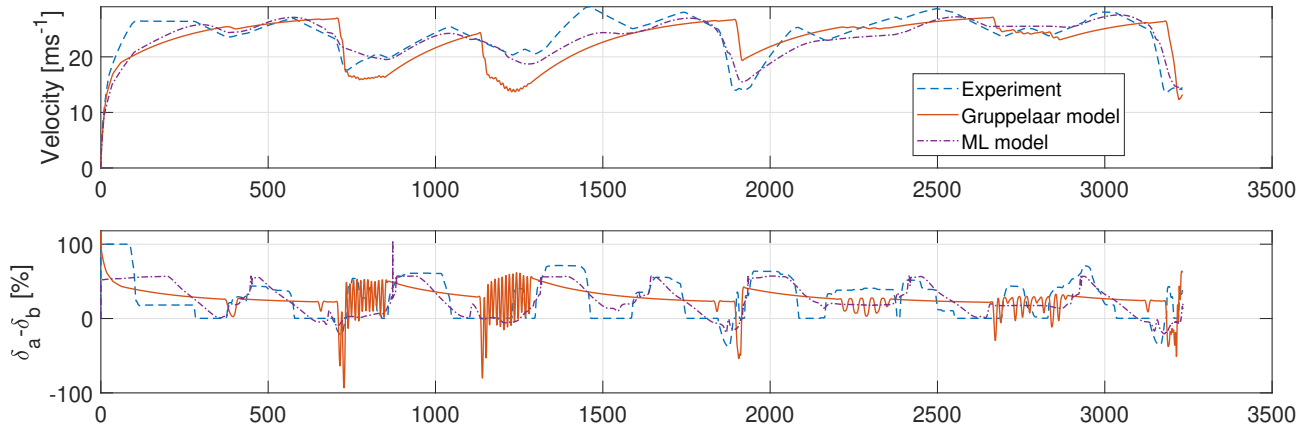


Figure 19. Comparison between classical model, machine learning approach and the measured data (Participant 6, Road 4)

the MSE stdDev (2.04 instead of 2.61) and a slightly higher value for the VAF stdDev (10.42 instead of 9.00).

In the analysis in the work of Gruppelaar, an average VAF value of 38.1 was achieved on validation roads driven by participants on which the heuristic model was developed (presented in [15]). The performance of this model increases in performance when only a single isolated turn is taken for analysis (average VAF = 63.85 [15]). His one-size-fits-all model produces quite variable results for the different participants (VAF standard deviation = 29.12 [15]). Furthermore, as discussed in his work, oscillations in pedal deflections close to turn exit are found due to a problematic implementation of a hard threshold on the rate of change of the TETP that determines which phase the vehicle should be in (see Fig. 19). Working without a model that first determines the driving phase, this phenomenon does not occur in the machine learning approach.

6. Discussion

6.1. Model Performance

Overall, the speed control algorithm shows its capability of capturing general longitudinal driving behaviour trends of the participants. The one-size-fits-all model can be applied to almost all participants with an average VAF of almost 75% and a MSE of just over 7. The sensitivity of the TP angle input seems to respond correctly to the driving phase shifts close to and in the different curves. There are however a few flaws in the model still. First of all, no different driving phases are separated explicitly in order to avoid hard thresholds on model inputs. In the case of the test tracks from Section 4, this model still works relatively well as the curves are only 250m apart, causing the model inputs to never be insensitive for a prolonged period of time. The insensitivity of the inputs can already be seen for some short periods of time between curves, as this results in an relatively constant model output in these stages (see $\delta_a - \delta_b$ in Fig. 11). In order to achieve higher accuracy on all road types, including longer straight parts, it would be better to characterise different driving phases, each dependent on different inputs. In long straight roads, the visual cues discussed in this research remain fairly fixed. As shown in the work of Gruppelaar [15] above a certain TETP threshold, speed adaptation is most likely governed by the vehicle velocity. This is an input for drivers which can be obtained by the optical flow field (roughly), or by looking at the speed indicator inside the vehicle. Also, inside a curve, speed control is most likely governed by the maximum allowable lateral acceleration drivers set for themselves for different velocities [28].

On the other hand, it is not the case that no driving phases are identified with the used model approach. By setting up the neural network and the second optimization loop (Section 2), it gives the model the possibility to come up with a driving phase determination by itself. The speed control algorithm is able to, for example, distinguish between whether to be in the braking phase or not without having to set hard thresholds on model inputs. Without hard thresholds, the oscillation problem in model output does not occur since no fluctuations between driving phases are possible (see Fig. 19). A trade-off can be made in terms of model independence and dimensionality. Due to the various driving phases being present and being identifiable by different model inputs, it becomes difficult to develop a fully independent model (without pre-determining phases) as more inputs are required. This leads to the curse of dimensionality [3], as with more model inputs the available data quickly becomes sparse.

A second change that could improve model performance with respect to reality, is by introducing car dynamical cues to the driver. In a research from Reymond et al. [28] the role of lateral acceleration while driving is investigated. In their work they show that there exists a relation between velocity and maximum lateral acceleration. The faster a subject drives, the lower their maximum allowable lateral

	Validation Road						
MSE	1	2	3	4	5	6	**Mean**
S01	11.56	9.05	8.70	9.63	14.16	4.94	9.67
S02	11.49	11.15	13.10	10.45	10.31	14.35	11.81
S03	12.84	15.15	9.62	10.47	12.28	10.34	11.78
S04	8.81	7.71	6.08	7.99	7.29	7.88	7.63
S05	7.02	8.16	11.35	12.57	7.00	8.03	9.02
S06	15.29	8.74	9.32	12.17	10.80	16.24	12.09
S07	12.61	9.56	9.66	11.14	9.91	6.46	9.89
S08	10.32	9.68	11.63	9.39	11.06	10.01	10.35
S09	16.65	21.81	13.65	16.36	7.99	8.72	14.20
S10	13.57	6.68	15.86	12.26	13.15	9.62	11.86
S11	17.83	10.91	12.65	10.55	9.65	12.36	12.32
S12	11.23	11.33	15.17	15.57	14.54	8.89	12.79
S13	11.51	12.96	9.31	12.93	11.41	10.43	11.43
S14	13.66	14.89	19.83	13.52	12.67	18.02	15.43
Mean	12.46	11.27	11.85	11.78	10.87	10.45	11.45
stdDev	2.89	3.95	3.53	2.33	2.38	3.67	2.04
VAF							
S01	52.65	75.28	68.29	66.75	70.25	87.19	70.07
S02	39.73	73.29	45.51	75.16	62.95	74.69	61.89
S03	16.19	52.26	39.76	51.70	57.25	34.15	41.89
S04	25.51	55.88	49.42	54.96	53.73	43.52	47.17
S05	67.56	77.33	56.42	52.84	76.87	55.84	64.48
S06	31.89	52.64	70.37	45.58	66.56	73.75	56.80
S07	48.87	54.14	51.85	49.15	66.11	69.30	56.57
S08	45.16	29.36	38.92	46.36	34.41	12.29	34.42
S09	51.32	65.86	62.14	48.61	74.18	61.58	60.61
S10	41.82	57.74	32.29	46.26	52.01	62.78	48.82
S11	57.98	75.92	64.22	74.02	76.28	54.67	67.18
S12	46.14	84.86	57.11	45.53	75.12	83.08	65.31
S13	57.45	74.28	64.71	58.67	65.70	54.29	62.52
S14	40.34	59.32	23.53	55.35	74.17	46.78	49.91
Mean	44.47	63.44	51.75	55.07	64.69	58.14	56.26
stdDev	13.54	14.58	14.22	10.15	11.98	19.99	10.42

Figure 20. Performance (MSE and VAF for velocity) of classical model applied to validation road 1-6

acceleration levels happens to be. In their work they also show the different driving behaviour for static and dynamic simulators. From the results, one can conclude that when lateral acceleration cues are included, drivers show higher sensitivity to these cues by adjusting the maximum allowable levels.

Finally, in order to improve model performance further, a separate off-road model can be implemented. During test runs, identification of some road parts have been challenging for participants, causing them to drive off-road for a period of time. During the off-road parts, the visual cue algorithm was not able to generate a correct visual cue distance and time, causing the algorithm to output a constant predetermined value. Because participants tend to brake when driving off-road, the optimized model also decelerates quickly. Since the model inputs are constant off-road, the speed control algorithm often brakes back to zero velocity. Oscillations in velocity close to zero velocity can be seen in Fig. 16 for an off-road part.

6.2. Model Approach Comparison

Using a neural network approach has lead to some advantages and some disadvantages compared to the classical modeling approach. An advantage of the classical approach is that it is more robust. When defining equations to build the model, one knows the output value for each possible given input, and can check whether this value is accurate. With the neural network approach, it is more difficult to validate each possible model input, as not all input combinations exist in the validation and training data. Secondly, when using the classical approach, it is easier to implement more model inputs as the curse of dimensionality does not influence the model development as much as for a neural network. When data becomes sparse, a neural network may produce unrealistic output values in places where little to no training data exists.

An advantage of using the machine learning approach is that one does not have to set hard phase determination thresholds. Furthermore, a neural network makes it convenient to implement various model inputs without changing the entire model architecture.

6.3. Future Work

As discussed before, future improvements directly related to this research can be the implementation of driving phases and he use of a dynamical driving simulator. As explained in Section 1, comfortability and trust is a mayor factor in human-automation interaction. A next step would therefore be to implement the developed speed control algorithm into the driving simulator. Participants would not need to accelerate themselves, but only steer the simulated vehicle. A human-machine interaction research could be conducted on these experiments and experiences of the participants in terms of comfortability can be investigated.

The model can also be used to work towards improvements in current ADAS systems such as a haptic pedal assistance system, similar to the one developed by Mulder et al. [24] for car following purposes. Such a system would be able to give force feedback on the gas pedal and or brake brake pedal in such a way that the driver can feel the severity of upcoming turn. Development of such systems can greatly improve the safety and driving performance on curved roads.

Further development of the speed control algorithm can also include steering behaviour in order to build towards more autonomous driving systems. These systems would need more models developed in various environments. Instead of just high speed curved roads without any traffic, one could include traffic, road signs, different speed limits, traffic lights, elevated roads, multiple road lanes etc.

7. Conclusion

The objective of this research was to develop a one-size-fits-all speed control model that uses inputs from the

observable visual domain. This research is a follow up project on the work of Gruppelaar [15] in which a heuristic model was developed relating the TETP to vehicle pedal deflections. Input examples that have been found observable by drivers are visual angles and time margins to visual points. The TP angle has shown increased sensitivity at moments where the sensitivity of the TETP decreases. Using these two observable cues as inputs to a neural network, the developed speed control algorithm is able to accurately capture longitudinal driving behaviour from the participants. The advantage of using a neural network model architecture is that no hard thresholds are set on the input variables, removing the phase oscillations earlier present in the pedal actuation. As opposed to almost all previous research, by using the pedal actuation as output, the model could be directly applied to the self-accelerating algorithm. In order to improve the model further, driving phases that are dependent on different input variables could be implemented. A dynamic simulator could be used to increase the fidelity of the performed experiments. improved models can be used in the future to increase autonomy in driving by developing Advanced Driver-Assistance Systems such as haptic pedal assistance.

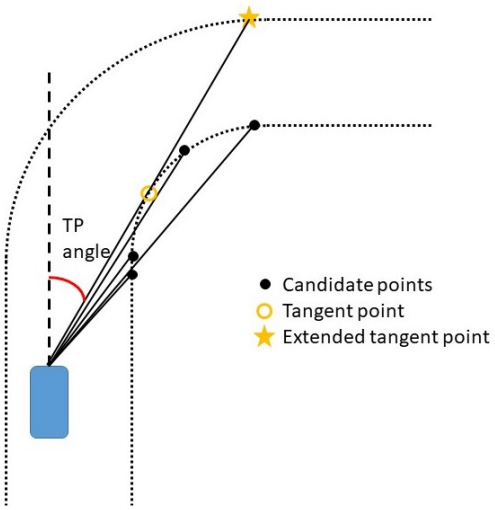


Figure 21. Detection of TP and ETP distance and angle

A. ETP detection

In order to have the speed control algorithm accelerate, it must been given the inputs TETP and the TP angle. In real life this could be done using one or more cameras mounted to the front of the car. An algorithm uses these camera images in order to generate the TETP and TP value. Such an algorithm is developed by Gallen and Glaser [14]. Their algorithm samples various angle points on both the right and left road edges. it calculates the derivative of these angles and finds the zero-crossing. This means that the TP angle is getting smaller further away, where after it gets larger at some point in the inside of a curve. The point of zero-derivative is the angle where the TP point is. the TP is found by calculating the distance of the road edge at this angle with respect to the vehicle location.

The same method is used mathematically, without camera, for the visual cue algorithm discussed in Section 2. The TETP is found by extending the line with the same TP angle until it intersects with another road edge. Note that this is usually the other side of the road, but can also be the same side of the road when two opposite curves are close together. A visual representation is shown in Fig. 21.

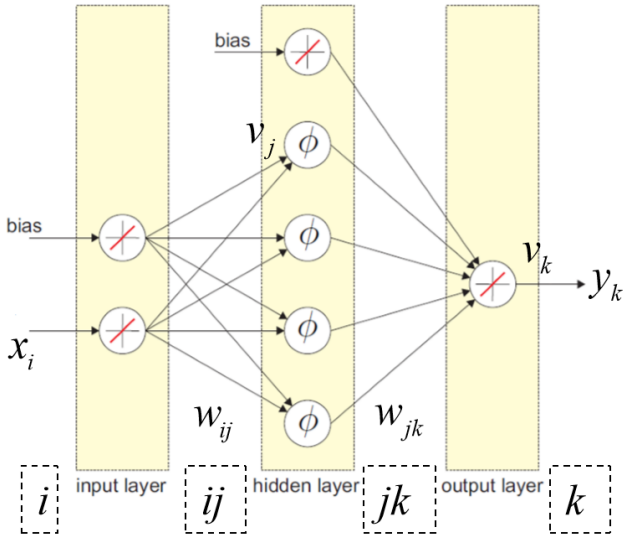


Figure 22. General description of single hidden layer neural network

B. Neural Network

B.1. Network Structure

For optimization, a single hidden layer feed-forward neural network structure is chosen with symmetric sigmoidal activation function. The general representation of such a network is shown in Fig .22. The input x_i consists of the TETP and the TP angle. The output y_k equals the accelerator minus the brake pedal deflection ($\delta_a - \delta_b$). First the input data is normalized by Eq. 5a. After normalization, the input x_{norm} is multiplied with weight matrix W_{ij} and the bias b_1 is added ($= x_j$). x_j is used as input to the hidden layer symmetric sigmoidal activation function (see Eq. 6). The output value of the activation function (y_j) is multiplied with weight matrix W_{jk} and bias b_2 is added resulting in the normalized neural network output. The normalized output is de-normalized using Eq. 5b.

$$x_{norm} = (x - x_{min}) \cdot \frac{y_{max} - y_{min}}{x_{max} - x_{min}} + y_{min} \tag{5a}$$

$$y = (y_{norm} - y_{min}) \cdot \frac{x_{max,2} - x_{min,2}}{y_{max} - y_{min}} + x_{min,2} \tag{5b}$$

$$x_{max,min} = \begin{bmatrix} TETP_{max,min} \\ \angle TP_{max,min} \end{bmatrix}$$

$$x_{max2,min2} = \left[(\delta_a - \delta_b)_{max,min} \right]$$

$$y_{min} = -1 \qquad y_{max} = 1$$

$$\phi = \frac{2}{e^{-2x_j} + 1} - 1 \tag{6}$$

B.2. Optimization

The method in used in order to train the data is the Levenberg-Marquardt (LM) method. This method is a second order method and in order to update the weights for the next iteration, Eq. 7 is used.

$$W_{t+1} = W_t - \left(J^{\mathrm{T}} J + \mu I \right)^{-1} J^{\mathrm{T}} e \tag{7}$$

In which W is the vector containing all weights to be updated. e is a vector containing the error ($\delta_{measured} - \delta_{sim}$). This column vector has the size of the number of data points for $\delta_{measured}$. μ is the (adaptive) damping parameter. Finally, J is the Jacobian matrix. The general definition for J can be found in Eq. 8 in which N is the number of data points, K the number of input weights (inputs x $N_{neurons}$) and M the number of output weights (outputs x $N_{neurons}$). The partial derivatives in Eq. 8 are calculated using Eq. 9 and 10. These equations are used for all data points in combination with all weights.

$$\mathbf{J} = \frac{\partial \mathbf{e}}{\partial \mathbf{W}_t} = \begin{bmatrix} \frac{\partial \mathbf{e}(1)}{\partial \mathbf{W}_{ij}(1)} & \frac{\partial \mathbf{e}(1)}{\partial \mathbf{W}_{ij}(2)} & \cdots & \frac{\partial \mathbf{e}(1)}{\partial \mathbf{W}_{ij}(K)} & \frac{\partial \mathbf{e}(1)}{\partial \mathbf{W}_{jk}(1)} & \frac{\partial \mathbf{e}(1)}{\partial \mathbf{W}_{jk}(2)} & \cdots & \frac{\partial \mathbf{e}(1)}{\partial \mathbf{W}_{jk}(M)} \\ \frac{\partial \mathbf{e}(2)}{\partial \mathbf{W}_{ij}(1)} & \frac{\partial \mathbf{e}(2)}{\partial \mathbf{W}_{ij}(2)} & \cdots & \frac{\partial \mathbf{e}(2)}{\partial \mathbf{W}_{ij}(K)} & \frac{\partial \mathbf{e}(2)}{\partial \mathbf{W}_{jk}(1)} & \frac{\partial \mathbf{e}(2)}{\partial \mathbf{W}_{jk}(2)} & \cdots & \frac{\partial \mathbf{e}(2)}{\partial \mathbf{W}_{jk}(M)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{e}(N)}{\partial \mathbf{W}_{ij}(1)} & \frac{\partial \mathbf{e}(N)}{\partial \mathbf{W}_{ij}(2)} & \cdots & \frac{\partial \mathbf{e}(N)}{\partial \mathbf{W}_{ij}(K)} & \frac{\partial \mathbf{e}(N)}{\partial \mathbf{W}_{jk}(1)} & \frac{\partial \mathbf{e}(N)}{\partial \mathbf{W}_{jk}(2)} & \cdots & \frac{\partial \mathbf{e}(N)}{\partial \mathbf{W}_{jk}(M)} \end{bmatrix} \tag{8}$$

$$\frac{\partial e}{\partial w_{jk}} = \frac{\partial e_{k,q}}{\partial y_{k,q}} \frac{\partial y_{k,q}}{\partial v_{k,q}} \frac{\partial v_{k,q}}{\partial w_{jk}} = -1 \cdot 1 \cdot y_j \tag{9}$$

$$\frac{\partial e}{\partial W_{ij}} = \frac{\partial e_{k,q}}{\partial y_{k,q}} \frac{\partial y_{k,q}}{\partial v_{k,q}} \frac{\partial v_{k,q}}{\partial y_{j,q}} \frac{\partial y_{j,q}}{\partial v_{j,q}} \frac{\partial v_{j,q}}{\partial W_{ij}} = \cdot - 1 \cdot 1 \cdot w_{jk} \cdot \frac{\partial \phi_j \left(v_j \right)}{\partial v_j} \cdot x_i \tag{10}$$

C. Vehicle Dynamics

TABLE 3. VEHICLE CHARACTERISTICS

Vehicle mass	1600 kg
Front tire cornering stiffness	30,000 N·rad^{-1}
Rear tire cornering stiffness	30,000 N·rad^{-1}
Vehicle yaw moment of inertia	2000 kg· m^2
Front wheelbase to c.g.	1.4 m
Rear wheelbase to c.g.	1.4 m
Height of c.g.	0.5 m
Steering wheel rotational inertia	0.5 kg·m^2
Steering wheel damping	5 kg·m
Steering gear ratio	25

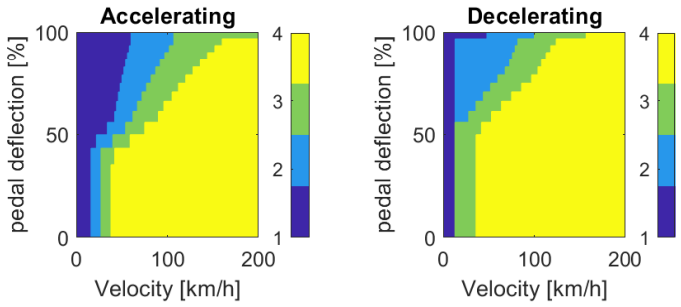


Figure 23. Gear-shift map for accelerating and decelerating vehicle

The main vehicle characteristics are shown in Table 3. The longitudinal vehicle dynamical model inputs the pedal deflections and outputs the vehicle acceleration. The gear in which the vehicle is in is calculated first using Fig. 23. In these plots, one can find the moment of switching gears for an accelerating and decelerating vehicle. Secondly the engine rpm is calculate using Eq. 11. Finally, the acceleration of the vehicle is calculated using Eq. 12 and Table 4. In which T_q is the engine torque in Nm, d_e is the rotation angle to engine input and G and B are the gas and brake contributions respectively.

TABLE 4. LONGITUDINAL VEHICLE PARAMETERS

Parameter	Value	Condition
$K_{rpm,1}$	0.1186	
$K_{rpm,2}$	0.00001252	
$K_{g,1}$	3.944358	
$K_{g,2}$	-5.12919	
$K_{g,3}$	2.178	
$K_{b,1}$	0.5	
$K_{b,2}$	29.43	
$K_{b,3}$	14.0283	
$\delta_{b,threshold}$	0.05	
$p_{b,threshold}$	30	
p_b	0	if $\delta_b < \delta_{b,threshold}$
	$200 \cdot \delta_b$	else
p_r	p_b	if $p_b < p_{b,threshold}$
	$30 + K_{b,1}(p_b - 30)$	else
M	1600kg	
K_{res}	1.7504	
K_{in}	1.65289256198347	

$$gearratio = [11.37 \quad 6.31 \quad 4.08 \quad 2.83];$$

$$rpm = \frac{100 \cdot gearratio(gear)}{\pi \cdot V} + 700 \tag{11}$$

$$a = \frac{4 \cdot (G - B) - K_{res} * V^2 + 200}{M + 100 + K_{in} * gearratio(gear)^2} \tag{12a}$$

$$G = 0.9 \cdot gearratio(gear) \cdot T_q \cdot d_e \tag{12b}$$

$$B = k_{b,2} \cdot p_b + k_{b,3} \cdot p_r \tag{12c}$$

$$T_q = K_{rpm,1} \cdot rpm - K_{rpm,2} \cdot rpm^2 \tag{12d}$$

$$d_e = K_{g,1} \cdot \delta_a + k_{g,2} \cdot \delta_a^2 + k_{g,3} \cdot \delta_a^3 \tag{12e}$$

Acknowledgments

The author would like to thank Dr. René van Paassen and Dr. Max Mulder for their support, encouragement, assistance and feedback during the entire research project.

The author would also like to thank Dr. Anahita Jamshidnejad for providing valuable insights related to machine learning and optimization methods.

References

[1] National Highway Traffic Safety Administration. "The Evolution of Automated Safety Technologies". In: *Traffic Safety* (1993).

[2] C. Basu et al. "Do you want your autonomous car to drive like you?" In: *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*. IEEE. 2017, pp. 417–425.

[3] R. Bellman. "Dynamic programming princeton university press princeton". In: *New Jersey Google Scholar* (1957).

[4] E. Bertolazzi et al. "Supporting drivers in keeping safe speed and safe distance: the SASPENCE subproject within the European framework programme 6 integrating project PReVENT". In: *IEEE Transactions on Intelligent Transportation Systems* 11.3 (2009), pp. 525–538.

[5] E.R. Boer. "Car following from the driver's perspective". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 2.4 (1999), pp. 201–206.

[6] E.R. Boer and M. Mulder. "To brake or not to brake: Scaling the curve". In: *Proceedings of the Vision in Vehicles (VIV) Cenference*. 2001, pp. 1–10.

[7] P. Bosetti, M. Da Lio, and A. Saroldi. "On the human control of vehicles: an experimental study of acceleration". In: *European Transport Research Review* 6.2 (2014), pp. 157–170.

[8] S. Cafiso and G. Cerni. "New approach to defining continuous speed profile models for two-lane rural roads". In: *Transportation research record* 2309.1 (2012), pp. 157–167.

[9] S. Ceelen. "Preliminary Thesis - Modeling drivers behaviour for speed adaptation in curves". In: 2021.

[10] M.T. Dzindolet et al. "The role of trust in automation reliance". In: *International journal of human-computer studies* 58.6 (2003), pp. 697–718.

[11] G. Ekman. "Weber's law and related functions". In: *The Journal of Psychology* 47.2 (1959), pp. 343–352.

[12] B.N. Fildes and T.J. Triggs. "The Effect of Road Curve Geometry on Curvature Matching Judgements". In: *Australian Road Research* 12.5 (1984).

[13] R. Fletcher. "Practical methods of optimization john wiley & sons". In: *New York* 80 (1987), p. 4.

[14] R. Gallen and S. Glaser. "Vision Based Tangent Point Detection Algorithm, Evaluation and Validation." In: *MVA*. Citeseer. 2009, p. 518.

[15] V. Gruppelaar et al. "A perceptually inspired driver model for speed control in curves". In: *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2018, pp. 1257–1262.

[16] W. Hock and K. Schittkowski. "A comparative performance evaluation of 27 nonlinear programming codes". In: *Computing* 30.4 (1983), pp. 335–358.

[17] Delft Haptics Lab. *Fixed-base Driving Simulator*. https://delfthapticslab.nl/device/fixed-base-driving-simulator/.

[18] M. Land and J. Horwood. "Which parts of the road guide steering?" In: *Nature* 377.6547 (1995), pp. 339–340.

[19] D. Lechner and C. Perrin. "The actual use of the dynamic performances of vehicles". In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 207.4 (1993), pp. 249–256.

[20] D.N. Lee. "A theory of visual control of braking based on information about time-to-collision". In: *Perception* 5.4 (1976), pp. 437–459.

[21] J.D. Lee and K.A. See. “Trust in automation: Designing for appropriate reliance”. In: *Human factors* 46.1 (2004), pp. 50–80.

[22] E. Lehtonen, O. Lappi, and H. Summala. “Anticipatory eye movements when approaching a curve on a rural road depend on working memory load”. In: *Transportation research part F: traffic psychology and behaviour* 15.3 (2012), pp. 369–377.

[23] A. Montella et al. “Prediction of drivers’ speed behavior on rural motorways based on an instrumented vehicle study”. In: *Transportation Research Record* 2434.1 (2014), pp. 52–62.

[24] M. Mulder, R.M.M. Van Paassen, and D.A. Abbink. “Haptic gas pedal feedback”. In: *Ergonomics* 51.11 (2008), pp. 1710–1720.

[25] A.M.C. Odhams and D.J. Cole. “Models of driver speed choice in curves”. In: *Proceedings of the 7th International Symposium on Advanced Vehicle Control*. Citeseer. 2004.

[26] F.N. Postema. *HMI Lab*. http://cs.lr.tudelft.nl/facilities/hmi-lab/.

[27] M.J.D. Powell. “Variable metric methods for constrained optimization”. In: *Mathematical programming the state of the art*. Springer, 1983, pp. 288–311.

[28] G. Reymond et al. “Role of lateral acceleration in curve driving: Driver model and experiments on a real vehicle and a driving simulator”. In: *Human factors* 43.3 (2001), pp. 483–495.

[29] D.D. Salvucci and R. Gray. “A two-point visual control model of steering”. In: *Perception* 33.10 (2004), pp. 1233–1248.

[30] M. Shino et al. “Deviated state detection method in driving around curves based on naturalistic driving behavior database for driver assistance systems”. In: *International Journal of Automotive Technology* 15.5 (2014), pp. 749–755.

[31] B. Turner, J. Woolley, and P. Cairney. “An analysis of driver behaviour through rural curves: Exploratory results on driver speed”. In: *Journal of the Australasian College of Road Safety* 26.4 (2015), p. 31.

[32] W. Van Winsum and H. Godthelp. “Speed choice and steering behavior in curve driving”. In: *Human factors* 38.3 (1996), pp. 434–441.

[33] A. Waytz, J. Heafner, and N. Epley. “The mind in the machine: Anthropomorphism increases trust in an autonomous vehicle”. In: *Journal of Experimental Social Psychology* 52 (2014), pp. 113–117.

[34] J. Xie et al. “A personalized curve driving model for intelligent vehicle”. In: *2017 IEEE International Conference on Unmanned Systems (ICUS)*. IEEE. 2017, pp. 301–306.

[35] D. Zhang et al. “Driver curve speed model and its application to ACC speed control in curved roads”. In: *International journal of automotive technology* 14.2 (2013), pp. 241–247.

Part II - Appendices

CONTENTS

A. Participant Instructions

1). Informed Consent Form

Taking part in the study	**Yes**	**No**
I have read and understood the study information dated 25/02/2021, or it has been read to me. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	☐	☐
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason.	☐	☐
I understand that taking part in the study involves collecting car state data using a fixed-base driving simulator in order to model human behaviour in curve driving.	☐	☐
I understand that there is currently a global pandemic (Covid-19) and that I have to take the proper safety measures (distance, mask and disinfect hands) to minimize infection risks.	☐	☐
Risks associated with participating in the study		
I understand that taking part in the study involves the following risks: physical discomfort (simulation sickness) and mental fatigue.	☐	☐
Use of the information in the study		
I understand that information I provide will be used for a MSc thesis project	☐	☐
I understand that personal information collected about me that can identify me, such as [e.g. my name or where I live], will not be shared beyond the study team.	☐	☐
Future use and reuse of the information by others		
I give permission for the simulation data that I provide to be archived at TU Delft so it can be used for future research and learning.	☐	☐

Signatures

2). Information Sheet

Purpose of research

The purpose of the research is to model human behaviour in curve driving on high speed roads. The goal is to investigate whether it is possible to make a generic model that includes the general trends of human behaviour while driving. Such a model could be used in the future to enhance comfort while automating vehicle (sub)systems.

Risks of participating

- Mental fatigue: This is a condition triggered by prolonged cognitive activity. Basically, it sends your brain into overdrive, leaving you exhausted. During and in between experiments, the participant will be asked if any symptoms are present. The participant may also stop the simulation at any time.
- Simulation sickness: This is a subset of motion sickness. Simulation sickness includes: Nausea (symptoms are sweating, increased salvation, stomach awareness, and burping) and Disorientation (symptoms are vertigo1, dizzy (eyes open), dizzy (eyes closed), and blurred vision). During and in between experiments, the participant will be asked if any symptoms are present. The participant may also stop the simulation at any time.
- Covid-19: when in public places during this pandemic, there always exists a risk of getting infected by the Covid-19 virus. Precaution will be taken by disinfecting hands, wearing a face mask, and to keep at least 1.5m distance.

Procedures for withdrawal from the study

When the participant wants to withdraw from the study, he/she has the right to do so without any reason. Please contact the researcher and/or supervisors (contact details are below), and all collected data and personal information will be deleted permanently.

Personal information about the participant

Personal information that will be collected is the information on the Informed Consent form. This information will be stored separately in a Project Storage (staff-umbrella) within the TU Delft network, accessible only by the project supervisor. Other information such as age, gender and years of driving experience will be anonymized.

Retention period for the research data

The retention period of the research data is at least 10 years.

File a complaint

Send an email to the project researcher or to the project supervisor.

Instructions & info for Experiment

- There will be 4 Training road which take approximately 10 minutes each. There are 6 validation roads that will take 5 minutes each. In total the time spend (with breaks) will be about 2-2.5 hours.
- First, you will have a few familiarisation runs, to get used to the simulation controls.
- Try to drive in a comfortable way as you would normally without any time constraints.
- The maximum speed is 100km/h. Just as on public roads, try to not exceed the limit.

- In case of emergency , there is a red emergency button next to you one the right that will stop the simulation.

Contact details of the researcher & supervisors

Stef Ceelen —– s.ceelen@student.tudelft.nl

Max Mulder —- M.Mulder@tudelft.nl

René van Paassen —- M.M.vanPaassen@tudelft.nl

Institution

TU Delft
Faculty of Aerospace Engineering
Control and Simulation Department

B. Additional Results

1). One-size-fits-all Model Results

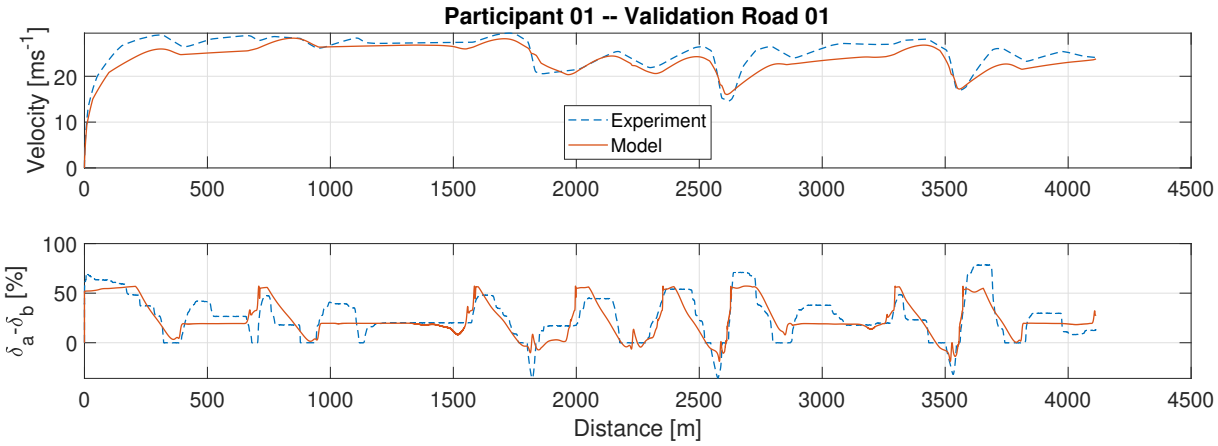


Figure B.1: One-size-fits-all model results, participant 01 on validation road 1

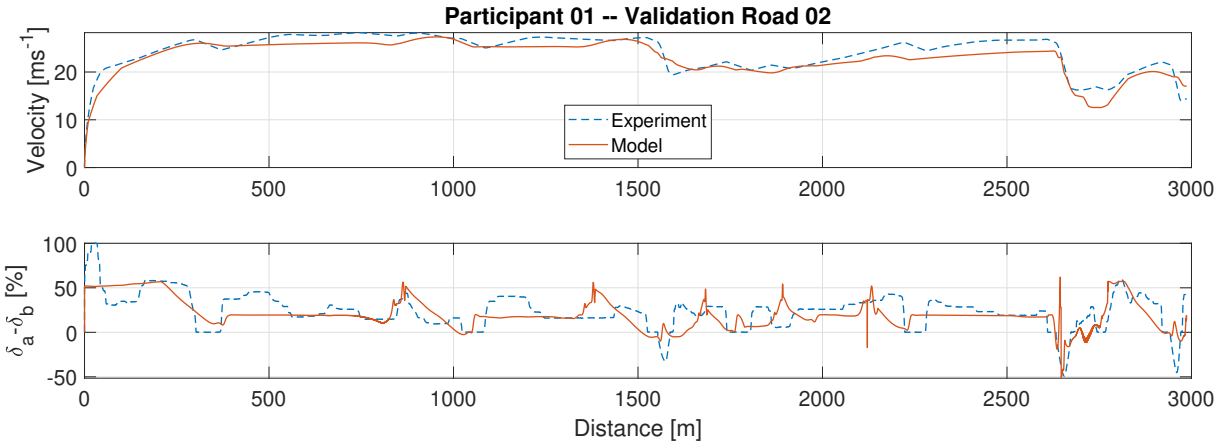


Figure B.2: One-size-fits-all model results, participant 01 on validation road 2

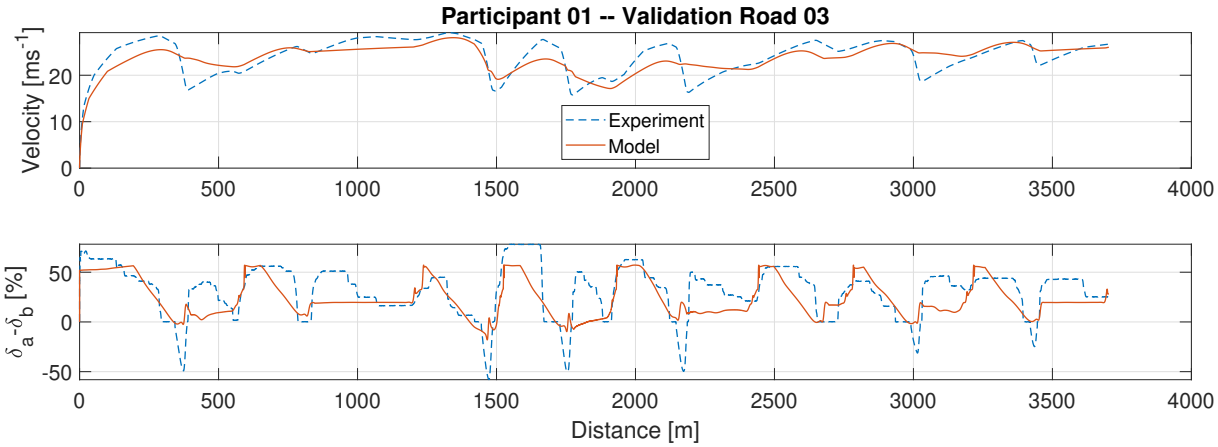


Figure B.3: One-size-fits-all model results, participant 01 on validation road 3

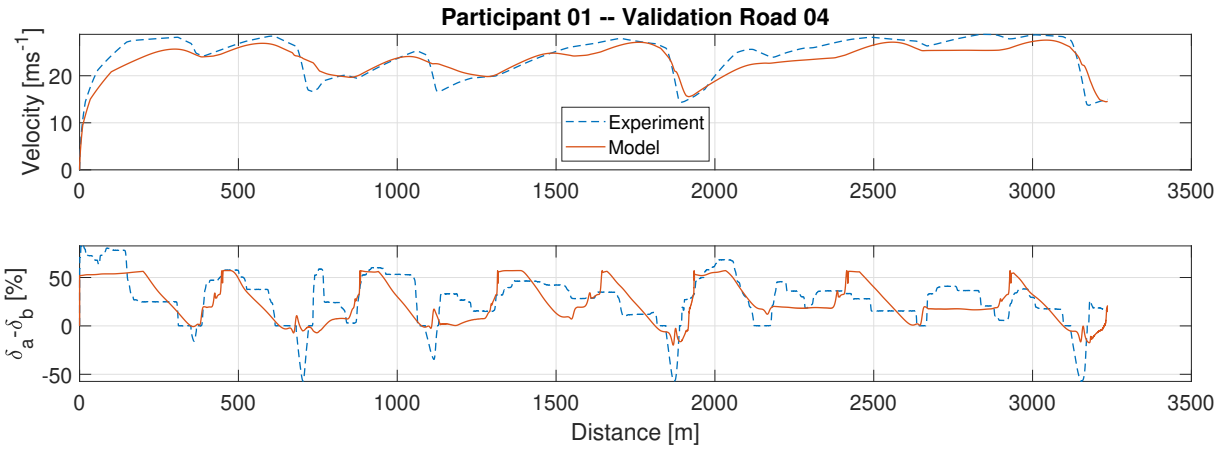


Figure B.4: One-size-fits-all model results, participant 01 on validation road 4

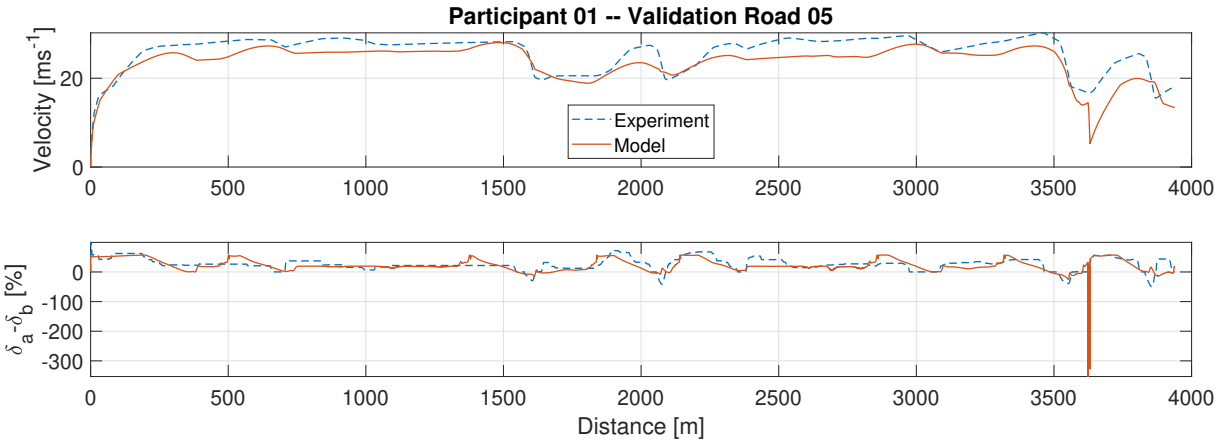


Figure B.5: One-size-fits-all model results, participant 01 on validation road 5

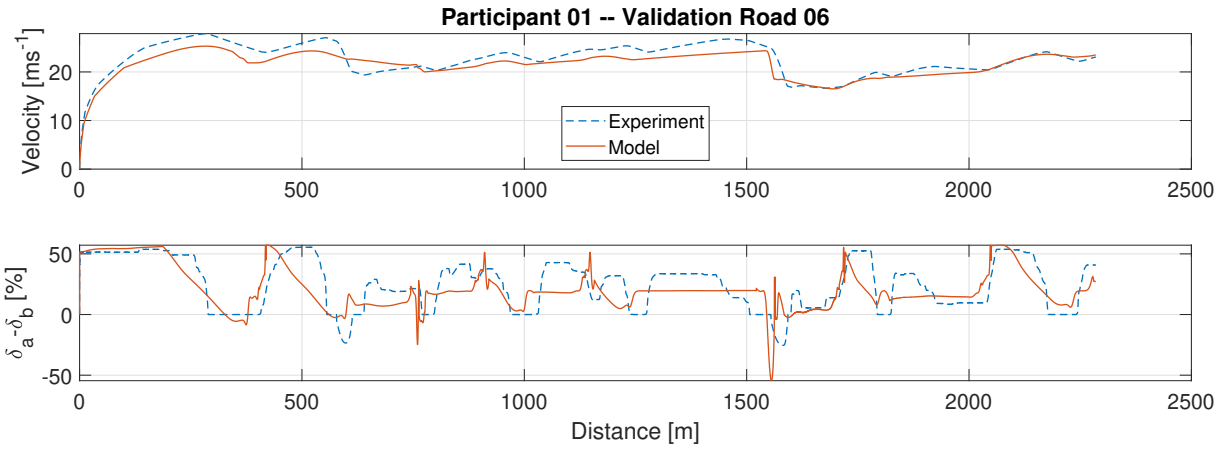


Figure B.6: One-size-fits-all model results, participant 01 on validation road 6

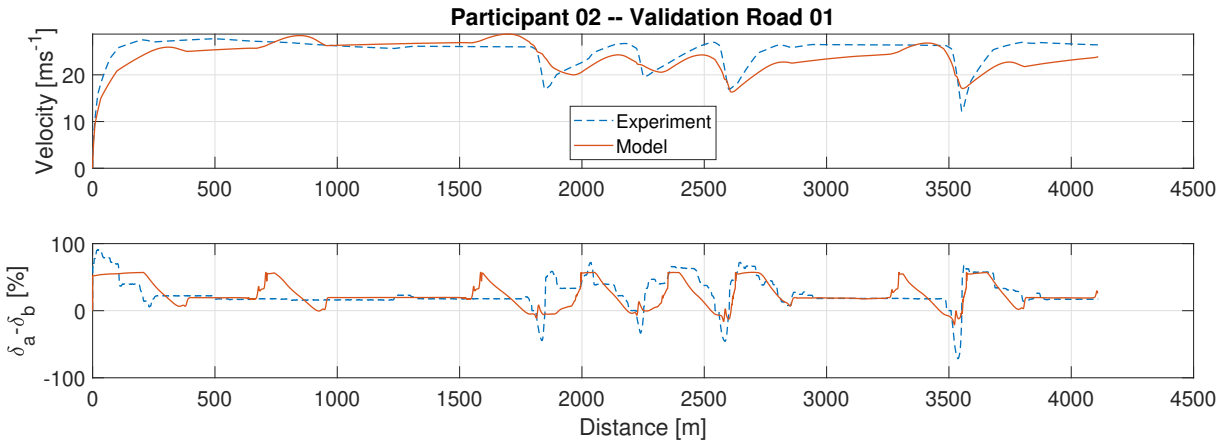


Figure B.7: One-size-fits-all model results, participant 02 on validation road 1

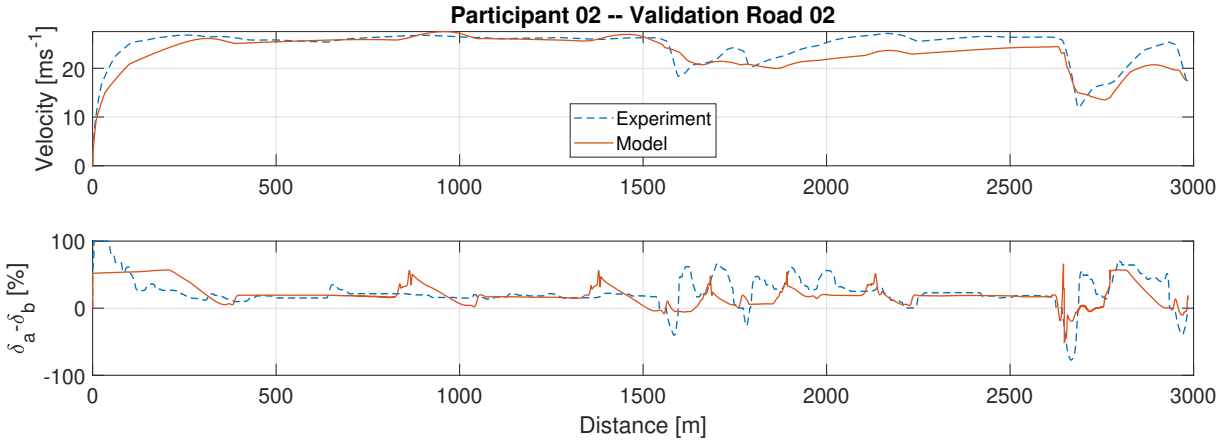


Figure B.8: One-size-fits-all model results, participant 02 on validation road 2

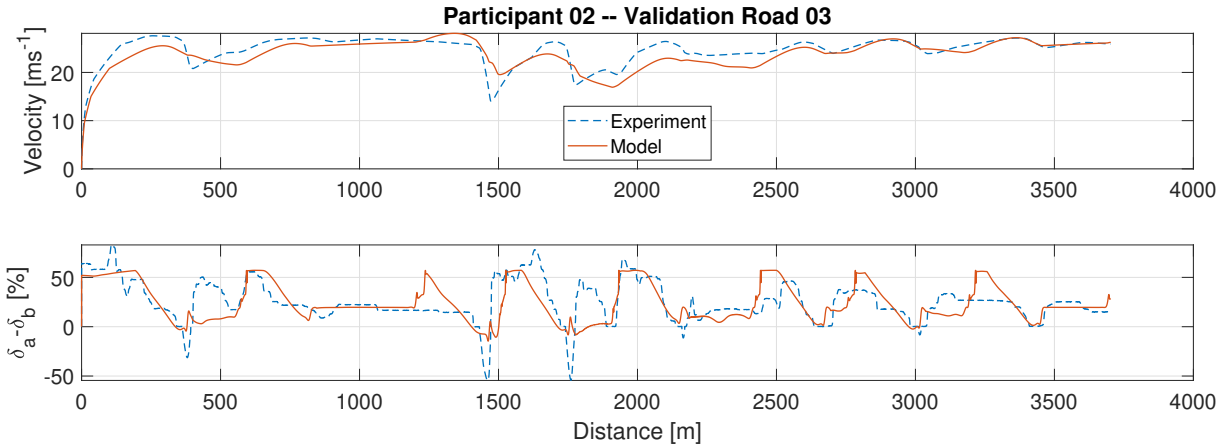


Figure B.9: One-size-fits-all model results, participant 02 on validation road 3

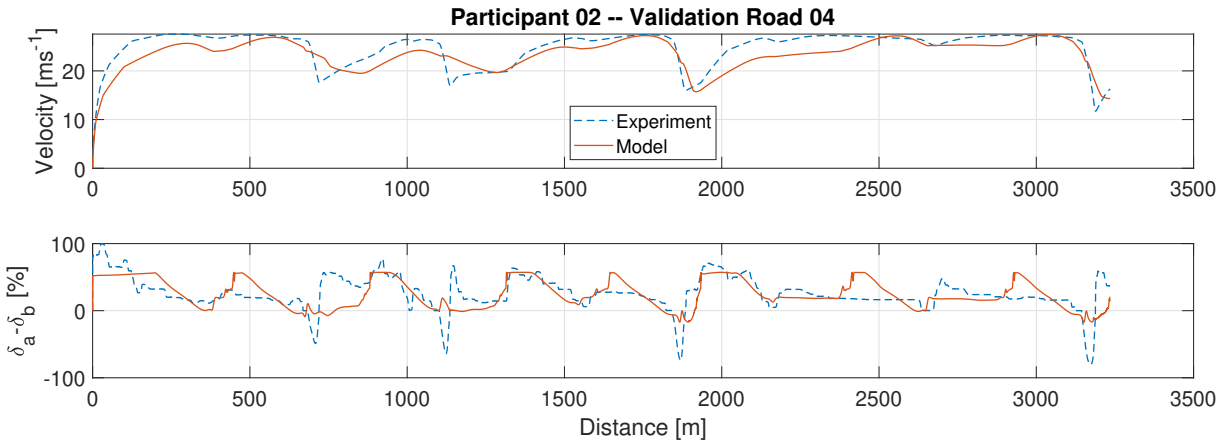


Figure B.10: One-size-fits-all model results, participant 02 on validation road 4

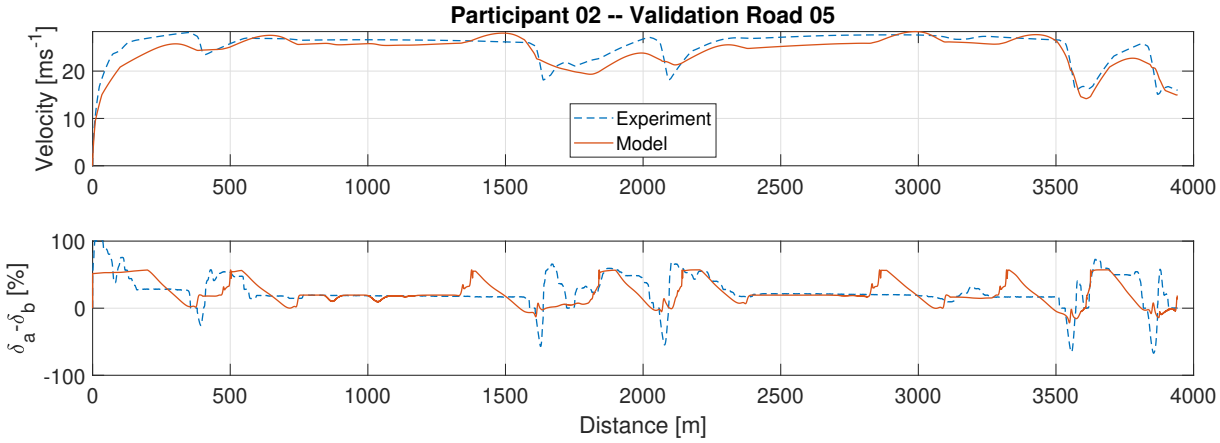


Figure B.11: One-size-fits-all model results, participant 02 on validation road 5

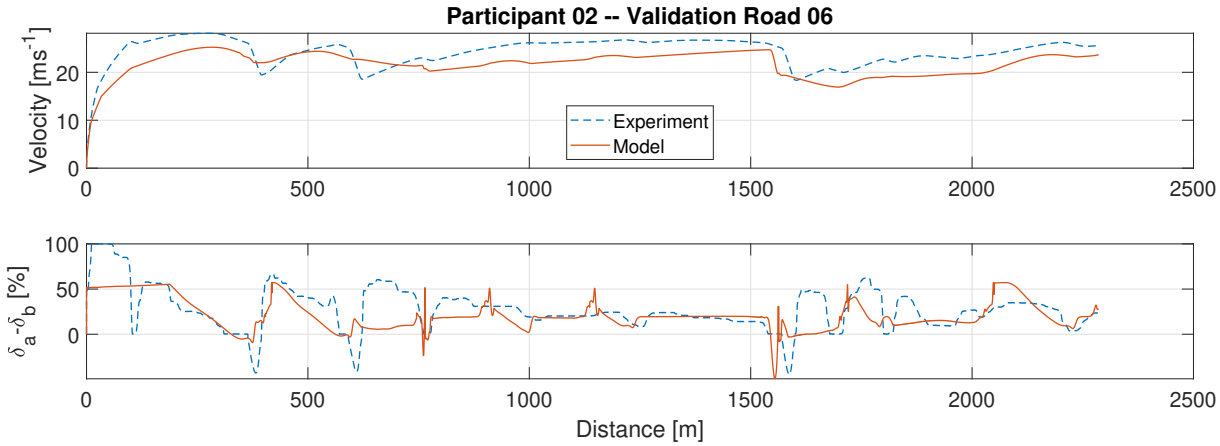


Figure B.12: One-size-fits-all model results, participant 02 on validation road 6

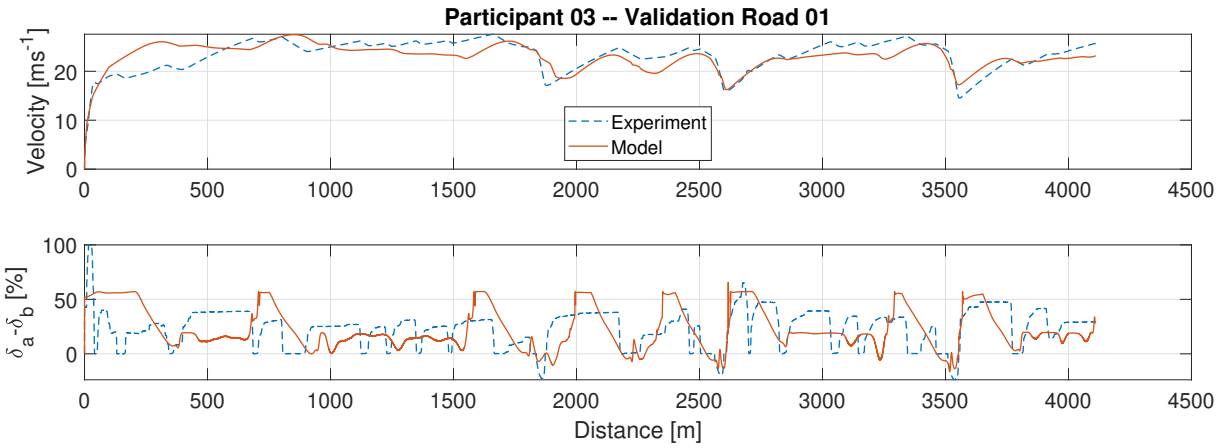


Figure B.13: One-size-fits-all model results, participant 03 on validation road 1

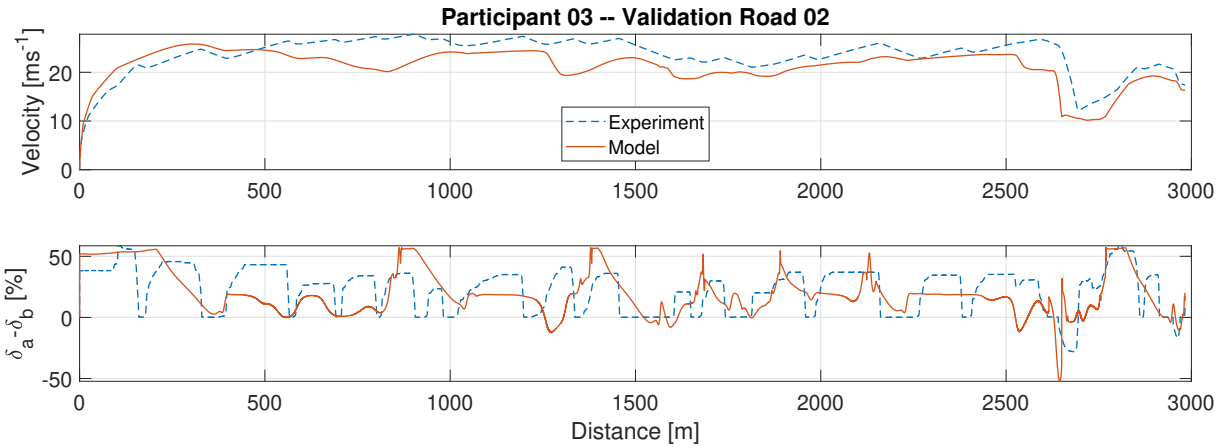


Figure B.14: One-size-fits-all model results, participant 03 on validation road 2

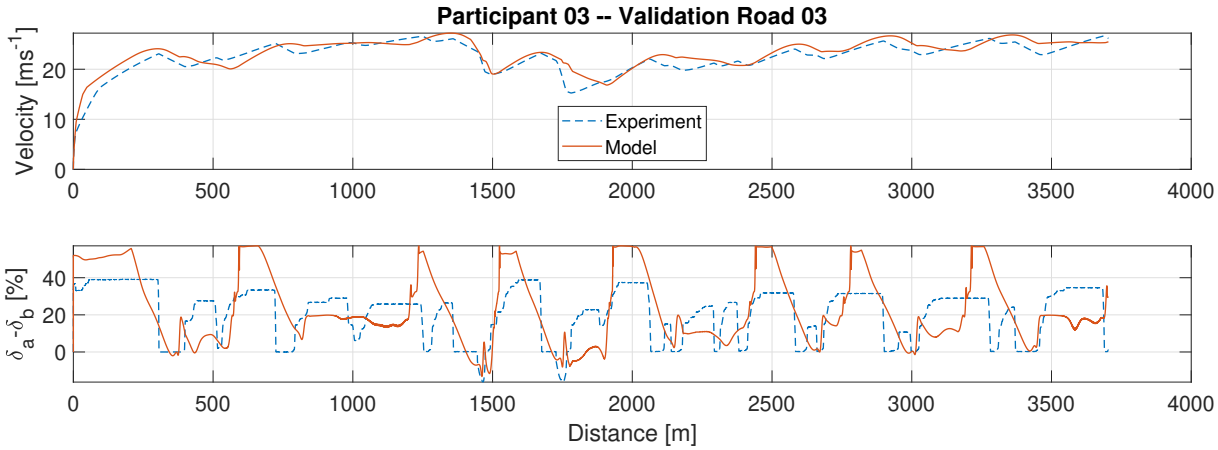


Figure B.15: One-size-fits-all model results, participant 03 on validation road 3

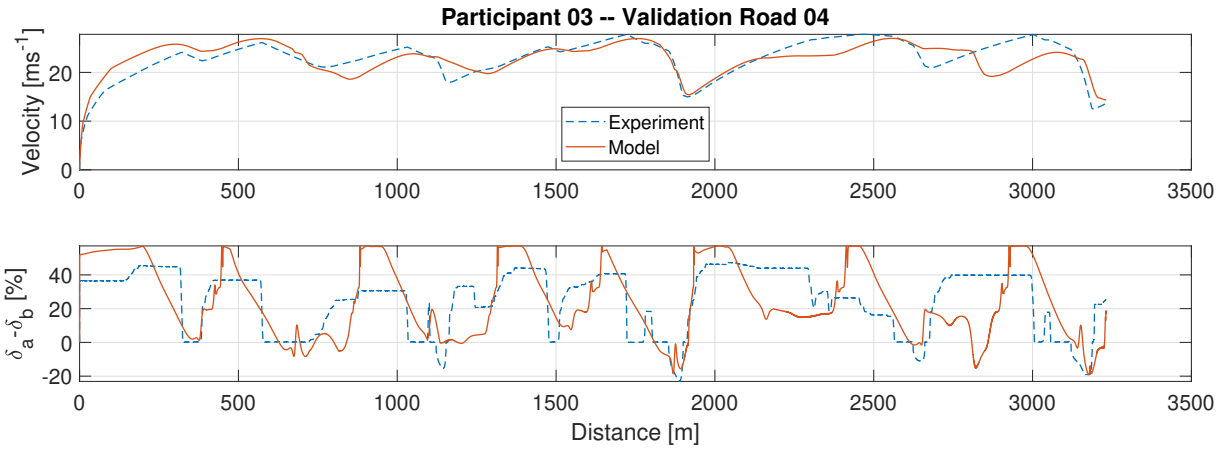


Figure B.16: One-size-fits-all model results, participant 03 on validation road 4

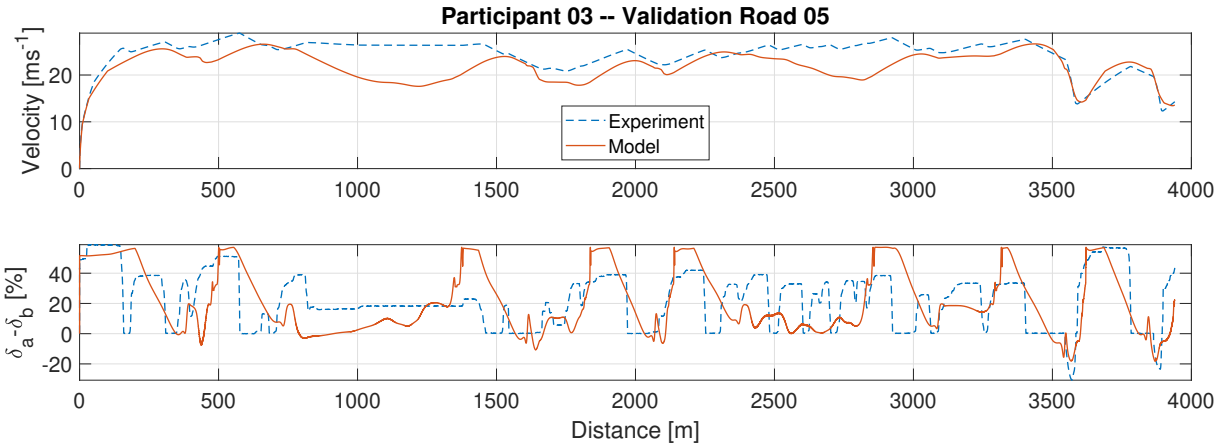


Figure B.17: One-size-fits-all model results, participant 03 on validation road 5

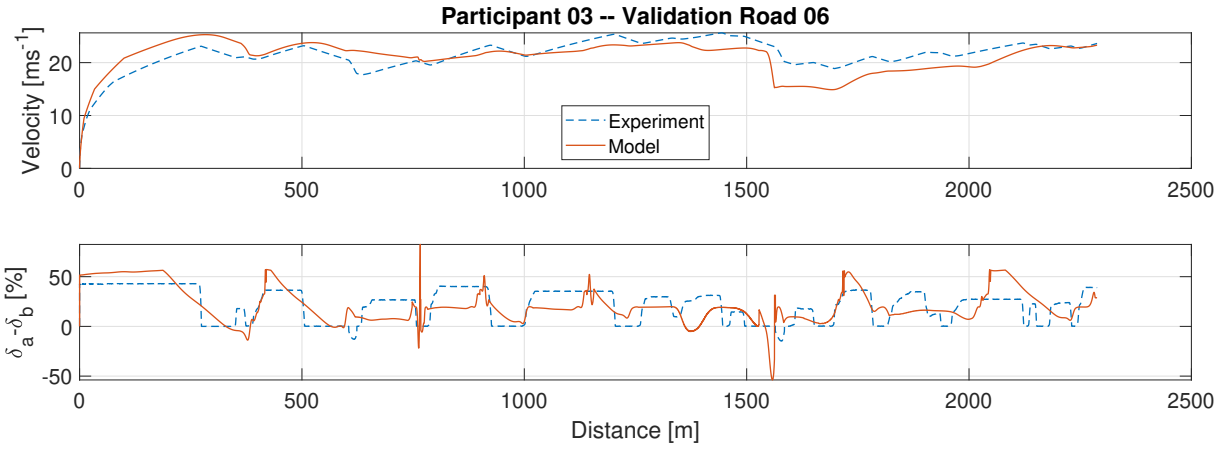


Figure B.18: One-size-fits-all model results, participant 03 on validation road 6

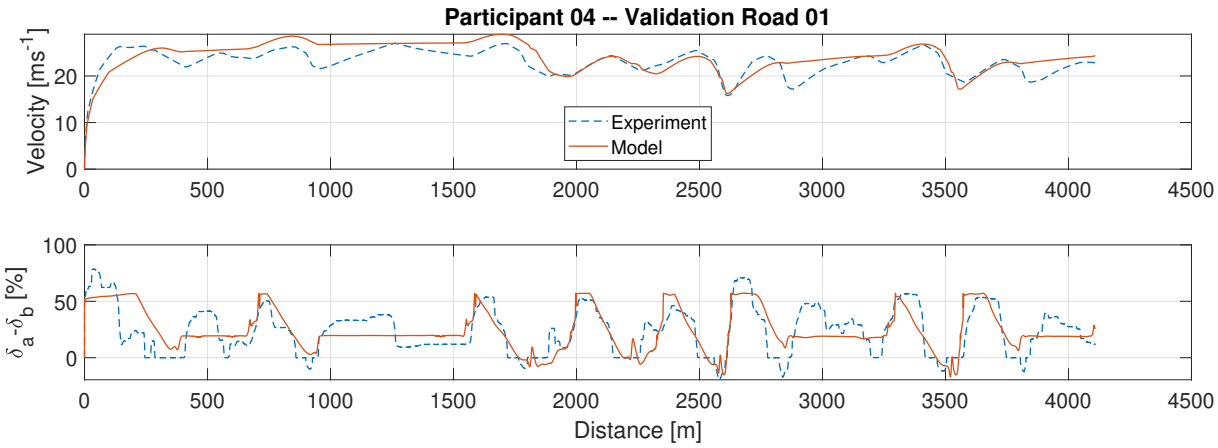


Figure B.19: One-size-fits-all model results, participant 04 on validation road 1

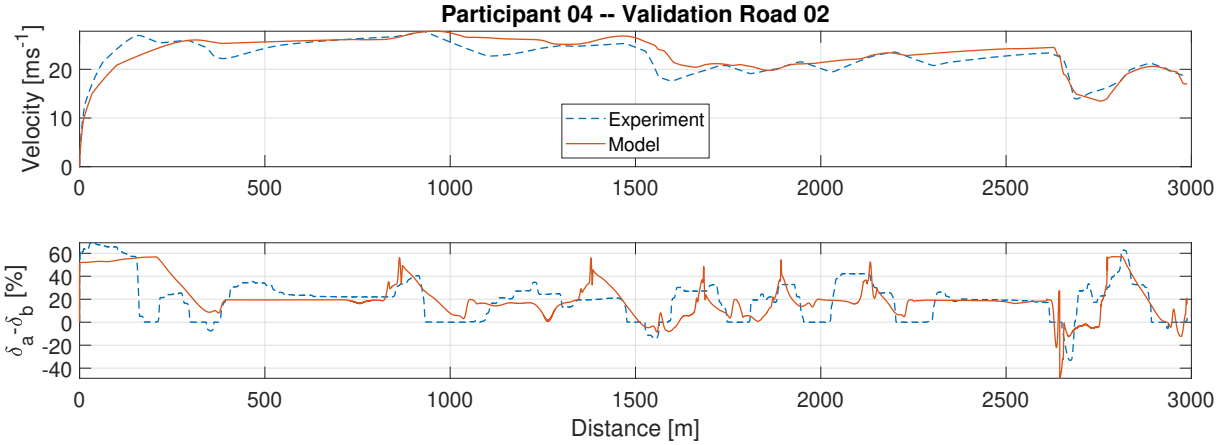


Figure B.20: One-size-fits-all model results, participant 04 on validation road 2

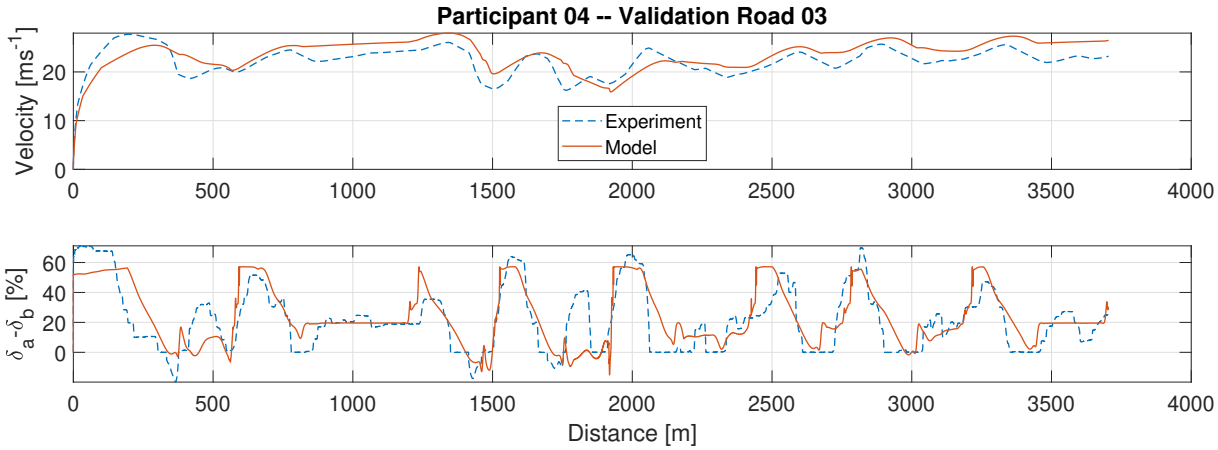


Figure B.21: One-size-fits-all model results, participant 04 on validation road 3

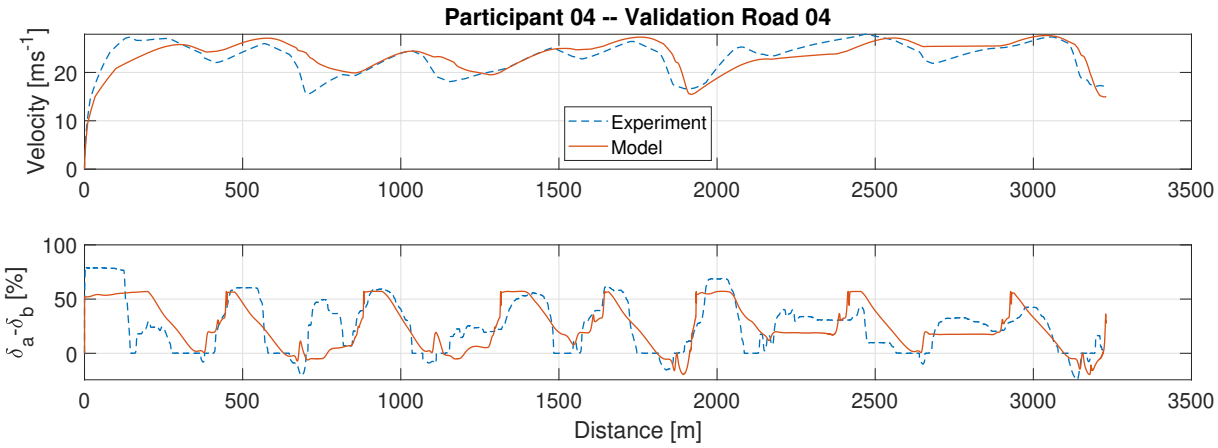


Figure B.22: One-size-fits-all model results, participant 04 on validation road 4

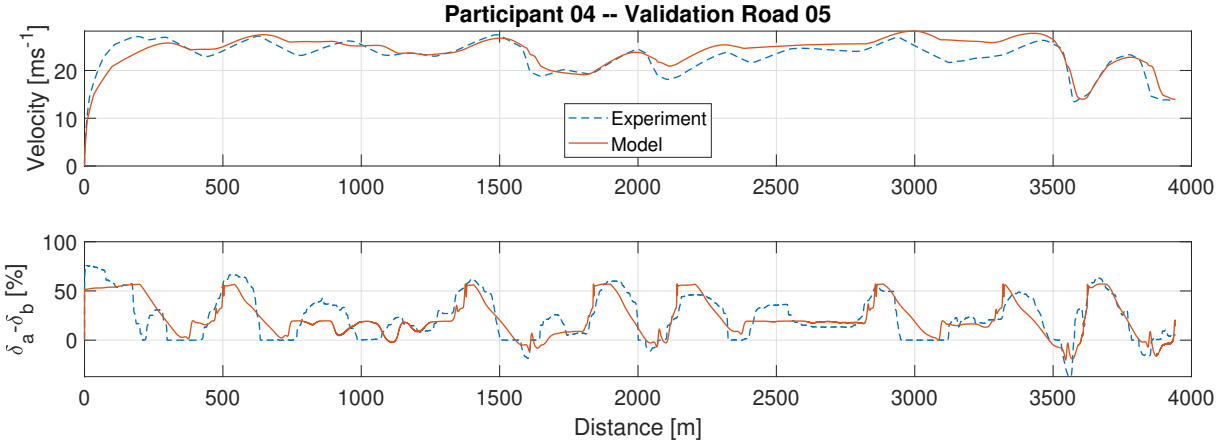


Figure B.23: One-size-fits-all model results, participant 04 on validation road 5

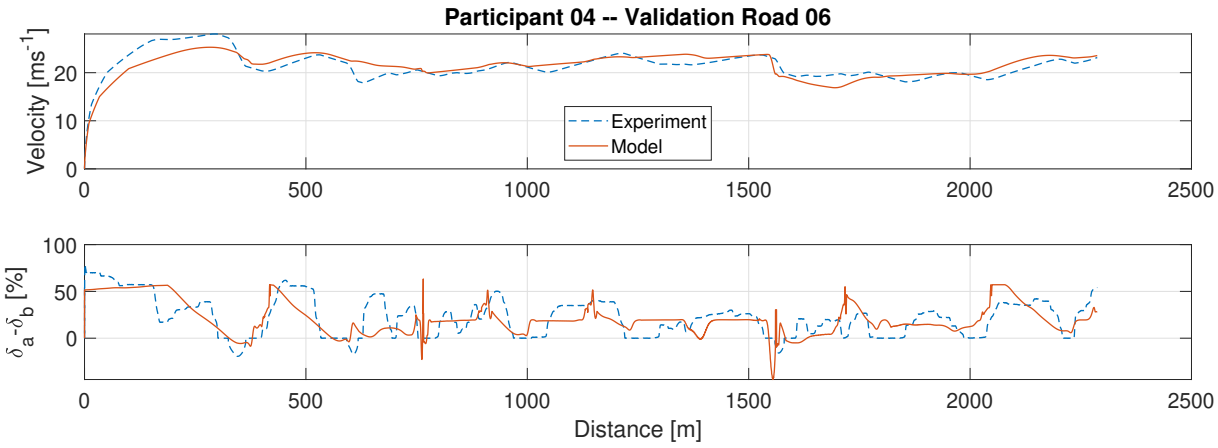


Figure B.24: One-size-fits-all model results, participant 04 on validation road 6

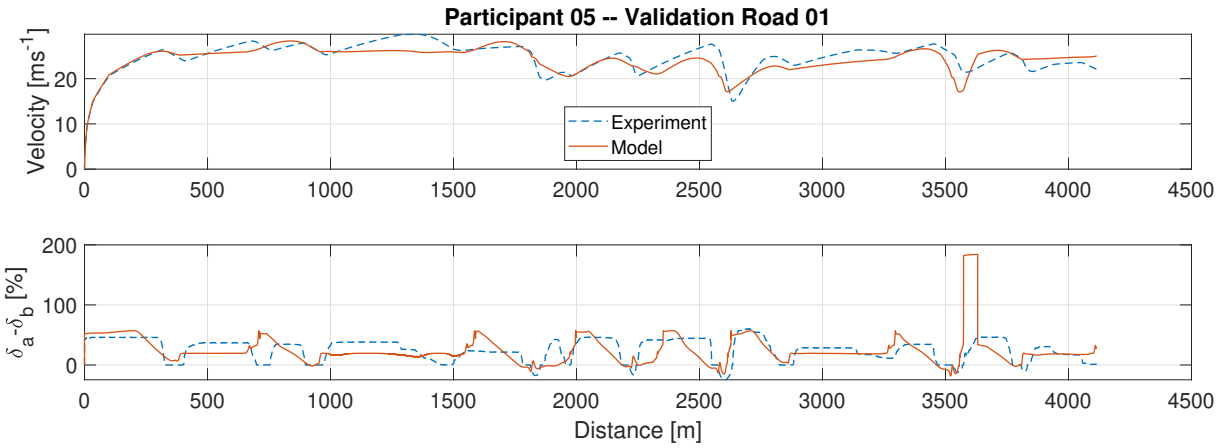


Figure B.25: One-size-fits-all model results, participant 05 on validation road 1

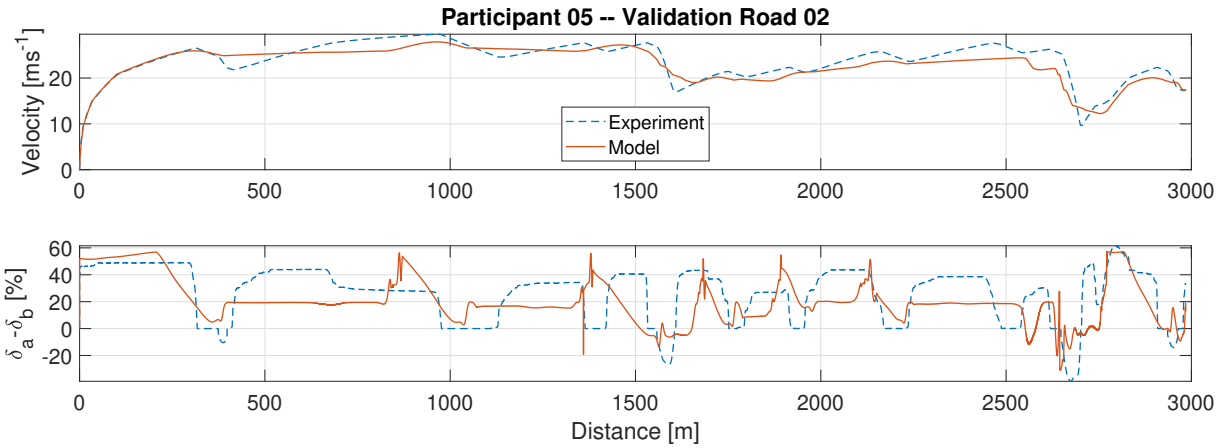


Figure B.26: One-size-fits-all model results, participant 05 on validation road 2

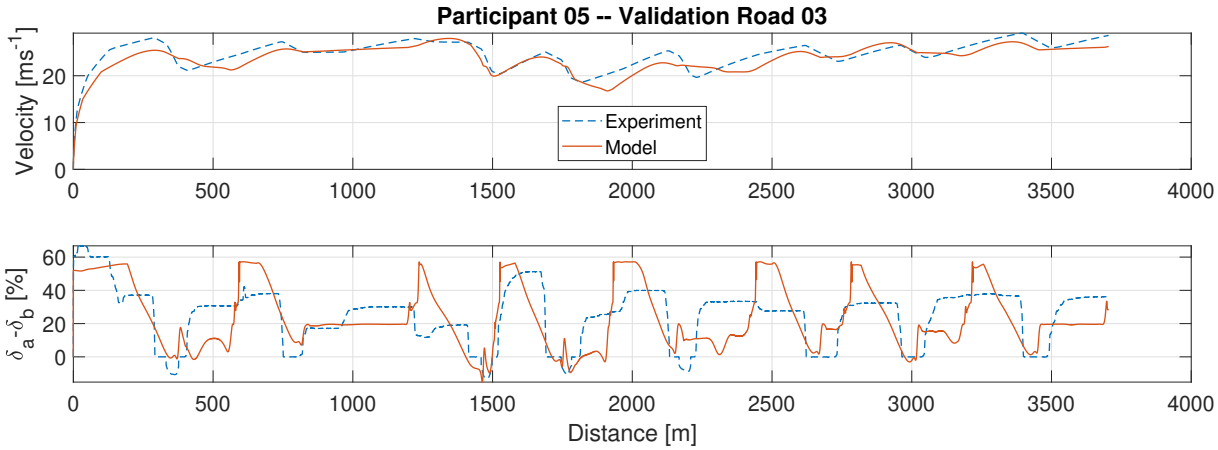


Figure B.27: One-size-fits-all model results, participant 05 on validation road 3

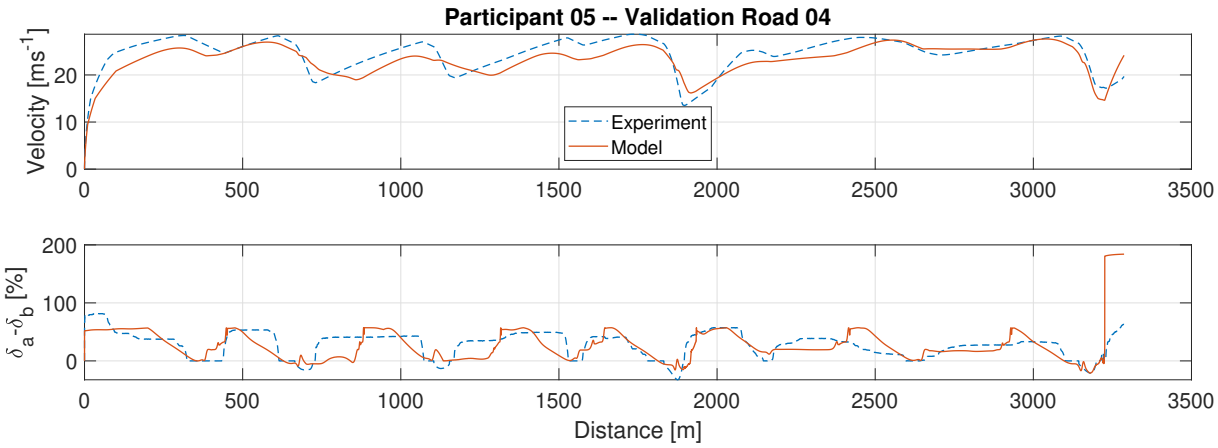


Figure B.28: One-size-fits-all model results, participant 05 on validation road 4

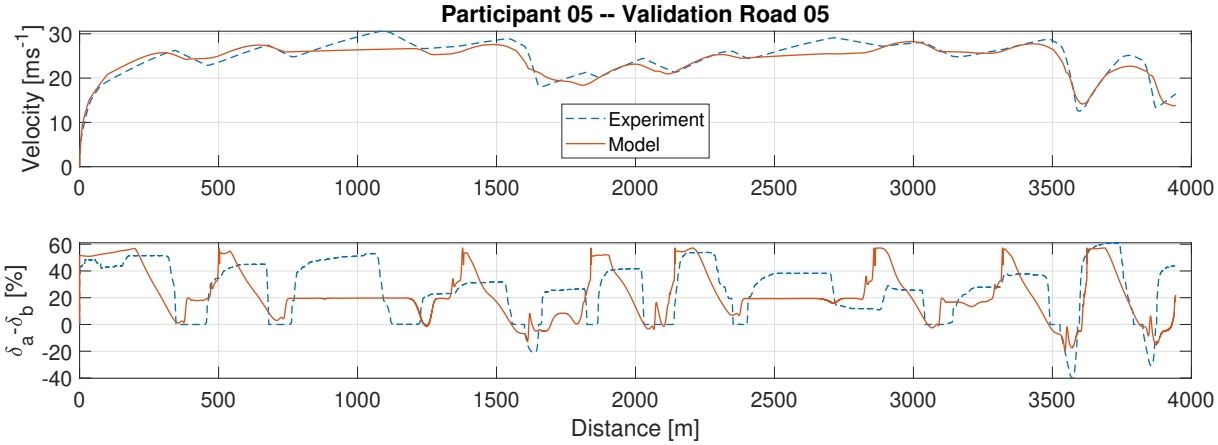


Figure B.29: One-size-fits-all model results, participant 05 on validation road 5

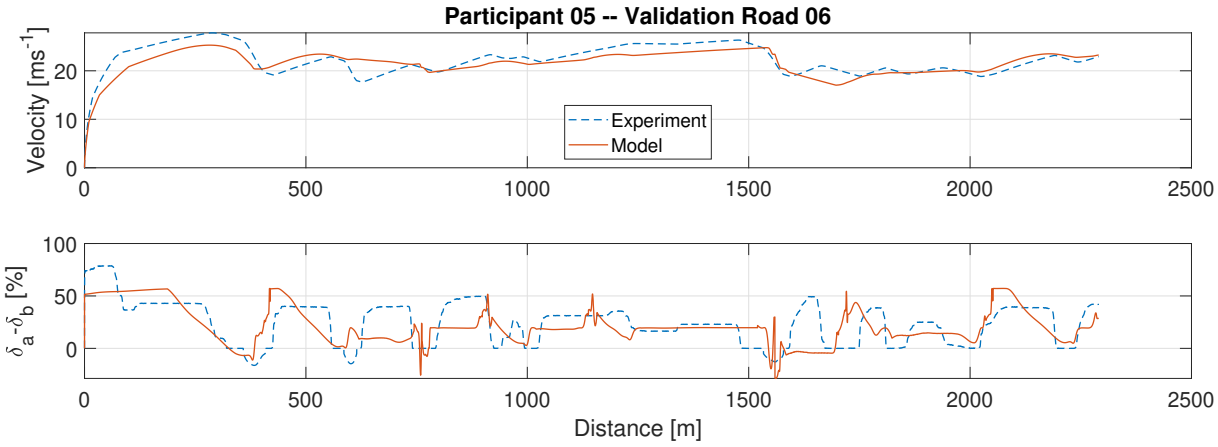


Figure B.30: One-size-fits-all model results, participant 05 on validation road 6

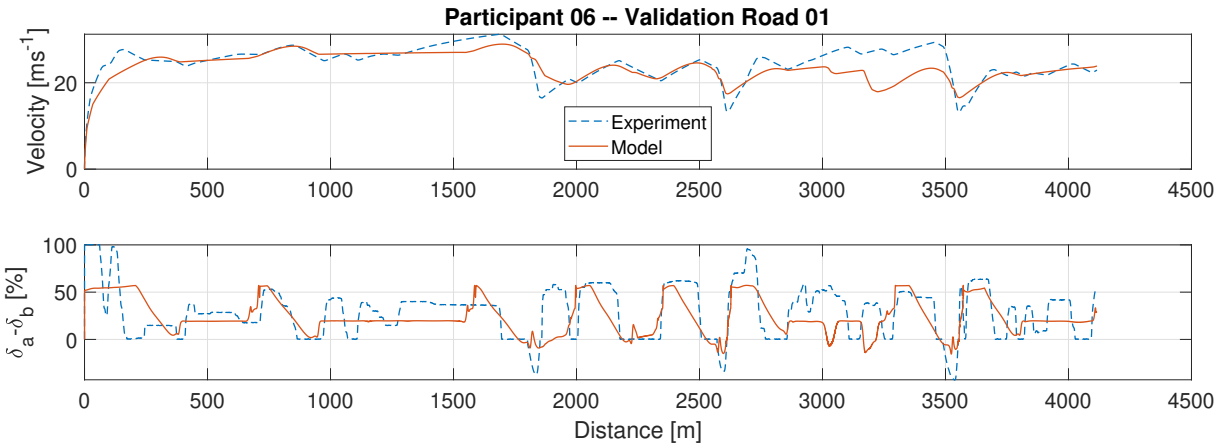


Figure B.31: One-size-fits-all model results, participant 06 on validation road 1

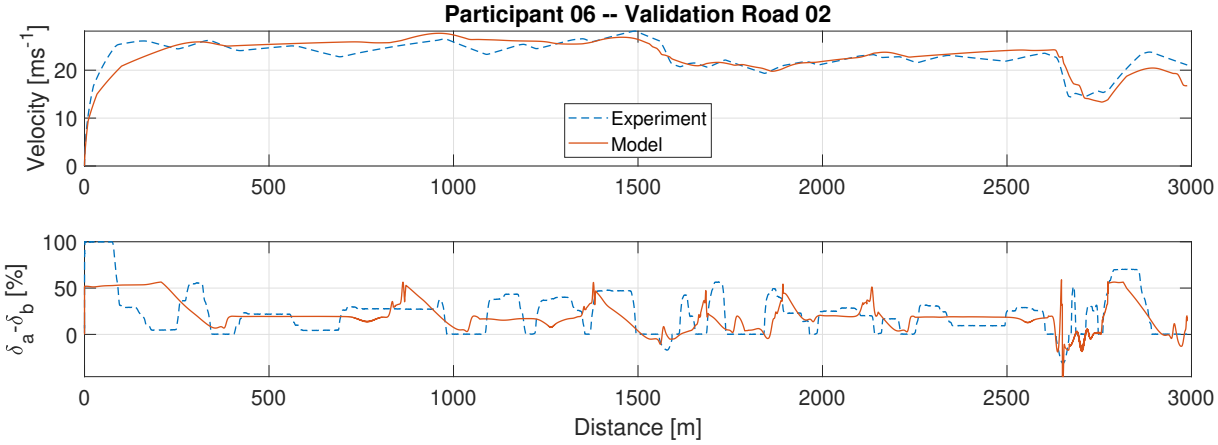


Figure B.32: One-size-fits-all model results, participant 06 on validation road 2

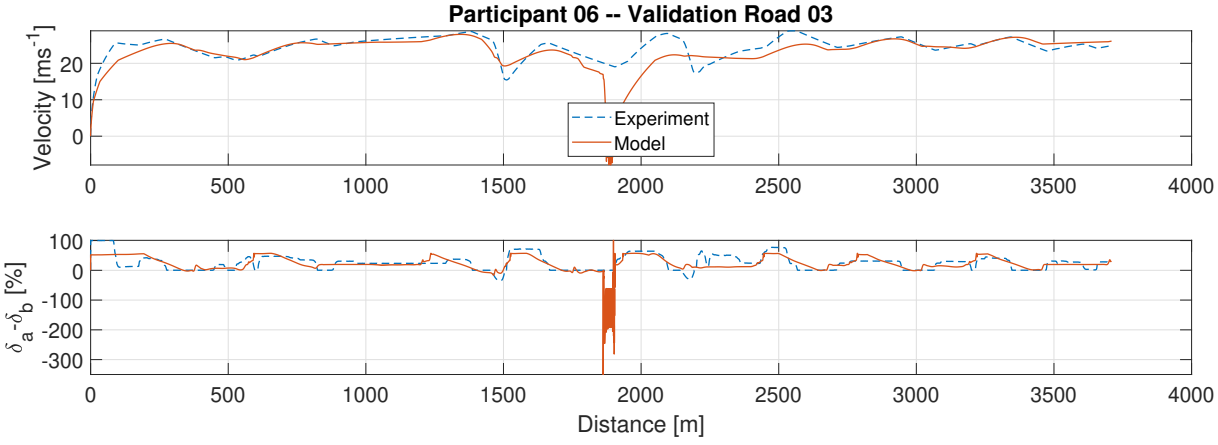


Figure B.33: One-size-fits-all model results, participant 06 on validation road 3

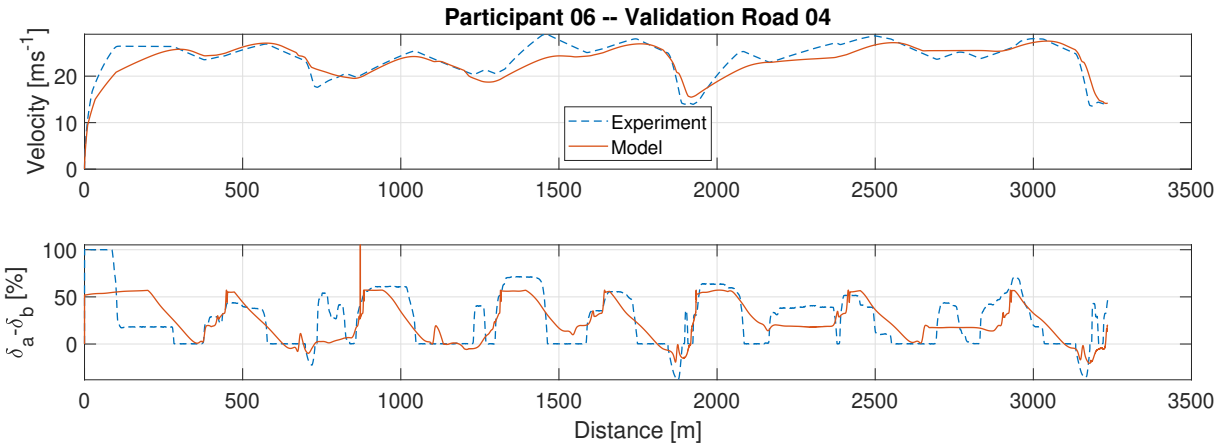


Figure B.34: One-size-fits-all model results, participant 06 on validation road 4

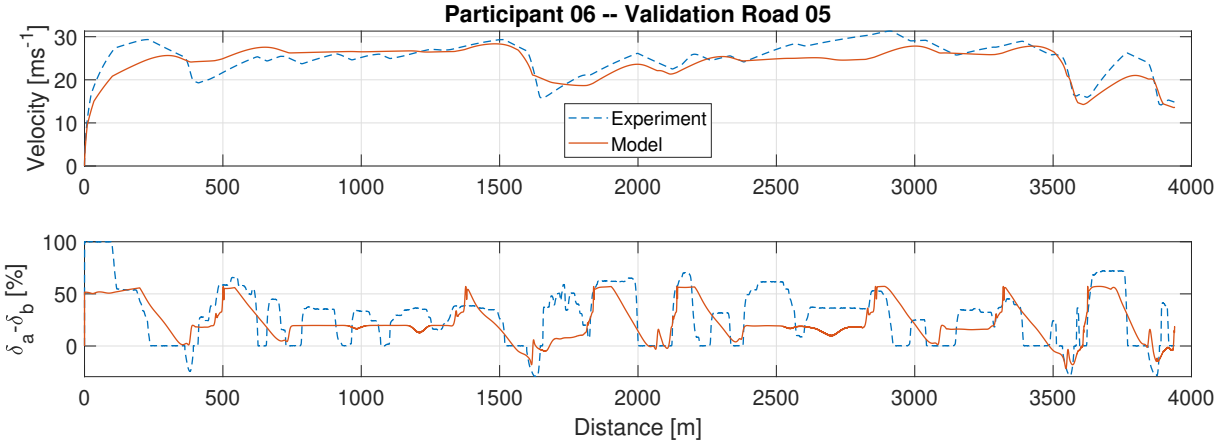


Figure B.35: One-size-fits-all model results, participant 06 on validation road 5

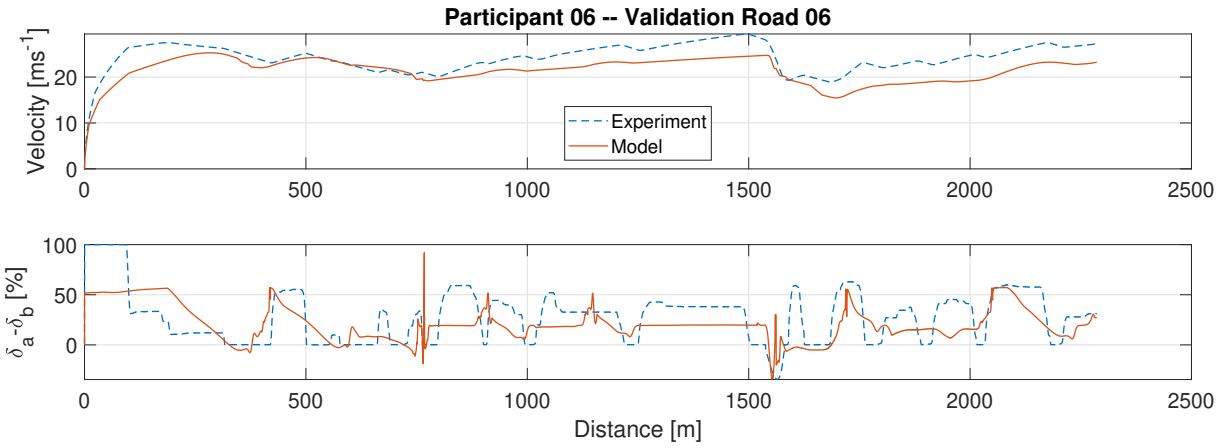


Figure B.36: One-size-fits-all model results, participant 06 on validation road 6

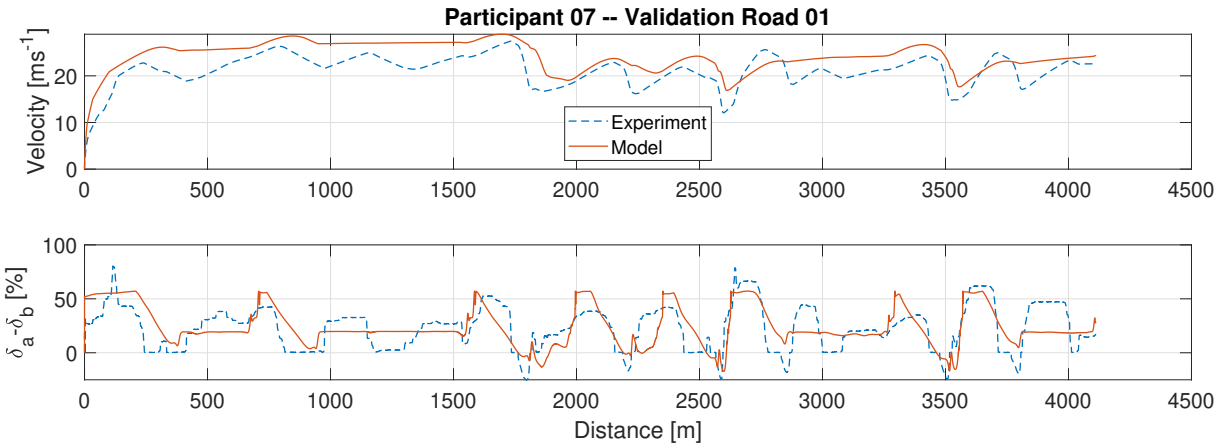


Figure B.37: One-size-fits-all model results, participant 07 on validation road 1

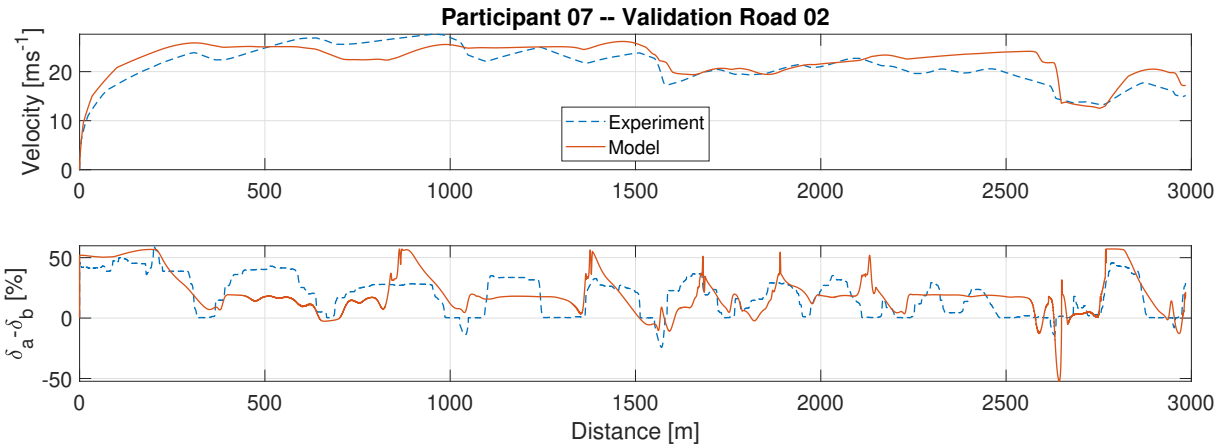


Figure B.38: One-size-fits-all model results, participant 07 on validation road 2

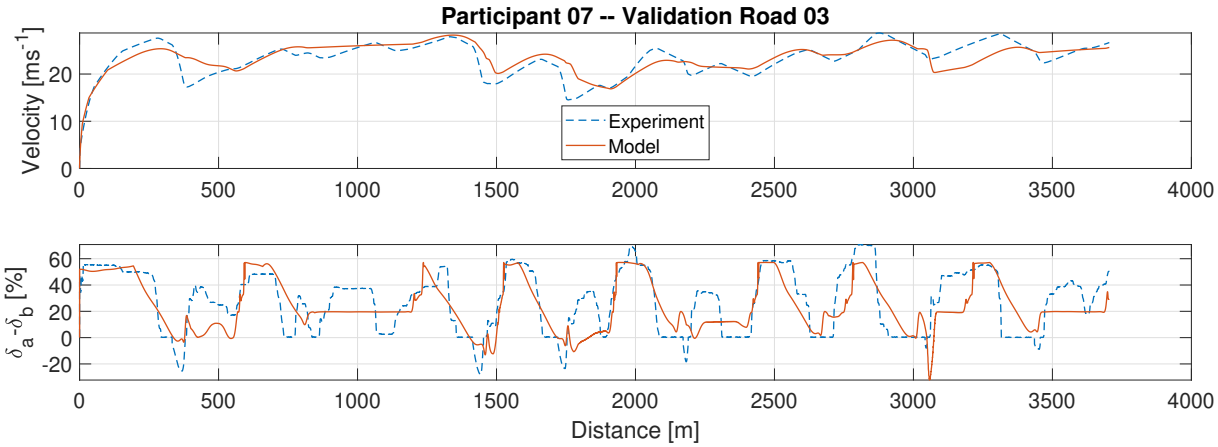


Figure B.39: One-size-fits-all model results, participant 07 on validation road 3

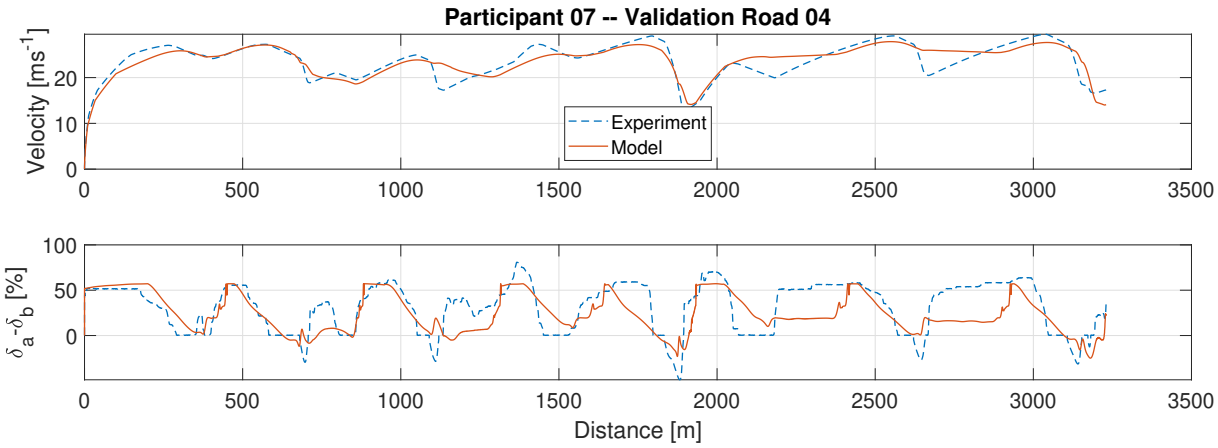


Figure B.40: One-size-fits-all model results, participant 07 on validation road 4

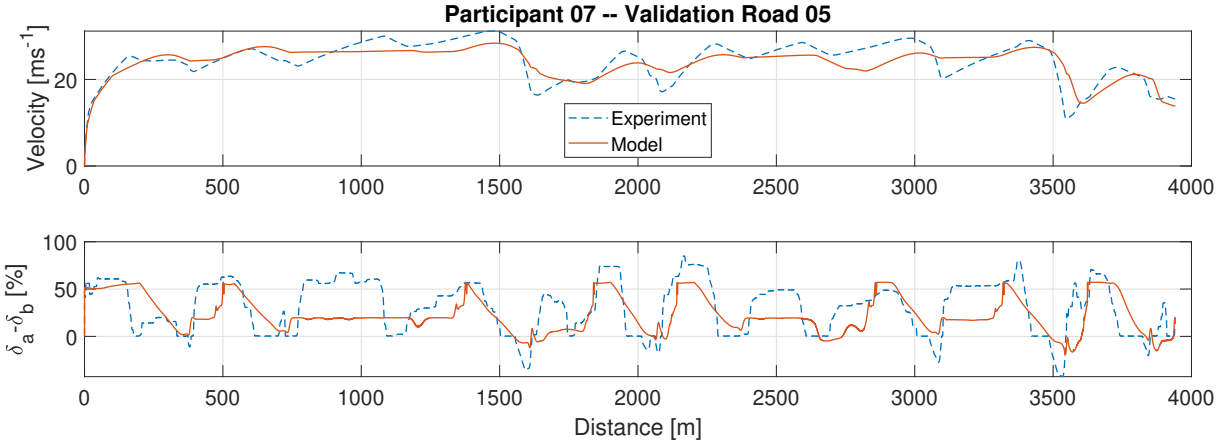


Figure B.41: One-size-fits-all model results, participant 07 on validation road 5

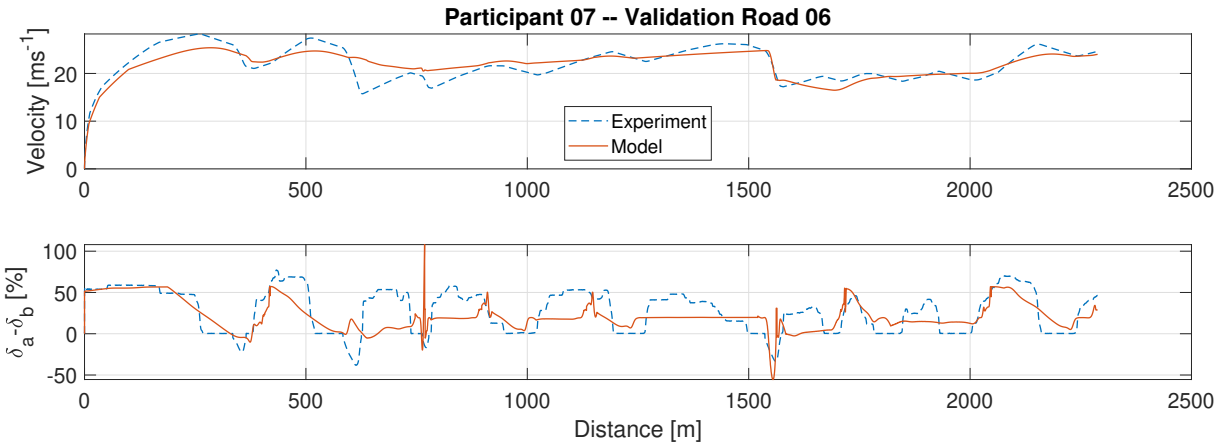


Figure B.42: One-size-fits-all model results, participant 07 on validation road 6

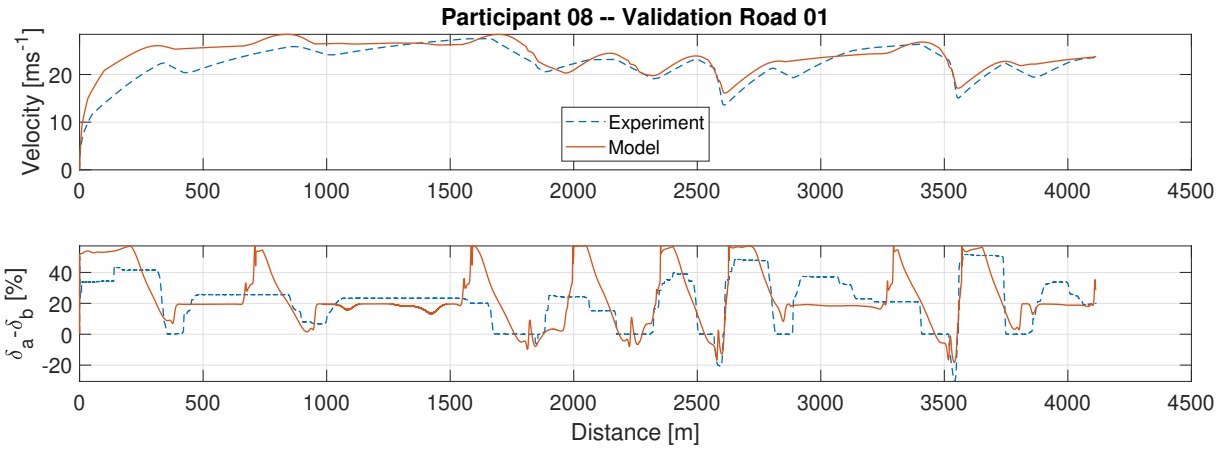


Figure B.43: One-size-fits-all model results, participant 08 on validation road 1

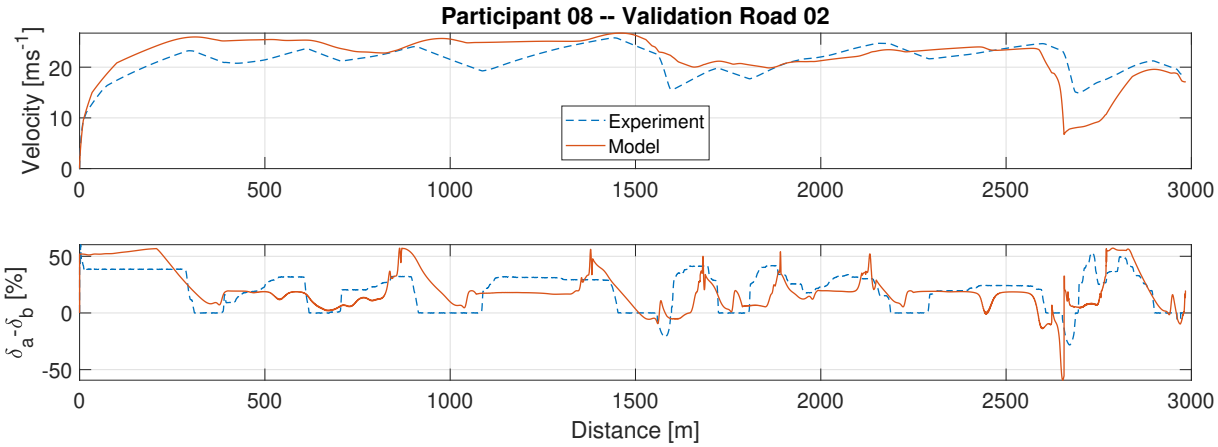


Figure B.44: One-size-fits-all model results, participant 08 on validation road 2

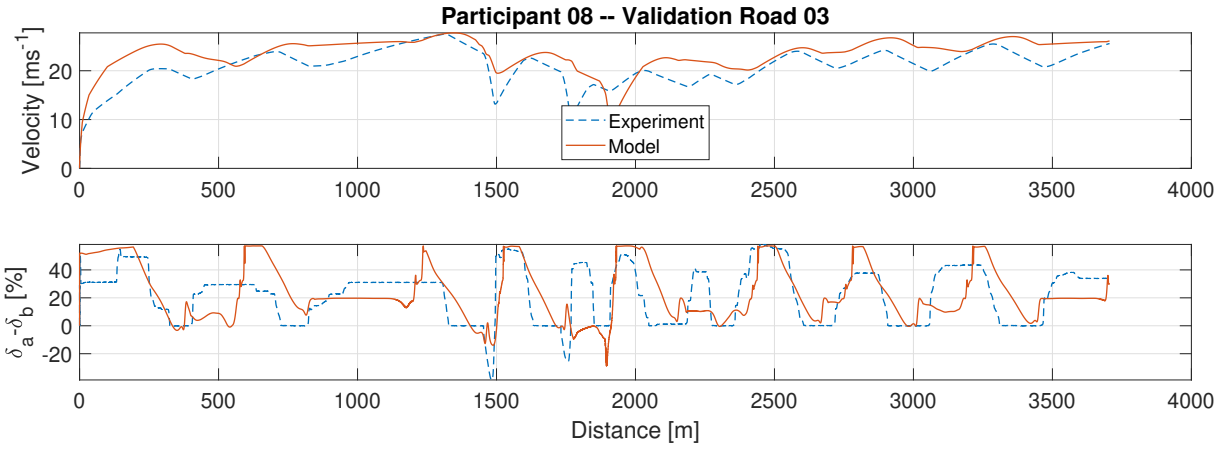


Figure B.45: One-size-fits-all model results, participant 08 on validation road 3

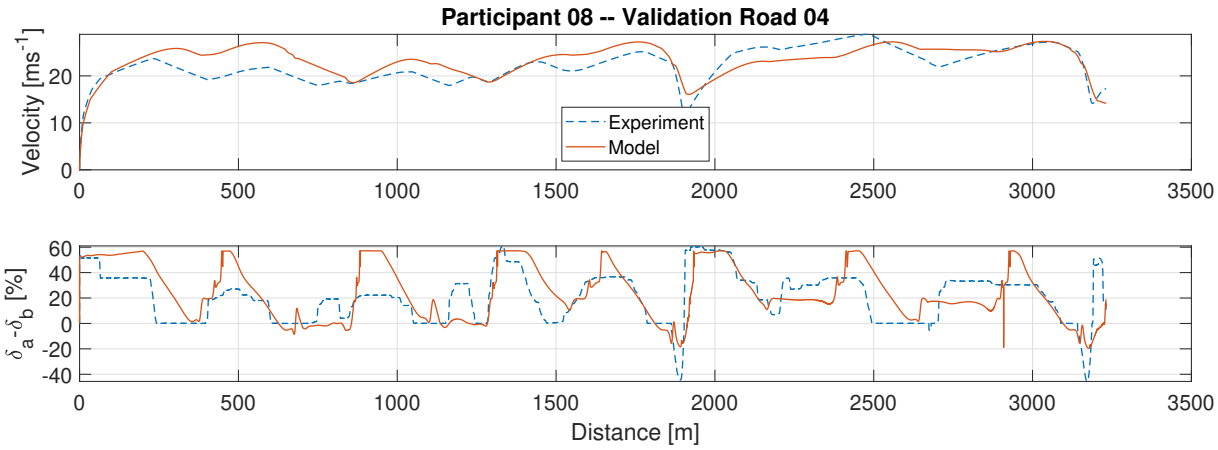


Figure B.46: One-size-fits-all model results, participant 08 on validation road 4

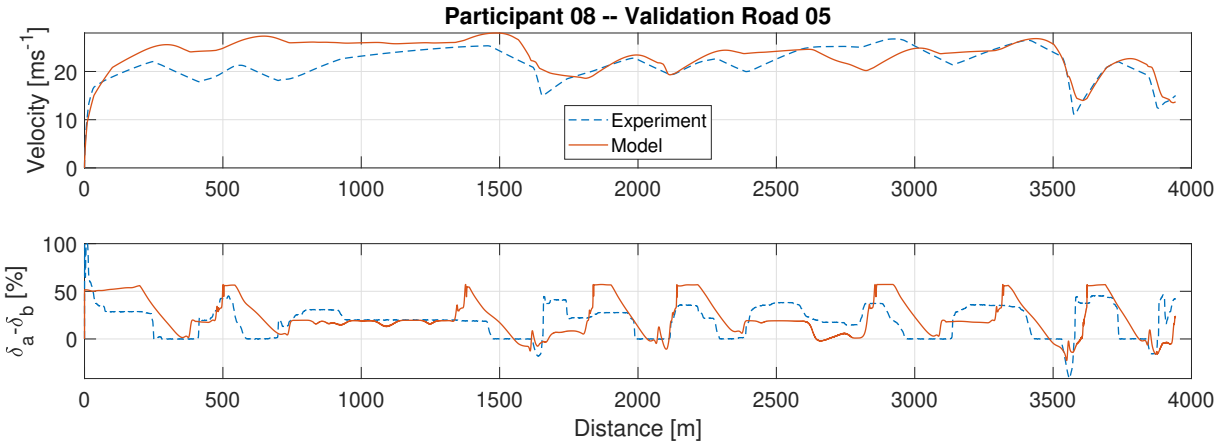


Figure B.47: One-size-fits-all model results, participant 08 on validation road 5

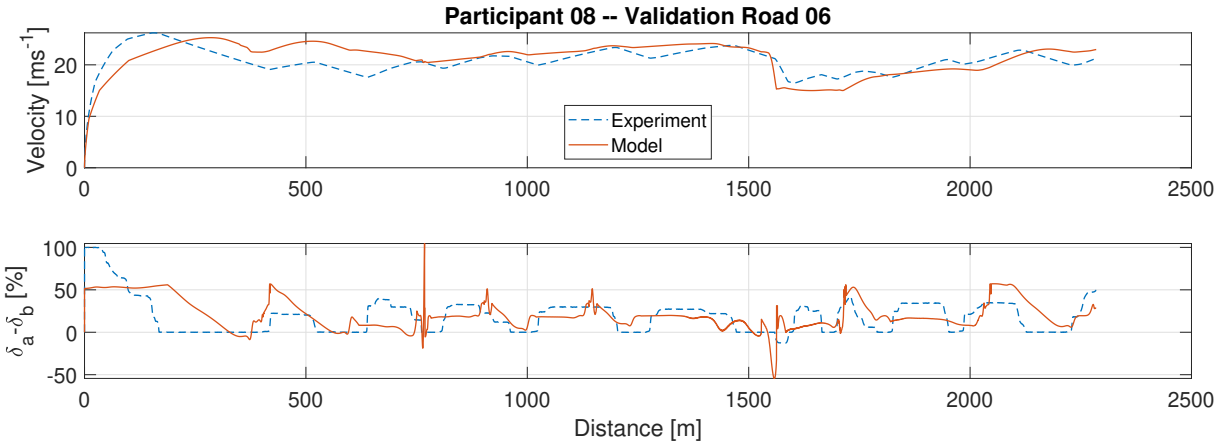


Figure B.48: One-size-fits-all model results, participant 08 on validation road 6

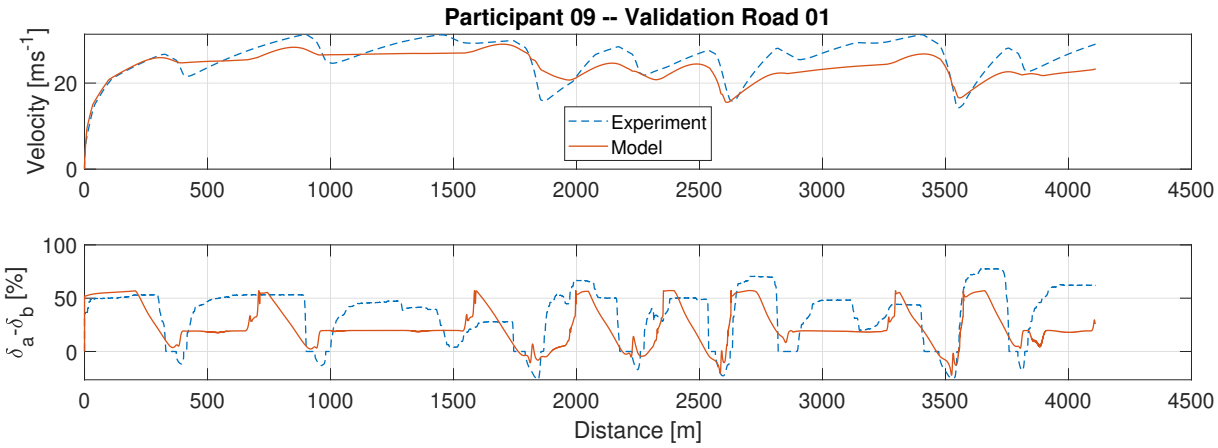


Figure B.49: One-size-fits-all model results, participant 09 on validation road 1

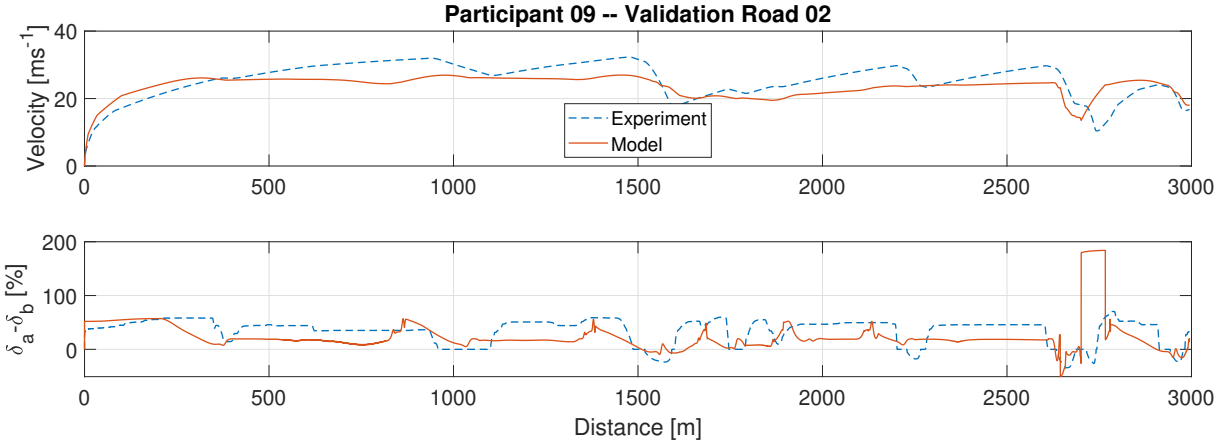


Figure B.50: One-size-fits-all model results, participant 09 on validation road 2

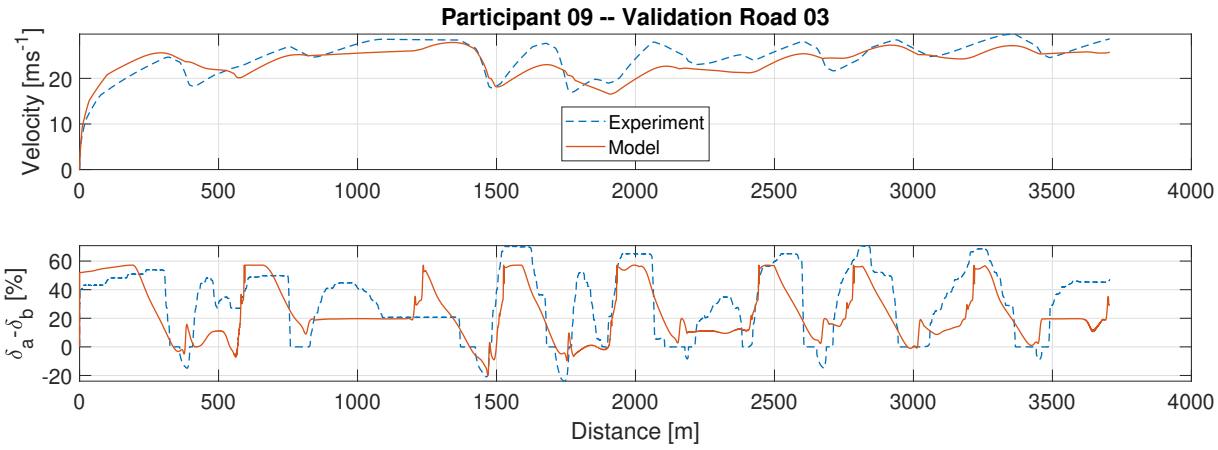


Figure B.51: One-size-fits-all model results, participant 09 on validation road 3

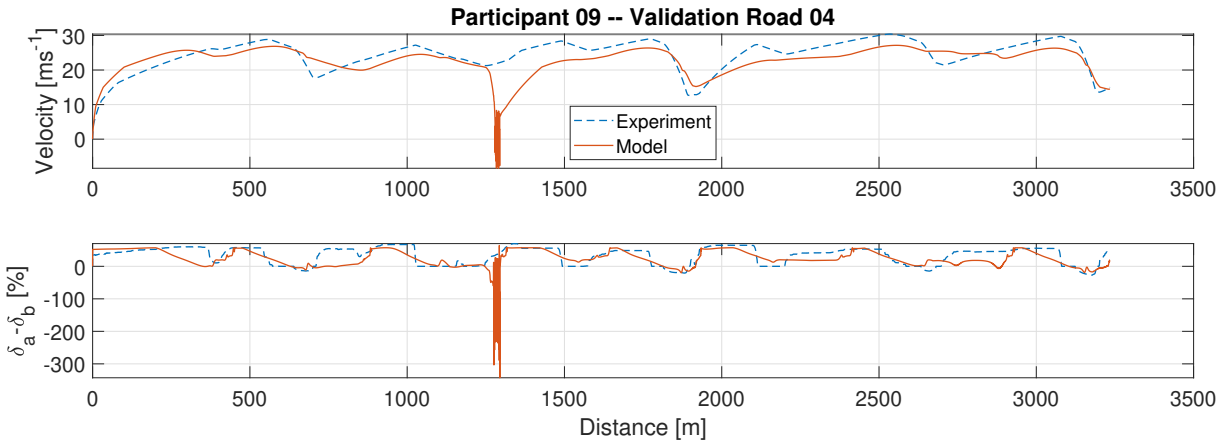


Figure B.52: One-size-fits-all model results, participant 09 on validation road 4

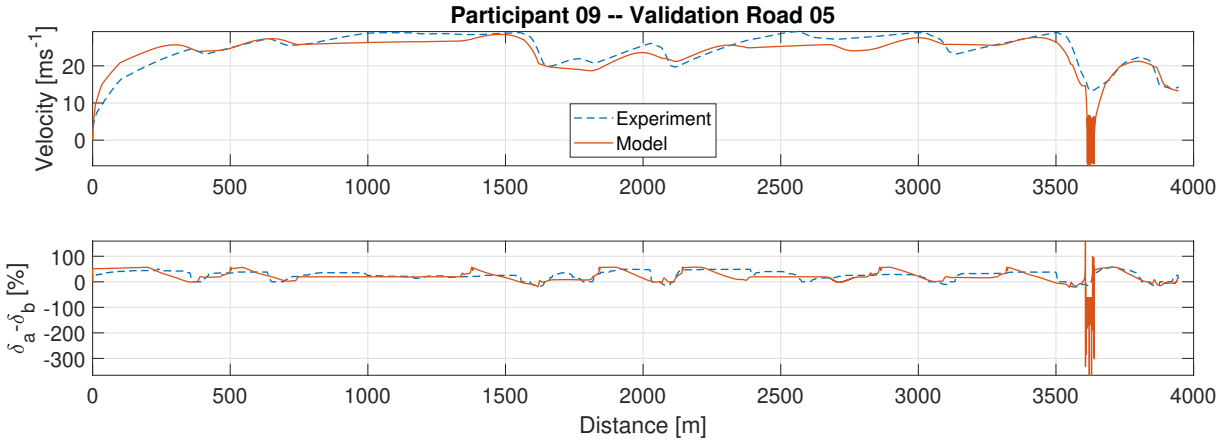


Figure B.53: One-size-fits-all model results, participant 09 on validation road 5

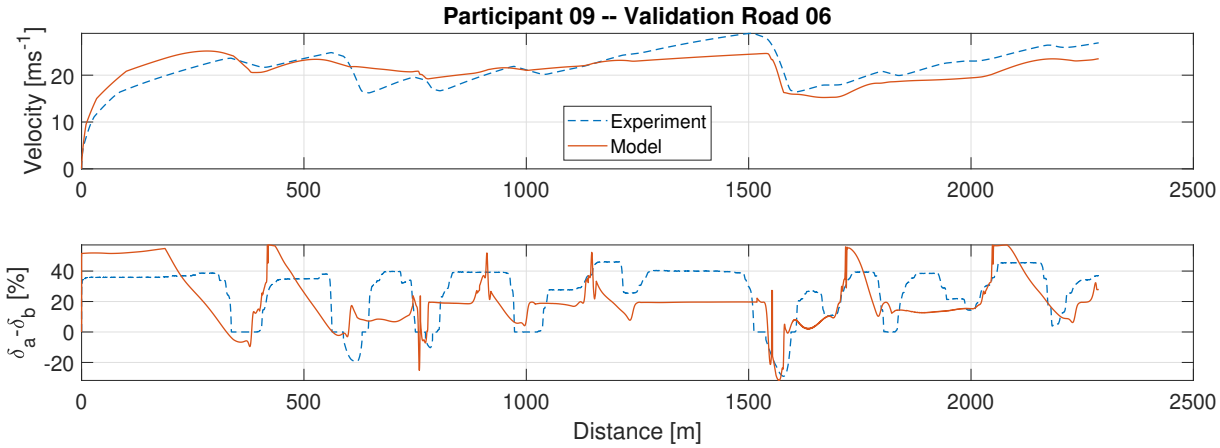


Figure B.54: One-size-fits-all model results, participant 09 on validation road 6

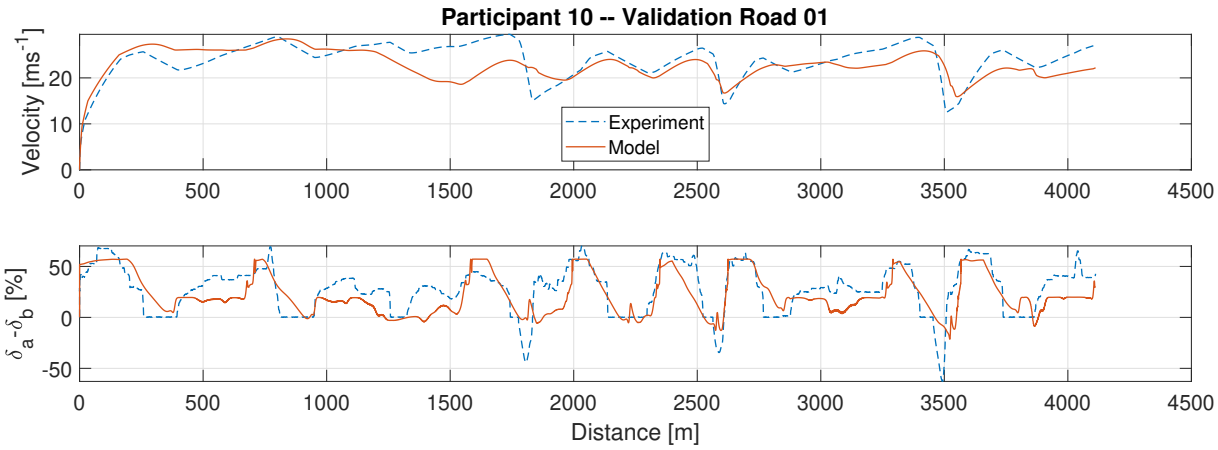


Figure B.55: One-size-fits-all model results, participant 10 on validation road 1

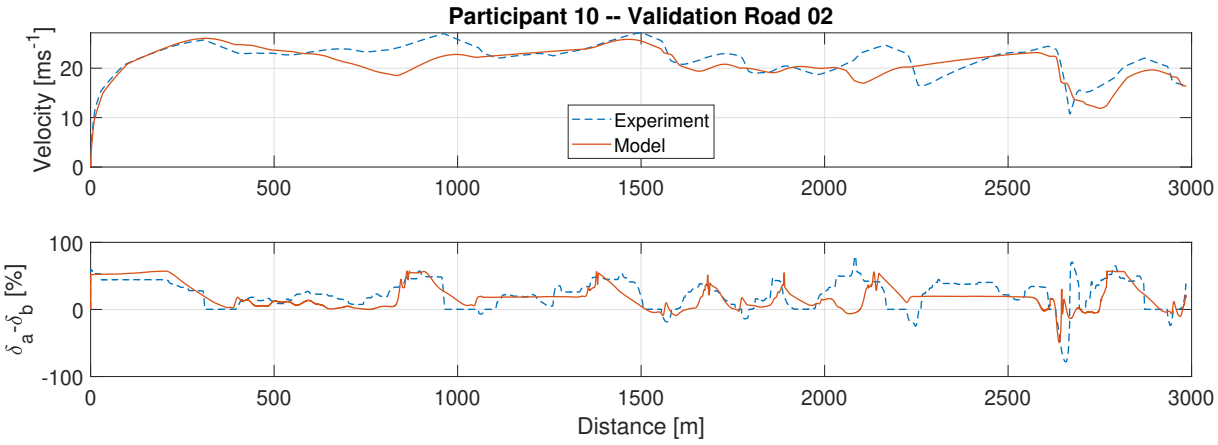


Figure B.56: One-size-fits-all model results, participant 10 on validation road 2

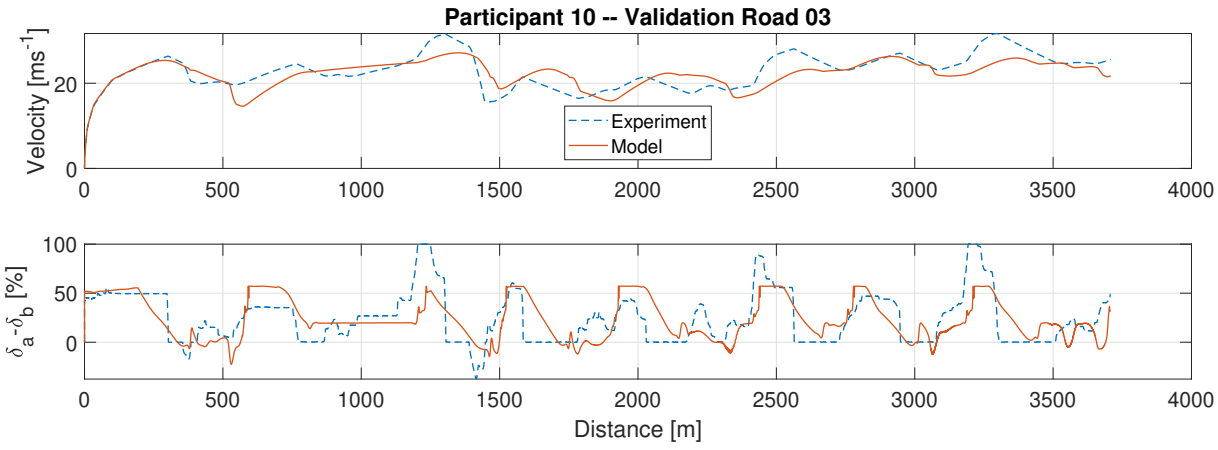


Figure B.57: One-size-fits-all model results, participant 10 on validation road 3

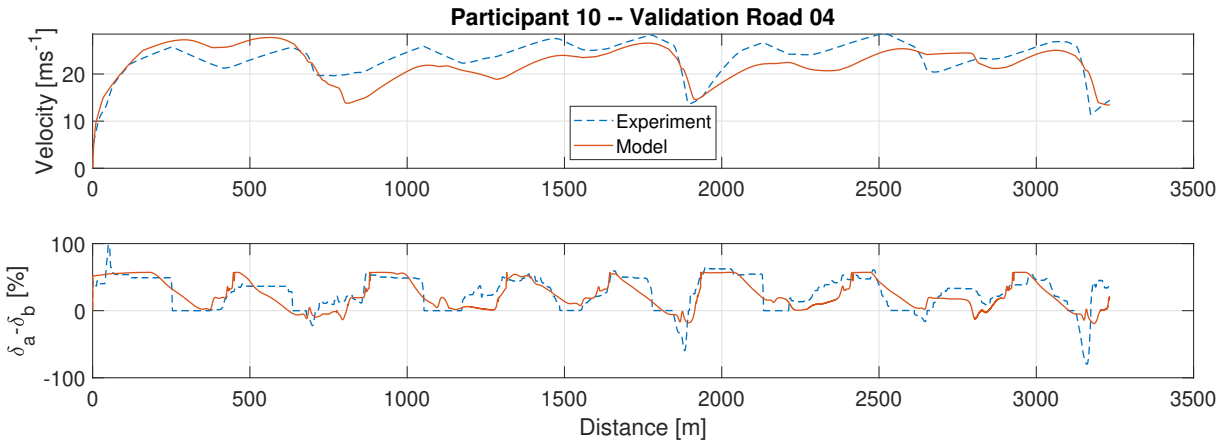


Figure B.58: One-size-fits-all model results, participant 10 on validation road 4

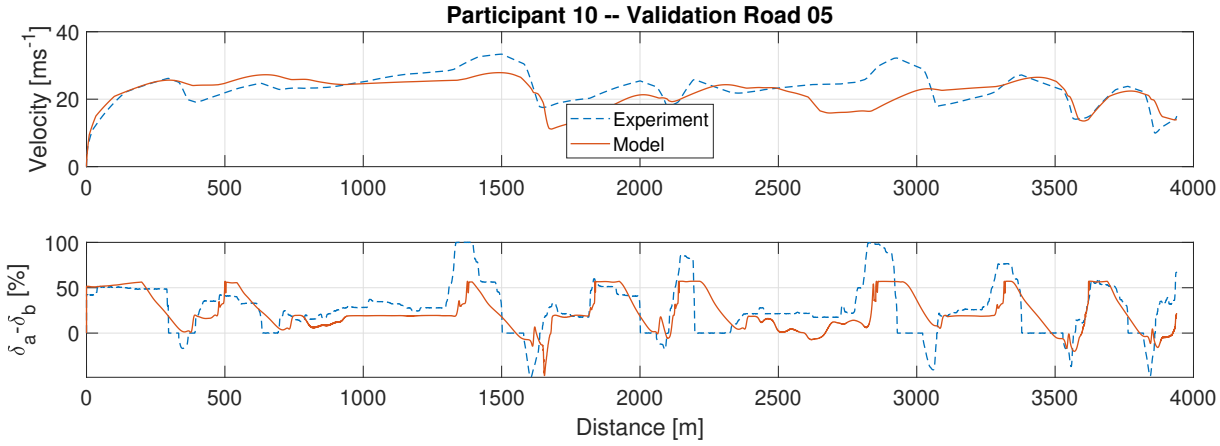


Figure B.59: One-size-fits-all model results, participant 10 on validation road 5

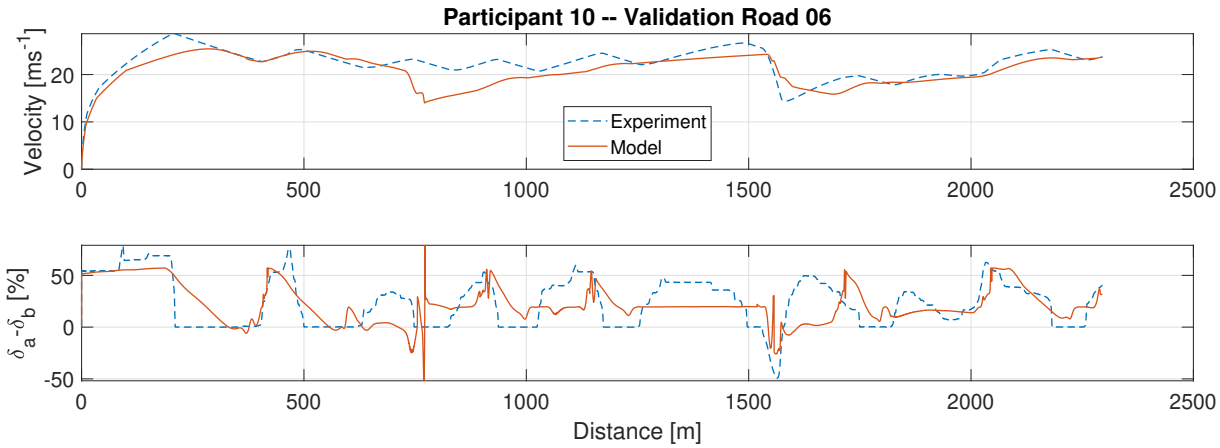


Figure B.60: One-size-fits-all model results, participant 10 on validation road 6

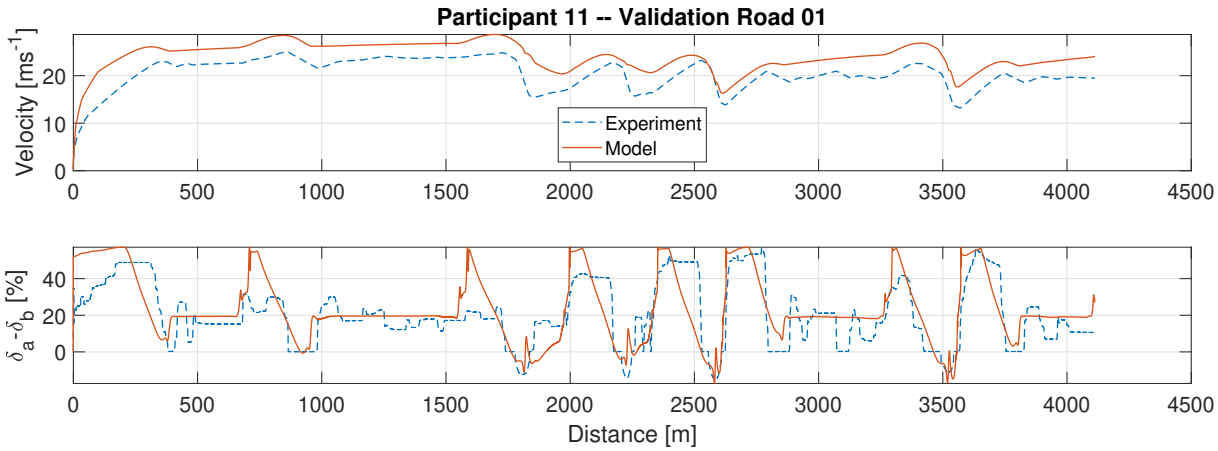


Figure B.61: One-size-fits-all model results, participant 11 on validation road 1

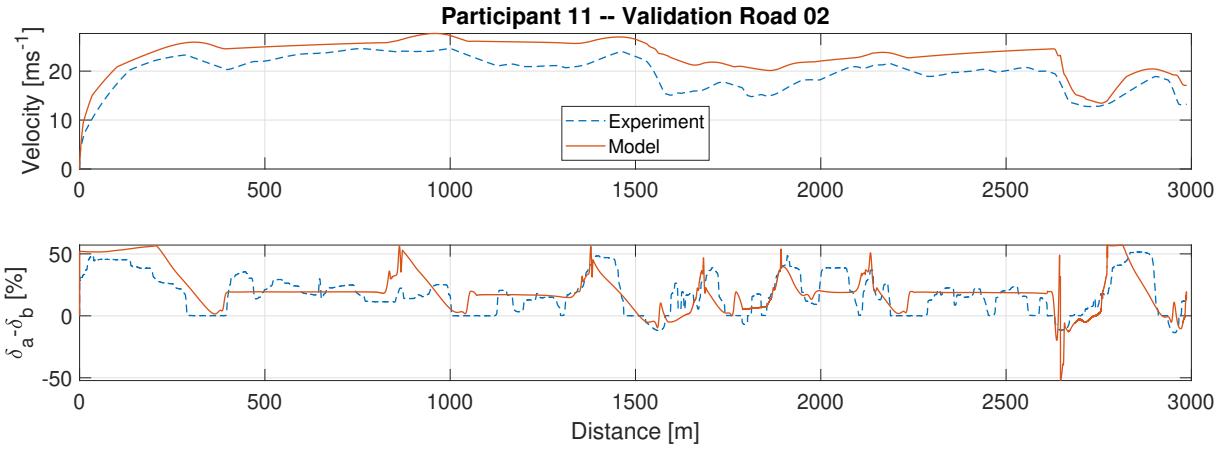


Figure B.62: One-size-fits-all model results, participant 11 on validation road 2

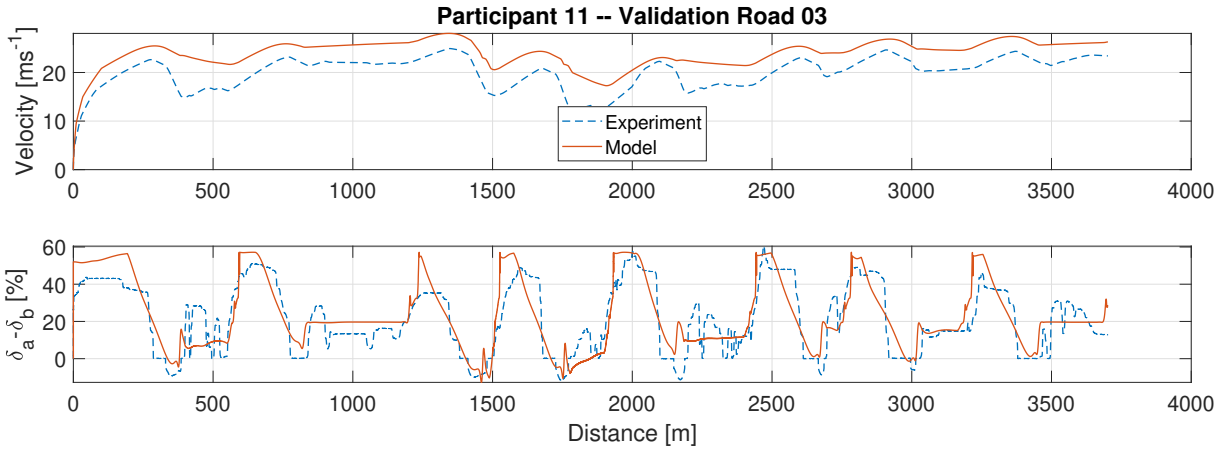


Figure B.63: One-size-fits-all model results, participant 11 on validation road 3

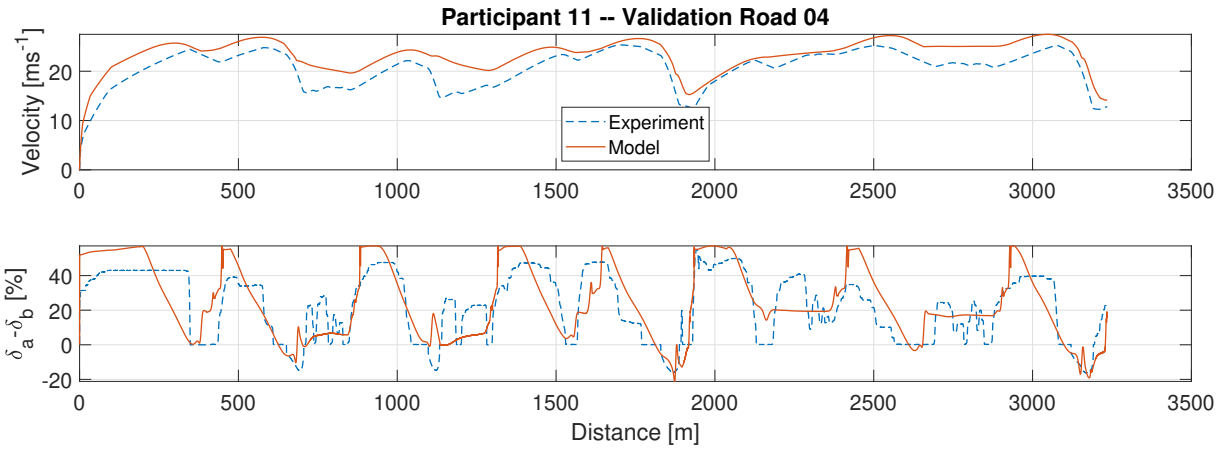


Figure B.64: One-size-fits-all model results, participant 11 on validation road 4

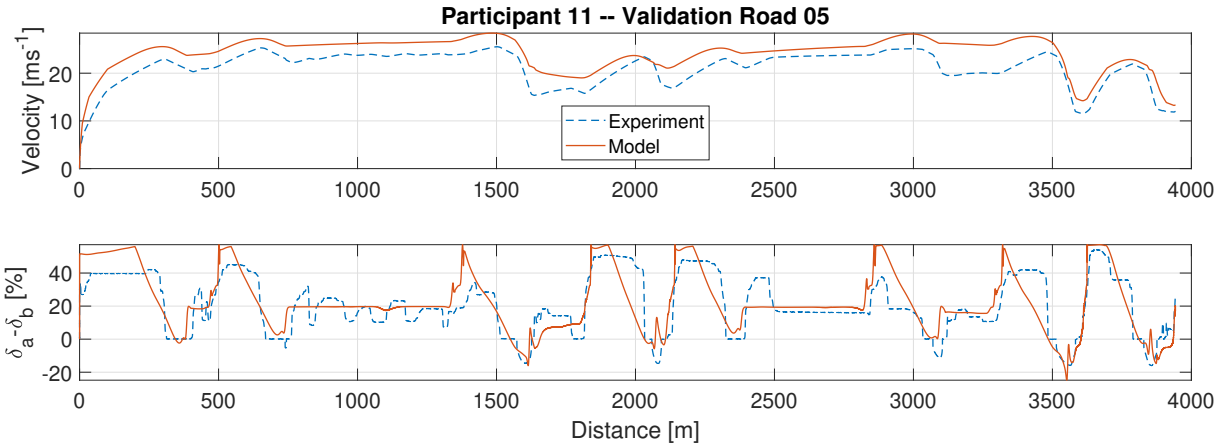


Figure B.65: One-size-fits-all model results, participant 11 on validation road 5

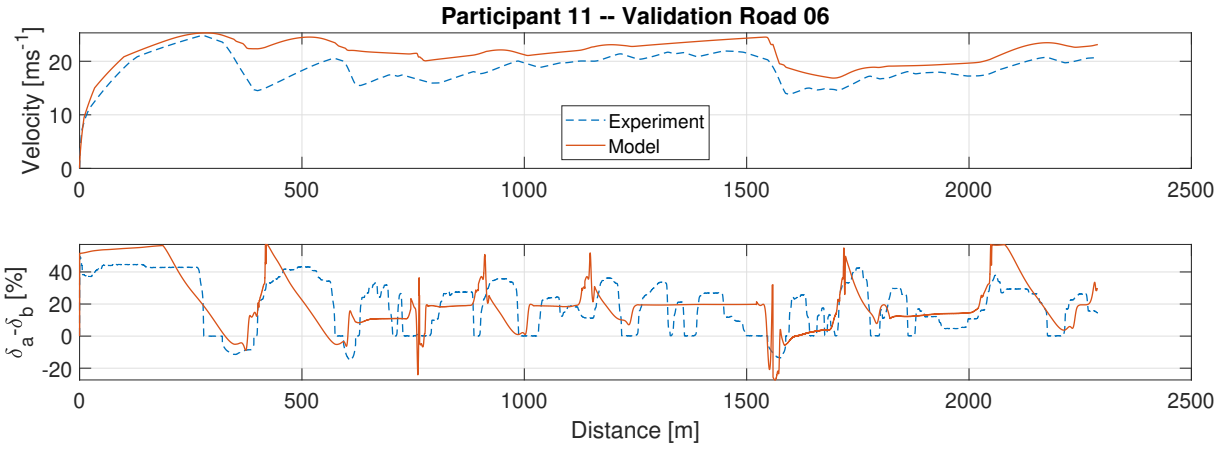


Figure B.66: One-size-fits-all model results, participant 11 on validation road 6

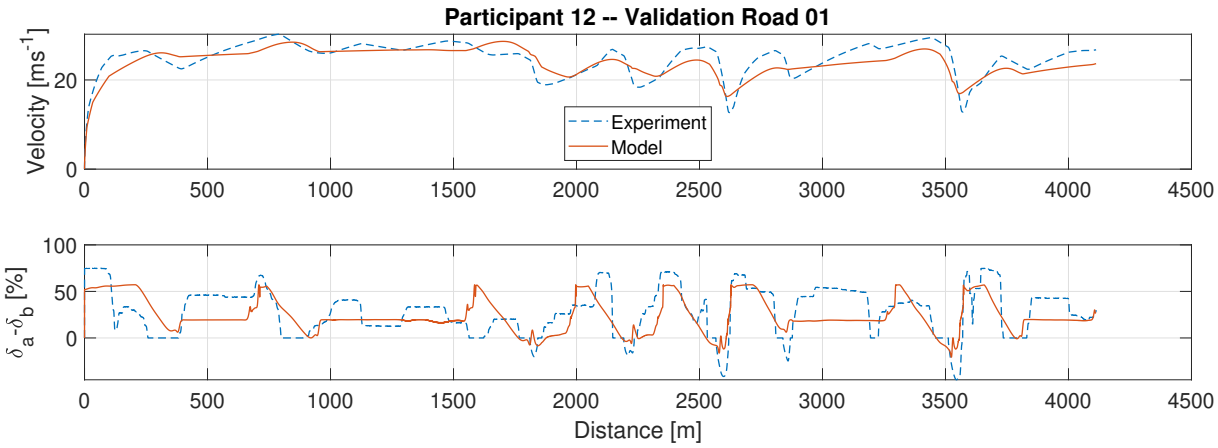


Figure B.67: One-size-fits-all model results, participant 12 on validation road 1

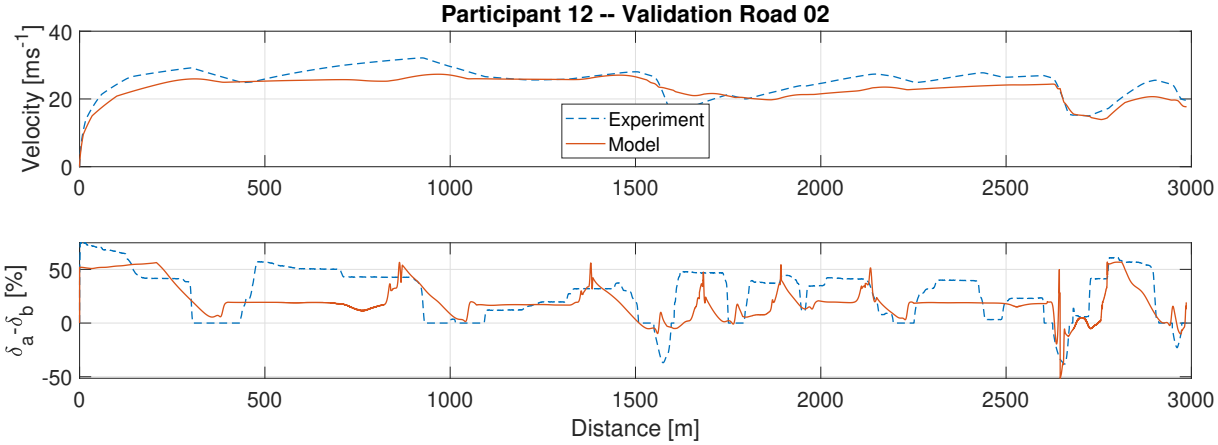


Figure B.68: One-size-fits-all model results, participant 12 on validation road 2

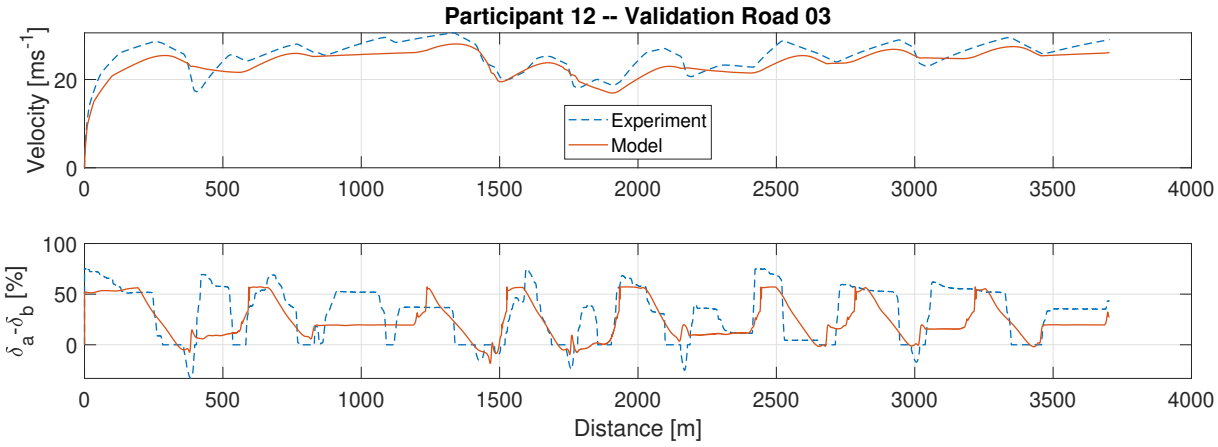


Figure B.69: One-size-fits-all model results, participant 12 on validation road 3

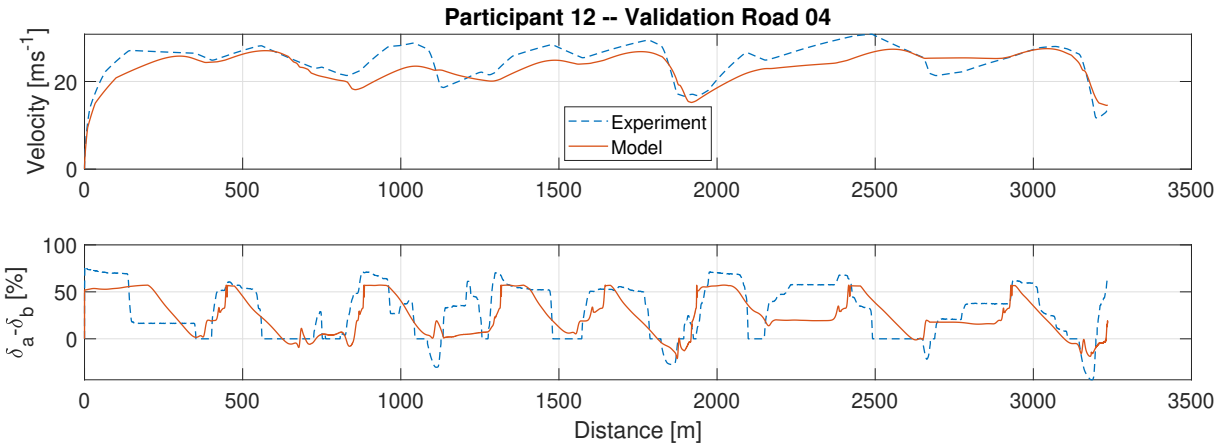


Figure B.70: One-size-fits-all model results, participant 12 on validation road 4

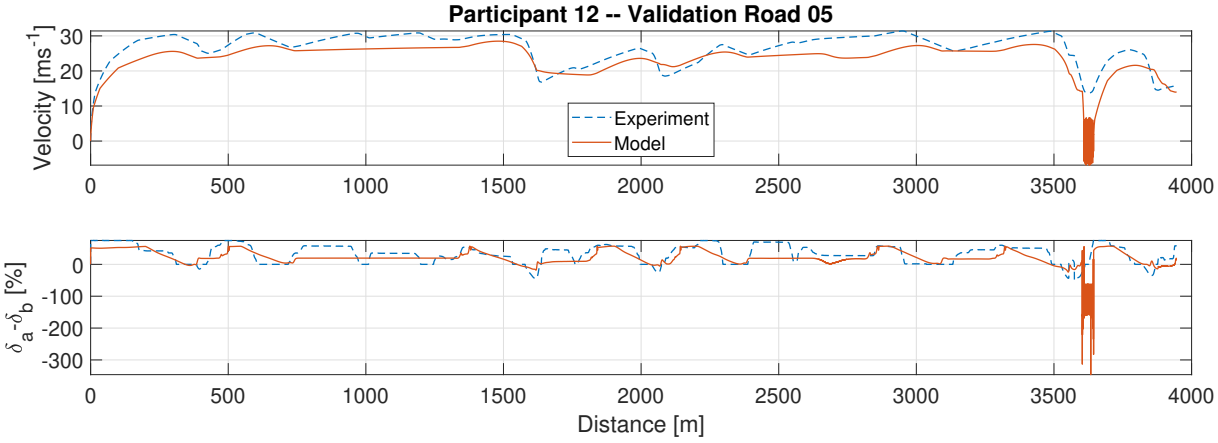


Figure B.71: One-size-fits-all model results, participant 12 on validation road 5

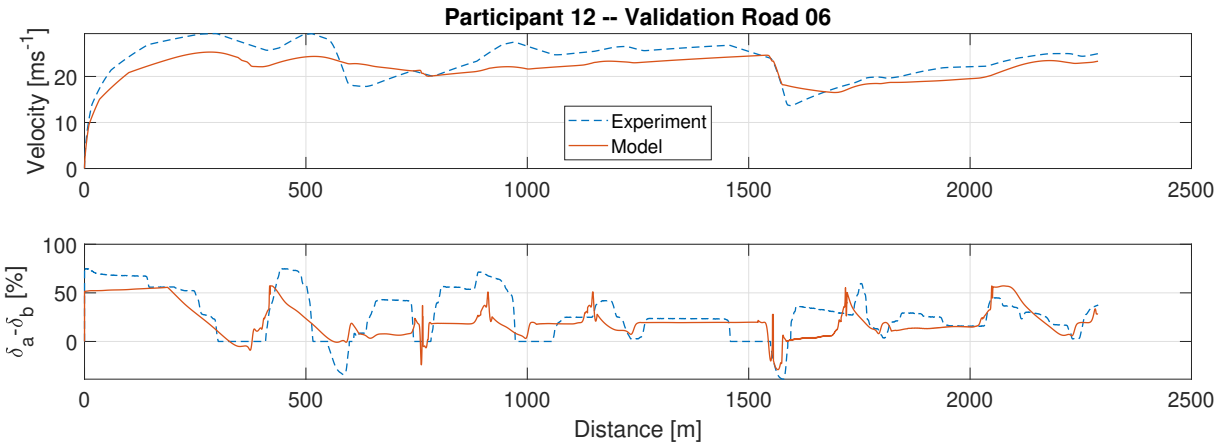


Figure B.72: One-size-fits-all model results, participant 12 on validation road 6

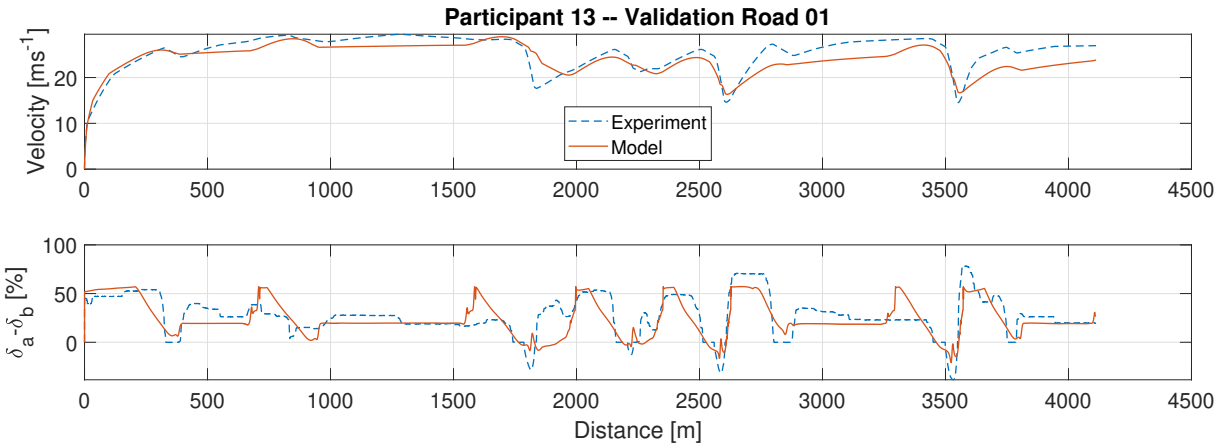


Figure B.73: One-size-fits-all model results, participant 13 on validation road 1

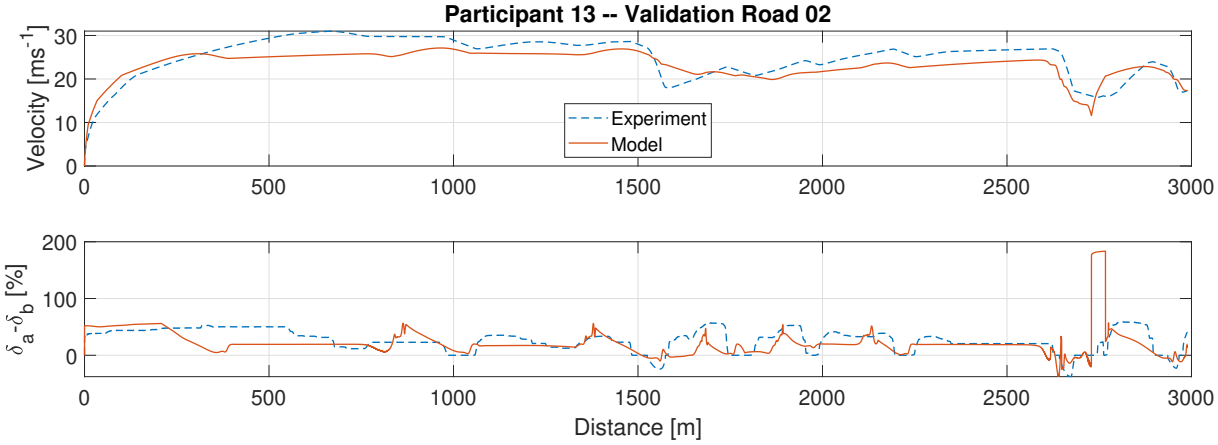


Figure B.74: One-size-fits-all model results, participant 13 on validation road 2

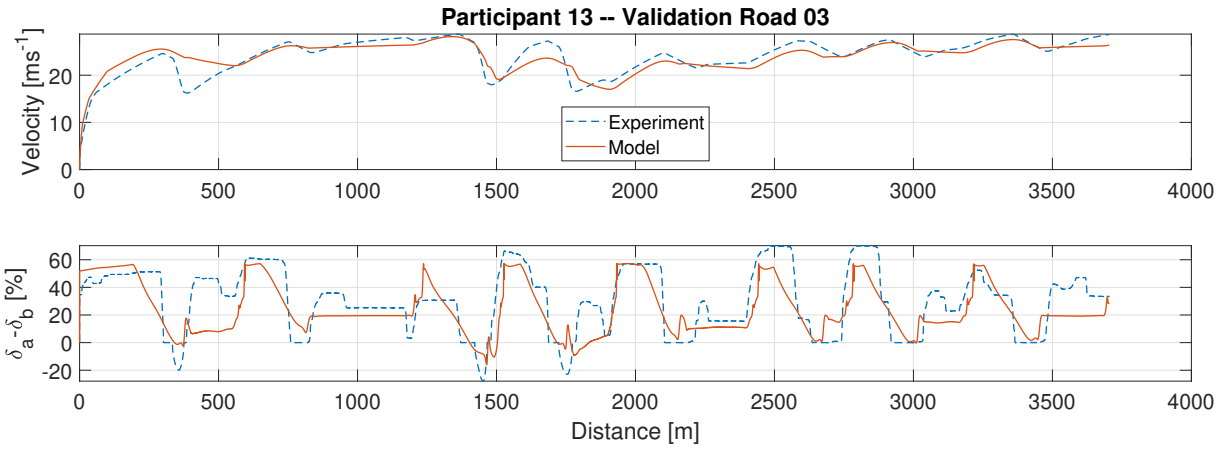


Figure B.75: One-size-fits-all model results, participant 13 on validation road 3

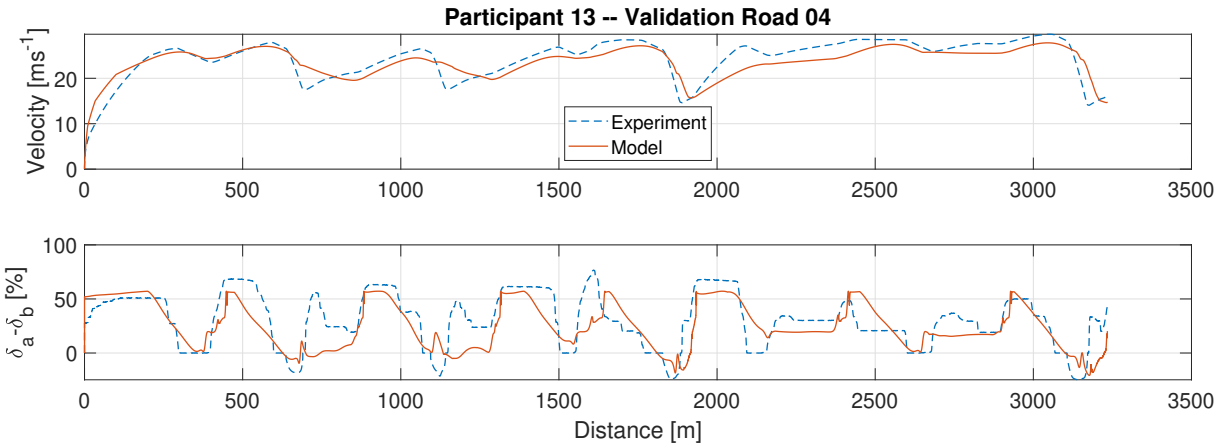


Figure B.76: One-size-fits-all model results, participant 13 on validation road 4

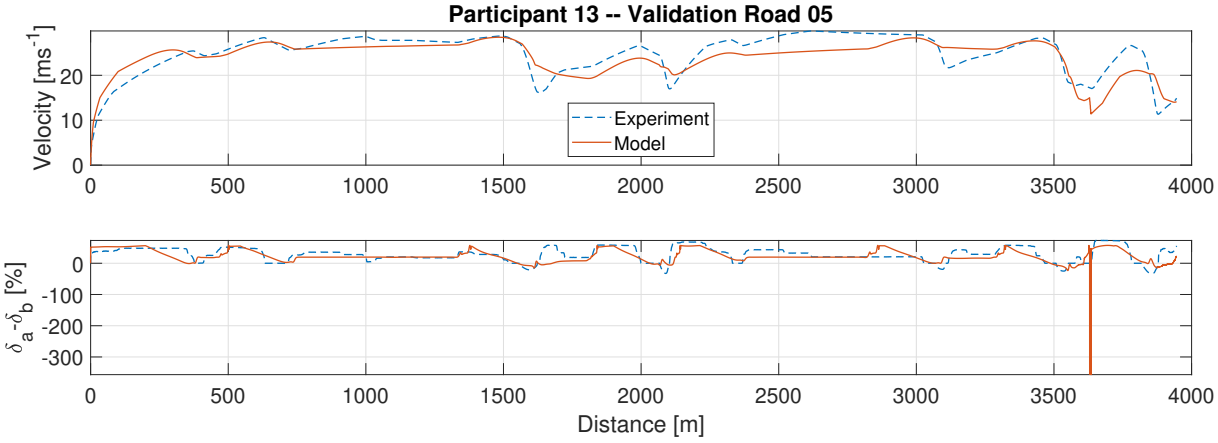


Figure B.77: One-size-fits-all model results, participant 13 on validation road 5

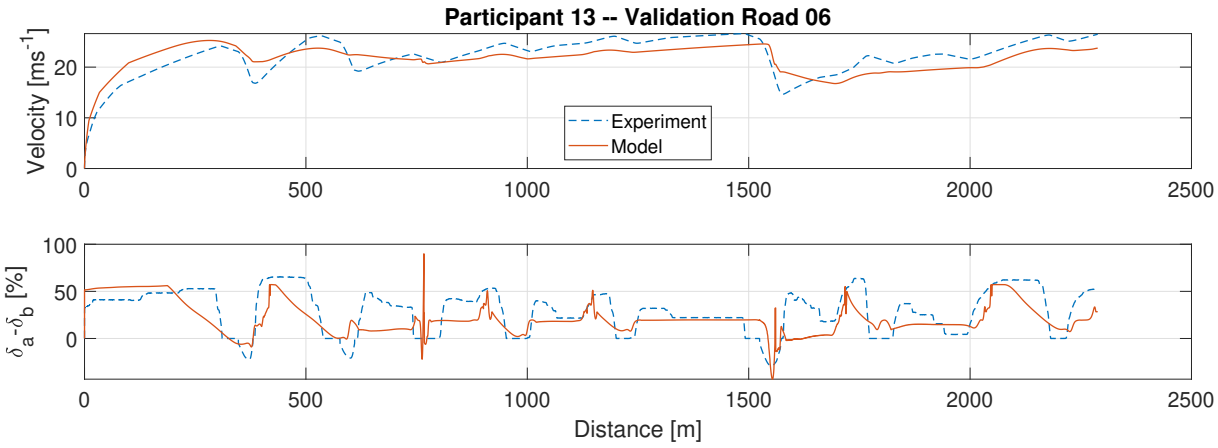


Figure B.78: One-size-fits-all model results, participant 13 on validation road 6

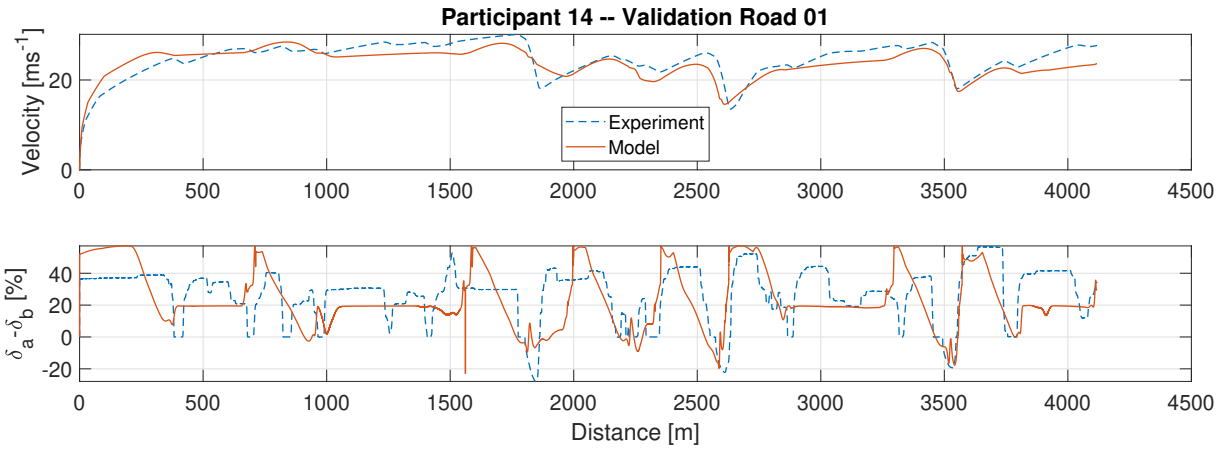


Figure B.79: One-size-fits-all model results, participant 14 on validation road 1

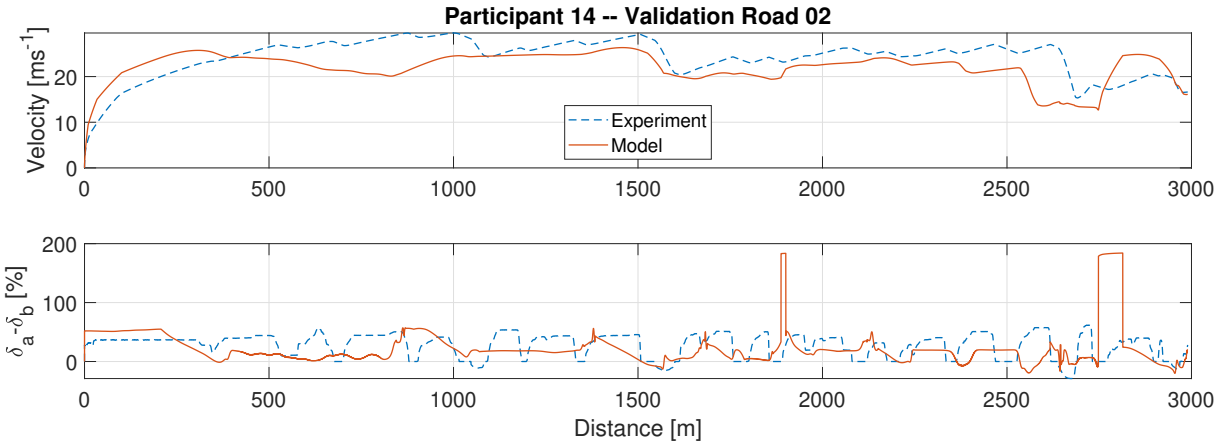


Figure B.80: One-size-fits-all model results, participant 14 on validation road 2

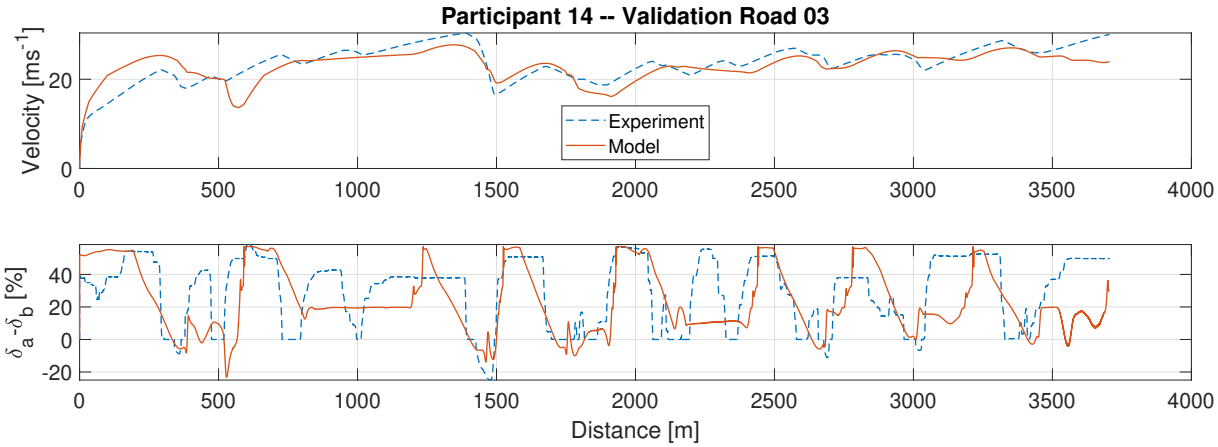


Figure B.81: One-size-fits-all model results, participant 14 on validation road 3

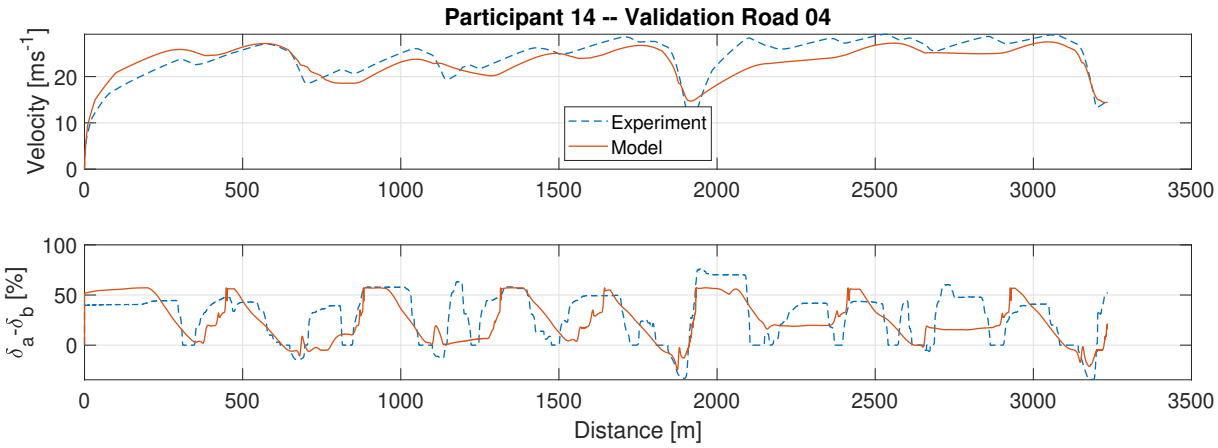


Figure B.82: One-size-fits-all model results, participant 14 on validation road 4

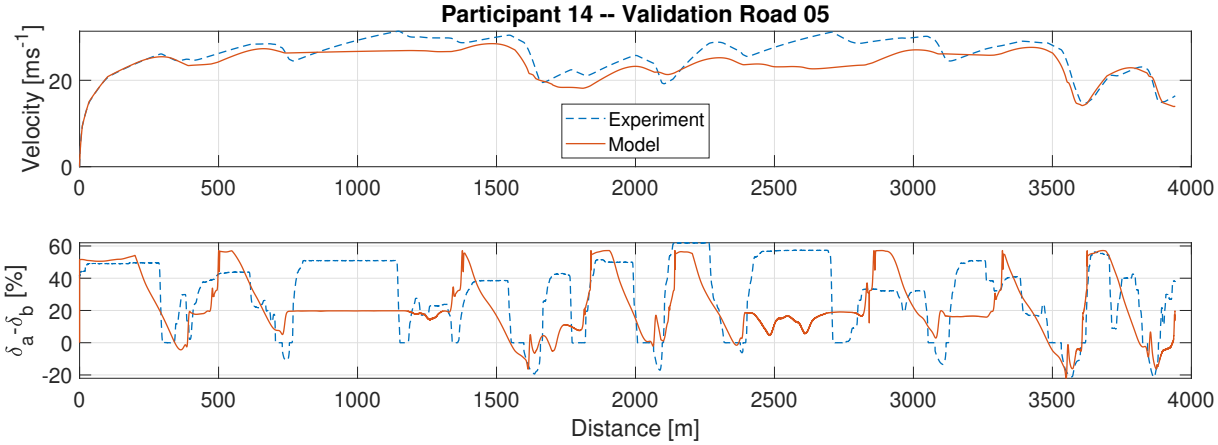


Figure B.83: One-size-fits-all model results, participant 14 on validation road 5

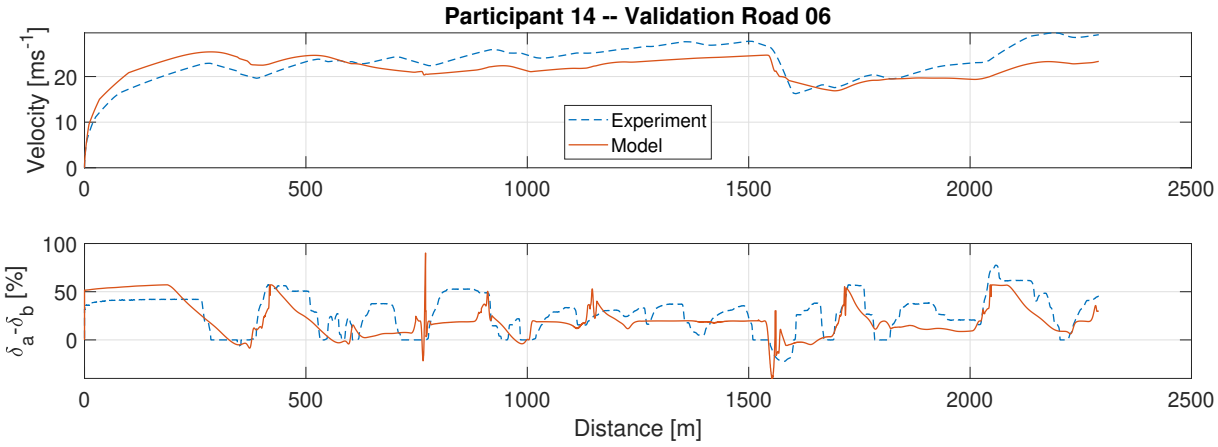


Figure B.84: One-size-fits-all model results, participant 14 on validation road 6

2). Variability Tables

		VRoad1 - MSE													
participant	test 1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
model 1	4.17	6.97	4.66	7.69	11.40	8.41	16.30	8.85	8.13	13.40	18.38	5.57	5.26	4.68	8.85
2	2.86	3.18	8.84	5.40	11.03	9.27	14.93	10.99	8.59	12.40	23.24	4.63	3.91	7.38	9.05
3	15.01	16.04	4.76	6.65	8.31	12.78	6.46	1.31	16.99	10.74	8.63	12.49	12.98	7.45	10.04
4	12.12	11.81	13.16	3.65	20.41	13.34	8.32	11.54	19.79	19.25	11.41	11.36	14.41	17.48	13.43
5	35.90	31.44	24.73	20.56	44.45	35.38	18.70	18.80	38.44	30.60	22.33	32.65	33.88	29.36	29.80
6	2.99	5.39	8.11	8.91	11.56	6.33	21.30	14.14	10.77	10.38	22.42	7.94	6.87	6.44	10.25
7	21.00	20.51	12.53	7.12	28.66	18.12	5.31	3.13	23.03	25.36	7.14	17.57	17.34	14.65	15.82
8	20.79	23.56	6.64	14.28	11.92	15.27	11.10	3.30	20.63	14.32	8.80	16.47	17.60	10.02	13.91
9	10.68	9.95	4.76	6.81	16.58	10.44	9.57	2.20	11.00	10.92	11.56	8.23	6.88	5.39	8.93
10	23.29	22.68	8.77	9.73	28.55	18.95	5.77	3.08	24.95	16.81	5.31	18.89	20.77	13.28	15.77
11	16.93	17.29	13.17	7.30	13.97	12.80	5.06	5.31	20.40	19.56	4.14	12.89	15.24	12.81	12.63
12	4.28	6.16	4.85	5.01	3.18	7.23	13.87	6.60	8.90	10.13	15.83	5.23	5.57	4.92	7.27
13	6.97	8.85	9.42	9.93	12.02	13.15	17.28	6.33	7.83	14.73	20.64	7.19	4.99	3.92	10.23
14	6.26	9.99	3.53	10.67	12.64	7.51	17.13	4.89	6.45	5.83	20.74	6.17	4.57	2.31	8.48
Mean	11.33	12.49	7.94	7.93	14.63	11.82	11.72	6.28	14.42	14.14	13.71	10.36	10.49	8.52	11.13

	VRoad1 - VAF														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
1	84.39	57.80	73.23	47.70	38.67	59.04	76.38	76.78	77.10	43.23	87.81	72.79	81.32	80.38	8.33
2	89.46	81.98	45.18	76.23	37.18	53.29	81.54	66.31	72.83	48.45	79.79	75.85	83.78	67.61	68.53
3	60.57	38.90	79.90	42.59	68.72	55.24	75.49	93.44	71.56	69.22	79.06	58.31	76.25	87.07	68.31
4	89.19	75.06	26.90	68.12	24.88	59.08	73.93	43.06	64.41	40.63	61.02	70.27	70.04	44.36	57.93
5	0.00	0.00	0.00	0.00	0.00	0.00	5.77	8.28	18.07	1.97	0.00	0.00	12.88	16.17	4.51
6	85.75	65.01	58.84	58.31	32.78	67.89	70.83	69.74	64.23	54.09	83.06	58.73	70.99	72.39	65.19
7	60.19	41.44	60.22	42.59	24.52	48.98	77.05	86.63	69.93	50.26	71.19	55.36	74.15	76.96	59.96
8	43.48	2.28	74.19	0.00	55.00	55.72	47.99	84.39	71.76	60.96	64.98	49.66	69.63	87.10	54.80
9	69.26	55.19	74.61	41.50	32.61	56.66	79.14	91.38	78.54	64.57	88.83	66.75	84.67	85.69	69.24
10	55.98	41.75	78.13	35.81	36.21	51.60	70.69	90.55	70.31	62.60	75.85	54.03	73.72	86.67	63.14
11	85.71	63.03	61.83	46.63	75.84	77.31	76.82	75.05	80.65	60.13	89.59	78.12	85.42	80.66	74.06
12	88.43	68.60	69.91	64.74	81.80	67.03	78.30	82.13	76.31	57.46	88.77	75.41	81.80	80.48	75.80
13	66.23	43.93	44.38	38.63	36.27	34.91	80.65	91.98	75.73	42.14	81.19	62.52	80.20	83.77	61.61
14	69.83	36.13	79.29	22.96	29.85	62.32	68.57	91.98	81.05	74.91	81.52	67.96	82.54	90.50	67.10
Mean	72.96	51.62	63.59	45.06	44.18	57.62	73.65	80.26	73.42	56.05	79.44	65.06	78.04	78.74	65.69

Figure B.85: Variability table (MSE and VAF on velocity) for validation road 1

		VRoad2 - MSE													
participant	test 1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
model 1	3.05	7.88	10.11	4.72	2.97	6.90	10.84	10.53	17.83	8.40	12.10	8.76	10.58	40.98	11.12
2	3.21	4.41	14.40	3.07	5.24	2.61	7.82	11.57	22.69	5.52	16.96	8.87	12.89	39.33	11.33
3	13.61	17.84	10.99	8.20	11.26	10.10	5.71	4.65	27.54	8.46	7.72	24.06	17.57	14.26	13.00
4	12.73	15.10	27.20	3.37	12.98	7.00	11.02	13.05	36.80	14.69	5.71	19.03	26.42	60.32	18.96
5	34.76	35.85	29.59	25.53	33.99	24.15	24.62	9.88	52.38	22.14	26.67	47.75	45.09	59.54	33.71
6	4.34	3.68	5.93	5.66	3.63	5.63	9.14	11.51	17.72	5.10	18.74	6.18	10.83	32.96	10.07
7	21.36	23.39	22.76	11.17	17.99	14.56	13.09	6.30	41.00	22.73	11.91	31.03	30.19	62.72	23.59
8	14.45	21.04	12.72	12.80	12.00	15.51	8.27	8.81	23.43	11.93	9.74	24.01	16.13	13.11	14.57
9	8.56	11.38	10.69	8.63	6.76	8.81	6.00	11.02	20.62	9.37	12.34	16.94	16.22	31.65	12.78
10	21.40	25.44	17.02	13.36	17.63	15.17	9.04	6.31	40.13	12.57	7.56	34.01	34.19	37.10	20.78
11	15.34	19.97	27.49	6.16	12.88	10.45	8.17	13.93	29.43	16.19	2.11	23.27	20.66	35.87	17.28
12	2.98	4.72	10.95	3.32	3.21	3.06	6.33	10.46	16.62	5.38	13.67	8.35	7.77	17.93	8.20
13	13.87	9.51	16.29	8.74	7.27	12.90	10.42	6.41	21.83	10.98	19.34	17.37	16.68	37.71	14.95
14	3.82	8.08	7.16	6.46	3.59	6.87	6.10	14.96	15.29	6.08	17.59	9.53	11.52	28.40	10.39
Mean	10.67	13.26	14.90	7.36	9.03	9.20	8.61	9.96	25.46	10.57	11.96	17.80	17.82	34.80	14.39

	VRoad2 - VAF														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
1	90.70	70.81	69.69	72.60	89.05	53.65	50.74	29.59	71.37	48.63	88.58	82.38	78.69	14.21	65.05
2	88.92	83.60	54.87	83.29	81.00	82.50	67.02	17.01	59.80	64.01	84.92	77.40	69.27	8.43	65.86
3	77.52	63.36	91.07	56.18	83.59	53.66	72.34	64.58	57.08	67.20	71.22	59.58	75.83	75.48	69.19
4	73.74	70.21	34.80	84.57	69.88	69.96	46.21	0.00	48.73	43.21	79.48	76.60	57.32	0.00	53.91
5	0.00	0.00	30.82	0.00	0.00	0.00	0.00	41.71	50.35	0.00	0.00	0.00	33.63	42.81	14.24
6	79.61	82.13	75.99	80.43	85.82	63.41	76.27	49.68	63.65	64.37	91.24	81.71	72.07	21.94	70.59
7	44.68	29.24	62.97	33.67	54.10	21.59	36.79	68.81	73.85	17.29	47.12	35.68	68.82	39.15	45.27
8	71.98	48.33	86.69	25.91	81.01	20.00	59.34	30.55	64.26	33.79	58.50	61.26	77.57	75.63	56.77
9	80.09	65.33	87.38	42.92	82.38	44.46	70.79	12.08	80.17	52.94	68.87	64.01	77.69	54.82	63.14
10	71.80	64.09	89.39	42.97	78.44	50.11	62.53	64.92	76.92	57.46	60.52	52.94	71.21	69.63	65.21
11	92.02	77.08	67.34	79.21	88.35	71.08	69.44	10.30	78.28	60.38	94.15	85.97	86.63	55.60	72.56
12	92.23	84.56	71.39	81.05	89.57	79.56	76.99	23.23	63.14	67.52	89.07	80.69	81.25	49.36	73.54
13	51.96	55.24	74.73	49.61	73.16	14.96	49.44	48.83	75.58	32.70	59.72	50.45	70.51	44.99	53.71
14	86.60	67.19	82.33	59.17	87.95	53.90	77.57	0.00	79.31	57.73	75.53	73.66	76.56	36.25	65.27
Mean	77.06	66.24	72.97	60.89	80.33	52.22	62.73	32.28	68.63	51.32	74.53	67.87	74.11	41.96	63.08

Figure B.86: Variability table (MSE and VAF on velocity) for validation road 2

						VRoad3 - MSE								
participant	test 1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
model 1	7.22	7.47	3.84	7.69	4.43	11.11	4.54	12.48	6.86	6.06	15.48	7.32	4.52	9.39	7.74
2	5.78	1.76	6.31	5.74	2.03	3.52	5.45	15.05	6.39	9.56	24.97	2.29	4.61	10.52	7.43
3	12.19	13.26	2.08	8.08	11.23	11.60	7.80	4.07	9.33	9.75	10.17	15.18	6.57	6.44	9.12
4	8.65	10.12	7.95	1.33	10.17	9.62	4.96	10.72	14.88	9.32	7.27	14.88	9.43	21.60	10.06
5	12.28	9.86	4.37	12.68	8.91	9.97	11.13	10.71	4.88	11.02	23.22	11.43	4.79	4.92	10.01
6	7.21	3.96	5.89	8.91	2.70	10.58	6.35	19.36	6.70	7.39	27.47	4.64	6.78	11.38	9.24
7	13.28	14.04	3.90	7.75	13.00	14.05	8.44	3.72	10.65	11.20	9.81	18.06	7.03	11.05	10.43
8	17.01	19.55	4.26	12.94	16.88	16.44	11.41	5.02	13.87	13.06	8.90	20.89	10.65	8.56	12.82
9	9.82	8.12	2.03	9.87	7.67	8.78	6.91	8.02	5.43	9.55	15.83	9.80	2.72	6.61	7.94
10	15.92	17.32	3.76	10.21	15.27	16.24	10.21	3.99	12.62	12.33	7.93	19.61	9.27	8.26	11.64
11	12.32	16.26	6.54	5.93	14.78	14.36	8.16	5.56	15.93	12.61	3.11	18.15	10.64	14.39	11.34
12	5.80	3.81	2.55	6.06	2.70	10.04	4.63	12.77	5.59	7.76	18.03	5.06	3.75	7.42	6.86
13	9.40	7.82	3.43	12.96	6.45	15.67	9.13	10.61	2.26	12.29	22.86	8.38	2.45	8.40	9.44
14	8.26	6.11	2.62	10.60	5.25	6.19	6.20	10.63	4.04	7.78	22.53	5.59	3.45	4.97	7.44
Mean	10.22	9.97	4.24	8.31	8.66	11.40	7.24	9.38	8.81	9.90	14.95	11.53	6.30	9.92	9.35

						VRoad3 - VAF								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
1	62.58	58.08	82.35	43.91	76.59	30.74	76.72	72.08	73.70	71.71	85.10	78.76	80.07	59.78	68.01
2	71.14	88.16	71.42	69.57	85.57	74.36	75.59	58.64	73.38	51.97	86.56	88.54	83.94	53.07	73.71
3	50.51	52.63	90.51	27.47	66.25	54.05	64.55	81.26	85.68	60.52	69.78	66.42	85.79	87.00	67.32
4	75.17	81.27	54.36	89.28	80.02	73.64	81.29	48.79	60.98	60.86	84.50	84.29	69.11	23.27	69.06
5	34.27	38.38	76.08	1.53	41.75	37.04	41.98	70.11	84.89	44.65	47.51	49.20	78.73	80.35	51.89
6	63.10	72.34	82.53	64.13	80.31	24.19	75.72	64.08	70.97	65.22	82.68	76.36	72.72	49.15	67.39
7	54.58	57.79	83.86	31.30	59.29	53.76	61.17	82.80	89.27	58.86	64.41	68.59	85.23	79.08	66.43
8	34.60	26.70	82.51	0.00	44.51	32.36	53.74	73.85	76.51	51.04	64.13	51.17	76.60	86.68	53.89
9	49.45	57.44	88.04	17.74	58.49	48.44	62.81	72.65	86.82	56.83	75.04	66.86	88.18	80.34	64.94
10	41.73	46.30	88.04	12.50	58.35	45.24	59.71	79.08	83.35	55.65	68.57	58.73	83.53	87.45	62.02
11	72.93	73.20	79.44	59.02	78.82	75.72	79.77	70.24	78.85	67.45	92.76	84.08	81.75	70.48	76.04
12	68.86	78.05	89.54	62.42	86.22	35.99	76.19	70.22	78.61	61.78	88.43	83.92	83.20	67.17	73.62
13	49.34	47.31	86.42	9.05	55.39	0.00	53.51	76.69	91.45	39.08	64.23	62.63	89.43	64.32	56.35
14	56.05	58.90	88.88	25.44	64.45	56.64	68.55	72.88	84.19	61.24	79.19	71.54	85.14	79.15	68.02
Mean	57.69	61.40	82.15	39.37	68.79	46.55	68.41	71.02	79.52	58.63	77.34	72.45	81.90	68.23	66.67

Figure B.87: Variability table (MSE and VAF on velocity) for validation road 3

						VRoad4 - MSE								
participant	test 1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
model 1	4.07	5.62	3.49	4.70	10.54	4.52	4.05	7.62	14.68	11.05	8.38	9.44	5.36	7.24	7.20
2	4.67	3.77	8.63	5.01	9.41	3.23	3.53	8.66	13.65	8.27	14.74	6.25	5.68	6.98	7.32
3	16.78	18.55	6.37	11.50	16.51	14.04	11.87	7.08	16.91	12.30	6.43	19.61	12.63	12.08	13.05
4	8.57	12.71	13.30	3.70	16.56	6.16	6.81	6.93	20.37	13.56	6.25	14.58	12.75	16.96	11.37
5	21.32	18.99	10.64	14.87	20.35	17.39	14.56	12.14	16.51	12.19	16.92	20.61	13.31	12.79	15.90
6	4.86	3.90	7.66	8.12	9.74	4.70	5.85	12.63	14.03	8.85	16.27	6.67	7.19	6.58	8.36
7	20.85	21.25	10.73	11.35	25.91	15.94	13.33	8.73	20.24	14.30	7.58	23.09	13.38	16.41	15.94
8	20.05	24.97	9.88	17.61	22.78	19.73	17.77	10.11	22.34	18.16	5.46	28.03	17.51	17.43	17.99
9	10.92	12.26	3.88	9.08	14.71	7.49	6.42	7.67	12.57	8.83	6.93	14.10	5.41	5.47	8.98
10	22.22	24.85	9.78	16.31	27.94	19.42	16.82	10.31	21.99	16.12	6.09	25.88	17.81	16.79	18.02
11	14.22	16.00	11.82	8.42	16.80	13.28	10.10	7.92	19.62	16.01	1.54	20.37	12.21	16.75	13.22
12	5.20	5.59	5.54	5.76	4.70	3.96	3.76	7.77	14.31	9.62	10.13	7.65	5.96	6.11	6.86
13	11.36	9.75	8.01	10.06	13.48	8.07	6.64	10.25	15.27	10.06	15.45	9.09	5.28	4.82	9.83
14	7.54	8.72	3.73	8.81	14.05	7.08	5.54	7.29	13.91	6.96	10.23	12.22	6.32	5.91	8.45
Mean	11.64	12.92	7.91	9.26	15.63	9.82	8.65	8.69	16.92	11.85	8.88	15.15	9.81	10.73	11.28

						VRoad4 - VAF								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
1	85.77	82.20	84.87	71.78	60.54	78.73	81.64	61.91	58.91	55.23	91.45	72.15	83.75	77.30	74.73
2	81.08	84.02	62.70	76.59	56.59	83.90	83.83	63.71	59.36	65.10	85.55	76.78	82.57	75.92	74.12
3	53.95	56.46	82.83	42.17	55.61	57.17	66.67	60.95	72.91	71.26	75.39	57.86	77.59	83.38	65.30
4	85.75	83.22	47.94	87.15	57.92	88.16	80.74	58.15	55.76	57.90	75.32	79.32	72.15	59.20	70.62
5	16.29	19.11	52.00	8.27	13.47	16.10	35.03	30.31	57.47	50.83	46.10	26.81	61.06	62.17	35.36
6	80.98	81.48	73.89	63.84	53.48	78.09	77.69	55.29	55.92	61.57	89.25	72.06	78.93	78.58	71.50
7	41.97	44.25	66.08	41.49	30.84	47.16	59.70	49.78	70.63	67.51	65.74	54.19	75.11	73.36	56.27
8	60.50	53.54	82.79	23.15	51.90	52.44	59.10	53.33	73.26	62.28	74.34	45.60	77.16	84.53	60.99
9	62.58	58.86	83.90	43.87	47.81	68.15	72.63	55.35	75.36	71.14	87.14	62.02	86.08	88.20	68.79
10	48.45	49.98	81.10	27.81	38.51	51.28	58.72	51.21	71.80	66.67	71.71	51.36	73.27	82.09	58.85
11	83.04	82.10	69.50	70.79	81.54	76.52	81.79	60.81	73.55	67.69	94.00	70.34	83.22	77.28	76.58
12	80.71	80.03	74.94	66.04	82.72	81.69	82.54	63.26	59.72	61.40	89.27	76.27	81.78	80.83	75.80
13	54.01	55.23	63.81	44.91	36.18	59.98	70.02	58.99	57.21	60.51	77.54	63.70	83.63	83.64	62.10
14	71.81	69.75	83.34	45.71	46.85	69.08	76.64	58.67	66.03	72.70	83.69	60.22	82.44	84.68	69.40
Mean	68.51	67.78	73.67	54.25	53.88	68.64	73.21	57.80	65.42	64.69	81.57	64.76	79.82	79.15	68.08

Figure B.88: Variability table (MSE and VAF on velocity) for validation road 4

						VRoad5 - MSE								
participant	test 1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
model 1	5.79	5.63	22.89	5.83	2.40	6.27	10.57	15.72	6.43	16.02	13.72	11.09	7.36	5.56	9.66
2	7.81	2.82	13.24	3.35	4.43	6.45	8.43	11.62	7.95	17.31	13.55	10.07	7.60	10.29	8.92
3	16.56	13.43	12.29	8.27	8.13	14.08	12.86	4.87	6.44	11.76	6.18	23.44	10.67	14.63	11.69
4	20.69	13.36	28.11	3.32	13.39	13.38	14.43	9.68	20.29	27.51	6.32	26.63	18.59	26.64	17.31
5	24.57	18.84	35.33	16.57	16.98	20.69	23.72	12.06	12.36	30.79	13.76	31.57	16.72	27.65	21.54
6	4.92	2.50	9.45	7.13	3.05	6.26	10.18	14.50	7.45	15.91	17.41	10.25	8.56	5.16	8.77
7	26.11	19.14	39.30	10.39	13.82	19.75	16.26	7.07	13.44	31.62	5.97	32.74	13.34	28.48	19.82
8	19.60	19.26	15.89	13.99	10.07	18.51	15.41	9.74	7.30	13.68	9.21	25.27	15.00	15.49	14.89
9	12.33	9.89	14.57	8.08	5.20	12.03	10.03	8.92	3.96	19.12	7.70	18.39	6.28	12.33	10.63
10	25.89	20.44	17.99	11.44	14.06	21.13	19.34	6.30	11.30	15.79	5.01	32.55	17.43	22.81	17.25
11	18.70	13.14	31.46	5.03	9.03	14.78	12.91	10.91	9.79	21.44	2.46	22.75	11.21	26.23	14.99
12	8.32	3.27	12.76	4.15	2.49	6.62	7.25	10.74	6.85	16.44	11.13	13.66	7.12	7.25	8.43
13	11.76	6.55	27.28	9.63	5.50	11.59	13.63	13.06	6.29	18.73	15.13	22.54	5.49	13.36	12.90
14	5.77	6.57	4.70	10.27	2.57	8.36	7.65	13.11	2.18	11.27	16.62	9.01	6.48	3.44	7.72
Mean	14.17	10.46	19.23	7.76	7.24	12.25	12.23	10.48	8.44	18.20	10.03	19.88	10.40	14.74	12.54

						VRoad5 - VAF								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
1	84.85	74.69	33.01	71.22	90.47	71.17	61.61	15.86	80.67	41.76	88.68	80.04	76.96	88.54	68.54
2	84.79	90.28	56.47	82.50	82.54	71.87	68.44	45.85	75.74	37.42	84.78	82.13	76.17	73.16	72.30
3	76.55	63.07	79.50	48.52	85.62	62.16	62.49	67.01	91.70	67.45	80.34	66.55	78.96	85.68	72.54
4	74.56	82.87	34.94	88.51	70.03	74.87	66.04	31.73	53.36	20.55	73.61	80.45	60.89	60.16	62.33
5	47.74	32.92	0.00	0.00	45.11	23.68	18.42	15.74	76.24	4.33	47.29	48.60	56.35	37.96	32.46
6	85.08	86.09	59.77	77.79	88.44	70.21	64.86	43.86	77.49	41.60	90.66	75.35	73.28	85.47	72.85
7	71.15	61.73	18.20	45.59	71.83	51.00	60.36	51.63	77.93	26.62	73.63	59.84	77.79	63.75	57.93
8	66.13	34.33	68.15	12.29	80.58	43.87	55.10	31.87	90.40	58.90	63.81	61.88	69.35	84.49	58.65
9	85.24	67.11	67.07	47.56	87.63	56.23	65.95	43.06	92.05	40.54	82.79	72.11	85.31	82.87	69.68
10	74.93	61.67	75.58	42.35	83.49	56.98	52.87	57.83	89.63	61.48	76.84	61.24	75.89	82.49	68.09
11	91.14	83.49	48.71	77.57	89.83	70.45	72.54	25.97	81.68	47.91	91.95	78.68	82.45	74.76	72.65
12	83.71	87.33	61.61	78.95	90.84	72.26	72.94	48.63	79.99	41.68	91.53	76.80	78.38	84.13	74.91
13	72.27	66.70	7.77	41.47	79.07	47.70	49.84	25.20	85.11	36.20	76.54	53.01	82.74	65.48	56.36
14	83.33	63.46	82.81	48.44	89.80	60.57	72.25	46.56	93.68	58.29	81.24	75.60	79.55	91.33	73.35
Mean	79.52	70.99	53.35	58.67	83.86	62.26	63.48	41.16	82.26	44.65	81.26	71.05	76.75	78.64	67.71

Figure B.89: Variability table (MSE and VAF on velocity) for validation road 5

						VRoad6 - MSE								
participant	test 1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
model 1	2.65	11.07	5.72	3.78	3.06	7.48	4.47	5.55	6.02	3.32	11.01	8.38	7.04	9.26	6.34
2	3.00	7.39	9.39	2.75	3.69	13.47	3.31	5.40	9.05	13.94	13.56	5.93	6.36	14.51	7.98
3	13.69	25.02	5.60	10.27	12.85	24.41	12.73	8.55	8.44	14.09	7.59	22.94	9.69	15.31	13.65
4	15.06	28.21	18.93	7.41	11.61	27.43	9.54	6.91	24.77	20.47	4.77	24.33	19.80	29.66	17.78
5	14.97	18.33	3.62	12.52	14.05	18.63	14.77	10.12	6.92	13.26	18.53	22.02	7.09	10.81	13.26
6	2.63	10.62	7.99	3.88	4.63	9.27	4.31	7.73	6.78	4.65	16.22	9.63	9.20	14.93	8.03
7	20.18	25.38	6.70	12.91	16.86	26.39	15.84	8.69	9.91	17.10	9.85	26.98	11.63	18.81	16.23
8	16.94	31.33	8.85	15.87	15.81	27.61	16.35	13.43	8.29	18.37	9.08	26.05	11.02	17.64	16.90
9	7.19	17.44	3.64	9.33	8.32	17.81	7.10	9.38	4.78	9.98	9.42	15.97	3.08	9.16	9.47
10	16.76	29.37	7.92	13.65	16.19	28.93	16.11	11.43	9.48	17.72	8.41	26.03	11.36	17.98	16.52
11	12.77	27.35	11.58	9.42	11.27	27.13	10.70	9.94	10.02	18.96	2.99	18.81	12.20	24.63	14.84
12	2.65	8.89	5.64	2.66	3.28	10.46	3.64	5.59	6.58	5.94	11.34	7.93	5.17	8.56	6.31
13	8.56	11.47	4.66	7.90	8.74	14.45	9.68	7.41	4.20	9.38	15.58	17.26	3.58	6.16	9.22
14	4.91	12.49	3.87	7.00	6.31	14.25	5.93	9.08	2.88	12.05	16.81	10.63	3.56	14.06	8.85
Mean	9.77	18.93	7.73	8.22	9.43	19.16	9.21	8.39	8.55	12.77	10.51	16.99	8.75	15.44	11.70

						VRoad6 - VAF								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean
1	90.88	78.11	61.32	70.35	78.29	83.85	75.06	48.12	74.40	83.03	67.71	77.10	70.57	69.92	73.48
2	84.61	84.38	36.95	78.86	76.38	80.88	81.55	45.49	61.46	42.65	80.30	80.19	72.72	50.44	68.35
3	65.14	60.86	88.17	40.66	41.71	63.16	53.59	28.62	79.49	64.93	45.92	45.95	80.20	82.20	60.04
4	71.79	68.69	0.00	83.61	67.00	68.18	75.46	54.51	17.53	58.78	57.73	66.70	36.00	22.97	53.50
5	20.09	34.73	78.92	1.75	4.30	33.54	17.43	0.00	72.77	34.03	3.28	15.35	71.44	73.11	32.91
6	85.79	63.81	47.83	70.63	67.82	69.08	76.58	31.05	71.18	71.70	56.06	66.84	60.55	42.54	62.96
7	23.48	45.36	81.61	17.56	9.55	41.93	20.96	27.29	75.58	41.91	22.30	24.04	68.81	65.16	40.40
8	57.33	45.65	75.57	5.06	24.62	56.92	44.32	0.00	83.35	49.72	26.68	44.32	82.87	86.01	48.74
9	72.90	58.90	80.36	30.39	52.21	67.23	61.88	6.48	86.49	62.50	61.55	54.53	91.49	82.84	62.13
10	57.83	54.81	85.11	24.24	28.09	57.13	44.98	11.42	78.91	55.47	33.73	41.29	83.01	85.91	52.99
11	90.85	79.20	67.11	68.70	77.89	85.42	84.70	40.10	79.02	77.46	79.17	81.52	79.64	62.86	75.26
12	90.45	82.08	61.98	79.28	78.21	82.24	79.48	46.23	72.19	74.77	76.08	77.31	78.35	69.88	74.89
13	52.00	60.33	73.28	38.41	36.48	48.58	46.03	25.09	82.51	50.47	45.68	33.92	84.75	79.93	54.10
14	75.71	63.91	75.39	45.11	54.39	71.92	70.08	8.72	87.69	51.75	52.14	61.42	85.57	63.54	61.95
Mean	70.67	65.08	64.21	50.22	53.28	67.43	62.67	28.70	73.06	60.40	54.23	58.09	74.96	66.48	60.68

Figure B.90: Variability table (MSE and VAF on velocity) for validation road 6

C. Code Generation

1). Process Simulation Data

With the matlab file 'ProcessRawData.m' the raw data from the simulator is processed to the input variables needed for further analysis. First one has to define which participant (1-15) on which road type (V= validation, T = training) and number needs to be processed.

The simulation data is stored as hdf5 file, which can be displayed with 'h5disp' and stored as variable in Matlab with 'h5read'. The road data is stored in a csv file and is opened using 'load'. The road data in the csv file consists of 8 columns shown below:

$$road = [x_{right} \quad y_{right} \quad x_{mid} \quad y_{mid} \quad x_{left} \quad y_{left} \quad \psi \quad R]$$

Due to a different definition of ψ that is used in further analysis (in radians) it is transformed in line 37. The curve radius is also transformed to road curvature (= R^{-1}).

For each sample point the distance and time values to the TP and ETP are generated using function 'findTP'. Also the angle to the TP and the change in ψ is generated using the same function. These values are stored in lists and after the 'end-of-road' data is deleted. The end-of-road data is defined as the last point where no TP values can be found until the last sample point.

2). TP & ETP determination

The distances, times and angles to the TP and ETP are found with function 'findTP.m'. First the actual location of the vehicle with respect to the road is established. After this it is checked whether the end of road is reached and if the car is located outside the lanes of the road. If one of these conditions applies, a fixed predetermined values is set as TETP, DETP, TTP, DTP and TPangle. If these conditions do not apply, the TP search starts.

First the left side of the road is checked, and TPsearch is set to 'false'. The '90deg-TP angle' is calculated using simple trigonometry for the first and second road lane point. When the difference in angle crosses zero, the TP angle must lay between these points. Note that when interpolating between these point to find the more exact TP angle, a value of 0.5 must be subtracted to both road lane point index numbers. This is due to the fact that the derivative between two points (e.g. point 1 and 2) is the derivative at 1.5 when linearly interpolating. When the TP index point left is found (or end of road is reached), the same is done for the right side of the road. When on both sides a TP point is found, priority is given to the one closest tot the vehicle. Finally, the TTP, DTP and TP angle are calculated using Eq. C.1.

$$DTP = \sqrt{(x_{car} - x_{TP})^2 + (y_{car} - y_{TP})^2}; \tag{C.1a}$$

$$TTP = \frac{DTP}{V_{car}}; \tag{C.1b}$$

$$TP_{angle} = wrapToPi(atan2((y_{TP} - y_{car}), (x_{TP} - x_{car})) - \psi); \tag{C.1c}$$

The distance and time to the ETP is found in a different way since the TP is already established. The line towards the TP is linearly extended using Eq. C.2 in which X and Y and sample x and y locations of the road lane.

$$Y = \frac{\Delta y}{\Delta x} \cdot (X - x_{car}) + y_{car} \tag{C.2a}$$

$$\Delta y = y_{TP} - y_{car}; \tag{C.2b}$$

$$\Delta x = x_{TP} - x_{car}; \tag{C.2c}$$

Using this line, the difference in y between the (left) side of the road at location x1 and the linearly extended line at location x1 is calculated. This is also done at the next location x2. These two differences are multiplied with each other. When this multiplication turns out to be negative, is must be that these two lines have crossed. Using 'interp1', linear interpolation is done to find the ETP index point. This is done for both the right and left side of the road. The ETP index closest to the vehicle is set at ETP location. Similar as in Eq. C.1, the TETP and DETP are calculated.

3). Neural Network

The neural network is trained in the file 'modelData.m'. First the desired processed files are opened and variables are put into one array. After this a neural network setup is made using 'feedforwardnet'. The training algorithm (LM), the number of epochs and the minimum allowable gradient are all set in the 'net.trainParam' structure. The network is trained by using the input (TETP and TP angle) and output ($\delta_a - \delta_b$) values. Finally, the weights and normalization values are retrieved from the network to be used in further analysis. The normalization values are used before putting in the inputs and and after generating the outputs into/out the neural net using Eq. C.3a and C.3b respectively.

$$x_{norm} = (x - x_{min}) \cdot \frac{y_{max} - y_{min}}{x_{max} - x_{min}} + y_{min} \tag{C.3a}$$

$$y = (y_{norm} - y_{min}) \cdot \frac{x_{max,2} - x_{min,2}}{y_{max} - y_{min}} + x_{min,2} \tag{C.3b}$$

$$x_{max,min} = \begin{bmatrix} TETP_{max,min} \\ \angle TP_{max,min} \end{bmatrix}$$

$$x_{max2,min2} = \left[(\delta_a - \delta_b)_{max,min} \right]$$

$$y_{min} = -1 \qquad y_{max} = 1$$

The neural network complete structure can be found in the Simulink model 'NNsetup.slx'. This model includes normalization and the network weights and activation functions. Note that these values are set for a specific training case, and do not change automatically when a new network is trained.

4). Initial Conditions for Optimization

In order to generate the initial conditions needed for the fmincon optimizer, a matlab file called 'GetInitialValues.m' is made. In this file one needs to define which model to use (produced in 'ModelData.m') and which participant/road to use the model on.

First the distance travelled ('dRoute') will be calculated by linearly attaching all vehicle location sample point, and calculating the distance between each point using simple trigonometry. Sample points at the start when the vehicle is standing still are removed using the 'unique' function since these are not of interest.

The second thing in this script is the running of the speed control algorithm using the defined model. In order to save much time, Δt is held variable while the location points of 'dRoute' are held fixed. When these location points are held fixed the DETP and TP angle values at these points remain the same as when processed in 'processRawData.m'. The TETP values can

simple be calculated using Eq. C.4 instead of the 'findTP' algorithm which needs much more processing power.

$$TETP(i) = \frac{DETP(i)}{V(i)} \tag{C.4}$$

When the visual cues at time t are determined the pedal deflection is calculated using one equation (line 76). This equation represents the entire neural network which uses inputs TETP and TP angle, and outputs the pedal deflection (positive = accelerator, negative = brake). The pedal deflections are used as input to the longitudinal vehicle dynamical model. This entire model is explained in Appendix C of the thesis article. Using this model the vehicle acceleration between sample point i and i+1 is calculated. The 'if' statement in the loop is there in case the vehicle crosses the side of the road and the vehicle decelerates to negative velocity (not possible in real life without 'reverse' gear). The reason this could happen is due to the fact that Δt is variable and calculated using Eq. **??**. When inside the square root becomes negative, a complex number would be given to Δt. The if statement re-sets the visual cues to starting values so that it can move forward until the vehicle is back on the track. After the speed control algorithm loop, the neural network parameters and the modeled test drive variables are saved to be used as initial conditions for the second optimization.

5). fmincon Optimization

The last step in model development is the fmincon (outer) optimization. This optimization minimizes an objective function that is defined at the end of the script. Because variables are not in the workspace when the script runs the objective function, global variables are defined first. These are the visual cues that remain fixed when the vehicle location sample point are kept fixed (DETP and TP angle). Also the neural network parameters and the measured car speed (to calculate MSE) are set as global variables. In a similar way as before, the initial conditions generated in 'GetInitialValues.m' are loaded into the workspace. In order to set up this optimization the 'optimoptions' funtion is used (line 48). Chosen is the 'fmincon - active-set' algorithm. It is also possible to change the constraint and step tolerance depending on the optimization problem. The to be determined variables are the neural network weights (defined as x). x0 are the weights after training the neural network. Furthermore are there no (non)linear constraints and no upper and lower limits (ub & lb) for the weights. The vehicle dynamics are integrated in the objective function instead as set as constraints. The objective function is minimized during optimization and is found in Eq. **??** for n sample points.

$$\underset{lb<x<ub}{\text{minimize}}\, f(x) = \frac{\sum_{n=1}^{n_{max}} (V_{sim} - V_{model})^2}{n} \tag{C.5}$$

The simulated velocity V_{sim} is calculated using the speed control algorithm which is integrated in the objective function and is a function of x, therefore, the objective is a function of x. Finally, the optimized neural network weights are saved as a new model.

6). Validation

Validation of models is done using 'ValidateData.m'. This script is basically the same as the speed control algorithm used to get initial values for the fmincon optimization. The only difference between the two is that this time Δt is set to a constant value (0.01 default). This results in different vehicle location samples compared to the measured values. Therefore, the 'findTP.m' algorithm is used again in order to compute the new visual inputs. This also results in a different array length when comparing the model and measured velocity. At the end the modeled velocity is interpolated to match the array size of the measured velocity so that the MSE and

VAF can be calculated.

In the folder with the code files, an example is given using the training road of participant 1 (this is participant 2 in the thesis article since numbering started at S00 during measurements). The saved results after each script is saved into the same folder.

Part III - Preliminary Thesis

Modeling drivers behaviour for speed adaptation in curves

Preliminary Thesis

by

S. Ceelen

As part of obtaining the degree of Master of Science
at the Delft University of Technology,

Student number: 4445961
Project duration: September, 2020 – June , 2021
Thesis committee: Dr. Ir. R. van Paassen, TU Delft, supervisor
Prof. Dr. Ir. M. Mulder TU Delft, supervisor

TU Delft

List of Acronyms

ACRONYM	FULL NAME
ABS	Anti-lock Braking System
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance System
CC	Cruise Control
DOF	Degree Of Freedom
DP	Driver's eye Point
DSA	Dynamic Speed Adaptation
ETP	Extended Tangent Point
FN	False Negative
FOE	Focus of Optical Expansion
FP	Far Point
FP	False Positive
FSD	Full Self Drive
GPS	Global Positioning System
HMI	Human-Machine Interface
HMI lab	Human-Machine Interaction Laboratory
IVIS	In-Vehicle Information System
LCP	Lane Crossing Point
LIDAR	Light Detecting and Ranging
LM	Levenberg-Marquardt
MSE	Mean Squared Error
NP	Near Point
OP	Occlusion Point
RADAR	Radio Detecting and Ranging
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
SLCP	Straight Lane Crossing Point
TETP	Time to Extended Tangent Point
TLC	Time to Lane Crossing
TP	Tangent Point
TTP	Time to Tangent Point

List of Symbols

ROMAN SYMBOL	FULL NAME	UNIT
a_{lat}	Lateral acceleration (local reference plane)	m/s^2
a_{long}	Longitudinal acceleration (local reference plane)	m/s^2
C	Curvature	m^{-1}
$C_{N,X}$	Number of combinations with group size N and outliers X	–
D	Distance to road edge	m
e	Neural network individual error between target and output	–
N_x	Number of neurons in x –	
P_o	Outlier participants fraction	–
r	Radius	m
R	Coefficient of correlation	–
R^2	Coefficient of determination	–
t	Time	s
V	Velocity	m/s
v_x	Neural network input of layer x	–
V_{lat}	Lateral velocity (local reference plane)	m/s
V_{long}	Longitudinal velocity (local reference plane)	m/s
w_{xy}	Neural network weight between layer x and y	–
X	Lateral distance (global reference plane)	m
$\dot{X}$	Time derivative of X	–
Y	Longitudinal distance (global reference plane)	m
y_x	Neural network output of layer x	–

GREEK SYMBOL	FULL NAME	UNIT
Γ	Lateral acceleration (= a_{lat})	m/s^2
δ	Texture density cue	m/deg
δ_a	Accelerator pedal deflection	%
δ_b	Brake pedal deflection	%
δ_s	Steering wheel deflection	rad
η	Learning parameter	–
ϕ	Activation function	–
Ψ	Yaw angle	rad

List of Figures

List of Tables

Contents

1 Introduction

1.1. Background Information

Many car manufacturers are investing their time and money into the development of autonomous vehicles, also known as the 'Self Driving Car'. This car is able to monitor its surroundings and to move safely through its environment with little to no input of a human. At the moment there is a slow transition happening from manual cars to fully autonomous cars. This transition started a long time ago. The goal of this transition is to enhance safety and driving convenience. The first form of automation was cruise control. This form of automation was implemented in the late 1950s on the Chrysler Imperial [1]. After cruise control, the Anti-Lock Braking System (ABS) was developed. The first ABS system as we know it was implemented in a Mercedes-Benz in 1978 [1]. After the year 2000 more advanced safety features were developed for cars. These features include 'Blind Spot Detection', 'Forward Collision Warning' and 'Lane Departure Warning' [1]. After 2010, more driving assistance features were implemented in cars. Vehicles could now assist with 'Rearview Video Systems', 'Automatic Emergency Braking' and give 'Rear Cross Traffic Alerts'. In the last 5 years of development towards more automation, cars became equipped with partially automated systems such as 'Lane Keeping Assist', 'Adaptive Cruise Control' (ACC), 'Traffic Jam Assist' and 'Self-Parking' [1]. Many of these implementations could still be improved in performance.

Apart from partially automated systems in cars, there also exist projects focusing on fully automated cars. An example of a project is Waymo. Waymo is an American autonomous driving technology development company. It is a subsidiary of Alphabet Inc, the parent company of Google. This company was founded in 2009, where it started as the self-driving car project from Google. By 2018, Waymo had tested its system in six states and 25 cities across the United States over a span of nine years. By October 2018, the autonomous cars had driven 10 million miles (16 million km) on public roads and simulations had completed 7 billion miles (11.2 billion km) 'virtual' miles [37]. By 2020, the experience on public roads exceeded 20 million miles (32 million km) [32].

The main priority of driving autonomously is of course safety of the passengers and its environment.however, next to safety, passenger comfort is also an important factor to take into account. People have much experience in driving and often drive through curves in a way that they themselves find comfortable. For automation there is a lot to learn from the way people drive through curved roads. This thesis will focus on creating models that capture human behaviour in order to drive through curves in a safe and comfortable way. There are multiple studies done regarding speed adaptation in curve driving. Most often, these studies create models relating road curvature to vehicle velocity. However, humans themselves are often poor judges of curvature. Yet, most of the time drivers are able to manoeuvre safely and comfortable through curves using nothing but their senses. Therefore, the main focus of this thesis will be the visual field as input for the speed adaptation model.

1.2. Research Objective and Questions

It is expected that the visual field must provide the main inputs for the speed choice behaviour of humans. Research with gaze detection equipment [11] and theoretical research indicate several features from the visual field that might play a role. In order to investigate which features contribute to human drivers' speed choice, the following main research objective was formulated:

"Measuring and modeling drivers behaviour for speed adaptation in curves using neural networks with visual field inputs

The reason that a neural network will be used is due to the number of possibilities of in- and outputs. There are many different cues from the visual field that can be used for speed determination. In order to easily switch between combinations of these cues, a neural network is used. It is not always possible however to produce a perfect model with the given in- and outputs. There is a large possibility that human driving behaviour is not only related to what they perceive through their eyes but also other inputs such as what is felt with the vestibular system. Maybe those inputs could also be related back to flows in the visual field. There is also the inconsistency and randomness of human behaviour within a person and between persons making it more difficult if not impossible to get one general model. Because of all these possibilities, the research objective will be split up by means of smaller research questions. The questions that will be answered in order to move towards a more detailed conclusion are as follows:

- Which visual points represent road geometry best?
- Which visual point flows represent accelerations best?
- Is there a correlation between visual cues and a comfortable speed profile in curve driving? If yes, how much and which cues?
- Is there a correlation between visual cues and the moment / magnitude of braking/accelerating? If yes, how much and which cues?
- How much influence has an upcoming curve on the driving behaviour in a current curve?
- How much does human behaviour vary between participants?
- How much does human behaviour vary for one participant?

1.3. Outline

This report is divided into two parts. The first part is a literature study. Many different aspects will be included in this study and it is important to know the background of these aspects and up-to-date conclusions from previous research on this matter. In chapter 2, a detailed explanation of the Advanced Driver Assistance System will be given. This includes the most recent developments towards autonomy. Many Advanced Driver Assistance Systems require (or will in the future require) speed adaptation. This research may provide insight that could be used in the development of such systems. In chapter 3 previous research will be discussed about the influences on desired driving speed. Based on the information gathered in this chapter, an (preliminary) experimental design can be developed. The plan of this research is to use a simulator to collect data. The types of simulators that exist and how they could potentially influence the data will be discussed in chapter 4. Finally, a small part is dedicated to the basic principles of neural networks in chapter 5 since this type of modeling will be used for this thesis.

In the second part of this report a preliminary research will be performed. In chapter 6 some already gathered data will be analysed. With the results from this analysis, it becomes clear which methods will or will not work and what data is missing on order to answer the research questions properly. With this information an experimental design can be thought of. This design is worked out in detail in chapter 7.

I

Literature

In part I of this report a literature study is done. This study includes research that has been done regarding speed adaptation in curve driving as well as closely related topics (chapter 2 and 3). Next to this there is also information given on the resources that will be needed in order to collect and process data. In chapter 5 the description of a neural network will be given that is used to build a speed adaptation model. The data for this network will be gathered with a driving simulator. Detailed information on this simulator is given in chapter 4.

2

Advanced Driver Assistance System

Advanced Driver Assistance Systems (ADAS) are electronic systems that help the driver with driving and parking functions. ADAS can be split into two categories, longitudinal and lateral control [29]. A lateral control example is the lane keeping system, which makes sure that the vehicle stays within the intended lane. An example for longitudinal control is Adaptive Cruise Control (ACC).

2.1. Adaptive Cruise Control

ACC was one of the first Advanced Driver Assistance System (ADAS) to be introduced to the market [14]. With Cruise Control (CC) the vehicle will continue to drive with constant velocity. ACC can also detect another vehicle in front and adapts its velocity in a safe way. Without any other vehicle in front, the ACC will work similar to the CC. The ACC usually consists of four functions [6]:

- Constant speed: without any other vehicle (close enough) in front of own car to detect a constant speed will be held.
- Deceleration control: When another vehicle is detected in front of the own car, the velocity will decrease in order to keep a certain (safe) distance from that vehicle.
- Acceleration control: When a detected vehicle in front accelerates or disappears, the car itself will accelerate to the desired initially set velocity.
- Following control: The car itself will accelerate and decelerate in order to keep a desired distance from the vehicle detected in front [30].

2.2. Curve Speed Adaptation

A subsystem of the ACC is the Curve Speed Adaptation (CSA). It is an important step towards an autonomous driving future. Without CSA a vehicle will try to remain at a constant speed. For some corners this can be very uncomfortable or even dangerous for the passengers. The goal of the CSA is to decelerate to a desired speed before a curve and accelerate to the initially set speed after the curve. In order to be able to do so, the CSA should be able to identify the upcoming curves [30].

Instead of only driving safely through curves, also passenger comfort can be taken into account. How comfortable people are, varies from person to person. Taking this in mind, a personalised ACC control algorithm for curved roads was developed by Zhang *et al.* [9]. For this research, experimental data regarding driver behaviour were collected from 40 different participants. To adapt the model to each driver's individual curve speed behavior, the coefficients of the model are identified in real time from the data sequences collected during drivers' manual operation stage by a self-learning algorithm based on a Recursive Least-Square (RLS) method with a forgetting factor [9]. The parameters from the model can be identified in the manual operation phase and the resulting models are applied in the ACC automatic control phase. The most suitable relation of which the parameters were to be identified can be found in Equation 2.1 [9].

$$v_{db} = v_0 - k_1 \exp(k_2 v_0 c) \tag{2.1}$$

In which v_{db} is the desired velocity, v_0 the initial velocity when entering a curve and c the road curvature. k_1 and k_2 are the to be identified parameters which are unique for each participant. A visualisation of this relation and the variation between two participants can be found in Figure 2.1.

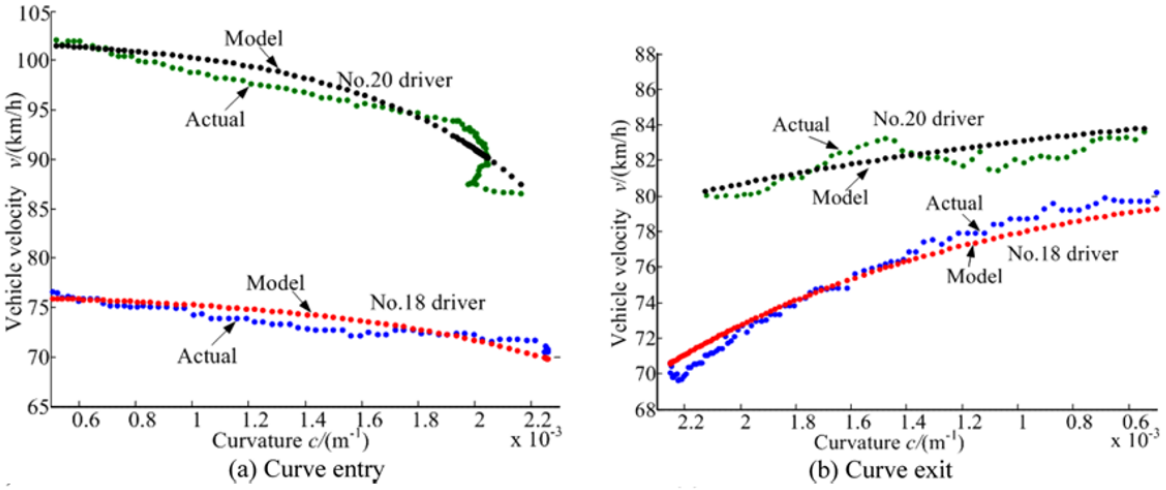


Figure 2.1: Velocity profile comparison when entering (left) and exiting (right) a curve

Instead of a constant velocity for a curve with a certain curvature, this relation includes human behaviour. As can be seen in the figures, almost all drivers slowed down into the curve and then sped up when exiting the curve. In his work he concludes that the driver curve speed model is adaptive to each driver's individual curve speed behaviour and can reflect drivers' specific characteristics and therefore can significantly improve the practicality and comfortablity of ACC systems [9].

Even though curve speed estimation has been considered in several works, only a few of them have been actually implemented and tested either in simulations or real environments [35]. Glaser et al. [28], and Park et al. [19] have been working on computing the ideal speed to drive through curves. This speed is based on the curvature of the road. Another research from Lee et al. [17] also computed curve speeds and speed limits. Their implementation was to an ACC system, in which the driver is still able to take control of the car. The work of Serna et al. [35], is targeted to become an autonomous solution. In his work he takes into account road curvature and speed limits with which automatically the vehicle adjusts its speed through their Dynamic Speed Adaptation (DSA) method. Part of his work consists of 'curve analysis extraction', where an algorithm analyses GPS information, identifies curvatures in the upcoming road and assesses a desired velocity to the curve. Another part focuses on 'speed limits database creation'. This database contains information about speed limits for all traveled roads. With both parts operating, the DSA ensures smooth transitions in velocity towards the ideal speed. [35].

2.3. Instrumentation

In order to assist the driver with the Advanced Driver Assistance Systems the vehicle will need specific sensors and instrumentation to collect the data needed. The main sensors available for cars at the moment are discussed in this section.

2.3.1. Digital Maps

Maps nowadays can provide support to the ADAS in predicting road geometry ahead and certain road attributes in front of the vehicle [13]. This is usually done using a Global Positioning System (GPS) together with an internal map [30]. The interaction between the GPS and internal map is called map matching. This is developed into the 'electronic horizon' which can then predict attributes such as road geometry, upcoming signs, traffic ahead and number of lanes [30].

2.3.2. Camera

Cameras on board of a vehicle are used for gathering data along the road. They are an important factor towards a fully autonomous driving future. The front facing camera helps the ACC by identifying objects, measuring distance towards it and also relative their velocity. The signal from the camera is transmitted to an image processor. The processor obtains individual images which are sent to the Electric Control Unit (ECU). The ECU assesses the information from each image and sends it further to the location where needed. For example, an object in front of the vehicle is needed by the ACC and maximum allowable speed will be shown on the Human Machine Interface (HMI) / dashboard.

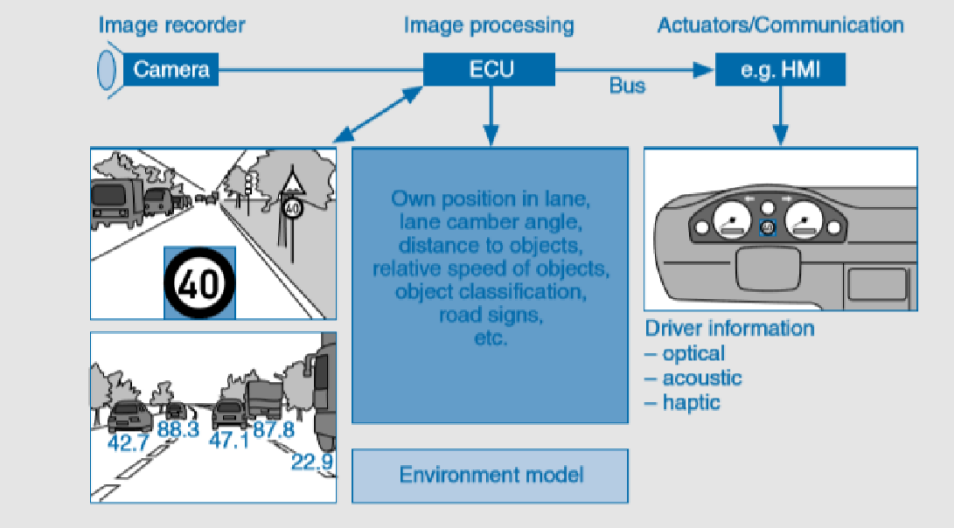


Figure 2.2: Principle of frontal camera sensor [30]

The object detection function works better when working together with a RADAR sensor. When both are used, the ACC range could be extended.

2.3.3. RADAR

The Radio Detecting And Ranging (RADAR) sensor is usually built in low in front of the vehicle. This sensor sends out a pulse which bounces back via attributes in front. Using the time it takes for the wave to return, the distance and relative speed to this object can be calculated. The RADAR used for the ACC usually has a maximum range of 120-250m [2].

2.3.4. LIDAR

Light Detection and Ranging (LIDAR) is used to make 3D digital representations of a target or surroundings. The working principle is similar to that of a RADAR. A target is illuminated with laser light, and the reflection is measured with a sensor. LIDAR is extremely accurate but also relatively expensive. Most companies, such as Waymo, Uber and Toyota, working on autonomous driving use LIDAR in order to accurately map all surroundings. The only company which does not use LIDAR is Tesla. The Tesla cars fully rely on vision.

2.4. Latest Developments

The latest development of the ADAS on the market came out in October 2020. In this month Tesla released the beta version of the 'Full Self Driving' (FSD) software. Even though the name says 'Full Self Driving', Tesla still requires active driver supervision at all times. In this package is included [38]:

- Traffic-Aware Cruise Control: Matches the speed of your car to that of the surrounding traffic
- Autosteer: Assists in steering within a clearly marked lane, and uses traffic-aware cruise control
- Navigate on Autopilot (Beta): Actively guides your car from a highway's on-ramp to off-ramp, including suggesting lane changes, navigating interchanges, automatically engaging the turn signal and taking the correct exit

- Auto Lane Change: Assists in moving to an adjacent lane on the highway when Autosteer is engaged
- Autopark: Helps automatically parallel or perpendicular park your car, with a single touch
- Summon: Moves your car in and out of a tight space using the mobile app or key
- Smart Summon: Your car will navigate more complex environments and parking spaces, maneuvering around objects as necessary to come find you in a parking lot.
- Traffic and Stop Sign Control (Beta): Identifies stop signs and traffic lights and automatically slows your car to a stop on approach, with your active supervision

3

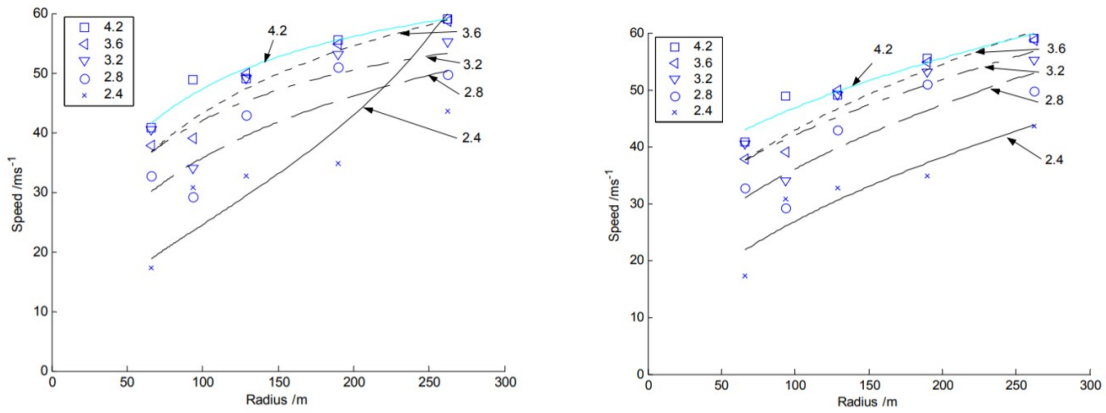
Velocity Determination

The velocity towards, in and out of a curve can be determined by many factors. Much research has been conducted finding a safe and comfortable speed using road geometry only. Apart from road geometry, one could also use more inputs such as accelerations felt by the driver. Humans themselves have shown to be poor judges of geometry values of the road ahead [25]. Instead, we use our visual field we have in front where we focus on specific points [11].

3.1. Road Geometry

Most research related to speed adaptation on curved roads is directly related to the geometry of the road. The personalised ACC feature developed by Zhang et al. [9] purely used road curvature as input. The curvature is defined as the inverse of the curve radius. Many other studies also show a direct correlation between curve radius and speed [21]-[24]. Those relations however focused mostly on average curvature and mean velocity throughout the turn.

Odhams et al. [7] points out the importance of taking both curve radius and also road width into account. In the article he concludes that "lane width and curve radius were both found to be significant determinants of speed choice, but several existing models of speed choice account for only one or other of these factors" [7]. Odhams compares various existing models in his work. But all models only relate one velocity to one curvature/width combination, as can be seen in Figure 3.1a and 3.1b where two of these models are shown [7].



(a) Raymond's lateral acceleration model

(b) TLC model

Figure 3.1: Two models compared from [7]

3.2. Accelerations

The second factor to take into account are the accelerations felt by the driver. The most relevant acceleration felt in a curve is the lateral acceleration. Lateral acceleration in a curve is defined by Equation 3.1.

$$a_y = \frac{V^2}{r} \tag{3.1}$$

where V is the vehicle velocity and r the radius of the curvature. A study from Reymond et al. [15] shows that drivers adjust their speed in curves such that the maximum lateral acceleration decreases with increasing velocity. In other words, turns with a small radius (and thus lower velocity) are driven with higher lateral acceleration than large radius curves where a higher velocity is possible (see Figure 3.2 [15]).

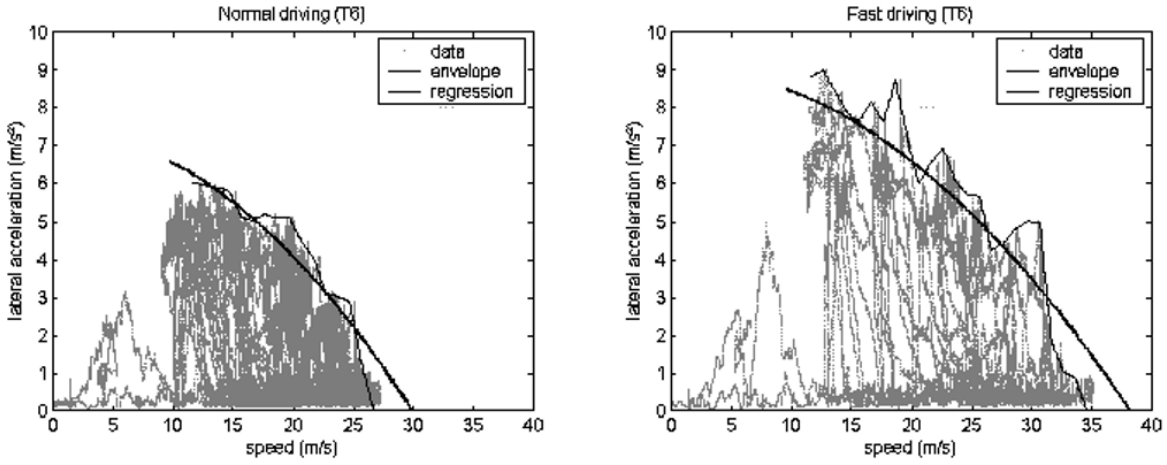


Figure 3.2: Maximum lateral acceleration as a function of velocity for normal driving (left) and fast driving (right)

It was also found that drivers use the lateral acceleration they felt while driving. The drivers in the experiment had driven the test track in a simulator once with the motion cues on and once with the cues off. It was found that when the motion cues were off, the upper limits of lateral acceleration decreased less steeply. This is interpreted as an underestimation of curvilinear speed due to the lack of inertial stimuli [15]. More on the difference in simulators will be discussed in chapter 4. Also in the work from Odhams et al. discussed before it was concluded that, the achieved lateral acceleration reduced at high speed. This was despite the use of a fixed-base simulator and the vehicle having no tyre saturation [7].

3.3. Visual Field

When humans drive, usually no information about road geometry is given to them. Also, humans have shown to be rather poor judges of road curvature [8]. Even though much research is carried out relating road geometry to velocity, humans probably make use of different cues. In the previous section it was already established that humans make use of the lateral acceleration felt while driving. Another cue that is used a lot while driving is in the visual field. A research from E. Lehtonen et al. [11] investigates the eye movement when approaching a curve on a rural road. In this study, a distinction is made between various points shown in Figure 3.3 [11].

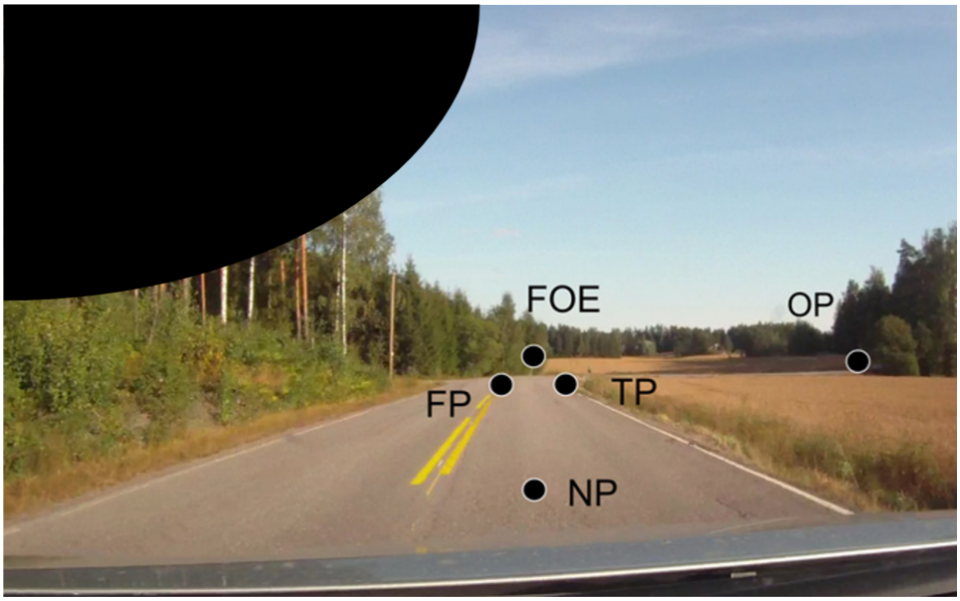


Figure 3.3: Main points in visual field

- NP/FP: the Near Point and Far Point. These are the points directly in front of the vehicle and further up the road [11]. Land et al. [20] and Salvucci et al. [10] have shown that these points might be used for steering the vehicle.

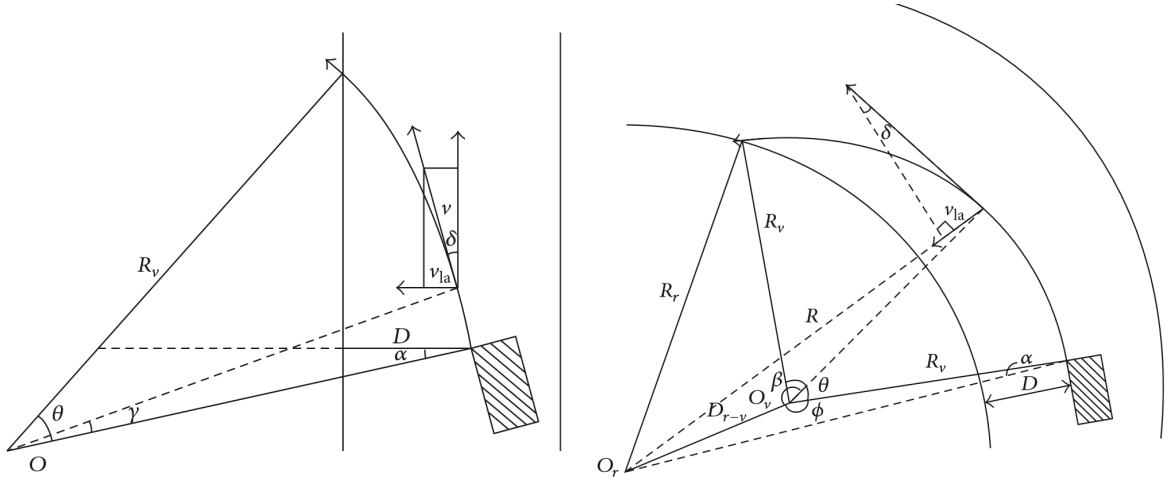
- FOE: the Focus of Optical Expansion point is a point laying directly ahead from which all apparent flow patterns in the visual scene symmetrically radiate from.

- TP: the Tangent Point is where is gaze direction is tangential to the road edge. This is only possible in curves.

- OP: the Occlusion Point is the nearest point where the view of the road is blocked by an obstacle. The oncoming traffic will emerge from this point.

It was found that drivers switch their focus between the on-road Far Point and Tangent Point and off-road Occlusion Point. The Occlusion Point would be used to monitor upcoming traffic, the Far Point for steering purposes and the Tangent Point while driving towards/in curves.

Most models directly relate road geometry to speed, while research into speed adaptation in car following and lane keeping has shown that speed control is likely governed by time margins to salient visual points [25]. Several studies have attempted to characterise speed choice as a function of the driver's anticipated path following error. In simulator tests by Van Winsum and Gothelp [26] the 'time to lane crossing' (TLC) was found to be kept above a constant minimum level by drivers. A way to determine the TLC was developed by L.H. Xu et al. [18]. In their work they derived equations in order to compute the TLC for all different variations of crossing the side of the road. This can either be while driving on a straight part (see Figure 3.4a) or driving on a curved part (see Figure 3.4b). The derivation of the TLC in their work is found below.



(a) Straight road

(b) Curved road

Figure 3.4: Schematic with TLC derivation definitions [18]

First, for the straight part of the road, the TLC is calculated using Equation 3.2.

$$\int_0^{TLC} V_{\text{lat}}\, dt = D \tag{3.2}$$

In which TLC is the parameter to be calculated, V_{lat} is the (time varying) lateral velocity with respect to the road (v_{la} in figure) and D the distance toward the side of the road. The lateral velocity is calculated using Equation 3.3

$$V_{\text{lat}} = V \sin\delta \tag{3.3a}$$

$$\delta = \gamma + \alpha \tag{3.3b}$$

Where γ is the degree as the vehicle is at time t (between 0 and θ) and α the yaw angle with respect to the road center line. γ can be calculated using Equation 3.4.

$$\gamma = \omega t = \frac{v}{R_v} t \tag{3.4}$$

In which ω is the vehicle angular speed and R_v is the vehicle radius. When everything is put together and the integral from Equation 3.2 is written out, the final equation to calculate the TLC becomes:

$$TLC = \frac{R_v\,(\arccos(\cos\alpha - D/R_v) - \alpha)}{v} \tag{3.5}$$

When the vehicle is driving through a curve, slightly different equations apply. Again, Equation 3.2 is used as starting point. However, this time, δ is calculated differently with Equation 3.6.

$$\delta = \arccos\left(\frac{\left(R^2 + R_v^2 - D_{r-v}^2\right)}{(2RR_v)}\right) \tag{3.6}$$

Where D_{r-v} is the distance between the center point of the road curve and the center point of the car curve (computed with Equation 3.7) and R the distance from the vehicle to the center point of the road curve (computed with Equation 3.8). R is the time variant variable. Note that in this derivation, R is the radius instead of r which is used in the rest of the report.

$$\cos\alpha = \frac{\left(R_v^2 + (R_r + D)^2 - D_{r-v}^2\right)}{(2R_v\,(R_r + D))} \tag{3.7}$$

$$\cos\beta = \frac{\left(D_{r-v}^2 + R_v^2 - R^2\right)}{(2D_{r-v}R_v)} \tag{3.8}$$

Again, α is the constant value between vehicle yaw and center line of the road. $\beta = 2\pi - \phi - (v/R_v)t$ (see Figure 3.4b). ϕ is computed using Equation 3.9.

$$\phi = \arccos\left(\frac{\left(D_{r-v}^2 + R_v^2 - (R_r + D)^2\right)}{(2 \cdot D_{r-v} \cdot R_v)}\right) \tag{3.9}$$

However, Gruppelaar [25] provides evidence that the TETP is a better candidate than the TLC for curve speed modeling applications. The TLC shows great variability, even on straight segments of road, as it depends also on the steering inputs of the driver. In the work from Gruppelaar et al. [25] the Time to Extended Tangent Point (TETP) was used to characterise driving behaviour. This point is defined as the time it takes to reach the Extended Tangent Point (ETP) when the velocity is held constant. The ETP is the extension of the Tangent Point (TP) to the other side of the road. In this study it was found that there exist a relation between certain time margins of the TETP (and its derivative) and the different phases in a turn. This relation can be seen in the flowchart of Figure 3.5 [25]. Unlike much other research, velocity is not the model output. In this study brake and accelerator pedal deflections were fit to a model. The comparison between the model and a participant can be found in Figure 3.6. Even though these two seem to be very different from each other, the model does capture the general trend of curve driving. In Figure 3.7 a velocity profile is plotted by iteratively deriving the acceleration and velocity from the model. This plot shows that the model captures the general trend of acceleration and deceleration in curve driving. Since the pedal deflection are the model output, this model is also easy to implement as input to a simulated or actual vehicle (with same dynamics) as those are also the actual vehicle's inputs to control acceleration. Another advantage of this method is that no unrealistic velocities and accelerations will be modeled as the pedal deflections are always between 0 and 100%.

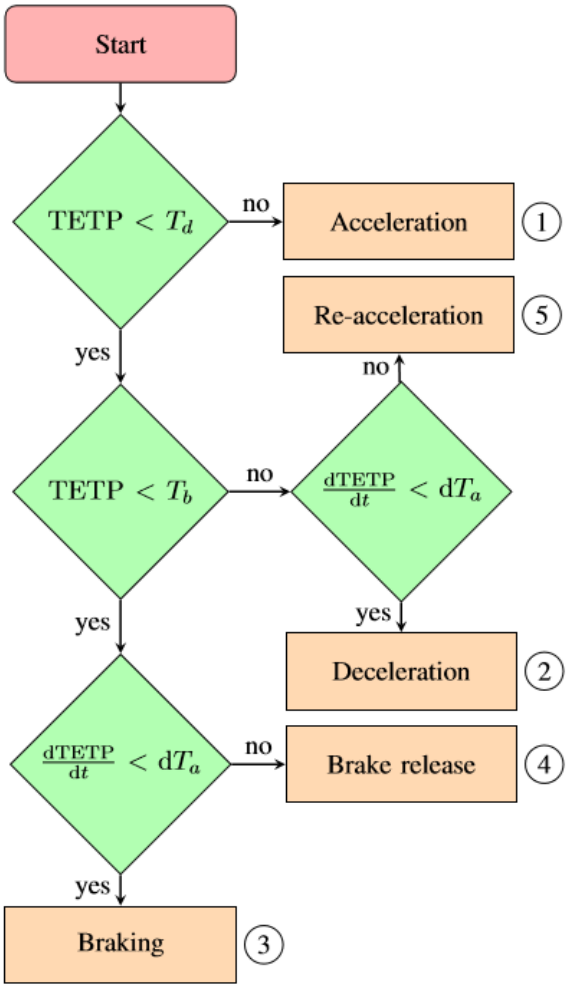


Figure 3.5: Model phase decision flowchart

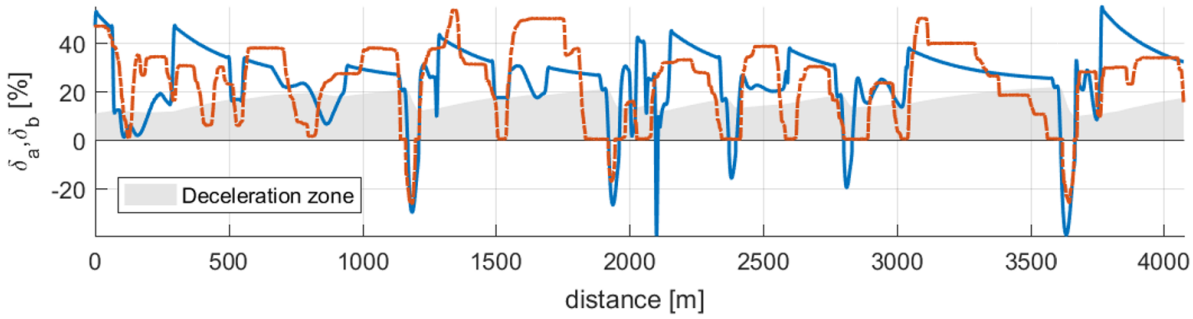


Figure 3.6: Pedal deflection comparison between measured data and model [25]

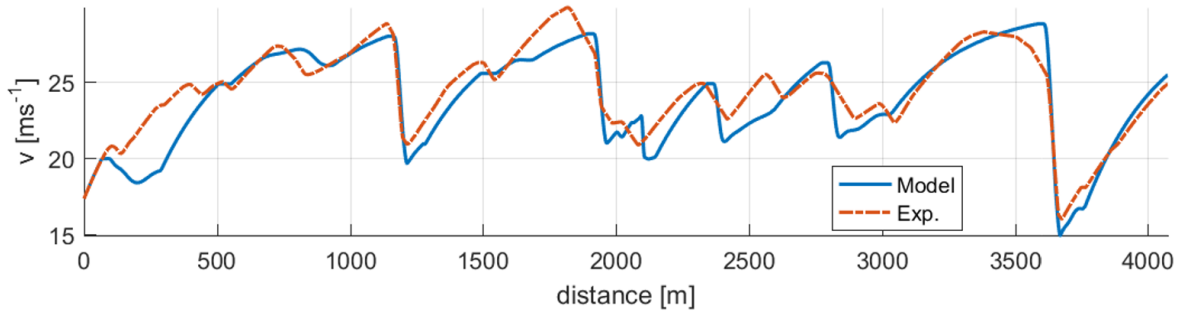


Figure 3.7: Integrated velocity comparison between measured data ans model [25]

In the study from Gruppelaar et al. [25] the ETP was computed mathematically with a known experimental road. It is also possible to find these visual points using a front camera. This is done in the work from Gallen and Glaser [22]. In their work they show how to find the TP and compute the TP Angle in real-time imaging with a monocular in-vehicle camera. Different methods were used to find the TP, and all methods made use of the Labayrade and Douret lane detection system [23]. The TP detection is shown in Figure 3.8 [22].

In another research from E.R. Boer and M. Mulder [12] it was investigated what cues drivers use in order to determine whether to brake before negotiating a curve. Curvature and curve length obviously play an important role. However, how drivers perceive these quantities remains largely unresolved [12]. In their work they demonstrate that the environment around a road provide more salient curvature cues. It was theoretically derived that a visual angle between the Lane Crossing Point and Extended Target Point could provide a more accurate estimate of curvature for long curves and that it may naturally bias drivers towards higher curve negotiation speeds when the curve is short. As seen in Figure 3.9 this cue is sensitive to changes in curve radius and its sensitivity to change in deflection angle is non-existent for large deflection angles but strong for short curves. In essence, the cue for short sharp curves is the same as for longer less sharp curves. This texture density cue is defined as the distance from the LCP to the ETP in world coordinates divided by the visual angle between to two points and calculated using Equation 3.10.

Figure 3.8: TP detection in a curve [22]

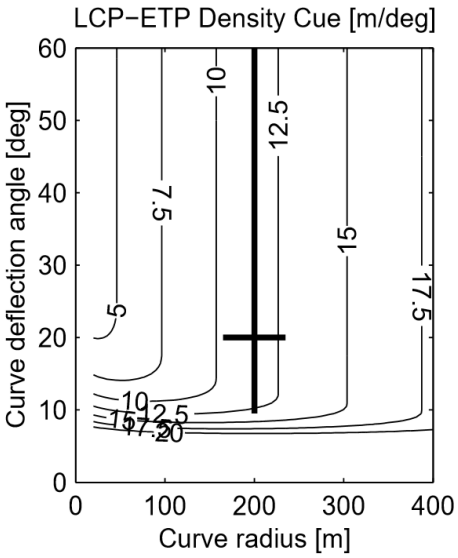


Figure 3.9: Texture density cue (δ) in the visual angle between the lane crossing point (LCP) and the extended tangent point (ETP). [12]

$$\delta = \left\| \begin{pmatrix} x_{ETP} - x_{LCP} \\ y_{ETP} - y_{LCP} \end{pmatrix} \right\| / \theta_{LCP,ETP} \tag{3.10}$$

In which

$$\theta_{LCP,ETP} = \arccos\left(\begin{pmatrix} x_{LCP} - x_{DP} \\ y_{LCP} - y_{DP} \\ 0 - h \end{pmatrix} \bullet \begin{pmatrix} x_{ETP} - x_{DP} \\ y_{ETP} - y_{DP} \\ 0 - h \end{pmatrix} / \left\| \begin{pmatrix} x_{LCP} - x_{DP} \\ y_{LCP} - y_{DP} \\ 0 - h \end{pmatrix} \right\| \left\| \begin{pmatrix} x_{ETP} - x_{DP} \\ y_{ETP} - y_{DP} \\ 0 - h \end{pmatrix} \right\| \right). \tag{3.11}$$

Where 'DP' means driver's eye point (see Figure 3.10).

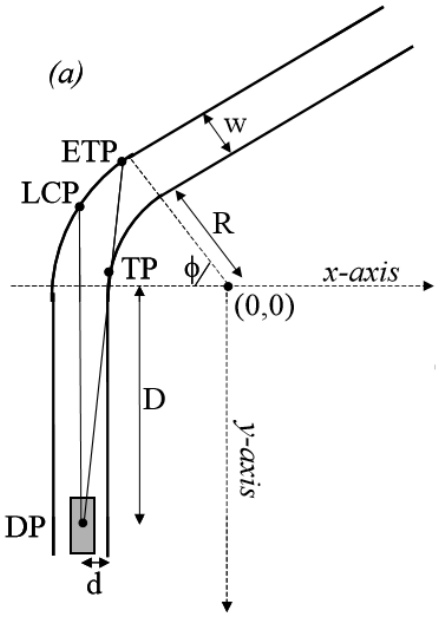


Figure 3.10: Schematic with definitions used in the work from Boer et. al. [12]

4

Driving Simulators

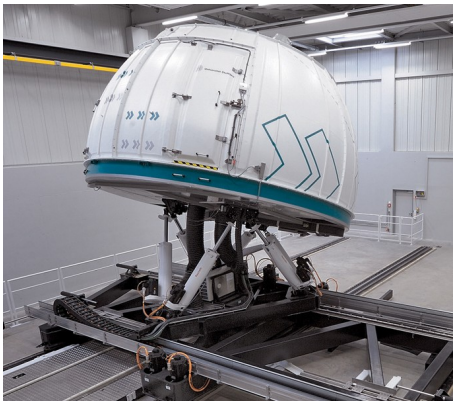
A simulation is an approximate imitation of reality. There exist loads of different simulators with different purposes. In the literature used in this report, often a driving simulator was used in order to collect data. Using a simulator can often be cheaper, and easier to reproduce many times without being dependent on outside variables (e.g. traffic and weather conditions). Driving simulators can be used for multiple purposes. Depending on the purpose, also the simulator type may vary. Some purposes require a high fidelity simulator while others will suffice with a lower fidelity. In this project, a simulator will be used to collect data. When analysing the data collected, it is important to understand the influence of the characteristics and fidelity of the used simulator.

4.1. Fidelity

The fidelity of a simulator determines the level of realism for a simulation. Fidelity is determined by the quantity and quality of visual, motion and sound cues. The visual cues can go from less to more realistic. Also, the screen can be either fixed, moving along with the car or head-mounted. The amount and type of screen will influence the viewing angle. A larger viewing angle is results in higher fidelity. To achieve a large viewing angle, either multiple screen can be put around the driver, or a larger convex screen can be used.

The next type of cues are motion cues. A driving simulator can have no Degree Of Freedom (DOF), also known as a fixed-base simulator. In this case, no motion cues are given to the driver and the simulator has a low-fidelity level. A high level fidelity simulator has at least 6 DOF [36]. These 6 DOFs are translation and rotation in all three axis (x, y and z). The largest driving simulators consist often of a dome inside which an actual size car is placed. This dome is placed on top of a turntable which is supported by a hexapod structure (see Figure 4.1a). This hexapod is fixed to a xy-table giving the simulator more space to move in the xy-plane (see Figure 4.1b).

(a) Hexapod configuration



(b) (Hexapod) Simulator on a xy-table

Figure 4.1: Degrees of Freedom in a simulator

Thirdly, also sound cues are important for the fidelity of a simulator. Sound cues mainly consist of the sound the engine of a car makes during driving. Next to these three major types of cues, also other simulator specific type of cues of have to be taken into account. For example the force feedback on the steering wheel and pedals and the vibrations felt by the driver (type of motion cue) are important factors to take into account for a high fidelity simulator.

4.2. Purpose

The three main purposes for driving simulators are training, entertainment and research. The fidelity needed for a simulator depends highly on the purpose. Most simulators used for entertaining purposes have a fixed screen and 0 or 1 DOF. The largest market share for entertaining simulators is owned by the video game industry [36]. Over the years the level of realism has increased due to increase of processing power. A typical entertaining driving simulator can be seen in Figure 4.2.

Figure 4.2: Gaming simulator

Another major purpose for simulators is training. Research has shown that driving simulators are proven to be excellent practical and effective educational tools to impart safe driving training techniques for all drivers [36]. The advantage of using a simulator is that it is possible to place the driver in a high risk situation without

actually putting the driver at risk. Also, environmental factors can be controlled ans the student's behaviour can be monitored very detailed. The fidelity of a training simulator is often required to be high. An example of a training simulator can be found in Figure 4.3.

Figure 4.3: Driving lesson in a training simulator

The last main purpose of a driving simulator is for research. This kind of simulator is also used in most articles discussed in this report. There is a wide range of research that has been done using driving simulators. Examples are the investigation of driving behaviour, the human-machine interface or to research the effects of drugs, tiredness, environmental issues or road design. Research simulators allow experiments to happen which would be illegal or unethical if done in a real environment. Vehicle manufacturers mainly use these type of simulators to test the design of the interior and the Advanced Driver Assistance System (ADAS) of a developing vehicle. Also, with the increase in In-Vehicle Information Systems (IVIS) such as navigation systems and cell phone use, research can be done investigating the effects of these devices on driving behaviour. Research simulators can have various levels of fidelity, depending on the specific research.

4.3. Simulator Driving Behaviour

It is clear that many different simulator types and purposes exist. For research sometimes a high fidelity is needed while for other research a low fidelity fixed based simulator will be enough. in a research from G. Reymond et al. [15] the role of lateral acceleration while driving is investigated. In their work they predict that there exist a relation between velocity and maximum lateral acceleration. This relation is defined by two inequalities. When driving through a curve at speed V, a change in curvature, ΔC, would result in a change in lateral acceleration $\Delta\Gamma$ (since $\Gamma = C \cdot V^2$). This change in lateral acceleration should not exceed $\Delta\Gamma_{max} = \Gamma_{max}$ - Γ to avoid reaching maximum acceleration Γ_{max}. This results in the first inequality shown in Equation 4.1a [15]. The second inequality is governed by other physical constraints. The maximum steering wheel angle ($\delta + max$) determines the maximum curvature a car can drive ($C_{max} = \delta_{\max}/L$). Therefore, the lateral acceleration will always be smaller than the maximum curvature times the velocity squared (see Equation 4.1b) [15].

$$\Gamma < \Gamma_{\max} - \Delta C \cdot V^2 \tag{4.1a}$$

$$\Gamma < (\delta_{\max}/L) \cdot V^2 \tag{4.1b}$$

A plot of the two inequalities combined can be found in Figure 4.4. The two inequalities together form an envelope in which a driver prefers to stay inside.

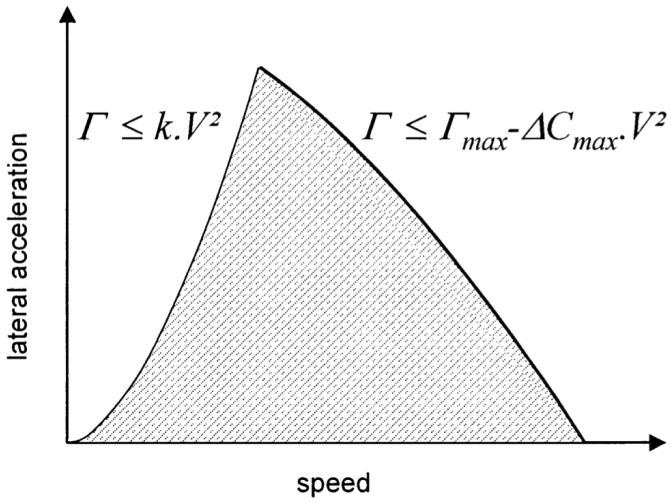


Figure 4.4: Lateral acceleration envelope [15]

They argue that a driver will adjust its speed before curves in such a way that the maximum lateral acceleration envelope will never be exceeded. Since the lateral acceleration is felt by the drivers and used as feedback, experiments are performed investigating the influence of feeling the lateral acceleration to the envelope. In order to do this, three tests are performed. First, a test in a real car, on a real road. This experiment took place on the Renault test track, located in Aubevoye, France (see Figure 4.5).

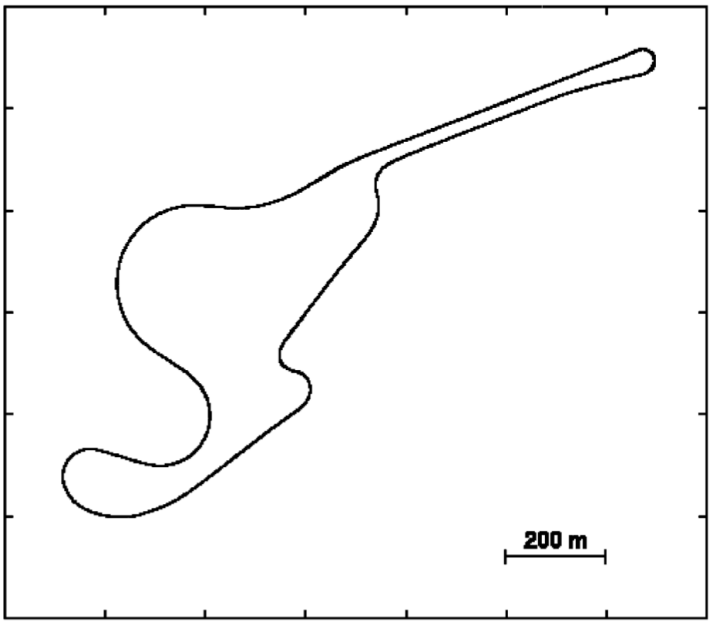


Figure 4.5: Renault test track

The drivers were asked to drive the test track a few times while driving normally, and a few laps while driving as fast as possible. The data showing the relation between velocity and lateral acceleration for one participant can be found in Figure 3.2.

clearly, when trying to drive as fast as possible, the maximum lateral acceleration increases compared to driving normally (and comfortable). Also, the envelope resulting from the first inequality is visible in the plot. After this experiment, another experiment was done. Test drivers had to drive a test track in a simulator as they would drive usually (same as before when asked to drive 'normal'). A few laps the motion base would be active, and a few laps the motion base would not be active. The differences between the dynamic and static

tests for one participant can be found in Figure 4.6 [15].

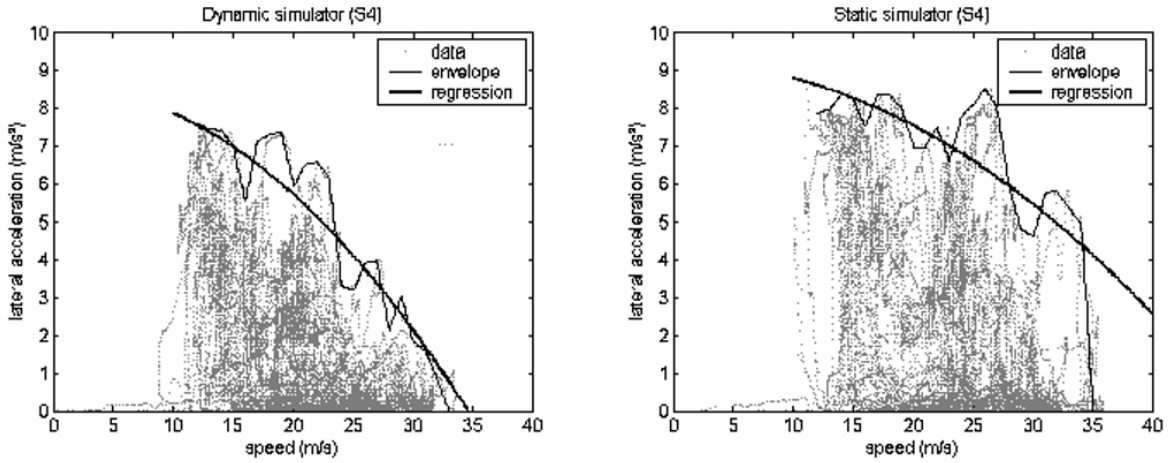


Figure 4.6: Dynamic (left) and static (right) simulation runs [15]

The major difference between these two plots is the increase in lateral acceleration when the simulator motion cues are inactive. The increase in parameter $\Gamma_{\max}$ was found to be small. The main change between static and dynamic experiments was caused by parameter ΔC. The change in these parameters could be interpreted as a change in driving behaviour when lateral acceleration is not felt. However, in experiment 1 (the real car), the drivers are asked to change their driving behaviour from normal to fast. When changing their behaviour, a relatively large variation in both parameters Γ_{max} and ΔC was found. Therefore, it is concluded by G. Raymond et al. [15] that "motion cuing did not change the driving strategy itself but altered the perceived motion variables that serve as inputs to this strategy. The apparent modification in the ΔC margin can in fact reflect a modification in perceived speed V corresponding to an underestimation in the static condition."

The small number of participants in the control group and their inexperience with the simulator produced dispersed results in Experiment 2, which did not allow to draw statistically valid conclusions regarding the comparison of driving strategies in real and simulated conditions for the same drivers. Despite the incompleteness of these tests, results from both simulated and real driving experiments are coherent with the proposed model, which thus provides a common framework of analysis by relating a driving behavioral pattern to the nature of physical stimuli (speed, acceleration) perceived by the drivers.

4.4. TU Delft Simulators

At the faculty of Aerospace Engineering, Control and Simulation Division, simulators are available for research purposes.

4.4.1. Fixed-Based Simulator (HMI Lab)

Operational since 2002 is the fixed-base simulator located in the Human-Machine Interaction (HMI) laboratory. This fixed-base simulator can be used for car and aircraft related research (see Figure 4.7). The screens and devices in the lab are controlled by up to 6 computers, linked together in a high-speed ethernet network. The lab is split up into an observation room and an experiment room [33].

Figure 4.7: Fixed-Based Simulator at HMI lab TU Delft [33]

The car side of the simulator contains an actuated steering wheel, an actuated gas pedal and a passive spring loaded brake pedal [31]. Furthermore it contain a luxury NISSAN car seat and a 12 inch LCD panel for the instrument displays (800 x 600 pixel resolution) [33]. Both sides share a unique outside visual projection system. That system is driven off one quad-core PC, with two dual head graphics cards driving a total of 3 DLP projectors with HD resolution. The projectors use a three-sided projection screen, and cover a field of view of over 180º (see Figure 4.8).

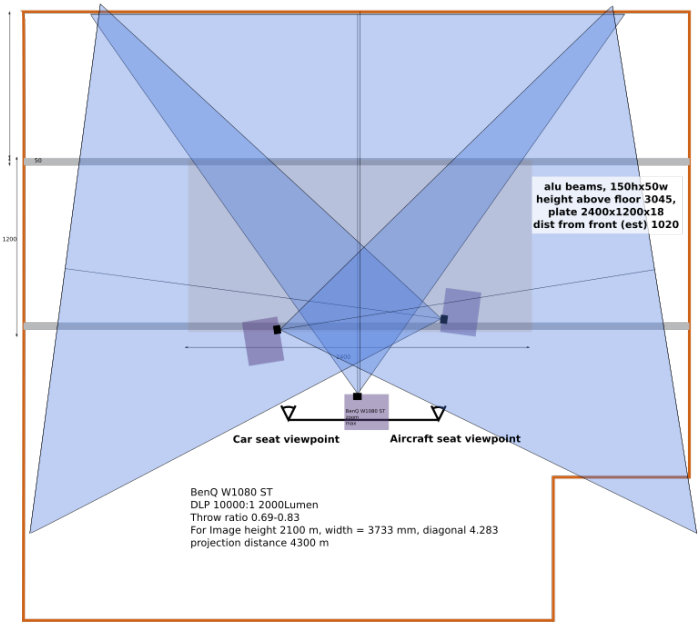


Figure 4.8: Fixed-Based Simulator projection system schematic [33]

4.4.2. SIMONA Research Simulator

The Simona Research Simulator can realistically simulate all types of aircraft, helicopters and even cars. The simulator was specially built for TU Delft and is used as a laboratory for education and research in the fields

of flight simulation technology and human-machine interaction [5].

The cabin of the SIMONA Research Simulator is made of lightweight materials and holds a two person flight deck. The SRS' glass cockpit is similar to a modern transport aircraft, and provides programmable instrumentation to match any aircraft type [5].

The six-degrees-of-freedom motion system is controlled by a multi-variable control system. The linear actuators are equipped with hydrostatic bearings and allow a 1.25 m stroke with advanced safety buffers. Robust actuator controllers compensate for internal variations in load, temperature, valve dynamics and other effects. The motion system can realize a maximum heave acceleration of 1.5 g and a minimum which lies below the threshold for human perception [5].

Integrated with the flight-deck structure, the lightweight outside world display system provides a collimated 180-degree horizontal by 40-degree vertical field of view [5].

(a) Simona hexapod base

(b) Inside view

Figure 4.9: The Simona research simulator [5]

4.5. Conclusions

Many different types of simulators exist for a logical reason: not all simulators have the same purpose. With driving behaviour research as purpose, the most obvious simulator to use to collect data in this research will be the fixed-base driving simulator in the HMI-lab at TU Delft. This simulator has a large field of view so that all potential visual cues can be observed. The downside of this simulator is the absence of (simulated) accelerations. As shown in Figure 4.6, the driving behaviour will most likely differ from the real world. However, the behavioural curve does only shift the equation parameters due to the absence of accelerations while the type of relation remains the same. This means that when this (static) behaviour can be captured using a specific method, the same method will in all likelihood also be able to capture driving behaviour including acceleration cues.

5

Neural Networks / Machine Learning

Even though the idea of artificial neural networks is quite old (1800s), the application of neural networks to model identification is relatively advanced and new. They are based on biological neural networks, also known as brains. The brain is made up of neurons, which are an extremely complex analog computer [4]. An artificial neuron is a much more simplified mathematical version of the biological neuron. Examples of some applications for neural networks are pattern recognition, speech recognition, stock market forecasting, aerodynamic model identification and games like Chess and Backgammon.

5.1. General Definition

A neural network consists of an input layer (i), hidden layer(s) (j) and output layer (k). The general description is shown in Figure 5.1. For each layer, v is the input and y the output. Each layer also contains an activation function (ϕ) relating input v to output y. Therefore, in one layer $y = \phi(v)$.

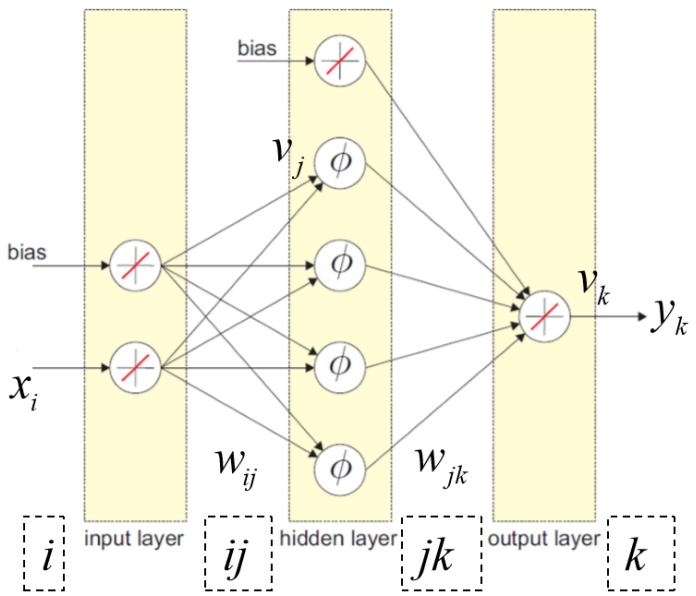


Figure 5.1: General Neural Network schematic [4]

In order to relate the output of the output layer (y_k) to the input of the input layer ($v_i = x_i$), Equation 5.1 is used. This is the most general description of a neural network containing one hidden layer. Usually, these equations are a bit simplified. The activation function in the in- and output layers is taken as a 'linear activation function' ($y = v$). Using this property, Equation 5.1 is simplified to Equation 5.2 [4].

$$\begin{aligned} y_k &= \phi_k(v_k) \\ v_k &= \sum_j w_{jk} y_j = \sum_j w_{jk} \phi_j(v_j) \\ v_j &= \sum_i w_{ij} y_i = \sum_i w_{ij} \phi_i(v_i) \end{aligned} \tag{5.1}$$

$$y_k = \sum_j w_{jk} \phi_j \left(\sum_i w_{ij} x_i \right) \tag{5.2}$$

Above a general description of a neural network is given. But in reality there are many different types of neural networks. All networks are designed for a specific task. There is not a single network which is optimal for all purposes. Things that can vary from one network to another are the activation functions, the amount of neurons, the in- and outputs, the training algorithm and more. For a specific task, the right combination of variables must be found to solve the problem as good as possible.

5.2. Activation Functions

The most popular activation functions can be found below in Table 5.1 and visualised in Figure 5.2 [4].

Table 5.1: Four popular activation functions

Name	**Function**
Linear (purelin)	$\varphi(x) = x$
Sigmoid (logsig)	$\varphi(x) = \frac{1}{1+e^{-x}}$
TanH (tansig)	$\varphi(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
Radial basis function (rbf)	$\varphi(x) = e^{-x^2}$

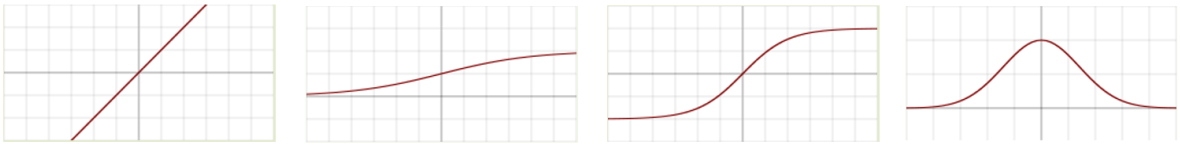


Figure 5.2: Activation functions left to right: 1) linear 2) sigmoid 3) TanH 4) rbf

The Theorem on Neural Networks from Cybenko (1989) is as follows:

"A feedforward neural net with at least one hidden layer with sigmoidal activation functions can approximate any continuous nonlinear function arbitrarily well on a compact set, provided that a sufficient number of hidden neurons are available. "

The sigmoidal activation function is one of the most popular ones. Another very popular one is the radial basis function. The description for a radial basis function neural network differs a bit from the general one of Equation 5.1 since the center position (c_{ij}) of the activation function is taken in consideration in the input layer instead of the usual bias term (see Equation 5.3) [4]. When substituting the center position into the general definition, Equation 5.1 becomes the following:

$$\begin{aligned} y_k &= v_k \\ v_k &= \sum_j w_{jk} \phi_j(v_j) \\ v_j &= \sum w_{ij} (x_i - c_{ij})^2 \end{aligned} \tag{5.3}$$

The main differences between a sigmoidal (tansig or logsig) and radial basis function neural network are:

- rbf is easier to model local complexity
- rbf has higher computational load
- rbf is easier to optimize

5.3. Network Training

Training of a neural network is done using error back propagation. The goal is to update the weights in such a way that the output (y_k) of the neural network comes closer to the target values (d_k) (known and correct output). Updates are performed using partial derivatives from error (E) to weights (w) so that a step is taken into negative gradient direction (see Equations 5.4a and 5.4b). Also a learning parameter η is used to increase or decrease the step size. The trade-off of a larger step size is instability.

$$w_{t+1} = w_t + \Delta w \tag{5.4a}$$

$$\Delta w = -\eta \frac{\partial E}{\partial w_t} \tag{5.4b}$$

$$E = \sum_k \frac{1}{2} \left(d_k - y_k\right)^2 = \sum_k \frac{1}{2} \left(e_k\right)^2 \tag{5.4c}$$

The partial derivatives used to find $\partial E / \partial w_t$ are given in Equations 5.5 and 5.6. These are the general partial derivatives used to update the weights for all 'q' data points. A similar approach is used to update the bias terms (feedforward) or the center terms (radial basis function).

$$\frac{\partial E}{\partial w_{jk}} = \sum_q \frac{\partial E}{\partial e_{k,q}} \frac{\partial e_{k,q}}{\partial y_{k,q}} \frac{\partial y_{k,q}}{\partial v_{k,q}} \frac{\partial v_{k,q}}{\partial w_{jk}} = \sum_q e_{k,q} \cdot -1 \cdot 1 \cdot y_j \tag{5.5}$$

$$\frac{\partial E}{\partial W_{ij}} = \sum_q \left[\sum_k \frac{\partial E}{\partial e_{k,q}} \frac{\partial e_{k,q}}{\partial y_{k,q}} \frac{\partial y_{k,q}}{\partial v_{k,q}} \frac{\partial v_{k,q}}{\partial y_{j,q}} \frac{\partial y_{j,q}}{\partial v_{j,q}} \frac{\partial v_{j,q}}{\partial W_{ij}} \right] = \sum_q \left[\sum_k e_k \cdot -1 \cdot 1 \cdot w_{jk} \cdot \frac{\partial \phi_j \left(v_j\right)}{\partial v_j} \cdot x_i \right] \tag{5.6}$$

The update approach from Equation 5.4a is a 'First Order Gradient Descent Algorithm'. There are also other ways to update the weights such as 'Resilient Backpropagation', 'Fletcher-Powell Conjugate Gradient' or the most popular one, the 'Levenberg-Marquardt' (LM) method. This method is a second order method and in order to update the weight for the next iteration, Equation 5.7 is used.

$$W_{t+1} = W_t - \left(J^{\mathrm{T}} J + \mu I\right)^{-1} J^{\mathrm{T}} e \tag{5.7}$$

In which W is the vector containing all weights to be updated. e is a vector containing the error ($d_k - y_k$). This column vector has the size of the number of data points for d_k. μ is the (adaptive) damping parameter. Finally, J is the Jacobian matrix. The general definition for J can be found in Equation 5.8.

$$\mathbf{J} = \frac{\partial \mathbf{e}}{\partial \mathbf{W}_t} = \begin{bmatrix} \frac{\partial \mathbf{e}(1)}{\partial \mathbf{W}_{ij}(1)} & \frac{\partial \mathbf{e}(1)}{\partial \mathbf{W}_{ij}(2)} & \cdots & \frac{\partial \mathbf{e}(1)}{\partial \mathbf{W}_{ij}(K)} & \frac{\partial \mathbf{e}(1)}{\partial \mathbf{W}_{jk}(1)} & \frac{\partial \mathbf{e}(1)}{\partial \mathbf{W}_{jk}(2)} & \cdots & \frac{\partial \mathbf{e}(1)}{\partial \mathbf{W}_{jk}(M)} \\ \frac{\partial \mathbf{e}(2)}{\partial \mathbf{W}_{ij}(1)} & \frac{\partial \mathbf{e}(2)}{\partial \mathbf{W}_{ij}(2)} & \cdots & \frac{\partial \mathbf{e}(2)}{\partial \mathbf{W}_{ij}(K)} & \frac{\partial \mathbf{e}(2)}{\partial \mathbf{W}_{jk}(1)} & \frac{\partial \mathbf{e}(2)}{\partial \mathbf{W}_{jk}(2)} & \cdots & \frac{\partial \mathbf{e}(2)}{\partial \mathbf{W}_{jk}(M)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{e}(N)}{\partial \mathbf{W}_{ij}(1)} & \frac{\partial \mathbf{e}(N)}{\partial \mathbf{W}_{ij}(2)} & \cdots & \frac{\partial \mathbf{e}(N)}{\partial \mathbf{W}_{ij}(K)} & \frac{\partial \mathbf{e}(N)}{\partial \mathbf{W}_{jk}(1)} & \frac{\partial \mathbf{e}(N)}{\partial \mathbf{W}_{jk}(2)} & \cdots & \frac{\partial \mathbf{e}(N)}{\partial \mathbf{W}_{jk}(M)} \end{bmatrix} \tag{5.8}$$

In which N is the number of data points, K the number of input weights (inputs x $N_{neurons}$) and M the number of output weights (outputs x $N_{neurons}$).

5.4. Effect of Hidden Layers and Neurons

Adding extra neurons to a neural network does not necessarily result in better performance. When more neurons are used, the network is able to model more complex systems. This can be an advantage or disadvantage. Usually the data are split up into train, test and validation data. While the network is trained the Mean Squared Error between target and output is computed for the train and test data. A way to find out whether too many neurons are used is to compare the MSE between the test and training data. With too many neurons, the network will 'overfit' the training data. Overfitting means that the training data will be fitted to exact by the model so that any future predictions will be unreliable. An example of overfitted data is shown in Figure 5.3. The point where overfitting occurs can be found when the test data MSE tends to increase instead of decrease further. When not enough neurons are used, the model will 'underfit' the data. This is usually easily spotted due to poor performance on the training data.

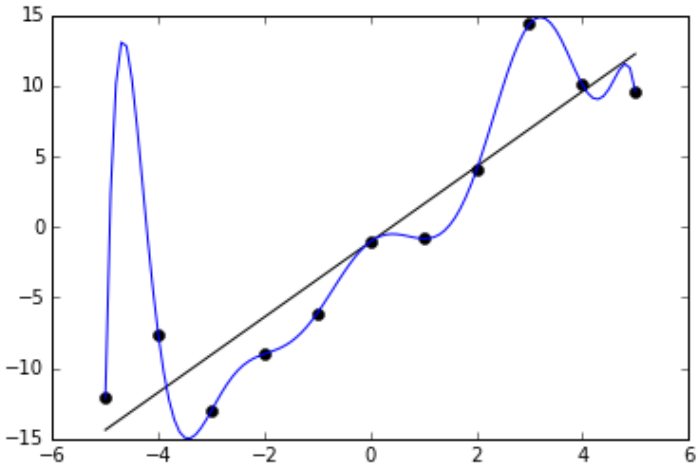


Figure 5.3: Overfitted data

Another way to make a neural network more complex is by adding more hidden layers. However, The Universal Approximation Theorem states that a neural network with 1 hidden layer and sigmoidal activation function can approximate any continuous function for inputs within a specific range. In the work from Jacques de Villiers and Etienne Barnard [3], they state that the complexity of a neural network is defined by its number of connections and thus by the number of weights. The number of weights used in a neural network can be calculated using Equation 5.9.

$$Weights = N_{inputs} * N_1 + \sum_{i=1}^{h-1} (N_i * N_{i+1}) + N_h * N_{outputs} \tag{5.9}$$

In which h is the amount of hidden layers and N_x the number of neurons in layer x (input, output or hidden layer 1, 2, 3 ...).

5.5. Conclusions

To conclude from this chapter, there are a few things that should be taken into account when a neural network will be used in order to produce models with the data from this research. First of all, it is important to keep the complexity between neural networks constant when a comparison is made in terms of performance. Next to this, an analysis should be done in which the optimal number of neurons/complexity of the network will be found. When too the model is too complex, over-fitting will happen. Without enough complexity, model performance will be bad. In order to find an optimum the data will be split into test and training data.

II

Preliminary Research

The preliminary research is split into two chapters. In chapter 6 existing data will be used in order to test the main assumptions that have emerged during the literature study. In this part the data will be analysed and relations between various in- and outputs will be tested. In the chapter 7 a thesis experiment will be designed. For this, the findings from the literature and from the preliminary experiment will be used.

6

Preliminary Experiment

In this chapter a preliminary experiment will be discussed. In the literature study many different approaches to driver behaviour modeling are discussed. The goal of this research is to only include visual cues in the driver model. The kind of data that are needed to construct this driver model is still unknown. Therefore, a preliminary experiment will be performed with previously collected data that are available. The objective of the preliminary experiment is to find how much certain visual inputs are related to driving behaviour. When these inputs are analysed, it is also important to find why they are (or aren't) related and which data is still lacking in order to draw a proper conclusion. With results from the preliminary experiment, the experimental setup of final thesis can be designed. This will be done in chapter 7.

6.1. Data

The data that was available to use for the preliminary experiment was data from the work of Gruppelaar et al. [25]. In his work he collected data using the Fixed-Based Simulator in the HMI Lab at the faculty of Aerospace Engineering of Delft University of Technology (see section 4.4). The simulated vehicle was equipped with an automatic 4-speed gearbox. In order to improve speed perception, the road was lined with poles spaced at regular intervals, and trees were placed in the scenery as can be seen in Figure 6.1 [25].

Figure 6.1: Fixed-based driving simulation with surroundings

The data was collected on 6 different roads (3 mirrored pairs), all consisting of 8 curves with straight parts in between them. The roads were approximately 4 kilometer long. Road pair 1/2 is the same as road pair 5/6, except from the width of the road. Road pair 5/6 used a width of 2.4 meters, and all other roads used a width of 3.6 meters. An example of a road (number 3) is given in Figure 6.2 [25].

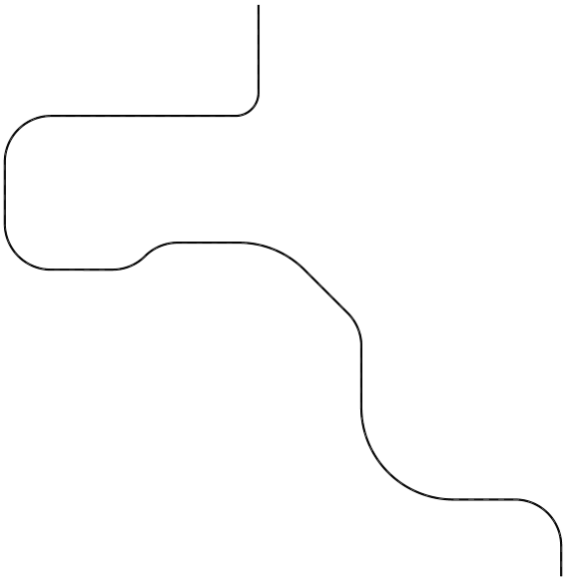


Figure 6.2: Experimental road example (number 3) [25]

The most essential data that was collected during these experiments were time series of the velocity and acceleration (both lateral and longitudinal), x and y position, yaw angle, gas/brake pedal deflection and force and steering wheel angle.

6.2. Experimental Goal

In most research discussed in the literature study, road geometry is used in order to compute a driver velocity in curve driving. within a curve, the road geometry (usually) remain constant. However, the velocity with which a driver goes through a curve is generally not constant. This is also established in previously discussed research. In Figure 2.1 the velocity time series is shown for vehicles entering and exiting a curve. Clearly, the velocity is not constant throughout the curve. In the work from Reymond et al. [15] it is stated that when entering a curve the speed choice strategy of a driver is based on adjusting a safety margin of lateral acceleration. When entering a curve, the driver will lower the car velocity to avoid reaching a maximum value for lateral acceleration (Γ_{max}). The value for Γ_{max} varies from person to person based on driving experience, handling performance and personal level of acceptable risk. Not reaching this value will make the driver more comfortable. A margin is taken while decelerating in case of an unexpected deviation (obstacle, steering error or increase in road curvature) [15]. After the entry of the curve, the driver could feel more comfortable and starts to accelerate towards an acceptable velocity for that specific curve. This kind of velocity profile is seen in other research too. In the work from Gruppelaar [25], a curve a split into 5 phases. These phases are acceleration, deceleration, braking, brake release and re-acceleration. A flowchart of the phases related to time margin can be found in Figure 3.5 [25]. The time series velocity and pedal deflections of these phases are shown in Figure 6.3.

In the research from Xie et al. [16] the road curvature was not split into 5 phases but into 3 phases. A curve would be split into curve entry, mid curve and curve exit. During curve entry a driver would decelerate, in the mid curve keep a constant speed, and while exiting the curve, acceleration would take place. These phases are very similar to those in Figure 6.3 if deceleration and braking would be put into one phase, and the acceleration phase before a curve would not be taken into account.

By now is has become pretty clear that drivers do not maintain a constant velocity between the start en end of a curve. The goal of this preliminary research is find potential visual points that could be used by drivers to control the vehicle velocity. Visual points, and time/distance towards those points do not only capture road geometry, but also vehicle position on the road and within a curve. The visual inputs will be fed to a neural network which will then try to predict the speed from the driver model. The in- and outputs and the details of this network will be discussed in the next section.

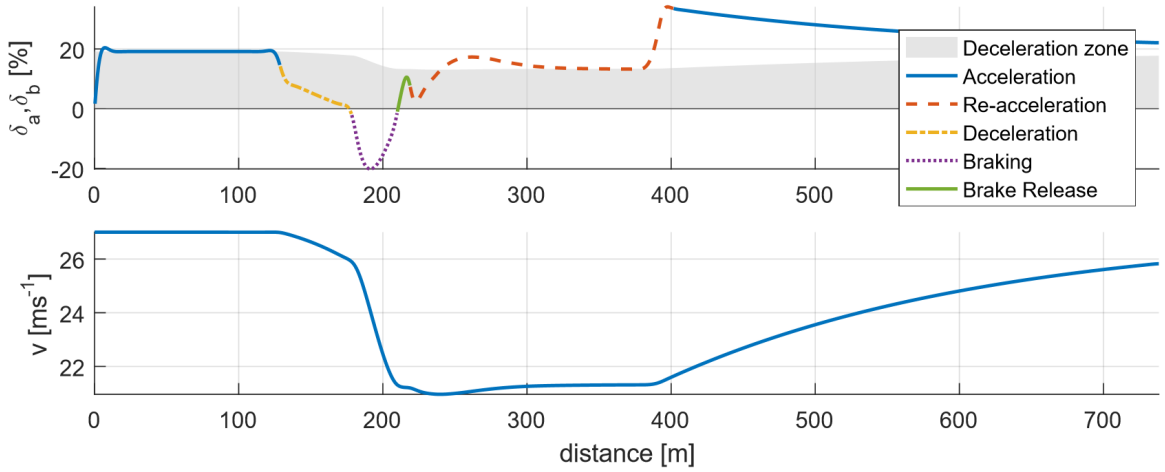


Figure 6.3: Five phases throughout a curve [25]

6.3. Model Inputs and Outputs

The inputs that will be analysed in the preliminary experiment are time and distances to visual points. The visual points that are used are the Tangent Point (TP), the Extended Tangent Point (ETP) and the Straight Lane Crossing Point (SLCP) (see Figure 6.4). The SLCP is the point where the vehicle crosses the road in case it continues to drive straight ahead. This is the Lane Crossing Point that is also used in the theoretical work from Boer [12]. For one visual point, various inputs can be thought of. From the ETP point, inputs Time to ETP (TETP), Distance to ETP (DETP) and Angle to ETP can be used. The TETP is defined as the DETP divided by the current velocity.

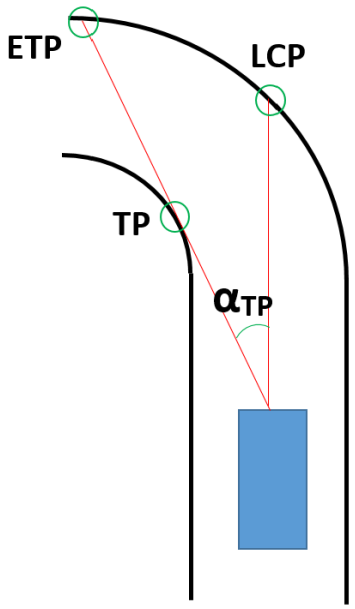


Figure 6.4: Top view schematic of specific visual points

The most straightforward output to determine driver behaviour through curves would be the (longitudinal) velocity. This output is most often used in literature to determine driver behaviour in curve driving. In order to analyse driver behaviour in a more detailed fashion, also gas and brake pedal deflections will be used as outputs. These pedal deflections can be used to compute a velocity profile iteratively using accelerations instead of directly from data. As concluded in the work from Gruppelaar [25], the TETP and its derivative have

a strong connection with the start of the braking/acceleration phase. When vehicle velocity is used as output it is important to chose the right inputs. When for example the distance and time to a visual point is used, the model will not learn a relation between visual points and desired velocity. Instead, it will learn the physical relation of V = distance/time since the time to a visual point is defined as distance/V and thus, indirectly, the output is given as "disguised" input. The output of such a model will always be the current vehicle velocity instead of the desired velocity.

From the literature study it has become clear that road geometry plays an important factor in speed determination. Even though humans are not very well capable of estimating curve radius, they can estimate times and distances to points in the visual field. It could be of interest to investigate which visual cues are the closest related to road geometry. In order to do so, combinations of inputs are plugged into the neural network used in section 6.5, with as output the (known) curvature. It turns out that a combination of distances to the TP, ETP and SLCP, together with the TP angle was best at predicting the curvature. The coefficient of determination (R^2) of network output and (perfect) linear fit between target and output equals 0.89 (R = 0.945). The result is plotted and shown in Figure 6.5. The highest errors are found on the positions close to a change in curvature. A reason for this is that the actual curvature of the road is a non-continuous function (has sudden jumps) while the neural network is a continuous function.

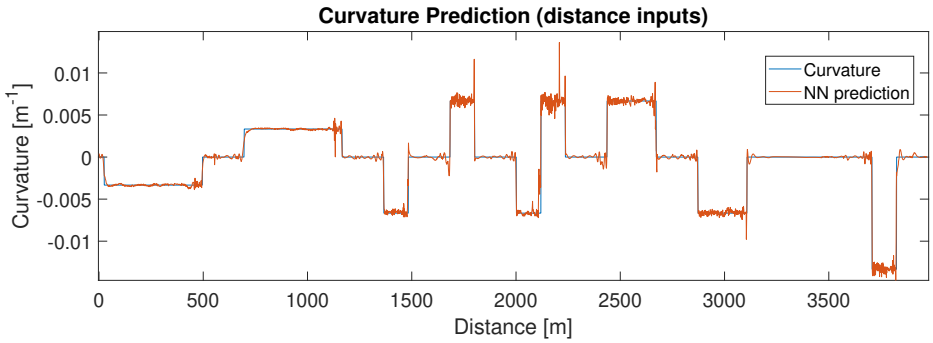


Figure 6.5: Neural Network curvature estimation

6.4. Neural Network

The link between the in- and outputs will be a feed-forward neural network. The activation function that is used is the TanH function (see Table 5.1), and the training method will be the second order Levenberg-Marquardt method. The number of neurons and hidden layers will be determined using trail and error while looking at the performance of the network. The performance is the Mean Squared Error (MSE) between model output and target value. For the preliminary experiment, the network does not need to be fully optimized since only potential candidates for the final experiment need to be found. The influence of the number of neurons when the vehicle velocity is chosen as output and with the four best performing inputs, can be found in Figure 6.6. It can be seen that after around 50 neurons, the increase in performance is relatively low. Due to the time consuming training with additional neurons, no network with more than 50 neurons was used for the preliminary experiment.

In Figure 6.6 the effect of additional neurons can be observed. Another way to change a neural network is by having more hidden layers. In the case of having 50 neurons in one hidden layer, and 4 inputs, the number of weights in the network equals 250 (see Equation 5.9). In the work from Jacques de Villiers and Etienne Barnard [3], they state that the complexity of a neural network is defined by its number of connections and thus weights. In order to compare the effect of extra layers properly, the number of weights should remain equal. According to Equation 5.9, the number of weights in a 50 neuron 1 hidden layer network is equal to one with 3 hidden layers of all 10 neurons. In order to analyse the performance of a these networks, three plots will be used.

In the regression plot (Figure 6.7), the relation between target and output value is shown. The goal of the neural network is to create a regression in which output equals target (y = t) with an R value of 1. The regression

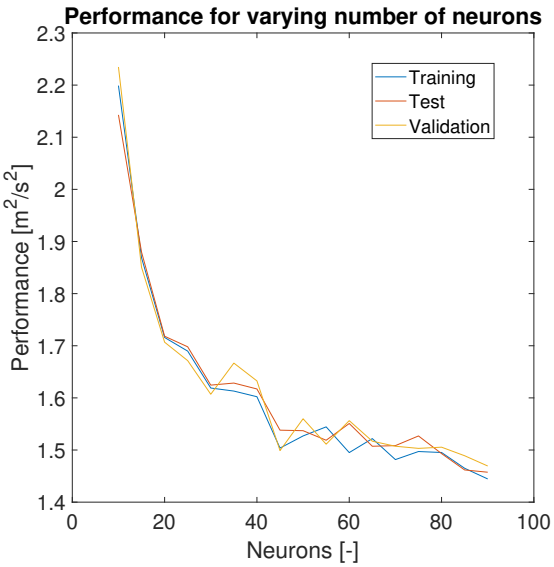


Figure 6.6: Relation between number of neurons and performance

equation on the y-axis gives an indicates of how far the regression of the output is from the target, the R value above the figure is the correlation coefficient. It shows the degree of correlation between input and output within the data. For example, multiple target values when the inputs are the same is an inconsistency in the collected data. This can either be related to noise, poor training or due to a missing input.

The performance plot (Figure 6.8) shows the learning curve of the neural network. In this plot the difference in performance between training, test and validation data is shown. When the performance of the training data is much higher, the network probably has 'over-trained' as explained in section 5.4. The validation performance is the Mean Squared Error (MSE) between the target- and output of the validation data. Also, the point where the best validation performance is found is shown, this does not have to be the last epoch due to over-training.

The output plot (see Figure 6.9) shows the difference between target and output values of a second validation set (also not used for training the neural net and kept separate). In this plot it becomes visible when the neural net is performing well, en when it is not. This plot could be analysed in order to improve data and training in the future. The red marked areas in this plot represent the locations of the corners of the driving track.

The results from the single hidden layer with 50 neurons network are shown in Figure 6.7 - 6.9. These have been compared with the results from a network with triple hidden layer with 10 neurons each (shown in section 6.5, Figures 6.11 - 6.10). In this case, the triple layered neural net (with the same complexity) showed a slightly higher R value, better performance and shorter training time (3m43s instead of 3m58s). Although the differences are very small, the triple layered neural net was chosen to be used for further analysis of the in- and outputs in the next section.

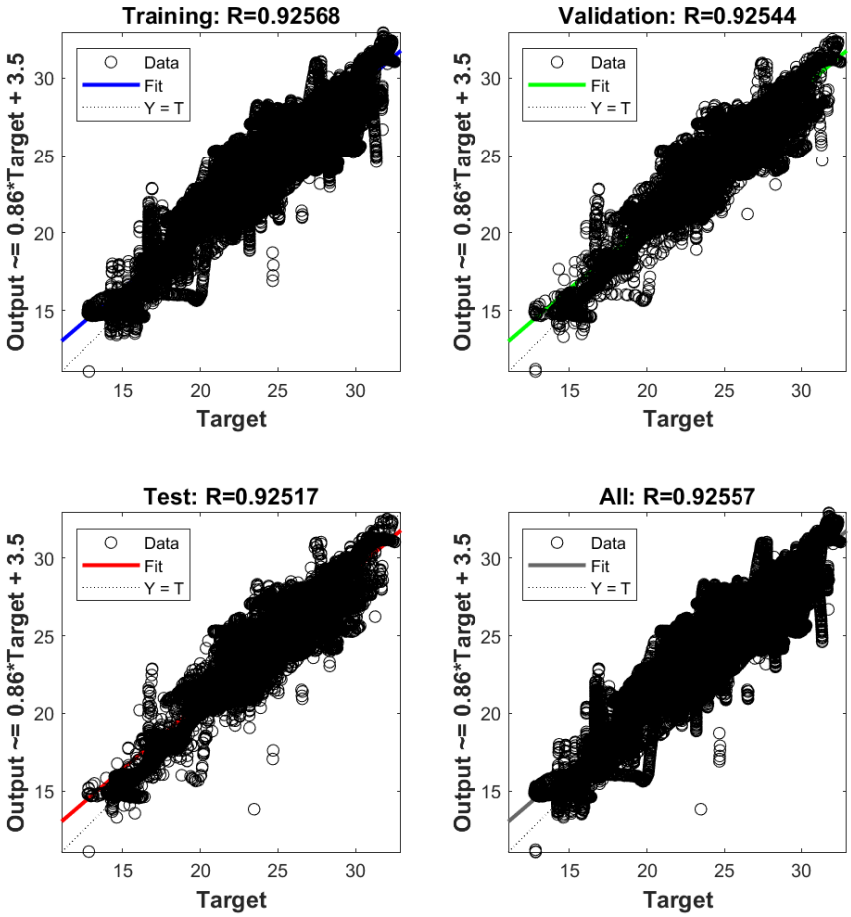


Figure 6.7: Regression plot, 50 neurons, 1 layer, complete road

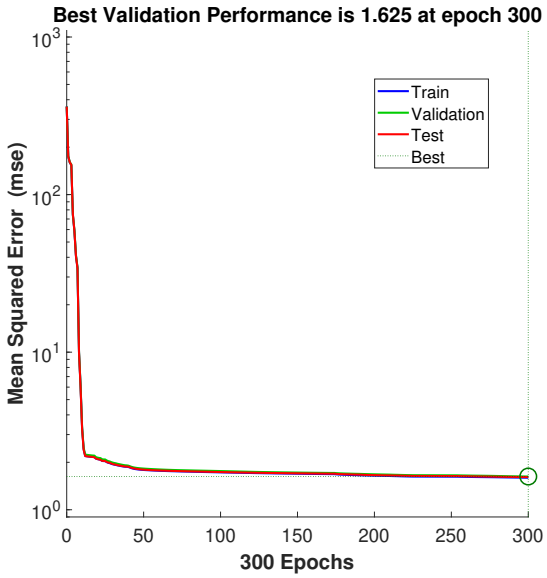


Figure 6.8: Performance plot, 50 neurons, 1 layer, complete road

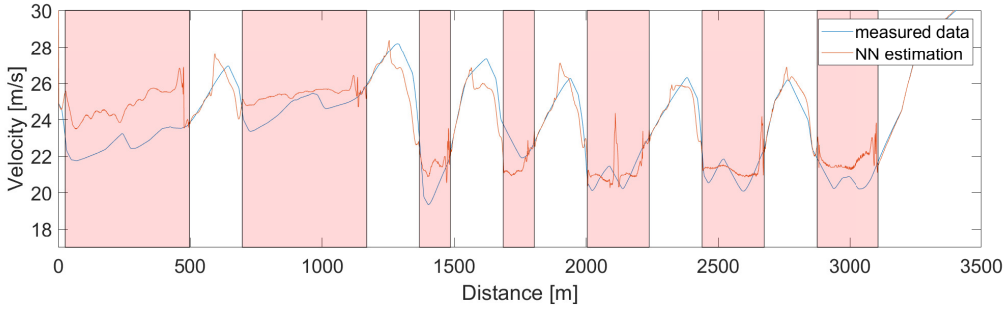


Figure 6.9: Result comparison plot, 50 neurons, 1 layer, complete road

6.5. Results

This section will present the results found with the data from Gruppelaar [25]. Different in- and outputs are tested in the first two subsections and the variability of the data is analysed in the final subsection.

6.5.1. Velocity Profile

For the first experiment, the output was chosen to be the (longitudinal) vehicle velocity. Since no time and distance inputs could be used simultaneously, a choice needed to be made between them. As expected, the velocity as output is more related to the distances of the visual points with respect to the time towards a visual point. This makes sense, as discussed in Gruppelaar's work [25], the time margins are often held constant. In other words, the time to visual points are related to the moment of braking/accelerating while the distances are related to road geometry, hence velocity. As inputs, a combination could be made from distances to the TP, ETP, SLCP and the TP-angle. The highest performance was found when all four inputs were used. The results of the trained neural network can be found in Figures 6.11, 6.12 and 6.10.

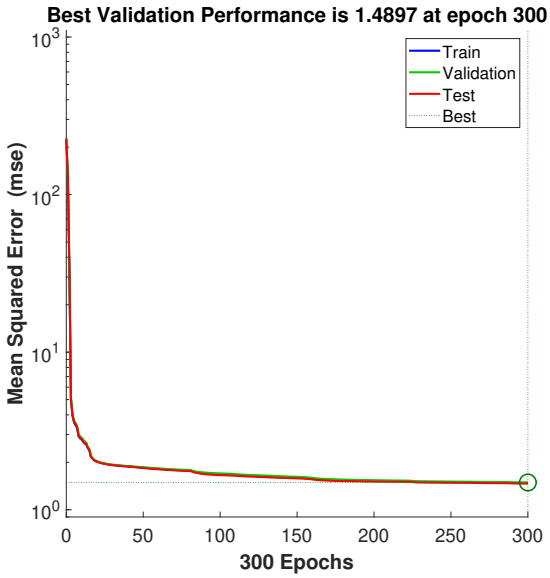


Figure 6.10: Performance plot, 10 neurons per layer, 3 layers, complete road

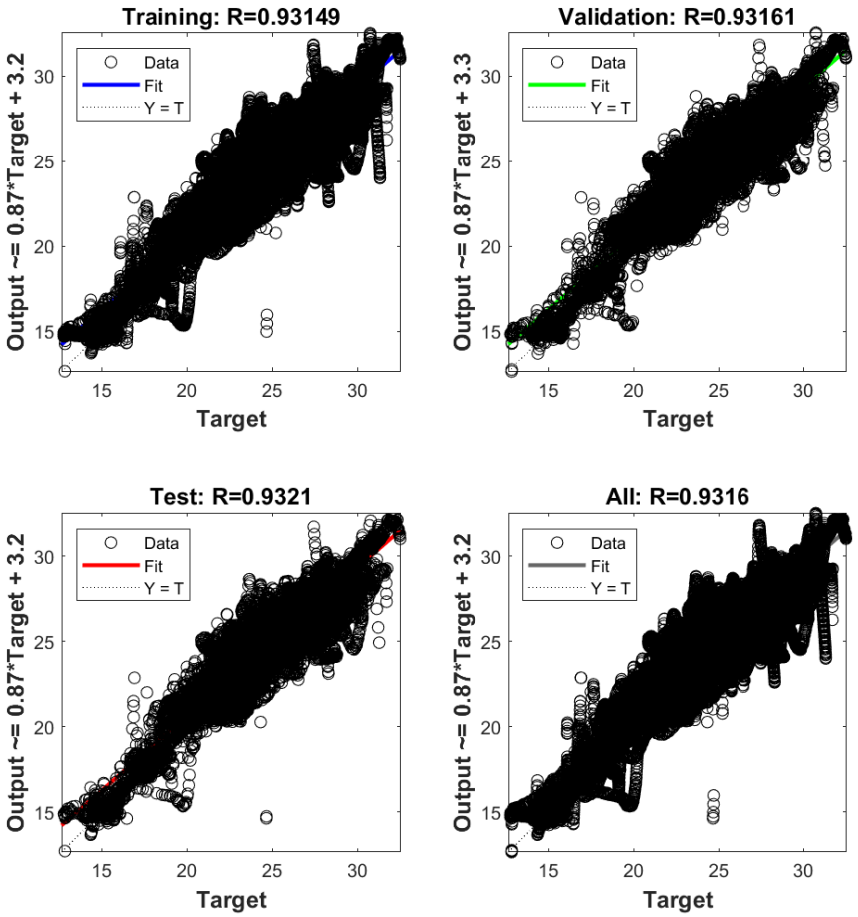


Figure 6.11: Regression plot, 10 neurons per layer, 3 layers, complete road

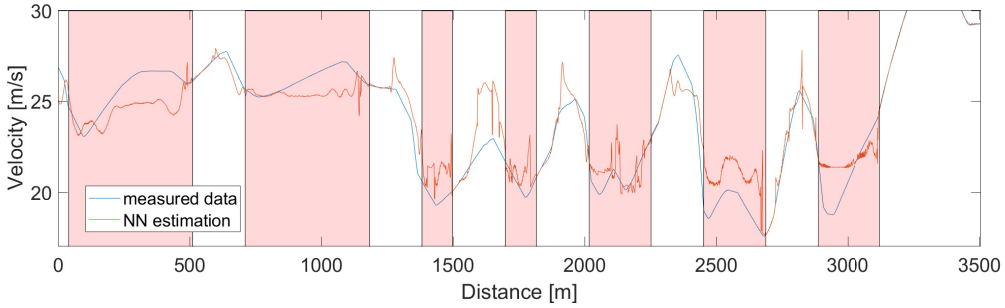


Figure 6.12: Result comparison plot, 10 neurons per layer, 3 layers, complete road

The first results that were found were relatively poor. The regression in Figure 6.11 shows quite a variability between input and output. The performance plot gives a MSE of 1.4897 which equals a Root Mean Square Error (RMSE) of 1.22 m/s. This variability and error between in- and output is visualised in Figure 6.12. In this plot also the position where the largest errors occur can be found. When looking at this plot, peculiarly, the performance is relatively high when the velocity is peaking, and relatively low when the velocity reaches the bottom. The reason that this happens was found using Figure 6.13. When the velocity is at the 'bottom' in the time series, the distance to visual points are all quite constant. When driving through a curve, in the mid phase the distances to those visual points do not change much. Hence, when the inputs do not change

(much), the output will not change much either. It is therefore that the velocity prediction is unable to predict the correct profile properly. It seems that drivers use more or different inputs (might be other than the visual field) to control their velocity in curves.

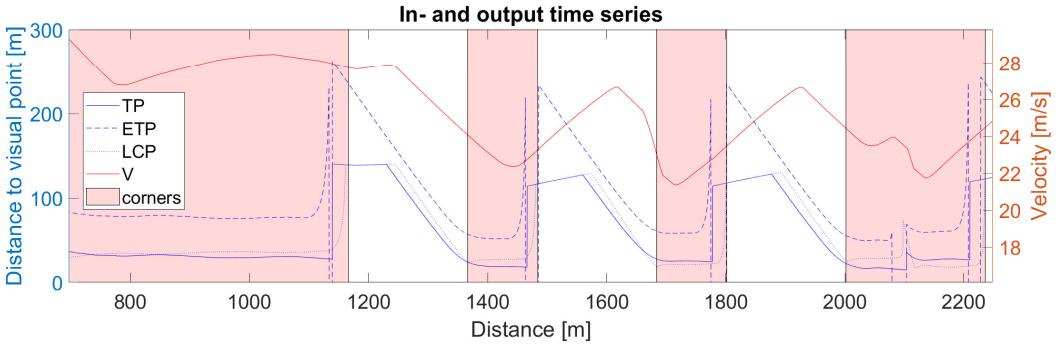


Figure 6.13: Velocity output, visual point inputs time series

In order to verify this hypothesis, the data will be split into data collected on straight parts of the road (driving towards a curve) and data collected driving inside a curve. The straight parts are defined as the parts where the curvature equals 0. The curved parts when the curvature does not equal 0. The results that are found from the data during cornering can be found in Figures 6.16 - 6.14. Note that the distance on the x-axis is not the actual track distance, but accumulated distance of all corners put directly after each other.

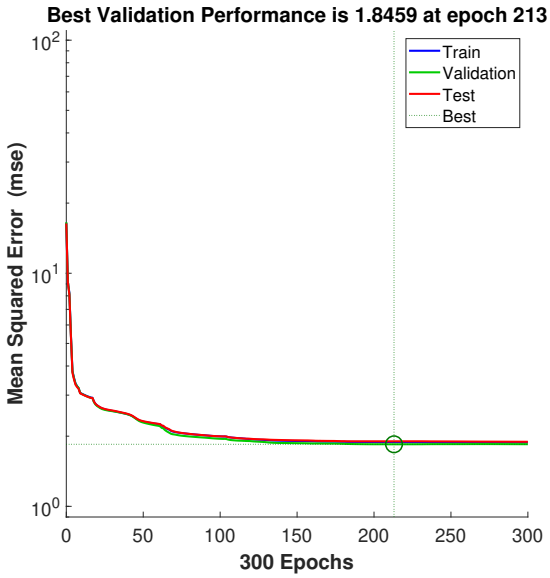


Figure 6.14: Performance plot, 10 neurons per layer, 3 layers, curved road parts

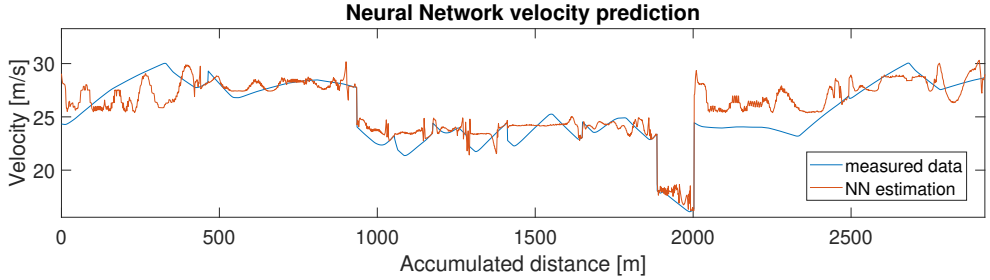


Figure 6.15: Result comparison plot, 10 neurons per layer, 3 layers, curved road parts

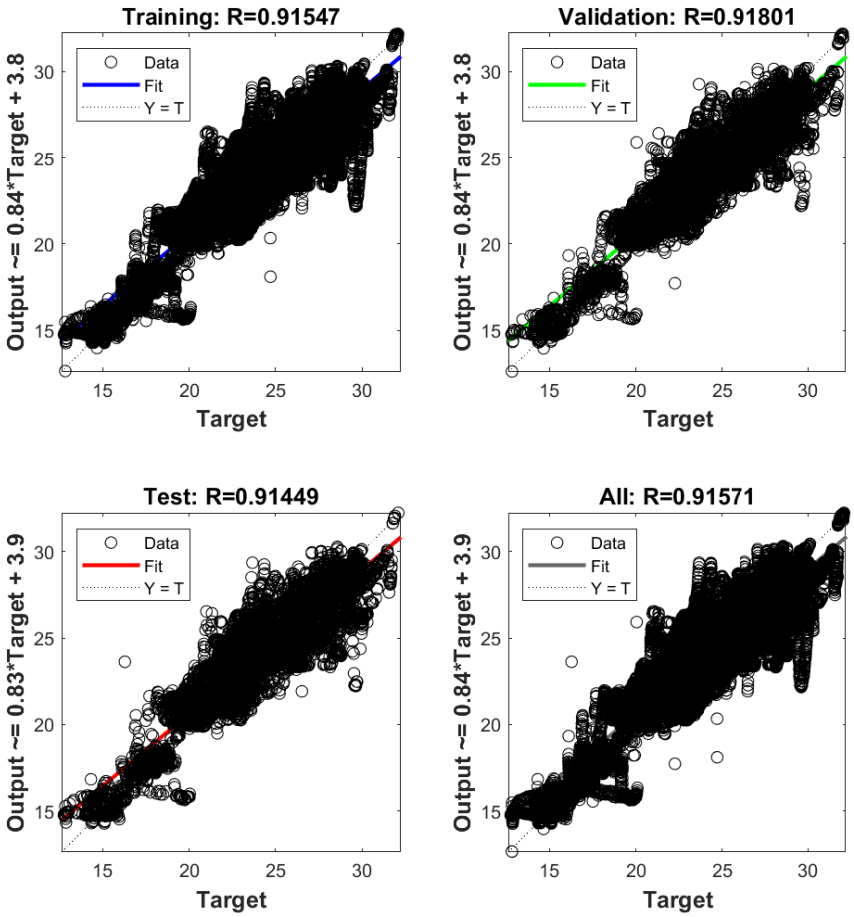


Figure 6.16: Regression plot, 10 neurons per layer, 3 layers, curved road parts

As predicted, the results are of even poorer quality compared to the results from all data analysed together. The validation R value has dropped to 0.918 and the MSE has increased to 1.846. In the output plot (Figure 6.15) it can be observed that the network output has troubles with following the target data. When zooming out on this figure, another notable encounter should be discussed. In Figure 6.17 a pattern is recognized. The output data of the neural network seems to cluster around 3 values. These values are approximately 25, 22 and 16 m/s. When looking at the driving track that was designed to collect this data, corners with 3 different radii are found which can indeed be matched to these 3 velocities. Distances to the visual points differ between corners with different radius. Therefore a difference in output velocity is found. However, as predicted, the neural network has troubles assigning the correct output velocity within one curve where the distances to the visual points remain fairly constant.

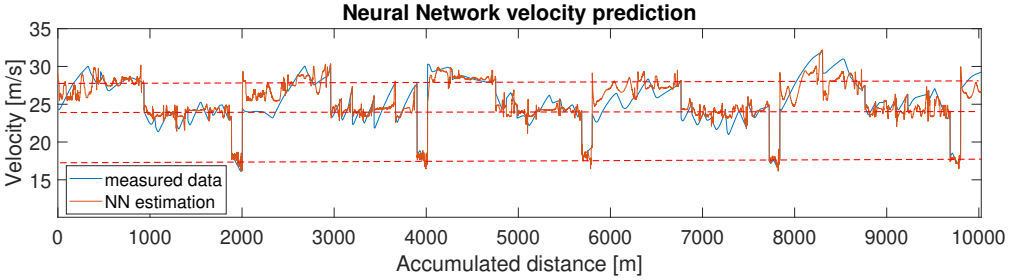


Figure 6.17: Zoomed out result plot from Figure 6.15

On the contrary, the outputs resulting from the neural network trained and tested on only the straight road parts are promising. The validation R value has increased from 0.92 to 0.97 and the MSE has dropped from 1.85 to 0.53 with respect to the results in corners. This is clearly a better network performance and higher correlation between in- and output.

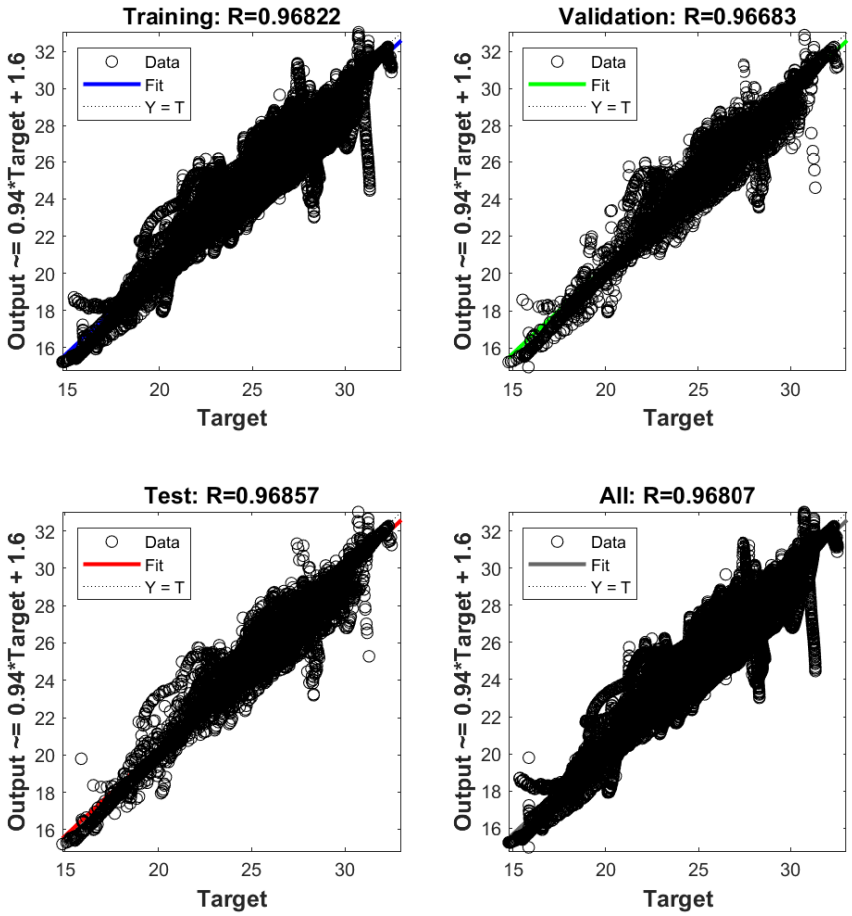


Figure 6.18: Regression plot, 10 neurons per layer, 3 layers, straight road parts

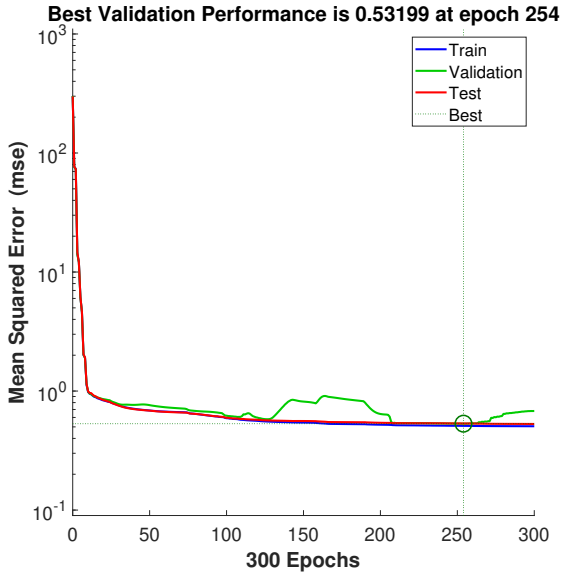


Figure 6.19: Performance plot, 10 neurons per layer, 3 layers, straight road parts

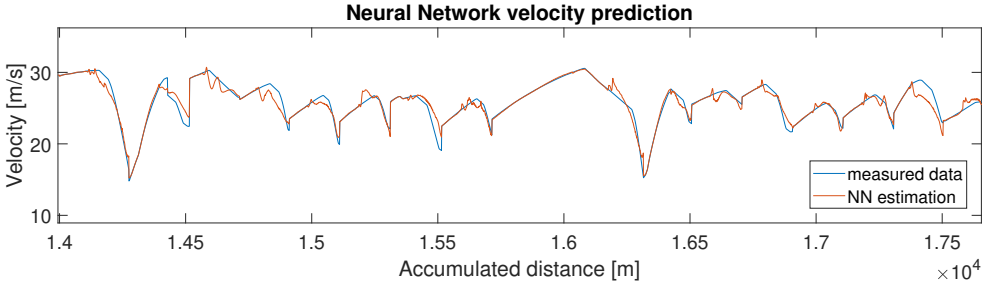


Figure 6.20: Result comparison plot, 10 neurons per layer, 3 layers, straight road parts

6.5.2. Pedal Deflections

Instead of using the velocity as output, also outputs brake- and accelerator pedal deflections will be analysed. In this case it is allowed to use time, distances and velocity as inputs simultaneously since there is no direct relation between these and the pedal deflections. The inputs that were used are TETP, TTP, TLC and longitudinal velocity. With these inputs, indirectly also the distances to visual points are used since the time values are calculated using t = d / V. The results of the brake pedal deflection on straight parts of the road can be found in Figure 6.21 - 6.22.

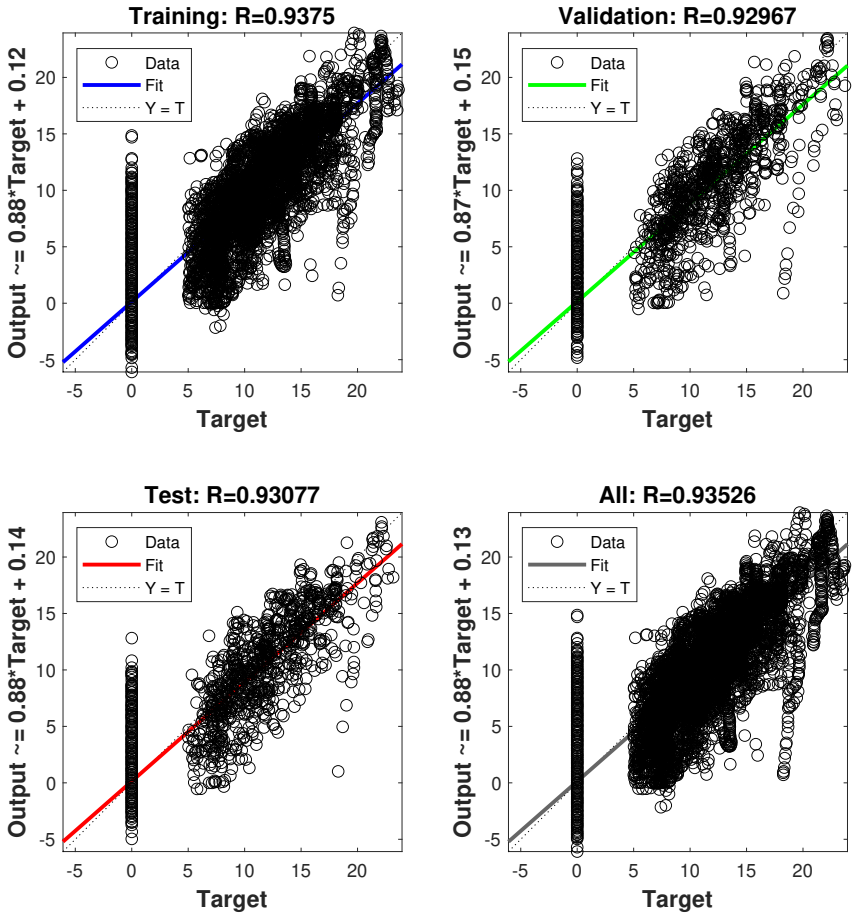


Figure 6.21: Regression plot, 10 neurons per layer, 3 layers, straight road parts, brake pedal deflection output

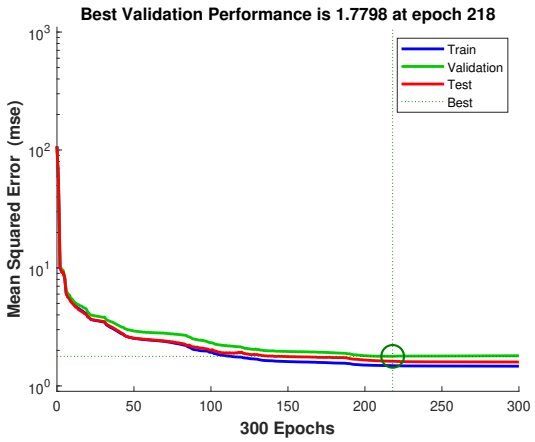


Figure 6.22: Performance plot, 10 neurons per layer, 3 layers, straight road parts, brake pedal deflection output

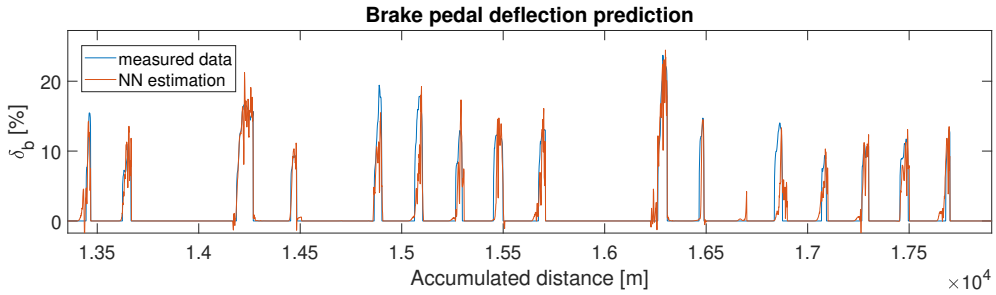


Figure 6.23: Result comparison plot, 10 neurons per layer, 3 layers, straight road parts, brake pedal deflection output

Overall, these results are not too bad. In Figure 6.21 an R value of 0.93 for the validation data is given. Another thing that is shown in this plot is the straight vertical line of data above target equal to 0. These output values correspond to 'False Positives' (FP) i.e. the output gives a positive value while the target is actually 0. The percentage of False Positives in the test data equals approximately 16.1%. This could be an error by the model, or an inconsistency in human braking behaviour. The number of 'False Negatives', i.e. no braking while the target would brake, is relatively low (1.5%). Even though the R value is not very large, the output of the network does not do a bad job identifying when braking occurs. This can be seen in Figure 6.23. The moment of braking is captured quite well by the model. Most of the errors and inconsistencies are due to not perfectly identifying the magnitude with which the driver brakes. Probably, this is due to the variance in human behaviour. But the degree of variance in human behaviour should be investigated further. The performance plot (Figure 6.22 shows a high performance. A MSE of 1.78 is equal to an RMSE of only 1.33%. However, this value is not very representative since most of the time the brake pedal deflection is 0%. Also, again the same difference in results is found between the straight and curved road parts. The R value of the regression plot from the data in the curve equals 0.75. The MSE cannot be compared properly since in corners the number of brake pedal deflection equal to 0 is much higher.

The results from the accelerator pedal as output on straight road segments are shown in Figure 6.24 - 6.25. It is clearly visible that the accelerator pedal deflection is less related to the visual inputs with respect to velocity and braking. A reason for this might be that accelerating is not controlled as actively as braking. When looking at Figure 6.26, it is seen that the magnitude of the deflection is often incorrectly predicted while the moment of pedal release (engine braking) is predicted well. The percentage of FPs in this case equals 4.7%. This is much lower than for braking. However, the cases where the accelerator pedal deflection equals 0 is also much lower (less chances for FPs to happen). The percentage of FN equals 6.8%, which is higher compared to braking. However, again, for accelerating the chances for a FN is much higher since it is almost never 0.

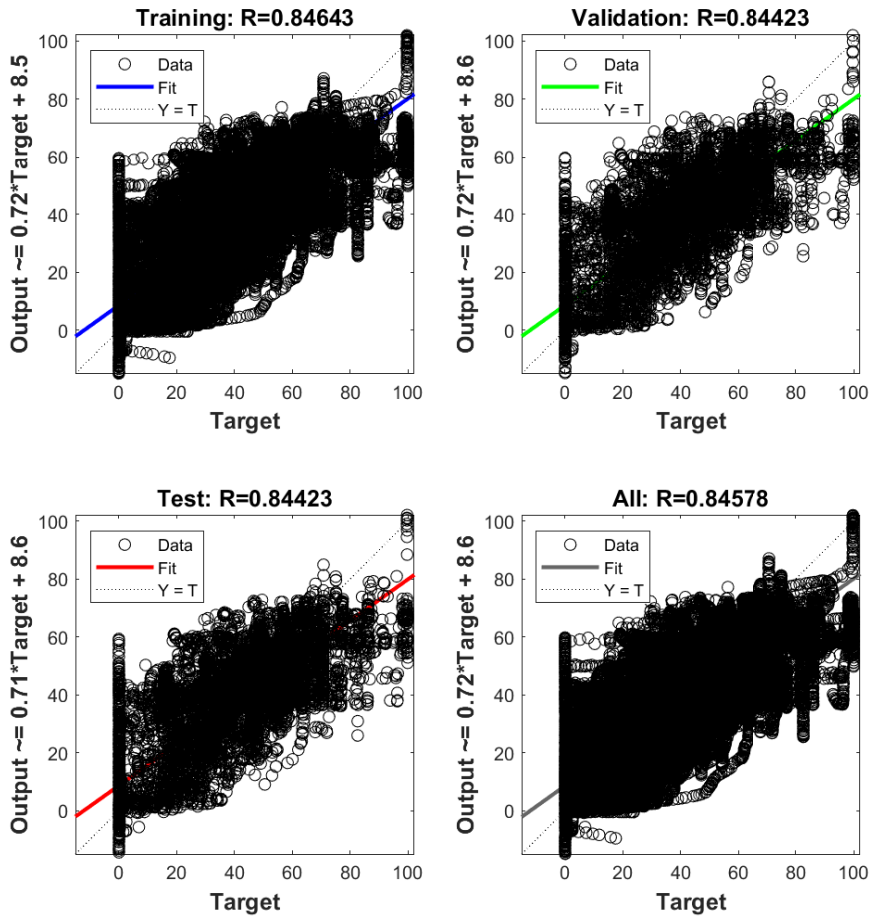


Figure 6.24: Regression plot, 10 neurons per layer, 3 layers, straight road parts, accelerator pedal deflection output

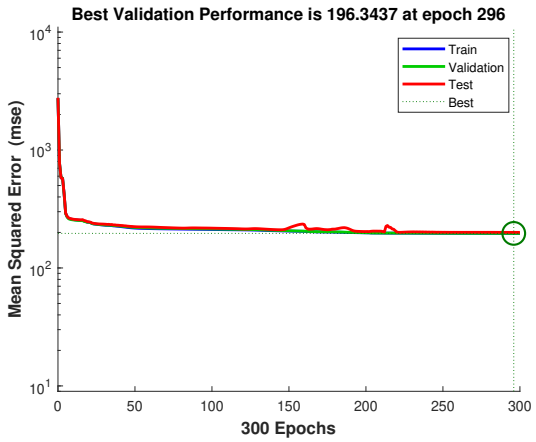


Figure 6.25: Performance plot, 10 neurons per layer, 3 layers, straight road parts, accelerator pedal deflection output

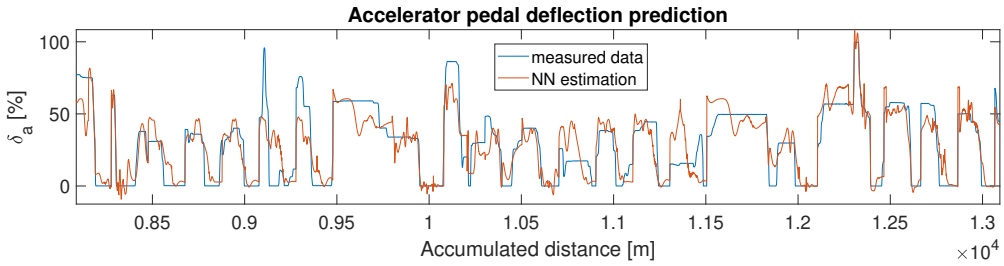


Figure 6.26: Result comparison plot, 10 neurons per layer, 3 layers, straight road parts, accelerator pedal deflection output

Another way to analyse the brake pedal deflections is to combine the two. Similar to the work of Gruppelaar [25], a combined model of brake and accelerator pedal deflection is made. This model is then iteratively transformed to an acceleration which can be integrated into velocity. An experimental model is made with only the data from participant 4 (same participant as shown in [25]). This predictive model can be compared with the actual participant in Figure 6.27. In order to convert this model to a self accelerating algorithm and compare the velocity profile, a model of the vehicle dynamics is needed. This model has not yet been available, but will be in the actual thesis experiment. To still have a rough idea of how this iterative process works, acceleration was taken as model output instead of pedal deflections. This acceleration is integrated to velocity and plotted in Figure 6.28. In this figure, the blue line represents a self accelerating algorithm. This plot does not have the same input values as the actual participant had since from the start the two deviate from each other. It seems that the algorithm does pick up the corners and decelerates accordingly.

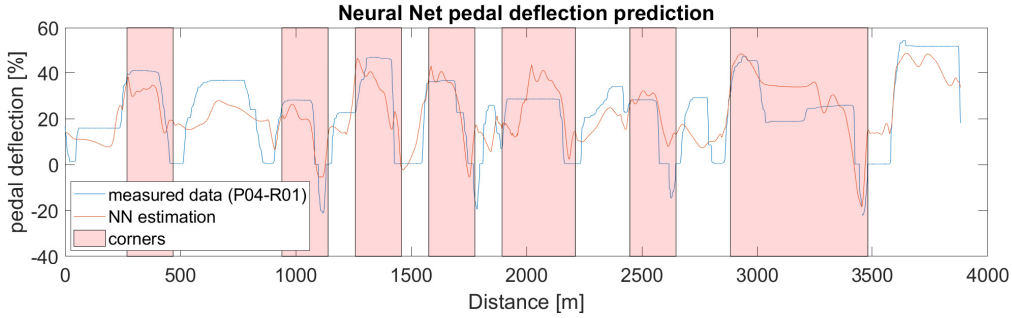


Figure 6.27: Pedal deflections (δ_a & δ_b) prediction

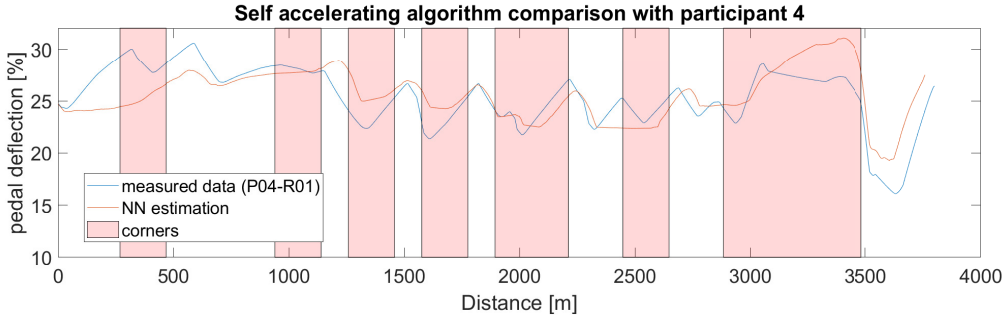


Figure 6.28: Integrated and iterated velocity profile from predicted acceleration, compared with measured data

The main finding in the results is the difference in performance between data inside a curve and on straight road segments. The reason for the poor performance of the network within a curve could be due to the fact that a certain visual point is missing, or due to the fact that the velocity determination is not fully dependant on visual points only. As discussed in the work from Reymond et al [15], lateral acceleration has an important role in velocity determination. Even if lateral acceleration could be introduced indirectly by for example visual flow rates, it is still difficult to model the surprising velocity profile. A driver tends to slow down while entering the curve up until a point where it feels comfortable. After this point, an acceleration is found

since the driver feels more and more confident while driving through the curve. if the curve is long enough, there should be a point where the acceleration stops. This point is however not found in the data generated in this experiment. In the new experimental set up there should be longer curves in order to test this hypothesis.

6.5.3. Variability

The desired outcome of this research would be a general model that would be sufficiently accurate to represent all humans. When human behaviour is too inconsistent this will not be possible. The variability between participants and within one participant is researched in this subsection. An example of an experiment is shown in Figure 6.29. From left to right the model input for training changes from participant 1 to 15. From up to down, the model remains fixed but the participant with which the model is tested changes from 1 to 15. The output of the model is the vehicle velocity. The diagonals are kept empty since in this table training is done with road 1, 2, 5 and 6 while testing is done only with road 1. The diagonals would not give a accurate representation since training and testing data would overlap. The numbers inside the table are the coefficients of determination (or R^2).

		TRAIN MODEL WITH PARTICIPANT															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AVG
	1		0.31502	0.51348	0.65525	0.44525	0.70478	0.63359	0.54342	0.09082	0.6537	0.20239	0.66978	0.38831	0.71791	0.18515	0.47992
	2	0.33689		0.31432	0.55807	0.53935	0.39587	0.67688	0.44474	0.40558	0.45885	0.38753	0.60315	0.5969	0.40167	0.50096	0.47291
	3	0.83798	0.6856		0.82571	0.83717	0.72903	0.81853	0.85442	0.30114	0.88565	0.33234	0.9115	0.72693	0.7092	0.37738	0.70233
	4	0.82411	0.58073	0.5457		0.58551	0.65228	0.72678	0.38668	0.14296	0.7828	0.22796	0.78984	0.58264	0.77736	0.28711	0.56375
	5	0.60292	0.66319	0.72258	0.7119		0.59463	0.68864	0.59916	0.44101	0.8336	0.47265	0.72625	0.74942	0.44839	0.31699	0.61224
TEST MODEL WITH	6	0.73968	0.53764	0.6731	0.72259	0.67753		0.65305	0.68594	0.23654	0.65365	0.28877	0.6853	0.60526	0.57309	0.36697	0.57851
PARTICIPANT	7	0.69109	0.56851	0.41664	0.72461	0.78047	0.57861		0.60584	0.30068	0.75557	0.40163	0.7624	0.66678	0.53167	0.30778	0.57802
	8	0.7217	0.55151	0.53339	0.65401	0.70623	0.6572	0.70774		0.16937	0.68291	0.2353	0.83972	0.61912	0.6924	0.44677	0.58696
	9	0.08776	0.41084	0.1957	0.1886	0.29314	0.05937	0.26944	0.14677		0.19236	0.83361	0.1931	0.32725	0.1265	0.19194	0.25117
	10	0.71556	0.42406	0.59902	0.73512	0.71906	0.72455	0.52831	0.41893	0.20596		0.2874	0.65492	0.57965	0.49697	0.23783	0.52338
	11	0.16853	0.44508	0.24363	0.2202	0.24391	0.20853	0.34885	0.34296	0.60342	0.19366		0.33595	0.38045	0.27925	0.32686	0.31009
	12	0.87922	0.56139	0.56951	0.76951	0.70962	0.80979	0.73544	0.82236	0.19203	0.79176	0.25405		0.61621	0.85761	0.37245	0.63864
	13	0.42047	0.83326	0.58041	0.69354	0.86491	0.41765	0.6412	0.52432	0.52468	0.76265	0.57894	0.67575		0.33119	0.45538	0.59317
	14	0.72524	0.50758	0.35855	0.7016	0.51773	0.73161	0.69823	0.75096	0.15271	0.56799	0.2241	0.73239	0.48177		0.4426	0.54236
	15	0.70987	0.715	0.55099	0.8446	0.64413	0.68631	0.71049	0.80589	0.21167	0.75237	0.32057	0.84368	0.68224	0.75783		0.65969
	AVG	0.60436	0.5571	0.48693	0.64324	0.61171	0.56787	0.63123	0.5666	0.28419	0.64054	0.36052	0.67312	0.57164	0.55007	0.34401	0.53954

Figure 6.29: R^2 values when training with road 1, 2, 5 and 6

From this table is becomes clear that when training the model is done with data from participant 9, 11 or 15, a model is created that does not fit well with the 12 other participants. The same is seen for participant 9 and 11 when their data is used for testing the models produced by the other participants. Surprisingly, when the data from participant 15 is used for testing, a relative high R^2 value is found. This indicates that data from this participant is bad for producing a model, but still fits reasonable in another model. In the work from Gruppelaar, he concludes that the behaviour of participants 6 and 9 are significantly different from the rest of the people. The variability of participant 6 was not found to be as high as was expected (R^2 value above average for testing and producing a model). However, participant 11 did show a high variability. When looking at the parameters from the model created by Gruppelaar [25], also participants 11 and 15 show to be significantly different from the rest of the participants. participant 6 does not show that much difference. The K-values and T-values are gains and safety margins respectively. In Figure 6.30 it can be seen that participant 9, 11 and 15 show most difference compared to the other drivers. An orange circle indicates the value is between 1 and 2 standard deviations, a red circle indicates that the value is outside 2 standard deviations.

A similar pattern is found when looking at the Mean Squared Error of the residuals. Again training with data from participants 9, 11 and 15 results in large MSE values relative to the other models (see Figure A.1). Another test was done by training the neural network with data collected on road 2, 5 and 6 while the model would be tested with data from road 1, a road the model has not yet seen before. The results are shown in Appendix A. Logically, the performance is slightly worse compared to models trained with data from road 1. Also, performance is better on the diagonals (mean R^2 = 0.65), indicating people are less variable with themselves compared to with other people. When even more training roads are deleted, performance drops more. In Figure A.4 and A.5, results are shown when the model is trained solely with data from roads which are 2.4m wide (road 5 and 6), while testing is done with a road width of 3.6m. Finally, a model is produced with data from all participants except numbers 9, 11 and 15 due to previously found results. The model is trained with data from road 2, 5 and 6 and again tested with data from road 1. The results are shown in Figure 6.31. Removing the 'outlier' participants for training, results on average in a much higher R^2 value and lower MSE.

Subject	T_d [s]	T_b [s]	K_d [-]	K_b-	K_a [-]	dT_a [-]
1	3.7780	3.0710	0.9989	5.1329	1.0031	-0.2500
2	4.8200	2.1230	0.9923	4.9843	0.9988	-0.2495
3	4.3380	2.8460	0.1649	4.1200	1.0027	-0.2480
4	3.8540	2.7940	0.8327	4.7101	0.9988	-0.2487
5	4.6340	3.5960	0.1579	4.3751	0.9930	-0.2496
6	7.1480	3.9040	1.0440	4.6086	0.9916	-0.2497
7	5.6200	3.8060	0.0595	4.9297	0.9983	-0.2500
8	3.2780	2.4930	0.5274	4.7574	0.9910	-0.2543
9	8.3930	6.2850	0.2233	4.9901	1.0001	-0.2500
10	4.8890	2.6760	0.7218	5.2828	1.0269	-0.2408
11	4.7600	2.2770	2.8536	0.9874	0.8727	-0.0118
12	3.9150	1.2150	0.0450	4.8248	0.9969	-0.2501
13	5.1480	4.7130	0.7592	4.9988	1.0000	-0.2500
14	3.4920	2.6490	0.1816	2.9510	0.9552	-0.2462
15	3.0750	1.9780	0.1936	0.5795	0.8721	-0.1391
Mean	4.6673	3.2597	0.6504	4.1488	0.9801	-0.2259
StdDev	1.4342	1.1286	0.7130	1.4776	0.0460	0.0657

Figure 6.30: Parameters from the model of Gruppelaar for each participant [25]

Without the outlier participants, it might eventually be possible to produce a model which reveals the general trends of human behaviour in curve driving.

TRAIN MODEL WITH ALL PARTICIPANTS EXCEPT 9, 11, 15	PAR	MSE	R2
	1	1.55292	0.67846
	2	1.31376	0.59856
	3	0.93591	0.86189
	4	0.75723	0.85633
	5	1.33121	0.76749
	6	1.31538	0.74834
	7	1.01296	0.79023
	8	1.24692	0.74358
	9	3.14299	0.21103
	10	1.19317	0.7496
	11	4.03334	0.36619
	12	1.14499	0.83868
	13	1.44539	0.6724
	14	0.99804	0.7785
	15	0.89757	0.83802
avg		1.48812	0.69995
avg(no 9&11&15)		1.18732	0.757

Figure 6.31: MSE and R^2 value for each participant when training the model with all participants except the outliers

7

Thesis Experimental Design

In this chapter the thesis experiment will be designed. In order to do so, the results from the literature study and experimental research will be used. A set of experimental scenarios will be created so that the research questions from chapter 1 can be answered. The number of participants needed to achieve proper results will be discussed in section 7.3. The data will be collected using the fixed-base driving simulator in the HMI lab at Delft University of Technology (section 7.4). In order to make sure that the experiments run smoothly, a sequence is created which the participants should follow. This sequence will result in familiarisation with the simulator dynamics but not with the road corner sequence. A detailed description is given in section 7.5. Before any data is collected it is also important to have a proper plan of which data to collect and how to process/analyse it. This plan is given in section 7.6. Based on the results from chapter 6 and the literature study, a hypothesis is formulated (see section 7.7).

7.1. Experimental Scenarios

In order to answer the research questions from chapter 1, the following scenarios are needed in the thesis experiment:

- One participant, one corner deflection, varying radius: to research the influence of a varying radius on visual cues and driving behaviour
- One participant, one radius, varying road heading deflection: to research the influence of a varying deflection on visual cues and driving behaviour
- One participant, one corner, varying road width: to research the influence of a varying road width on visual cues and driving behaviour
- One participant, one track, multiple runs: to research the variability/inconsistency within human behaviour
- Multiple participants, one track: to research the variability between participants.

When analysing the data throughout the experimental track of the preliminary research the velocity as a function of time almost never seems to be (reasonably) constant on the straight parts of the road. It seems that the corners are significantly close together so that before reaching the end of accelerating to the maximum speed, decelerating for the next corner is already required. It could be that these corners close enough together influence the driving behaviour as opposed to separate corners. Therefore, the design of this experiments will be with corners further apart so that each corner can be analysed solely too. The vehicle in the simulation will have the maximum allowable velocity when approaching a curve, and thus also accelerate towards this speed after exiting a curve.

other interesting scenarios to investigate would be the influence of road super-elevation in curves, and the impact of driving in a climb/descent. The way the visual cues are derived will change in this case, since a 3 dimensional model needs to be made instead of a 2 dimensional model.

7.2. Experimental Track

The experimental track has to be designed in such a way that the answers to the research questions will be clear and valid. The following variables will be tested on the experimental road:

- Radius = [50 100 200 300] (meters)
- Deflection = [30 60 90 120] (degrees)
- Direction = [L R] (-)
- Width = [2.8 3.2 3.6] (meters)

The widths of the road are based on the average rural road widths investigated by the SWOV Institute for Road Safety Research in The Netherlands [34]. The deflections and radii are based on the results from other research, and the preliminary experiment. If a test track has all combinations of radii, deflections and directions, it will have 32 corners. each corner will have a 400 meter straight part after it so that there will be no influence on the next corner. For this research it is important to investigate the use of the visual field only. A participant should not become accustomed with the test track since in that case not only the visual field is investigated, but also the influence of familiarisation. Therefore, the sequence of test track corners should always feel random for the participant. A random road generator is designed which produces a road which includes all linear combinations of corner variables which are given as input. The road will never cross itself, and each corner is randomly placed. Some example roads with the inputs given above can be found in Figure 7.1.

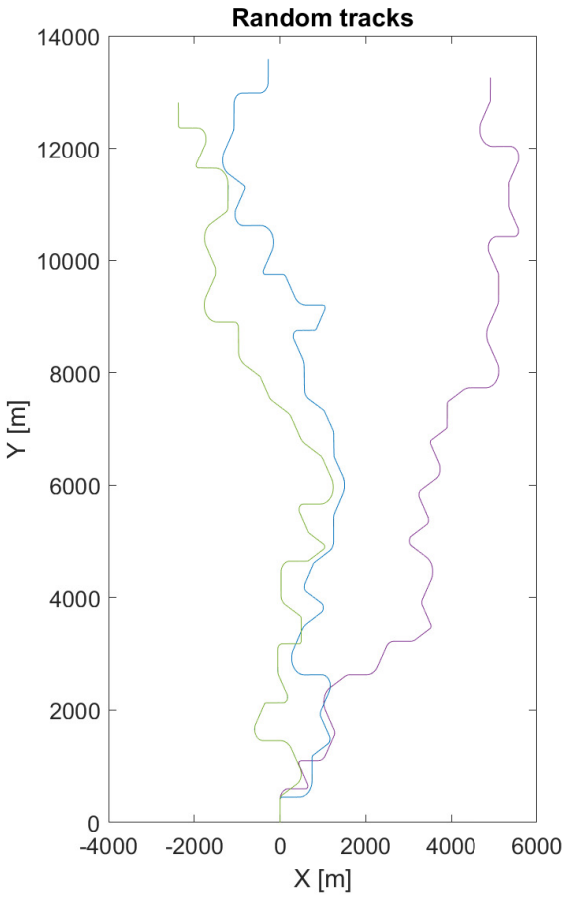


Figure 7.1: Randomly created test tracks with same corners

This track has a width of 3.6m, but roads of any width can be produced. The length of the road in this example is around 20 km, which will take the participant about 15 minutes to complete. The test track can also be split in two parts, in case the 15 minutes leads to weariness during a test drive.

Test tracks will also be used for validation. This will be done in a few different ways. First of all, randomly selected data points from the training data will be eliminated and stored as test or validation data. With this data the model results will be tested for over-fitting. Secondly, a test track with corners similar to tracks which are used for training will be held apart. This track is never been seen by the neural net and will be used to validate the model. Finally, a track with different corners compared to track for training will be held apart for validation. This way, the model can be validated to work as a general model for all corner radii and deflections (within range). In this last validation part, some corners will also be put closer together in order to create a more realistic road.

7.3. Participants

It is important to have the right amount of participants for an experiment. When too little participants are used, the change of significant results drops. When too many are used, much unnecessary time is wasted on collecting and analysing data. Also, the research facilities will be occupied longer, delaying other research that needs to use the same facilities. In order to find the amount of participants needed, a power analysis will be done. Generally speaking, there are two things that need to be considered. The state of the actual world and the findings from the study.

Table 7.1: Real world versus study finding: conclusions

		Study Finding	
		Effect present	Effect absent
Real World	Effect present	Reject H_0	False Negative (type II)
	Effect absent	False Positive (type I)	Retain H_0

There exist two types of errors, a type I and a type II error. A type I error is also known as a False Positive (FP). In this case the research results pick up an effect or correlation which does not exist in real life. The chance of a type I error occurring is represented by the α-value. An α-value of 0.05 indicates a 5% chance of wrongly rejecting the null-hypothesis. One could argue that it would be favourable to have α as low as possible. However, decreasing α will also lower the chances of detecting a true effect, hence increasing the probability of a type II error. The type II error is known as a False Negative (FN). In this case the null-hypothesis is retained while in real life this is not the case. The probability of a type II error is the β-value. This value is related to the power levels of a statistical test (power = 1 - β). It is possible to increase the power level, hence decreasing the probability of a type II error, by increasing the sample size.

The next important factor is the effect size. Where α only indicates whether an effect is statistically significant, the effect size indicates the magnitude of this effect. The effect size represents the correlation of the effect that is studied. It is also known as the square root of the coefficient of determination ($R = \sqrt{R^2}$). From the results of the preliminary experiment an average R value of 0.708 was found when comparing model and real life output from data runs never seen by the neural network. This included outlier participant behaviour (S9, 11 and 15). When the model is made without outlier runs, the R value increases to 0.83.

To compute the required sample size such that the α and β input requirements are met using data with an effect size R, a software called G*Power is often used. This software is developed by the Heinrich-Heine-Universität in Düsseldorf, Germany [27]. This software can be used for statistical power analysis. The type of statistical power analysis that is used before the study is done is the 'A priori analysis' (Bredenkamp, 1969; Cohen, 1988): *"The necessary sample size is computed as a function of user-specified values for the required significance level α, the desired statistical power $1-\beta$, and the to-be-detected population effect size."* [27].

However, in the case of the experiment that will be done in this research, this software can not be used. The simulator collects data in time series for multiple participants. The sample size will increase when more participants are used, but also when more types of corners are put into a certain track, or when a track is driven more often. It is very difficult if not impossible to determine the number of participants using G*Power with all these variables. Therefore another approach is taken to determine the minimum number of participants. When looking at the variability table (Figure A.1), a few outlier drivers can be identified. The histogram of the coefficients of determination (R^2) is shown in Figure 7.2.

The distribution of R^2 has a mean value of 0.501 and a standard deviation of 0.191. One standard deviation below the mean equals therefore 0.310. The number of occasions with a coefficient below 0.310 equals 44 and the total number of measurements equals 225 (15^2). This means that 19.56% of the coefficients have a value below 0.31. These values are almost solely divided over 3 participants (number 9, 11 and 15) that make up 20% of all participants. Therefore, one could say that the variability of these three participants is higher compared to the rest of the group. In this thesis it will be important to be able to say with a specific certainty that the outlier participants can be detected. In this case it would be preferable if it is possible to say with at least 95% certainty that the number of outlier participants is lower than 40%, which is a clear minority. For this a derivation is done. The number of combinations that exist for a group of N people including an X number of outliers equals:

$$C_{N,X} = \frac{N!}{X!(N-X)!} \tag{7.1}$$

Assuming the data to find the variability is correct, 20% of the participants are considered 'outliers' ($P_o = 0.2$). Therefore the chance of having more than 40% of the participants being an outlier equals:

$$chance = \sum_{X=ceil(0.4\cdot N)}^{N} C_{N,X}\left(P_o^X \cdot (1-P_o)^{N-X}\right) \tag{7.2}$$

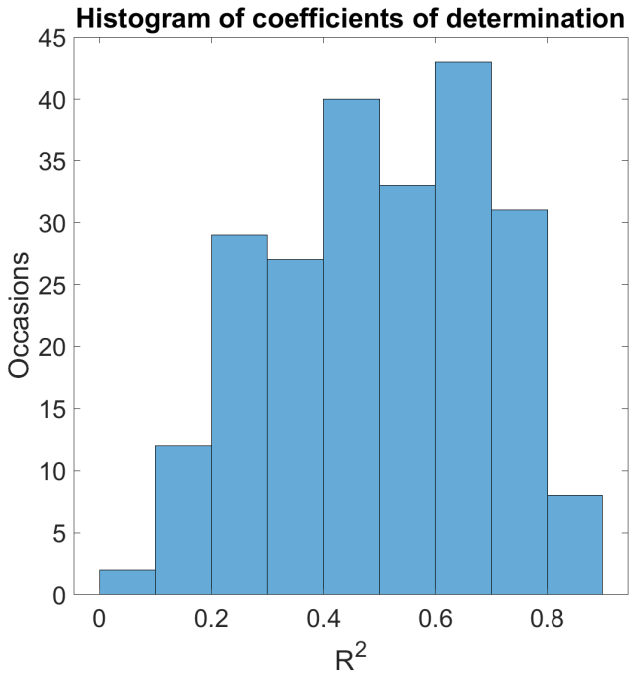
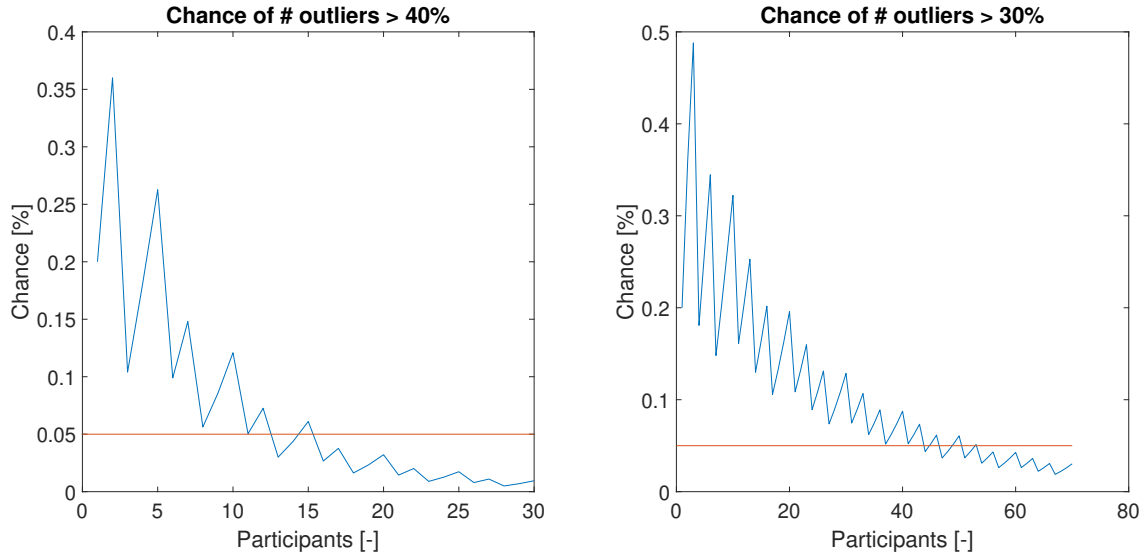


Figure 7.2: Histogram of coefficients of determination

This value will decrease with an increasing number of participants. In order to achieve the 95% certainty, this value should drop below 0.05. As can be seen in Figure 7.3a, for this 16 participants are required. This seems like a realistic number of participants in the scope of this research. When for example a certainty of 95% is required with number of outliers below 30%, much more participants (50+) would be needed (see Figure 7.3b)



(a) (b)

Figure 7.3: Chance of outlier participants above X% versus required minimum

7.4. Apparatus

The experiments will be done in the fixed-base simulator located in the Human-Machine Interaction (HMI) laboratory of the Aerospace faculty at Delft University of Technology. A detailed explanation of this simulator can be found in section 4.4. As explained in chapter 3, lateral acceleration does play an important role in curve

driving. The results of this experiment will therefore probably deviate from real life experiments. However, even though driver behaviour deviates compared to real life, it is still possible to capture driving behaviour based on visual cues. If there happens to be a significant correlation between visual cues and behaviour, more testing could be done in dynamic simulators and actual vehicles. The participants will be able to control the steering wheel and the gas/brake pedals. Furthermore, the road will have some scenery next to it in order increase realism and to give the participant an indication of speed without looking at the speedometer.

Figure 7.4: Fixed based driving simulation (HMI lab), different elements [33]

7.5. Sequence

To start with, each participant gets a few runs on the simulator in order to get familiar with the vehicle dynamics. This will be done on a road that will not be used for collecting the actual data. After being familiar with the simulator, the participant is asked to drive three different test tracks. These tracks will all have the same corners but the road width is varying. Also, for each track, the corner sequence is different. This is done in order to eliminate two effects as much as possible.

The first effect is getting familiar with the corner sequence, which should not happen. When this happens, driving behaviour will not be depending solely on visual cues. The second effect is the influence of corner sequence itself. For example, when multiple large radius corner are in front of a small radius corner, the participant might have a different judgement on the upcoming corner as opposed to a situation where the corners in front all have a small radius. Changing the test track randomly, increases the variability of corner sequence, and hence decreases the possibility of unwanted influences on the data. Note that each participant should drive the same test tracks. This way a better comparison can be made between each participant.

Finally, each participant is asked to drive validation runs. One run includes the same type of corners as the tracks driven before and one run will include a completely new set of corners. These validation runs may be much shorter as compared to the runs for model development.

Driving in a simulator can lead to simulator sickness. Symptoms of simulator sickness include discomfort, apathy, drowsiness, disorientation, fatigue, and nausea. During each run, the participant will be asked how he/she is feeling once every minute. The participant can name a number between 1 and 5. 1 means that the participant feels very sick and 5 means feeling very good. Once this number drops to 2 (below a neutral feeling), the simulation run will be stopped immediately in order to minimize the chances and magnitude of simulator sickness.

7.6. Data

7.6.1. Measurements

The most important variables that will be measured during the experiments are the following:

- Time (t) [s]
- Longitudinal distance (global plane of reference) (Y) [m]
- Lateral distance (global plane of reference) (X) [m]
- Longitudinal velocity (local plane of reference) (V_{long}) [m/s]
- Lateral velocity (local plane of reference) (V_{lat}) [m/s]
- Longitudinal acceleration (local plane of reference) (α_{long}) [m/s^2]
- Lateral acceleration (local plane of reference) (α_{lat}) [m/s^2]
- Yaw angle (ψ) [rad]
- Yaw rate ($\dot{\psi}$) [rad/s]
- Accelerator pedal deflection (δ_a) [%]
- Brake pedal deflection (δ_b) [%]
- Steering wheel angle (δ_s) [rad]

Next to these, there will also be more measurements done. Some of them can be used for validation of visual inputs, but would not be available in the actual vehicle (e.g. curve radius).

7.6.2. Analysis

The data will be analysed in different steps. First, all different corners will be split so that they can be analysed separately if necessary. Also, the straight and cornering parts of the road will be split for the same reason. After this, mathematical expressions for the distances, times, angles and their derivatives towards the following visual points will be derived:

- TP
- ETP
- LCP
- SLCP

With all variables derived, separate corner types will be investigated. The combined effects that road width, radius and deflection have on the outputs (velocity and pedal deflections) and visual cues will be checked. With a detailed analysis on this it will become easier in the future to select the right visual inputs needed to compute the desired output. When inputs turn out to be desirable that are not in the visual field directly, it will be examined if such a cue also can be derived in the visual field.

After this, linear combinations of inputs will be tested to produce various models. These will be models of velocity and pedal deflections. When the best working linear combinations of inputs are found, they will be used for further analysis. Participants will be analysed separately in order to check if there any 'outlier participants' that show very different behaviour compared to the rest. The results from these participants will not be taken into account for developing a general model.

Models will be developed for each individual. These models will be tested and validated with the validation runs from that driver and other drivers. This way the variability between people can be analysed. In the end, a general behavioural model will be made with data from all participants. With this model it can be checked whether it is possible to capture the general trends of human driving behaviour within one model that can be

used for all people.

The main model will have a similar form as used in the work from Gruppelaar [25]. This model will predict the pedal deflections based on the visual inputs. These deflections can then be transformed into an acceleration which will be integrated to a velocity. The difference between this thesis and the work from Gruppelaar is that more visual cues will be tested in order to try to obtain a more accurate results. The model will be a trained neural network instead of a set of pre-determined relations.

7.7. Hypotheses

The main research objective is to capture human behaviour in curve driving in a model using a neural network with visual inputs. In order to work towards this objective, several smaller questions need to be answered. Based on the literature and preliminary experiment, the following hypotheses are formulated:

- *A combination of distances to visual points represent road geometry best:*
  When testing the visual cues available at the moment as inputs to the neural network, the road curvature was predicted best when distances to visual points (TP, ETP, LCP) were used. Accelerations were not tried to be predicted yet, since they have not yet shown any improvement in the model as input.

- *There is a correlation between visual cues and speed profile in curve driving:*
  The highest correlation will be found with distances to specific visual points. This correlation is found to be stronger in the straight parts of the road as opposed to the cornering parts. Also, this correlation varies based on the data that is used for training and testing. When data is used from one participant only, the model correlation coefficient is estimated to be between 0.9 and 1. When more people (10+) are used, this correlation is estimated to be between 0.8 and 0.9. The reason for this is that the variability is higher between people than for one participant. The correlation coefficient will also drop when the test data is from a test run that is never been seen by the model before.

- *There exist a correlation between visual cues and moment of braking/accelerating:*
  The correlation is highest when times to specific visual points are used as inputs. This correlation is stronger for braking than for accelerating. Also, the moment of braking/acceleration can be modeled quite well, but the magnitude of pedal deflection not yet. However, when iterating and integrating the pedal deflection to a velocity, the model does seem to capture general acceleration and deceleration trends well. It is also not the goal to capture the pedal deflections magnitude perfectly since this is also very variable in human behaviour.

- *Variability between people is larger than for one participant:*
  When investigating the variability table, lower R^2 values are found when training and testing is done with data from different participants. Also, some 'outlier' participants can be present which have a different driving behaviour compared to the rest of the group.

- *When curves are close together, they will influence each other:*
  When curves are close together, a driver will be unable to accelerate at the end of the first curve as would be done without a curve ahead. Also, the driver will enter the second curve differently if it would have been approached by a straight road segment only. This is an hypothesis based on the data from Gruppelaar [25]. In the actual thesis experiment the curves will be further apart for training, while in the validation data also corners will exist which are close together. At these positions this hypothesis will be investigated. If the produced model will still work ate these positions, corners closer together do not influence the model.

8

Conclusion

The goal of the preliminary thesis was to collect enough information in order to be able to design a well defined thesis experiment. This information is both obtained through the use of available literature and a preliminary experiment. The research should be designed in such a way that the following researched objective will be met:

"Measuring and modeling drivers behaviour for speed adaptation in curves using neural networks with visual field inputs"

In the literature study, much research was found on modeling driver behaviour in curve driving. But still, there is a lot of variation between the methods used for modeling. Most research was based on relating speed to road geometry (radius and width). A flaw in this modeling is that often curves have a fixed radius and width, while the velocity through this curve does not remain constant. As a human, it is also very difficult to predict the radius of an upcoming curve, yet humans (usually) seem to drive safely and comfortably through curves. The main cue that people have available during driving is their visual field. There are specific points in the visual field where drivers seem to look most often such as the Occlusion Point and the Extended Tangent Point. Speed control is most likely governed by time margins to these points.

In the preliminary experiment, different types of models were produced. The distances to visual points best represent road geometry. It makes sense that therefore these same inputs also has the highest correlation to velocity directly. Strong relations between in- and output are found on straight road segments (driving towards a curve). These relations were weaker on curved road parts since also the distances to these points remain relatively unchanged. When predicting the brake and acceleration pedal deflections, the most promising inputs are time values towards specific visual points. The moment of braking and accelerating was found with high accuracy, while the magnitude was more difficult to predict. This might not be a problem since human behaviour is quite variable itself. When iterating the pedal deflections to a velocity profile, the model captures most acceleration and deceleration trends.

The main plan for the thesis is to collect data from about 16 participants driving roads with varying width and curve radii. The curves will be far enough apart so that they can be analysed separately. The model that will be trained will have as output the pedal deflections. These will be iterated and integrated to a velocity profile for validation. There are a few advantages when using this method. Firstly, no unrealistic accelerations will be modeled since deflections are always between 0 and 100%. Secondly, the pedal deflection inputs are very easy to implement into a vehicle as these are the actual inputs of the system. Finally, this method also shows promising results on the whole track, instead of only on the straight road segments.

Bibliography

[1] National Highway Traffic Safety Administration. The evolution of automated safety technologies. https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety.

[2] BOSCH. Automated mobility. https://www.bosch-mobility-solutions.com/en/highlights/automated-mobility/, 2020.

[3] J. de Villiers & E. Barnard. Backpropagation neural nets with one and two hidden layers. 1992.

[4] C. de Visser. Lecture notes ae4320 system identification. 2019.

[5] TU Delft. The simona research simulator. http://cs.lr.tudelft.nl/simona/facility/.

[6] A. K. Yadav et al. Safety problems in vehicles with adaptive cruise control systems. *AGH University of Science and Technology, Krakow, Poland.*, 2017.

[7] A.M.C. Odhams et al. Models of driver speed choice in curves. *Cambridge University*, 2004.

[8] B.N. Fildes et al. The effect of road curve geometry on curvature matching judgements. 12:63–70, 1984.

[9] D. Zhang et al. Driver curve speed model and its application to acc speed control in curved roads. *International Journal of Automotive Technology*, 2013.

[10] D.D. Salvucci et al. A two-point visual control model of steering. 33:1233–1248, 2004.

[11] E. Lehtonen et al. Transportation research part f: Traffic psychology and behaviour. 13:369–377, 2012.

[12] E.R. Boer et al. To brake or not to brake: Scaling the curve. *Delft University of Technology*, 2002.

[13] F. Jimenez et al. Measurement uncertainty determination and curve-fitting algorithms for development of accurate digital maps for advanced driver assistance systems. *University of Madrid*, 2008.

[14] G. F. B. Piccinini et al. Reaction to a critical situation during driving with adaptive cruise control for users and non-users of the system. *Chalmers University of Technology*, 2014.

[15] G. Reymond et al. Role of lateral acceleration in curve driving: Driver model and experiments on a real vehicle and a driving simulator. *Renault & Collège de France*, 2001.

[16] J. Xie et al. A personalized curve driving model for intelligent vehicle. 2017.

[17] J.W. Lee et al. A unified framework of adaptive cruise control for speed limit follower and curve speed control function. Technical report, SAE International, 2013.

[18] L.H. Xu et al. A new lane departure warning algorithm considering the driver's behavior characteristics. *South China University of Technology*, 2015.

[19] M. Park et al. Development of steering control system for autonomous vehicle using geometry-based path tracking algorithm. 37:617–625, 2015.

[20] M.F. Land et al. Which parts of the road guide steering? *University of Sussex*, 1995.

[21] P. Bosetti et al. On the human control of vehicles: an experimental study of acceleration. 6:157–170, 2013.

[22] R. Gallen et al. Vision based tangent point detection algorithm, evaluation and validation. 2009.

[23] R. Labayrade et al. A reliable and robust lane detection system based on the parallel use of three algorithms for driving safety assistance. *IEICE*, 2006.

[24] S. Cafiso et al. A new approach to define continuous speed profile models for two lane rural roads. 23: 1–22, 2012.

[25] V. Gruppelaar et al. A perceptually inspired driver model for speed control in curves. *TU Delft*, 2018.

[26] W.V. Winsum et al. Speed choice and steering behaviour in curve driving. 38:434–441, 1996.

[27] Franz Faul, Edgar Erdfelder, Axel Buchner, and Albert-Georg Lang. Statistical power analyses using g* power 3.1: Tests for correlation and regression analyses. *Behavior research methods*, 41(4):1149–1160, 2009.

[28] S. Glaser. Speed limitation based on an advanced curve warning system. *IEEE Intelligent Vehicles Symposium*, 2007.

[29] G. Gil Gómez. Towards efficient vehicle dynamics development: From subjective assessments to objective metrics, from physical to virtual testing. *KTH, Royal Institute of Technology*, 2017.

[30] J. Johansson. Specify the curve speed adaption parameters within the adaptive cruise control system. *Chalmers University of Technology*, 2020.

[31] Delft Haptics Lab. Fixed-base driving simulator. `https://delfthapticslab.nl/device/fixed-base-driving-simulator/`.

[32] T.B. Lee. Waymo is way, way ahead on testing miles—that might not be a good thing. `https://arstechnica.com/cars/2020/01/waymo-is-way-way-ahead-on-testing-miles-that-might-not-be-a-good-thing/`, 2020.

[33] F.N. Postema. Hmi lab. `http://cs.lr.tudelft.nl/facilities/hmi-lab/`.

[34] C.C. Schoon. Road design standards of medians, shoulders and verges. *SWOV Institute for Road Safety Research, The Netherlands*, 1995.

[35] C.G. Serna. Dynamic speed adaptation for path tracking based on curvature information and speed limits. *University Bourgogne Franche-Comté*, 2017.

[36] J.J. Slob. State-of-the-art driving simulators, a literature survey. *Eindhoven University of Technology*, 2008.

[37] Waymo Team. Where the next 10 million miles will take us. `Wherethenext10millionmileswilltakeus`, 2018.

[38] Tesla. Autopilot and full self-driving capability. `https://www.tesla.com/support/autopilot`, 2020.

A

Variability tables

		TRAIN MODEL WITH PARTICIPANT															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AVG
	1		2.798061	1.930685	1.709302	2.274557	1.2801	1.829336	1.884306	4.62185	1.660587	4.115734	1.595151	2.554939	1.279013	3.34509	2.348479
	2	2.341457		2.55678	1.284705	1.326131	2.402039	1.151912	1.742886	2.475973	1.711402	2.440969	1.33235	1.198385	2.149477	1.403923	1.822742
	3	1.166076	1.806958		1.123232	1.140535	1.487146	1.1283	0.968235	3.70299	0.880105	3.386714	0.752445	1.533059	1.413731	2.441908	1.63796
	4	1.328049	1.514183	1.988686		1.45328	1.763794	1.193701	2.274827	3.244115	1.097195	2.947327	0.965102	1.491229	1.177041	2.161114	1.757117
	5	1.884904	1.812741	1.554641	1.483849		1.971018	1.510507	1.700797	3.30714	1.110402	3.037163	1.409551	1.447086	2.095555	2.59599	1.922953
TEST MODEL WITH	6	1.155866	2.360076	1.431637	1.47471	1.708922		1.671296	1.346928	4.600781	1.687773	4.396785	1.430222	2.143083	1.375081	2.903346	2.120465
PARTICIPANT	7	1.522461	1.583159	2.176309	1.139136	1.036449	1.767003		1.459567	3.313061	1.147482	3.09949	1.090929	1.387114	1.7333	2.195409	1.760776
	8	1.407721	1.928491	1.779957	1.476382	1.429364	1.516018	1.460168		3.725035	1.408058	3.633998	1.001572	1.770803	1.35678	2.177567	1.86228
	9	4.505663	2.381169	3.888517	3.249775	3.107434	5.343821	2.993455	3.60351		3.714126	1.031178	3.33234	2.677221	4.313547	2.869258	3.35793
	10	1.26564	2.306707	1.631131	1.337169	1.38918	1.278036	1.78509	2.108811	3.996077		3.797521	1.400822	1.88974	1.599143	2.806146	2.04223
	11	5.996806	2.959472	5.037377	4.441675	4.577924	7.075359	4.246398	4.82136	1.601823	5.035147		4.112038	3.380526	5.730692	3.460429	4.462645
	12	0.928131	2.376754	1.895424	1.519199	1.672933	1.113999	1.647963	1.218935	4.387681	1.297083	4.089255		2.093984	0.939664	2.907879	2.006349
	13	2.499428	0.846807	2.202655	1.297923	0.924615	2.723579	1.461276	1.787407	2.123993	1.406942	1.969761	1.458931		2.545288	1.599676	1.774877
	14	1.257694	1.921781	2.094656	1.214263	1.692032	1.194061	1.356334	1.058728	3.862337	1.51636	3.564088	1.123155	1.921877		2.185209	1.85447
	15	1.730701	1.279874	2.018229	0.875289	1.427376	2.038217	1.331921	1.068958	3.043964	1.239758	2.639915	0.897332	1.343579	1.534329		1.60496
	AVG	2.070757	1.991159	2.299049	1.687615	1.797195	2.353871	1.769118	1.931804	3.429059	1.779459	3.153564	1.564424	1.916616	2.08876	2.503782	2.155749

Figure A.1: MSE values when training with road 1, 2, 5 and 6 and testing with road 1

		TRAIN MODEL WITH PARTICIPANT															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AVG
	1	0.849178	0.190191	0.518771	0.420543	0.501569	0.662046	0.396678	0.433025	0.093419	0.364876	0.230527	0.343193	0.431184	0.720264	0.186793	0.422817
	2	0.333231	0.587608	0.424246	0.564874	0.349868	0.46815	0.705379	0.263998	0.441251	0.353662	0.281826	0.562794	0.477621	0.2719	0.499262	0.439045
	3	0.835742	0.484965	0.842658	0.764876	0.822659	0.64557	0.783509	0.726244	0.287316	0.797004	0.419929	0.587879	0.804533	0.635407	0.316659	0.65033
	4	0.746562	0.36571	0.578849	0.629623	0.518671	0.65975	0.556753	0.356101	0.128011	0.220152	0.19492	0.430647	0.59836	0.657426	0.262565	0.460273
	5	0.68785	0.495806	0.738856	0.739437	0.661986	0.602728	0.770084	0.486592	0.429056	0.790207	0.463684	0.69488	0.662189	0.373129	0.245565	0.58947
TEST MODEL WITH	6	0.747945	0.442501	0.720066	0.614585	0.601088	0.772651	0.676353	0.478641	0.218547	0.429243	0.374539	0.492228	0.565672	0.653856	0.260092	0.536534
PARTICIPANT	7	0.694636	0.375065	0.477213	0.548791	0.778018	0.615606	0.635806	0.452484	0.282682	0.557601	0.466732	0.60565	0.543944	0.37741	0.296526	0.513878
	8	0.717162	0.361798	0.579573	0.55955	0.642042	0.648976	0.603311	0.692889	0.218458	0.568007	0.280014	0.500034	0.656309	0.623751	0.274971	0.528456
	9	0.101229	0.448751	0.174075	0.256029	0.249706	0.108315	0.378768	0.186227	0.839572	0.336826	0.610808	0.541561	0.265941	0.030584	0.47975	0.333876
	10	0.726839	0.294232	0.567692	0.526367	0.671086	0.699951	0.463372	0.339297	0.223357	0.42785	0.326869	0.436987	0.550966	0.471724	0.172491	0.459939
	11	0.100375	0.432888	0.20941	0.192152	0.313348	0.255167	0.361486	0.229554	0.636098	0.246218	0.309836	0.472982	0.366559	0.188121	0.482353	0.31977
	12	0.828027	0.376082	0.595388	0.643823	0.71967	0.775189	0.62044	0.757821	0.26285	0.645013	0.410911	0.508553	0.689122	0.748774	0.254251	0.589061
	13	0.44494	0.673535	0.559563	0.668348	0.742418	0.430965	0.710033	0.463853	0.423378	0.664852	0.714922	0.839347	0.753575	0.226224	0.437396	0.583556
	14	0.68691	0.359773	0.452463	0.540105	0.57112	0.712524	0.565916	0.711654	0.209077	0.401578	0.33254	0.361659	0.515281	0.707462	0.312767	0.496055
	15	0.642681	0.530439	0.560993	0.684783	0.643748	0.764797	0.605034	0.721544	0.23722	0.666191	0.465806	0.519058	0.713232	0.615887	0.509894	0.592087
	AVG	0.609554	0.427956	0.533321	0.556926	0.5858	0.588159	0.588862	0.486662	0.328686	0.497952	0.392258	0.526497	0.572966	0.486795	0.332756	0.50101

Figure A.2: R2 values when training with road 2, 5 and 6 and testing with road 1

		TRAIN MODEL WITH PARTICIPANT															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AVG
	1	0.934396	3.219561	1.821201	2.3197	2.154668	1.434065	2.313981	2.183015	4.787607	2.772525	4.008554	2.997429	2.192471	1.244183	3.814267	2.546508
	2	2.152605	1.226077	1.925786	1.197804	1.676585	2.138916	1.06769	2.11121	2.487234	1.977761	2.256016	1.433543	1.591564	2.463265	1.572136	1.818546
	3	1.095265	2.297056	0.987152	1.381621	1.149289	1.777443	1.287887	1.308735	3.777003	1.37179	2.849658	2.165703	1.166325	1.645239	3.103808	1.824265
	4	1.283524	2.005124	1.641849	1.359777	1.542736	1.626782	1.560527	2.426213	3.375343	3.507721	3.007608	2.101978	1.519215	1.660908	2.508213	2.075168
	5	1.614542	2.203425	1.431659	1.413828	1.580618	1.785049	1.303903	1.931794	3.497552	1.266925	2.795689	1.857736	1.588103	2.372949	3.365291	2.000604
TEST MODEL WITH	6	1.142867	2.796397	1.292169	1.775394	2.040892	1.120269	1.618513	1.905227	4.687782	2.338874	3.835207	2.792084	1.75792	1.264769	3.990977	2.290623
PARTICIPANT	7	1.43964	2.071013	1.819547	1.519362	1.056278	1.541695	1.350809	1.846538	3.255782	1.762943	2.807032	1.812223	1.554101	2.155772	2.722099	1.914322
	8	1.354173	2.394736	1.627708	1.761193	1.580415	1.481521	1.698953	1.461915	3.76258	1.83162	3.095544	2.413448	1.505376	1.53373	3.140944	2.042924
	9	4.102877	2.206717	3.802931	2.997392	3.100478	4.932978	2.686064	3.100682	1.041776	3.068018	1.703215	1.937402	3.036614	4.674459	1.893077	2.952312
	10	1.197703	2.679598	1.589493	1.792337	1.480427	1.241284	1.912852	2.367313	4.282146	2.601704	3.608234	2.540146	1.710836	1.727643	3.678322	2.294002
	11	5.680348	2.685021	5.27445	4.378865	4.531803	7.248149	3.756907	4.510258	1.528616	4.305092	2.98633	2.763175	3.722997	5.667776	2.405971	4.096384
	12	1.058512	2.843882	1.774274	1.879141	1.584365	1.255972	1.911259	1.480542	4.624433	1.961984	3.56464	2.746154	1.722243	1.32631	3.95174	2.245697
	13	2.263413	1.181874	1.986601	1.265535	1.194983	2.524064	1.273982	1.797035	2.29933	1.425342	1.372818	0.895323	1.216234	2.82772	1.806487	1.688716
	14	1.244007	2.368225	1.754675	1.66347	1.578287	1.169985	1.6097	1.193898	3.859533	2.025513	3.086741	2.526979	1.689307	1.236111	3.115072	2.0081
	15	1.699946	1.706745	1.93214	1.336101	1.328684	1.813412	1.597193	1.229896	3.008612	1.48459	2.147512	1.978967	1.260816	2.058383	2.026388	1.773959
	AVG	1.884255	2.25903	2.044109	1.869435	1.8387	2.206105	1.796681	2.056951	3.351689	2.246827	2.874986	2.197486	1.815608	2.257281	2.872986	2.238142

Figure A.3: MSE values when training with road 2, 5 and 6 and testing with road 1

		TRAIN MODEL WITH PARTICIPANT															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AVG
	1	0.804097	0.043644	0.219759	0.324015	0.434585	0.695116	0.416659	0.24388	0.076108	0.29877	0.003397	0.237255	0.092915	0.617355	0.092988	0.306703
	2	0.277491	0.309818	0.197274	0.469644	0.186714	0.556414	0.690975	0.598972	0.250526	0.336125	0.039127	0.486088	0.377625	0.248527	0.252375	0.351846
	3	0.84212	0.109621	0.583622	0.569872	0.440823	0.638537	0.731468	0.611909	0.196715	0.653648	0.078647	0.376245	0.278465	0.612892	0.268669	0.466217
	4	0.677143	0.115899	0.373979	0.579322	0.455375	0.629085	0.54741	0.316186	0.061424	0.186837	0.013377	0.354096	0.164527	0.756526	0.163091	0.359618
	5	0.520266	0.177413	0.577266	0.593856	0.338906	0.595203	0.740865	0.417782	0.337569	0.673824	0.097172	0.50129	0.305494	0.30552	0.27434	0.430451
TEST MODEL WITH	6	0.670352	0.139566	0.331769	0.544464	0.30573	0.696892	0.48994	0.473903	0.155471	0.539865	0.121306	0.383964	0.186677	0.531805	0.238924	0.387375
PARTICIPANT	7	0.510666	0.127566	0.376167	0.404762	0.507206	0.594073	0.684372	0.489377	0.240991	0.639683	0.053815	0.542055	0.252944	0.446218	0.208148	0.405203
	8	0.682304	0.040373	0.291753	0.371618	0.362829	0.668961	0.544285	0.5623	0.102327	0.732219	0.06844	0.380224	0.098748	0.435599	0.285548	0.375169
	9	0.072524	0.207019	0.305477	0.294725	0.129677	0.082987	0.328692	0.228247	0.642501	0.098872	0.236238	0.275385	0.343333	0.102966	0.588325	0.262465
	10	0.556304	0.090222	0.334335	0.482693	0.406215	0.42049	0.454139	0.219824	0.132484	0.289036	0.009761	0.415886	0.145621	0.370352	0.199174	0.301769
	11	0.143377	0.331794	0.296487	0.203646	0.253316	0.161783	0.357709	0.294878	0.326685	0.209597	0.192271	0.396564	0.48708	0.154551	0.432368	0.282807
	12	0.7511	0.168497	0.417379	0.516509	0.489315	0.735342	0.614047	0.589112	0.145736	0.710051	0.08229	0.439338	0.225378	0.728691	0.2773	0.459339
	13	0.390679	0.183023	0.515362	0.672572	0.385438	0.381309	0.752807	0.600783	0.370751	0.568833	0.142358	0.70764	0.462451	0.235058	0.3893	0.450558
	14	0.592474	0.149352	0.305656	0.467448	0.369496	0.695062	0.486696	0.575339	0.088543	0.588883	0.094151	0.328542	0.162011	0.761039	0.301585	0.397752
	15	0.658519	0.306174	0.373698	0.719287	0.449117	0.585026	0.620391	0.655931	0.092258	0.48819	0.130696	0.571504	0.330879	0.615727	0.362933	0.464022
	AVG	0.543294	0.166665	0.366666	0.480962	0.367649	0.542419	0.56403	0.458562	0.214673	0.467629	0.09087	0.426405	0.260943	0.461522	0.289005	0.380086

Figure A.4: R2 values when training with road 5 and 6 and testing with road 1

		TRAIN MODEL WITH PARTICIPANT															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AVG
	1	1.05881	4.435249	3.158609	2.538252	2.826225	1.336301	2.575015	2.992656	6.274906	3.048984	6.985037	3.544502	4.29306	1.491372	4.484179	3.402877
	2	2.602805	2.442672	2.413248	1.348312	2.81545	2.078058	1.04809	1.104863	5.324153	2.187918	4.631292	1.5868	2.368604	2.224273	2.244458	2.428066
	3	1.376156	4.1136	1.841577	1.887242	2.560527	1.61505	1.577829	1.754	5.523274	1.754634	4.408606	2.708303	3.587585	1.606675	3.281379	2.639762
	4	1.664806	3.042843	2.208085	1.453891	2.149943	1.668453	1.678046	2.172527	5.204142	3.722653	7.541697	2.304304	2.714715	1.172184	3.001208	2.779966
	5	2.180138	3.636143	1.908996	1.785606	2.707519	1.926406	1.439313	2.207577	5.16059	1.634813	4.640155	2.445575	3.204882	2.351154	3.408049	2.709128
TEST MODEL WITH	6	1.354715	4.751851	3.01018	2.110117	3.605584	1.241	2.400251	2.45813	6.918283	2.141609	5.435919	3.422607	4.790122	1.480591	4.187692	3.287244
PARTICIPANT	7	2.043089	3.342471	2.301052	1.841232	2.29017	1.635715	1.336697	1.730661	5.760101	1.566153	5.001313	2.137633	3.091208	1.775502	3.162004	2.601
	8	1.605046	4.005146	2.508332	2.219667	2.477871	1.390686	1.882881	1.844879	5.677454	1.378872	4.116506	2.736917	3.49535	1.833449	3.109827	2.685526
	9	4.786881	2.776934	3.118604	2.697241	3.586492	5.042036	2.833	2.81654	2.39898	4.35035	2.789523	2.505756	2.437195	4.347317	1.68662	3.211565
	10	1.645257	4.158826	2.453779	1.98122	2.539664	1.762955	2.079579	2.730047	6.175754	2.817467	6.911645	2.699029	3.978847	1.824828	3.826221	3.172341
	11	6.554578	2.362932	4.120385	3.851704	5.186879	6.442015	3.766957	3.940152	2.608983	5.763132	3.846422	3.182587	2.262793	6.237107	2.343611	4.164683
	12	1.34802	4.130522	2.547525	2.215329	2.461908	1.295785	2.03845	2.191818	5.881411	1.517059	5.001159	3.061694	3.862109	1.326689	3.994073	2.858237
	13	2.833743	2.584831	1.878359	1.20862	2.532486	2.740028	1.085981	1.292102	3.967218	2.082019	3.302876	1.274886	1.896392	2.624748	1.958097	2.217492
	14	1.643815	3.599531	2.418799	1.798811	2.52785	1.167597	1.868391	1.80152	5.562463	1.568626	4.656988	2.775642	3.51185	0.962683	3.372889	2.61583
	15	1.857315	2.412274	2.338214	1.250416	2.237072	2.22497	1.578793	1.377551	4.929823	2.10396	3.412519	1.794801	2.311724	1.747412	2.385757	2.264173
	AVG	2.303678	3.453055	2.548383	2.012511	2.833709	2.237804	1.945952	2.161001	5.157836	2.509217	4.845444	2.545402	3.187096	2.200399	3.096404	2.869193

Figure A.5: MSE values when training with road 5 and 6 and testing with road 1