Autonomous Driving Scenario Generation in Overtake Manoeuvres Through Data Fusion

Autonomous Driving Scenario Generation in Overtake Manoeuvres Through Data Fusion

Henrique de Carvalho Pinheiro1[0000-0001-8116-336X], Diego Cruz Stanke1, Alessandro Ferraris1 [0000-0003-0712-3399], Massimiliana Carello1[0000-0003-2322-0340], Giovanni Gabiati2, Isabella Camuffo2, Massimo Grillo2

1 Politecnico di Torino, Torino, Italy
2 Fiat Research Center, Torino, Italy massimiliana.carello@polito.it

Abstract. For an effective study of specific driving scenarios, in particular related to overtaking manoeuvres, developing well-thought-out manoeuvre databases from the acquired data will greatly improve the analysis process. The key point being that identifying clearly the studied scenario and clustering the manoeuvres based on specific sub-cases of this will bring an extra dimension of information that allows to visualise existing correlations between a given set of conditions and the manoeuvres performed under them. This paper expands on how to obtain these databases starting from only from overtaken manoeuvres extracted from experimental data. Consisting in two main procedures, the manoeuvre classification itself, in which a hybrid-classifier that combines both supervised and unsupervised algorithms generates the scenario sub-cases, and the database compilation stage, in where it displays the different types of databases that can be created, based on the aim of the study.
Keywords: Autonomous driving, Test scenarios, Overtake manoeuvres, Database generation, Vehicle dynamics.

Introduction

Many are the new technologies that are being discussed in the state of the art of automotive engineering, such as lightweight design [1-4], hybridization and electrification of vehicles [5-8] and, of course, autonomous driving [9,10].

One important goal that can potentially render autonomous driving vehicles more acceptable to clients is developing an overtaking manoeuvre model that can assimilate as close as possible the behaviour of a human driver, because these manoeuvres would feel more “familiar” resulting in greater levels of comfort than what would be perceived through an optimized trajectory. To achieve this, it is deemed crucial to find the parameters that influence how overtakes are performed and would allow to replicate these in a model [11]. Therefore, experimental tests were performed, in which two vehicles mounted with various sensors and a CAN logger were driven repeatedly on the highway by various drivers under controlled conditions (low to mid traffic hours, during sunny days for better camera visibility, and fairly straight highway stretches). This resulted in large amounts of data requiring pre-processing in order to isolate the overtaking manoeuvres so that an analysis could be performed. This study and the main results are reported in [12] that is an important background and starting point of this paper.

The final purpose of this paper was a database of isolated manoeuvres which included the time-series information for any parameter (i.e. vehicle speed at every

point in time), alongside single value data, such as the manoeuvres duration. It is of utmost importance to generate dedicated database for manoeuvre analysis, since it would allow for an efficient and structured study and comparison of the different overtaking manoeuvres. Not only, should this database include the information that allows for a profound analysis, but also identify similar manoeuvres, as to bring an extra dimension of understanding of how certain driving conditions affect human-driving behaviour.

Manoeuvre Classification

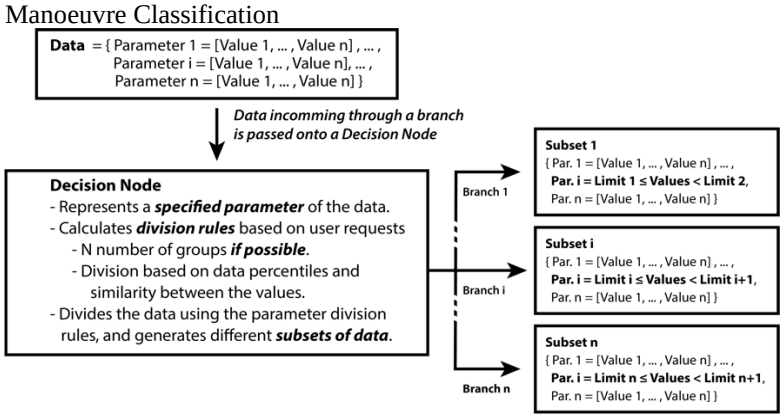

Fig. 1. Decision tree node

The hypothesis was that the manoeuvres must be clustered through a classification process that resembles the decision process of the driver in the specific driving scenario that is being analysed (i.e. initiating a cut-out manoeuvre). Based on this premise, the classifier's macro-structure had to resemble that of a decision tree (Fig. 1), where its decision nodes symbolize the driving-scenario's parameters, the branches represent different non-overlapping intervals of the values related to that parameter, and the leaf nodes represent a given range of conditions for each sub-case (which will now be referred as scenarios) of this macro-scenario. Another important decision was to make it a hybrid-classifier, meaning that it was composed of a supervised and unsupervised set of algorithms. The supervised portion has to do with the user setting the classification requirements, while the unsupervised algorithms oversees the classification by finding a way to comply with as most of them as possible. This structural setup allows it to statistically subdivide the manoeuvres through a set of user-defined parameters representing the requested scenario, thus resulting in a more data-driven clustering method.

Parameter Selection

The decisions represent how the driver evaluates a defined set of conditions related to a specific macro scenario and performs a given course of action [13]. A macro scenario refers to the global situation being analysed, like for example, the moment in which the driver activates the left turn signal to let the other drivers know that he intends to perform the overtake in the coming seconds. Each scenario will represent a specific sub case of this event, thus a specific combination of these conditions, such as a defined range of speeds, lateral displacements, etc. This definition makes it is easier to understand if a specific set of conditions has a determined influence on

the overtaking behaviour, especially when comparing it to other scenarios, but also by observing the similarity between the manoeuvres conforming them.

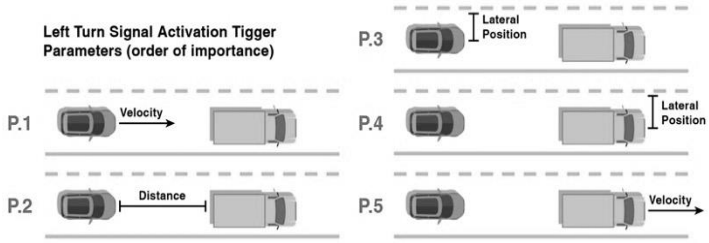

Fig. 2. Decision-making parameters of left-turn signal activation preceding a cut-out

Therefore, the first step towards building the classifier has to do with defining these conditions (these will be referred as classification parameters). Thus, the exact time index in which this event takes place needs to be identified, in order to extract the corresponding value inside a time-series parameter. Alongside defining the parameters, it is key to understand and establish their order of appearance in the decision hierarchy, since it will dictate how the manoeuvres are classified. **Error! Reference source not found.** illustrates the selected parameters and their order of succession that were selected to represent the moment the left-turn light is activated, which denotes the intention of performing a cut-out manoeuvre. It is believed that performing a similar thought process is good practice when studying any type of macro scenario that requires human decision-making. Hopefully, this is performed prior to designing the experimental tests, since it helps identifying which parameters and data needs to be measured and studied, so that in a future stage it is possible to recover the combination of triggers that led to performing the manoeuvre.

Configuration of Decision Tree Classifier

The main objective of the classifier was that in each step it could split the data as balanced as it deemed plausible, while avoiding separating data sharing the same values for the respective decision node parameter.

Since conventional decision trees have decision nodes with predefined splitting rules, they are not able to take into account how the data will be distributed in the future, tending towards an unbalanced splitting. Since an automatized approach was required as well, this lead designing a new approach. Because it had to be both flexible and adaptable to any type of dataset and its respective distribution, it was established that the splitting criteria would rely on the use of percentiles.

Its flexibility has to do with the easiness of selecting the splitting parameters governing each decision node and the desired number of branches into which to subdivide the data subsets at each node, without prior knowledge of it is even possible or wise. Remember that the branching of the preceding node influences the distribution of the variables conforming each sub-set of data reaching a node, making it very difficult to define splitting rules for each subset beforehand. Here is where the system's adaptability comes into play. It analyses the percentile information of the data arriving at the node, determines if the user's splitting request is possible or not, and based on that, it defines the splitting rules it deems most adequate through the following procedure:

Find the requested percentiles.

Calculates the bounds requested by the user (Equation 1). When requested for three divisions, it searches for the values matching the percentiles 0, 33, 66 and 100.

$$for\ i = 1, \dots, n_{branches} \rightarrow Branch(i) = \left[Percentile_{\frac{100\cdot(i-1)}{n_{branches}}}, Percentile_{\frac{100\cdot i}{n_{branches}}} \right] \quad (1)$$

Where nbranches is the number of branches

Analyse one subset at a time and correct their bounds.

Through the logic scheme shown in Fig. 3, the procedure can update each division node's clustering criteria so that all similar entries are grouped together while still maintaining clusters' dimensions as close as it seems wise. It will also reduce the amount of divisions when the requested division rule is too high.

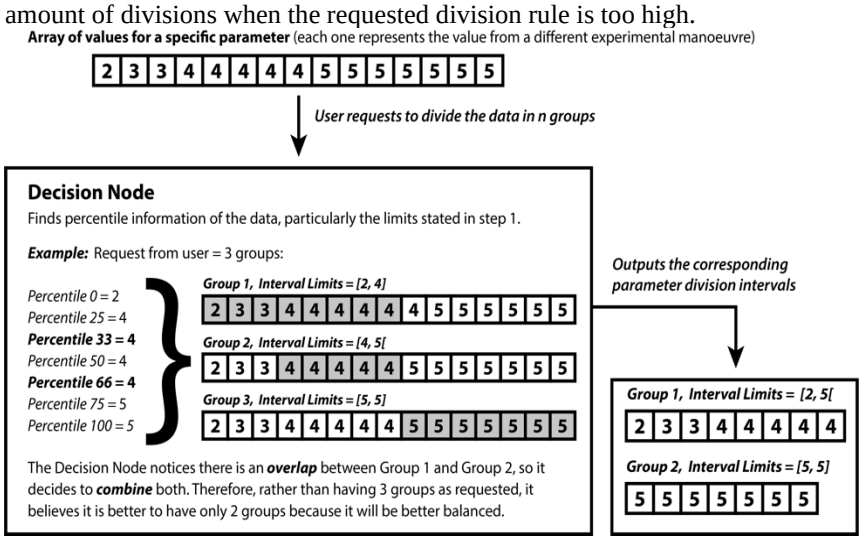

Fig. 3. Branching correction process

Subdivide the data.

Each division node uses the new division instructions to divide the subset's entries along their respective branches into to the subsequent level or leaf node (Fig. 1). Data subsets are essentially the result of grouping the data of the manoeuvres which values corresponding to the analysed parameter are inside the same interval limits (as shown here). In the end, the output would be the final scenarios. It is likely that the total amount of scenarios is lower than the amount requested by the user.

Classifying the Manoeuvres

By extracting the subset of the database that contains solely the information related to the classification parameters, the procedure becomes less memory intensive. In Table 1, it is exemplified how the subset would look like for the macro case scenario depicting the Left Turn Signal Activation used before (Fig. 2), and as it can be noticed, it has a lot less entries than if all the hundreds of measured parameters were included. This data will then pass through each of the decision nodes of the classifier, until it reaches the leaf nodes and the final scenarios are obtained.

Table 1. Example of Manoeuvre Database extract used for Scenario Classification

Manoeuvre Identifier	Decision Node Variables				
	P.1 [km/h]	P.2 [m]	P.3 [m]	P.4 [km/h]	P.5 [m]
1	121.28	65.34	-1.46	98.45	-1.63
2	118.97	102.61	-1.78	101.22	-1.94
⋮	⋮	⋮	⋮	⋮	⋮
n-1	126.73	95.34	-0.98	87.98	-1.16
n	123.64	87.48	-2.12	97.61	-1.59

The classifier then proceeds to add an extra parameter to a new copy of the Main Manoeuvre Database, thus linking each of the manoeuvres to their respective scenario. Alongside it, it will generate a datasheet for each scenario (exemplified in Table 2 for an arbitrary scenario) detailing the interval bounds of each classification parameter based on the data contained inside them, like the maximum and minimum vehicles speeds of the manoeuvres belonging to a scenario. This helps us understand what is going on in the scenario, like if in the case the driver is driving at high speeds and is relatively close to a slow-moving truck, what does he tend to do. It is also a very useful tool when comparing different scenarios, as it aids the user in understanding how these differ from one another and how their characteristics affect the manoeuvres differently.

Table 2. Datasheet regarding an arbitrary scenario

Scenario Number	Classification Parameter	Lower Limit	Upper Limit
8	P.1: EGO Vehicle Speed [km/h]	123.58	126.32
	P.2: Distance from Overtaken Vehicle [m]	76.35	95.62
	P.3: EGO Vehicle Lateral Position [m]	-1.53	-0.29
	P.4: Overtaken Vehicle Speed [km/h]	92.45	104.82
	P.5: Overtaken Vehicle Lateral Position [m]	-1.81	-1.45

Database Generation

The process for building the different databases depends on the objective of the scenario analysis, and in fact, two main classes of databases were identified and developed upon.

One is designed for scenario comparison, while the second is targeted towards analysing manoeuvres belonging to the same scenario and study its spectra.

Selection of Comparison Variables

This selection depends on the aspects of comfort and/or the manoeuvre of interest. For this study, it was important to at least include those variables that described the driver's aggressiveness and vehicle comfort. An indirect reference of the scenario characteristics (retrievable from the scenario datasheet) is presented through the in-

clusion of the scenario number. It is also possible to apply specific criterions (tolerances) to filter out outliers or irrelevant manoeuvres. For example, for the previously referenced left-turn-light activation scenario, the parameters deemed important were: Cut-Out Distance Travelled; Cut-Out Duration; Cut-Out Jerk (Mean, Minimum, Maximum, Peak to Peak); Cut-Out Lateral Acceleration (Mean, Minimum, Maximum, Peak to Peak).

Once these have been defined, the system proceeds to extract and/or calculate (i.e. Average value of a time interval of a parameter's time series) the requested information from the Main Manoeuvre Database and constructs the requested database.

Database for Comparison of Scenarios

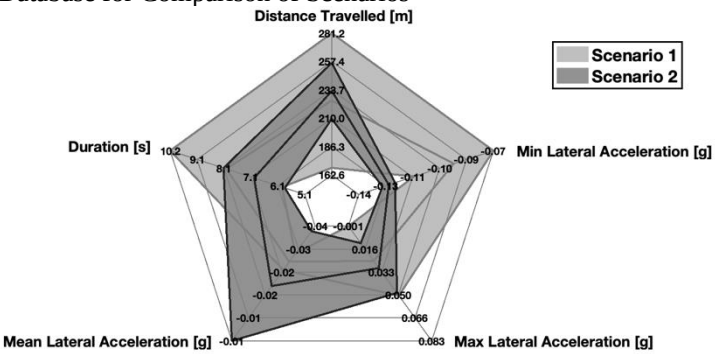

Fig. 4. Spider plot with percentile bands (25%, 50%, 75%) for a cut-out scenario

A simple working principle was established: scenario comparison shall be performed through their respective standard manoeuvres, which are the result of averaging out the manoeuvres belonging to them. These represent the most likely course of action when meeting a specific set of conditions. Unfortunately, it is not entirely true, since these omit information related to dispersion, that can back up how representative these are, as well as putting in evidence how a changing a scenario parameter's range of values may alter the randomness of the driver's behaviour. This dispersion information is represented through a statistical table displaying the requested percentiles. As a consequence, selecting percentile 50 results in the inclusion of the standard manoeuvre, and the dispersion information will get richer by every extra percentile that is included.

A spider plotting tool (Fig. 4) was designed in order to display the information contained in this statistical table. It allows for a visual multi-variable analysis between different scenarios in a very compact and friendly way. This type of plot was chosen since it is commonly used to compare and contrast different attribute between two or more objects (ex. racing games, when comparing two vehicles), making its comprehension easier. The key difference with the typical plots, is that this one can display an attribute's dispersion by plotting the percentile data together. Each of the polygon's contour lines represents a percentile requested by the user, and each of the axis running through the polygon's vertices represents one of the scenario's parameters. Therefore, the intersection between a contour lines and an axis, will indicate a given percentile of that parameter. This is very powerful, because the distance between these contours can indicate that parameter's dispersion, and is what can help

visualize the effect a variation in the scenario's defining parameters (ex. Fig. 2) has on the manoeuvre's randomness.

Database for Studying Individual Scenarios

There are two approaches on the design of the databases for comparing manoeuvres belonging to the same scenario. The first is like before, but without generating the standard manoeuvre data, nor distribution information, and only extracting the manoeuvres from the requested scenario. The second, allows the user to include time-series from the Main Manoeuvre Database, and proceeds to generate a new database consisting of structures, rather than through a table. The first is aimed towards statistical comparison or value-to-value comparison, while the second is useful for comparing time histories and development of the different parameters, like the vehicle's longitudinal velocity or its lateral acceleration at different points along the trajectory.

Conclusions

With the advancements of autonomous driving a robust and precise procedure to test the reliability and representativeness of driving scenarios is necessary, and the literature does not present an established method so far. This paper brings up the discussion about how to organize and effectively divide manoeuvres to better understand the underlying phenomena of one driving condition: overtake manoeuvres in highways. Starting from an already built experimental campaign, described in previous paper, this paper proceeded on the essential steps of manoeuvre classification and the final database generation. The result is a comprehensible and clear method to go from scattered points to visualization tools between driving scenarios. Future developments on the field might include the application of the method in other driving conditions, as well as the development of real-case testing for autonomous vehicles.

Acknowledgments

This paper has been developed in a partnership between FCA and Politecnico di Torino within the topic of "Vehicle Dynamics Assessment for Autonomous Driving". The authors would like to thank Silvio Data and all the team for the support and suggestions.

References

Xu, S., A. Ferraris, A. Giancarlo Airale, and M. Carello. 2017. "Elasto-Kinematics Design of an Innovative Composite Material Suspension System." Mechanical Sciences 8 (1): 11-22. doi:10.5194/ms-8-11-2017

Ferraris, A., A. Messana, D. Multari, L. Sisca,et al. 2019. Steering System of a Low-Consumption Vehicle: From the Dynamics Analysis to the Design of the Wheel Assembly. Mechanisms and Machine Science. Vol. 68. doi:10.1007/978-3-030-03320-0_10

Carello, M., A. Bertipaglia, A. Messana, A. G. Airale, and L. Sisca. 2019. "Modeling and Optimization of the Consumption of a Three-Wheeled Vehicle." SAE Technical Papers 2019-April (April). doi:10.4271/2019-01-0164

Messana, A., Sisca, L., Ferraris, A., Airale, A.G., de Carvalho Pinheiro, H., Sanfilippo, P., Carello, M. 2019. "From Design to Manufacture of a Carbon Fiber Monocoque for a Three-Wheeler Vehicle Prototype". Materials 2019, 12(3), 332; doi: 10.3390/ma12030332

Ferraris, A., F. Micca, A. Messana, A. G. Airale, and M. Carello. 2019. "Feasibility Study of an Innovative Urban Electric-Hybrid Microcar." International Journal of Automotive Technology 20 (2): 237-246. doi:10.1007/s12239-019-0023-x

Ferraris, A., A. G. Airale, A. Messana, S. Xu, and M. Carello. 2018. "The Regenerative Braking for a L7E Range Extender Hybrid Vehicle.". doi:10.1109/EEEIC.2018.8494000

Carello, M., N. Filippo, and R. D'Ippolito. 2012. "Performance Optimization for the XAM Hybrid Electric Vehicle Prototype." SAE Technical Papers. doi:10.4271/2012-01-0773

Cubito, C., L. Rolando, A. Ferraris, M. Carello, and F. Millo. 2017. "Design of the Control Strategy for a Range Extended Hybrid Vehicle by Means of Dynamic Programming Optimization.". doi:10.1109/IVS.2017.7995881

Gelder E. and Paardekooper J. 2017. “Assessment of Automated Driving Systems Using Real-Life Scenarios”. IEEE Intelligent Vehicles Symposium 2017, CA, USA

Kalra N., Paddock S. 2016. “Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?”. Santa Monica, CA: RAND Corporation. https://www.rand.org/pubs/research_reports/RR1478.html.

Carello, M., Ferraris, A., Bucciarelli, L., Data, S. et al., "Customer Oriented Vehicle Dynamics Assessment for Autonomous Driving in Highway," SAE Technical Paper 2019-01-1020, 2019, doi:10.4271/2019-01-1020.

Carello, M., Ferraris, A., de Carvalho Pinheiro, H., Cruz Stanke, D. et al., "Human-Driving Highway Overtake and Its Perceived Comfort: Correlational Study Using Data Fusion," SAE Technical Paper 2020-01-1036, 2020, doi:10.4271/2020-01-1036.

Do H., Tehrani H., Mita S., Egawa M., Muto K., Yoneda K. 2017. “Human Drivers Based Active-Passive Model for Automated Lane Change”. IEEE Intelligent Transportation Systems Magazine. 9. 42-56. doi: 10.1109/MITS.2016.2613913.